BỘ GIÁO DỤC VÀ ĐÀO TẠO

TIN HỌC 10



NHÀ XUẤT BẢN GIÁO DỤC VIỆT NAM

PHẠM THẾ LONG (Tổng Chủ biên)
ĐÀO KIẾN QUỐC (Chủ biên)
BÙI VIỆT HÀ – LÊ CHÍ NGỌC – LÊ KIM THƯ

TIN HỌC

10

(Tái bản lần thứ tư)

KẾT NỐI TRI THỨC
VỚI CUỘC SỐNG

NHÀ XUẤT BẢN GIÁO DỤC VIỆT NAM

Hướng dẫn sử dụng sách

MỤC TIÊU: Giúp em biết sẽ đạt được gì sau bài học.



KHỞI ĐỘNG: Giúp em nhận biết ý nghĩa của bài học bằng cách kết nối những tình huống xuất hiện trong cuộc sống với nội dung bài học.

NỘI DUNG BÀI HỌC:

Các hoạt động: Giúp lớp học tích cực, bài học dễ tiếp thu, học sinh chủ động hơn trong quá trình nhận thức.



Kiến thức mới: Cung cấp cho học sinh nội dung chính của bài học, giúp em bổ sung kiến thức nhằm đạt được mục tiêu của bài học.

Hộp kiến thức: Ghi ngắn gọn hoặc tóm tắt kiến thức mới. Em có thể dùng hộp kiến thức, cùng với bảng giải thích thuật ngữ (ở cuối sách), để ôn tập hoặc tra cứu thuật ngữ mới.



Câu hỏi: Giúp em kiểm tra xem mình đã hiểu bài chưa.

Với các bài có cả lí thuyết và thực hành sẽ có thêm phần thực hành:



THỰC HÀNH Gồm những bài tập dưới dạng nhiệm vụ có hướng dẫn chi tiết.

Cuối mỗi bài là luyện tập và vận dụng:



LUYỆN TẬP Gồm những câu hỏi, bài tập để củng cố kiến thức, kĩ năng trong bài học.



VẬN DỤNG Gồm những câu hỏi, bài tập yêu cầu em dựa vào kiến thức, kĩ năng đã học để giải quyết tình huống thực tiễn.

Ngoài ra có một số bài đơn thuần là thực hành. Trong bài thực hành có nhiệm vụ và hướng dẫn từng bước.

Các bài theo định hướng Tin học ứng dụng (ICT) sẽ có nền tiêu đề của bài là màu xanh lơ, các bài theo định hướng Khoa học máy tính (CS) có nền tiêu đề của bài là màu xanh ngọc.

Hãy bảo quản, giữ gìn sách giáo khoa để dành tặng các em học sinh lớp sau!

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

Lời nói đầu

CÁC EM THÂN MẾN!

Sự phát triển mạnh mẽ của Tin học những năm qua đã và đang dẫn tới nhiều đổi thay to lớn trong mọi lĩnh vực đời sống xã hội. Việc có kiến thức, kĩ năng Tin học và vận dụng các hiểu biết ấy vào cuộc sống sẽ giúp các em sống và làm việc tốt hơn trong xã hội hiện đại.

Trong tay các em là cuốn sách giáo khoa (SGK) Tin học 10, bộ sách "Kết nối tri thức với cuộc sống", Nhà xuất bản Giáo dục Việt Nam.

Theo Chương trình Giáo dục phổ thông 2018 (Chương trình), lớp 10 là năm đầu tiên của giai đoạn giáo dục định hướng nghề nghiệp. Chính vì vậy, ngoài việc củng cố, hệ thống hoá và nâng cao năng lực tin học đã được hình thành, phát triển ở giai đoạn giáo dục cơ bản trước đây, SGK Tin học 10 sẽ cung cấp cho các em những tri thức khởi đầu mang tính định hướng nghề nghiệp thuộc lĩnh vực Tin học, cụ thể giúp các em:

– Có những hiểu biết sâu hơn về thông tin, các phương tiện và công cụ xử lí thông tin hiện đại cùng kiến thức về văn hoá, đạo đức, pháp luật mà mỗi công dân xã hội số trong Kỉ nguyên Cách mạng Công nghiệp 4.0 cần phải có.

– Củng cố và phát triển tư duy giải quyết vấn đề thông qua việc làm quen với lập trình cơ bản trên Python – một trong các ngôn ngữ bậc cao được yêu thích nhất hiện nay trên thế giới.

– Bước đầu tìm hiểu một số ngành nghề trong lĩnh vực Tin học để có thể chủ động và tự tin lựa chọn công việc trong tương lai cho bản thân.

Cũng theo Chương trình, từ lớp 10, môn Tin học sẽ được phân chia thành hai định hướng nghề nghiệp lớn: Tin học ứng dụng (ICT) và Khoa học máy tính (CS). Định hướng ICT sẽ cung cấp nhiều hơn các kĩ năng sử dụng phần mềm ứng dụng; định hướng CS sẽ cung cấp các kiến thức cơ bản sâu hơn liên quan tới lĩnh vực Khoa học máy tính. Tuy vậy, SGK được viết chung cho cả hai định hướng để các em có được một hình dung hệ thống tổng thể. Các em chỉ cần theo học những bài quy định tương ứng với định hướng của mình. Nếu có điều kiện, các em nên tìm hiểu thêm các bài không bắt buộc để mở rộng kiến thức.

Mỗi bài học đều được bắt đầu bằng một hoạt động khởi động gợi mở, các mạch kiến thức trong nội dung bài học đều được gắn với những hoạt động khám phá kiến thức giúp tăng cường tính chủ động của các em trong học tập. Các tình huống thảo luận, các ví dụ minh hoạ và các câu hỏi, bài tập đều được lựa chọn bám sát với thực tế, giúp các em dễ nắm bắt được kiến thức theo yêu cầu cần đạt của Chương trình. Sau khi hình thành và chốt kiến thức, các câu hỏi, bài tập luyện tập sẽ giúp các em củng cố những hiểu biết thu nhận được ngay trên lớp sau mỗi bài học. Bài tập vận dụng cuối mỗi bài học được coi là yêu cầu nâng cao, nhằm khuyến khích các em vận dụng tổng hợp kiến thức vào trong thực tế.

Các chủ đề của SGK được thiết kế tương đối độc lập, do vậy, trừ Chủ đề 6. Hướng nghiệp với tin học nên được học sau các Chủ đề 4. Ứng dụng tin học và Chủ đề 5. Giải quyết vấn đề với sự trợ giúp của máy tính, việc lựa chọn chủ đề nào học trước hay học sau hoàn toàn có thể được thực hiện tuỳ theo điều kiện thực tế.

Chúc các em học tốt!

Mục lục

Trang

Chủ đề 1.
MÁY TÍNH VÀ XÃ HỘI TRI THỨC 6

Bài 1. Thông tin và xử lí thông tin 6
Bài 2. Vai trò của thiết bị thông minh và tin học đối với xã hội 11
Bài 3. Một số kiểu dữ liệu và dữ liệu văn bản 16
Bài 4. Hệ nhị phân và dữ liệu số nguyên 20
Bài 5. Dữ liệu lôgic 24
Bài 6. Dữ liệu âm thanh và hình ảnh 28
Bài 7. Thực hành sử dụng thiết bị số thông dụng 33

Chủ đề 2.
MẠNG MÁY TÍNH VÀ INTERNET 38

Bài 8. Mạng máy tính trong cuộc sống hiện đại 38
Bài 9. An toàn trên không gian mạng 44
Bài 10. Thực hành khai thác tài nguyên trên Internet 50

Chủ đề 3.
ĐẠO ĐỨC, PHÁP LUẬT VÀ VĂN HOÁ TRONG MÔI TRƯỜNG SỐ 55

Bài 11. Ứng xử trên môi trường số. Nghĩa vụ tôn trọng bản quyền 55

Chủ đề 4.
ỨNG DỤNG TIN HỌC 63

Bài 12. Phần mềm thiết kế đồ hoạ 63
Bài 13. Bổ sung các đối tượng đồ hoạ 69
Bài 14. Làm việc với đối tượng đường và văn bản 75
Bài 15. Hoàn thiện hình ảnh đồ hoạ 82

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

Trang

Chủ đề 5.
GIẢI QUYẾT VẤN ĐỀ VỚI SỰ TRỢ GIÚP CỦA MÁY TÍNH 86
Bài 16. Ngôn ngữ lập trình bậc cao và Python 86
Bài 17. Biến và lệnh gán 91
Bài 18. Các lệnh vào ra đơn giản 97
Bài 19. Câu lệnh rẽ nhánh if 101
Bài 20. Câu lệnh lặp for 105
Bài 21. Câu lệnh lặp while 108
Bài 22. Kiểu dữ liệu danh sách 111
Bài 23. Một số lệnh làm việc với dữ liệu danh sách 115
Bài 24. Xâu kí tự 119
Bài 25. Một số lệnh làm việc với xâu kí tự 123
Bài 26. Hàm trong Python 127
Bài 27. Tham số của hàm 131
Bài 28. Phạm vi của biến 136
Bài 29. Nhận biết lỗi chương trình 141
Bài 30. Kiểm thử và gỡ lỗi chương trình 145
Bài 31. Thực hành viết chương trình đơn giản 149
Bài 32. Ôn tập lập trình Python 153

Chủ đề 6.
HƯỚNG NGHIỆP VỚI TIN HỌC 156
Bài 33. Nghề thiết kế đồ hoạ máy tính 156
Bài 34. Nghề phát triển phần mềm 160

Phụ lục 165
Bảng giải thích thuật ngữ 166

Chủ đề 1

Máy tính và xã hội tri thức

BÀI 1

THÔNG TIN VÀ XỬ LÍ THÔNG TIN

SAU BÀI NÀY EM SẼ:

- Phân biệt được thông tin và dữ liệu.
- Chuyển đổi giữa các đơn vị lưu trữ dữ liệu.
- Nêu được sự ưu việt của việc lưu trữ, xử lí và truyền thông tin bằng thiết bị số.

Tin học được định nghĩa là khoa học nghiên cứu các phương pháp và quá trình xử lí thông tin tự động bằng các phương tiện kĩ thuật, chủ yếu bằng máy tính.

Chúng ta đã biết ở lớp dưới, thông tin được biểu diễn trong máy tính bằng các dãy bit (gồm các kí hiệu 0, 1), máy tính xử lí dữ liệu là các dãy bit trong bộ nhớ.

Vậy dữ liệu và thông tin khác nhau như thế nào?

1. THÔNG TIN VÀ DỮ LIỆU

a) Quá trình xử lí thông tin

Thông tin là tất cả những gì mang lại cho chúng ta hiểu biết. Như vậy thông tin gắn với quá trình nhận thức. Máy tính không tự nhận thức được mà chỉ là công cụ hỗ trợ cho con người trong quá trình nhận thức.



Hình 1.1. *Quá trình xử lí thông tin/dữ liệu bằng máy tính*

Quá trình xử lí thông tin của máy tính gồm các bước sau:

Bước 1: Tiếp nhận dữ liệu. Máy tính tiếp nhận dữ liệu theo hai cách thường gặp là:

- Từ các thiết bị, ví dụ tệp hình ảnh từ máy quét là dữ liệu.
- Từ bàn phím do con người nhập, ví dụ khi soạn một văn bản, thông tin của người nhập chuyển thành dữ liệu.

Bước 2: Xử lí dữ liệu. Biến đổi dữ liệu trong bộ nhớ máy tính để tạo ra dữ liệu mới. Ví dụ, từ nhiệt độ trung bình của nhiều năm gần đây, các phần mềm chuyên dụng có thể cho chúng ta biết khuynh hướng Trái Đất đang nóng dần lên.

Bước 3: Đưa ra kết quả. Máy tính có thể đưa ra kết quả theo hai cách:

– Dữ liệu được thể hiện dưới dạng văn bản, âm thanh, hình ảnh,... mà con người có thể hiểu được. Như vậy dữ liệu đã được chuyển thành thông tin.
– Lưu dữ liệu lên một vật mang tin như thẻ nhớ hoặc chuyển thành dữ liệu đầu vào cho một hoạt động xử lí khác.

Quá trình xử lí thông tin/dữ liệu được minh hoạ như Hình 1.1, đầu vào có thể là dữ liệu hoặc thông tin và đầu ra cũng có thể là dữ liệu hoặc thông tin. Trong tin học, dữ liệu là thông tin (âm thanh, hình ảnh, văn bản, số,...) đã được đưa vào máy tính để máy tính có thể nhận biết và xử lí được. Trong máy tính, xử lí thông tin chính là xử lí dữ liệu.

b) Phân biệt dữ liệu và thông tin

Hoạt động 1 **Có thể đồng nhất thông tin với dữ liệu được không?**

Có các ý kiến như sau về dữ liệu của một bài giảng môn Tin học:

An: Bài ghi trong vở của em là dữ liệu.

Minh: Tệp bài soạn bằng Word của cô giáo là dữ liệu.

Khoa: Dữ liệu là tệp video ghi lại tiết giảng của cô giáo.

Theo em bạn nào nói đúng?

Thông tin và dữ liệu độc lập tương đối với nhau:

– Có thể có nhiều loại dữ liệu khác nhau của một thông tin, bài ghi trong vở của trò, tệp bài soạn của cô giáo hay video ghi lại tiết giảng đều là dữ liệu của bài giảng.
– Nếu dữ liệu không đầy đủ thì không xác định được chính xác thông tin. Ví dụ, dữ liệu "39°C" trong bộ dữ liệu về thời tiết mang thông tin "trời rất nóng" nhưng dữ liệu "39°C" trong bộ dữ liệu của bệnh án lại mang thông tin "sốt cao". Như vậy, thông tin có tính toàn vẹn, được hiểu đúng khi có đầy đủ dữ liệu, nếu thiếu dữ liệu thì có thể làm thông tin bị sai hoặc không xác định được.
– Với cùng một bộ dữ liệu, cách xử lí khác nhau có thể đem lại những thông tin khác nhau. Ví dụ dữ liệu thời tiết một ngày nào đó có thể được tổng hợp theo vùng để biết phân bố lượng mưa trong ngày, nhưng cũng có thể xử lí để cho dự báo thời tiết ngày hôm sau.
– Việc xử lí các bộ dữ liệu khác nhau cũng có thể đưa đến cùng một thông tin. Ví dụ, xử lí dữ liệu về băng tan ở Bắc Cực hay cường độ bão ở vùng nhiệt đới đều có thể dẫn đến kết luận về sự nóng lên của Trái Đất.

- Trong tin học, dữ liệu là thông tin đã được đưa vào máy tính để máy tính có thể nhận biết và xử lí được.
- Thông tin là ý nghĩa của dữ liệu. Dữ liệu là các yếu tố thể hiện, xác định thông tin. Thông tin và dữ liệu có tính độc lập tương đối. Cùng một thông tin có thể được thể hiện bởi nhiều loại dữ liệu khác nhau. Ngược lại, một dữ liệu có thể mang nhiều thông tin khác nhau.
- Với vai trò là ý nghĩa của dữ liệu, thông tin có tính toàn vẹn. Dữ liệu không đầy đủ có thể làm thông tin sai lệch, thậm chí không xác định được.

1. Em hãy cho ví dụ về một thông tin nhưng có nhiều cách thể hiện dữ liệu khác nhau.
2. Em hãy cho ví dụ về một dữ liệu có trong nhiều thông tin khác nhau. Tính toàn vẹn của thông tin được thể hiện như thế nào trong ví dụ này?

2. ĐƠN VỊ LƯU TRỮ DỮ LIỆU

Máy tính tổ chức lưu trữ và truy cập dữ liệu không theo từng bit mà theo nhóm bit. Byte là đơn vị dữ liệu nhỏ nhất mà máy tính truy cập được. Một byte gồm 8 bit.

Các máy tính ngày nay đều tổ chức bộ nhớ trong thành những đơn vị lưu trữ có độ dài bằng bội của byte như 2, 4 hay 8 byte. Cách tổ chức như vậy rất tiện cho lưu trữ và truy cập dữ liệu. Chẳng hạn, một đơn vị nhớ 4 byte có thể biểu diễn một số thực, hoặc hai số nguyên 16 bit hoặc vừa đúng 4 kí tự La tinh. Chính vì vậy có thể lấy byte, vốn là đơn vị tổ chức dữ liệu, làm đơn vị đo lượng lưu trữ dữ liệu (thường được gọi là đơn vị lưu trữ dữ liệu hoặc đơn vị lưu trữ thông tin).

Các đơn vị đo dữ liệu hơn kém nhau $2^{10} = 1024$ lần.

Bảng 1.1. *Các đơn vị lưu trữ dữ liệu*

Đơn vị	Kí hiệu	Lượng dữ liệu
bit	b	1 bit
Byte	B	8 bit
Kilobyte	KB	2^{10} B
Megabyte	MB	2^{10} KB
Gigabyte	GB	2^{10} MB
Terabyte	TB	2^{10} GB
Petabyte	PB	2^{10} TB
Exabyte	EB	2^{10} PB
Zettabyte	ZB	2^{10} EB
Yottabyte	YB	2^{10} ZB

1. Định nghĩa nào về Byte là đúng?

A. Là một kí tự.
B. Là đơn vị dữ liệu 8 bit.
C. Là đơn vị đo tốc độ của máy tính.
D. Là một dãy 8 chữ số.

2. Quy đổi các lượng tin sau ra KB:

a) 3 MB. b) 2 GB. c) 2048 B.

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

3. LƯU TRỮ, XỬ LÍ VÀ TRUYỀN THÔNG TIN BẰNG THIẾT BỊ SỐ

Hoạt động 2 Thiết bị số và ưu điểm của thiết bị số

1. Các thiết bị làm việc với thông tin số như lưu trữ, truyền dữ liệu hay xử lí thông tin số đều được gọi là *thiết bị số*. Trong các thiết bị dưới đây, thiết bị nào là thiết bị số? Nếu thiết bị không thuộc loại số thì thiết bị số tương ứng với nó (nếu có) là gì?



a) Đồng hồ cơ *b) Đĩa hát* *c) Bộ thu phát wifi* *d) Thẻ nhớ* *e) Máy tính xách tay*

Hình 1.2

2. Hãy so sánh thiết bị không thuộc loại số ở Hình 1.2 với thiết bị số tương ứng, nếu có.

Thẻ nhớ, bộ thu phát wifi, máy tính xách tay là các thiết bị số. Đồng hồ cơ, đĩa hát ghi các bài hát đều không dùng dữ liệu số.

Ngày nay, tin học được ứng dụng vào mọi lĩnh vực của đời sống đem lại thay đổi lớn về hiệu quả công việc và chất lượng cuộc sống. Những thành tựu đó có được là do có máy tính và các thiết bị số.

Về lưu trữ. Với thiết bị số, người ta có thể lưu trữ một lượng thông tin rất lớn trong một thiết bị nhỏ gọn nhẹ với chi phí thấp. Một đĩa cứng khoảng 2 TB có thể chứa một khối lượng thông tin ngang với một thư viện sách của một trường đại học.

Internet kết nối rất nhiều thiết bị số tạo thành những kho lưu trữ thông tin khổng lồ trong đó có các website chứa rất nhiều thông tin hữu ích. Trên Internet, có thể tìm được hầu hết mọi thông tin bằng các công cụ tìm kiếm.

Lưu trữ thông tin trên thiết bị số còn giúp cho việc tìm kiếm thông tin dễ dàng và nhanh chóng.

Về xử lí. Máy tính xử lí thông tin với tốc độ nhanh và chính xác. Tốc độ xử lí ngày càng được nâng cao. Một máy tính cỡ trung bình ngày nay có thể thực hiện vài chục tỉ phép tính một giây. Thậm chí, một số siêu máy tính đã đạt tốc độ tính toán lên tới hàng trăm triệu tỉ phép tính số học mỗi giây.

Để dự báo thời tiết một tuần, các máy tính phải xử lí các dữ liệu lưu trữ từ nhiều năm trước và các thông tin nhiệt độ, khí áp, độ ẩm, hướng và tốc độ gió từ hàng trăm trạm khí tượng chuyển về với hàng trăm tỉ phép tính cần thực hiện. Siêu máy tính là công cụ hỗ trợ hiệu quả để giải quyết các bài toán có khối lượng tính toán lớn như vậy.

Bên cạnh đó, máy tính có thể thực hiện các lệnh một cách chính xác, không sai sót; các máy chủ có thể làm việc nhiều tháng, nhiều năm cung cấp dịch vụ liên tục cho người dùng (người sử dụng).

Tóm lại, thực hiện tính toán nhanh, cho kết quả chính xác và ổn định là các ưu điểm tuyệt đối của máy tính so với tính toán thủ công của con người.

Về truyền thông tin. Trước đây, tốc độ truyền tin bằng telex chỉ đạt 60 từ một phút. Nhờ tin học và thiết bị số, công nghệ truyền thông tin (truyền tin) đã có những bước tiến dài, tốc độ và độ tin cậy truyền tin được cải thiện nhanh chóng. Ngày nay chúng ta đã có thể xem phim qua Internet, có thể tương tác với nhau trên mạng xã hội "một cách tức thời". Các gia đình có thể sở hữu các đường cáp quang với tốc độ vài chục Mb/s, tương đương với vài triệu kí tự một giây, điều khó tưởng tượng nếu so với đầu những năm 2000.

Thiết bị số có các ưu điểm:

- Giúp xử lí thông tin rất nhanh với độ chính xác cao và có thể làm việc liên tục.
- Có khả năng lưu trữ dữ liệu với dung lượng lớn, giá thành rẻ, tìm kiếm nhanh và dễ dàng.
- Có khả năng truyền tin với tốc độ rất lớn.
- Giúp thực hiện tự động nhiều công việc một cách chính xác, chi phí thấp và tiện lợi.

1. Em hãy so sánh việc gửi thư theo đường bưu điện và gửi thư điện tử theo các tiêu chí tốc độ, chi phí, khả năng lưu trữ.

2. Giả sử để số hoá một cuốn sách kể cả văn bản và hình ảnh cần khối lượng dữ liệu khoảng 50 MB. Thư viện của một trường có khoảng 2000 cuốn sách. Nếu số hoá thì cần khoảng bao nhiêu GB để lưu trữ? Có thể chứa nội dung đó trong thẻ nhớ 256 GB hay không?

LUYỆN TẬP

1. Từ dữ liệu điểm các môn học của học sinh, có thể rút ra những thông tin gì? Mô tả sơ bộ xử lí để rút ra một thông tin trong số đó.

2. Hình 1.3 là danh sách các tệp ảnh lấy ra từ thẻ nhớ của một máy ảnh số. Em hãy tính toán một thẻ nhớ 16 GB có thể chứa được tối đa bao nhiêu ảnh tính theo độ lớn trung bình của ảnh.

Name	Date	Type	Size
DSC0005	4/6/2022 6:51 AM	JPG File	10 144 KB
DSC00113	4/6/2022 7:04 AM	JPG File	9 888 KB
DSC00112	4/6/2022 7:11 AM	JPG File	10 016 KB
DSC00124	4/6/2022 7:15 AM	JPG File	9 440 KB

Hình 1.3. *Danh sách một số tệp ảnh*

VẬN DỤNG

1. Trong thẻ căn cước công dân có gắn chip có thông tin về số căn cước, họ tên, ngày sinh, giới tính, quê quán,... được in trên thẻ để đọc trực tiếp. Ngoài ra, các thông tin ấy còn được mã hoá trong QR code và ghi vào chip nhớ. Theo em, điều đó có lợi gì?

2. Hãy tìm hiểu và mô tả vai trò của thiết bị số trong việc làm thay đổi cơ bản việc chụp ảnh.

BÀI 2

VAI TRÒ CỦA THIẾT BỊ THÔNG MINH VÀ TIN HỌC ĐỐI VỚI XÃ HỘI

SAU BÀI NÀY EM SẼ:

- Nhận biết được một số thiết bị thông minh thông dụng. Nêu được ví dụ cụ thể.
- Biết được vai trò của thiết bị thông minh trong xã hội và cuộc cách mạng công nghiệp lần thứ tư.
- Biết vai trò của tin học đối với xã hội. Nêu được ví dụ.
- Biết các thành tựu nổi bật của ngành Tin học.

Chúng ta từng được nghe rất nhiều thứ gắn với từ "smart" như "smart TV", "smart phone", "smart watch",... Đó là tên gọi của các thiết bị thông minh.

Máy tính xách tay có phải là thiết bị thông minh không?

Chúng ta sẽ cùng tìm hiểu thiết bị thông minh và vai trò của chúng trong cuộc cách mạng công nghiệp lần thứ tư.

1. THIẾT BỊ THÔNG MINH

Hoạt động 1 Nhận biết thiết bị thông minh

Thiết bị nào sau đây là thiết bị thông minh?



a) Đồng hồ lịch vạn niên



b) Điện thoại di động



c) Camera kết nối Internet



d) Máy ảnh số

Hình 2.1

a) Thiết bị thông minh là một hệ thống xử lí thông tin

Có thể hiểu một cách đơn giản thiết bị thông minh là các thiết bị điện tử có thể hoạt động một cách tự chủ trong một mức độ nhất định nhờ các phần mềm điều khiển được cài đặt sẵn. Các thiết bị thông minh ngày nay đều có khả năng tương tác với các thiết bị khác một cách tự động qua mạng không dây như bluetooth, wifi,... để tiếp nhận, xử lí và truyền dữ liệu.

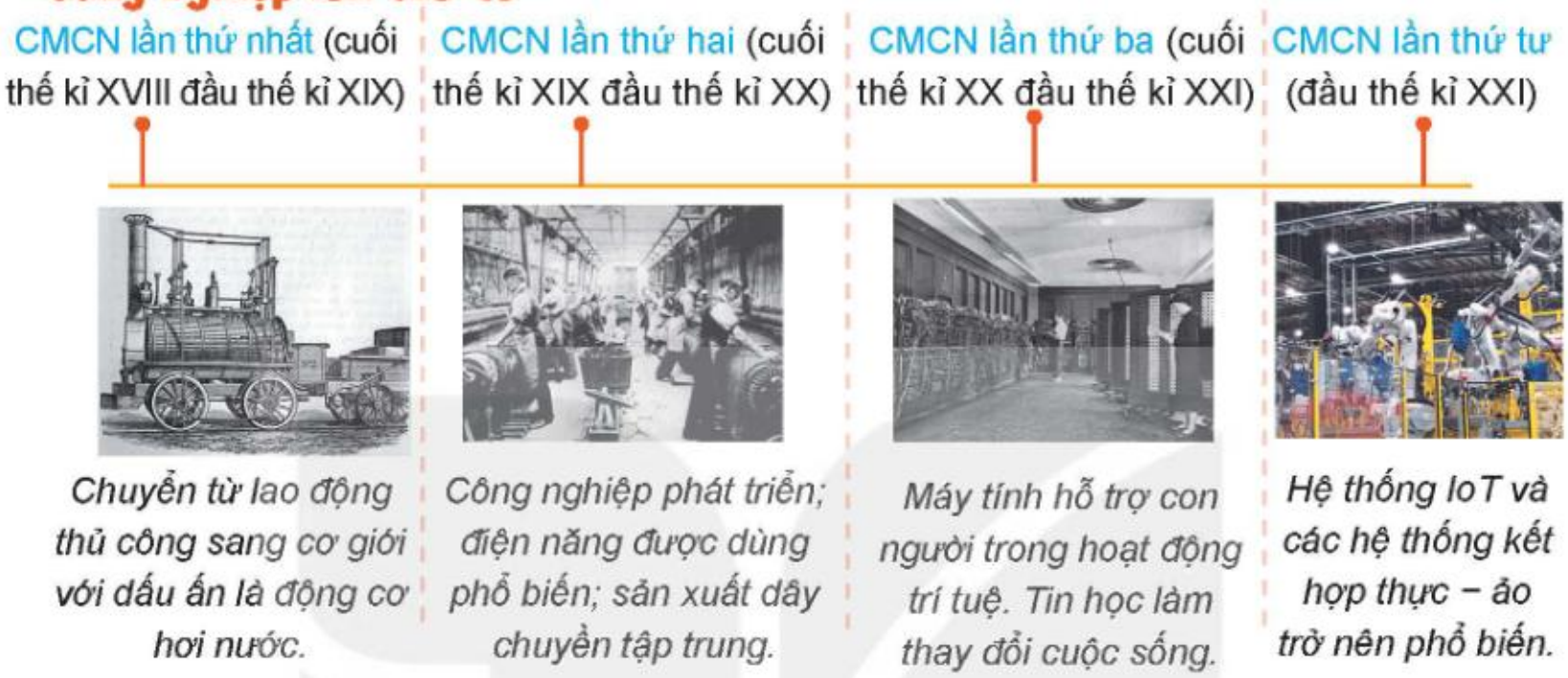
Đồng hồ lịch vạn niên không có khả năng kết nối, máy ảnh số không hoạt động tự chủ nên các thiết bị này không phải thiết bị thông minh.

Những camera kết nối Internet để truyền dữ liệu một cách tự động và có khả năng chọn lọc chỉ ghi hình khi phát hiện chuyển động là một thiết bị thông minh.

Những thiết bị thông minh thường gặp nhất là điện thoại thông minh, máy tính bảng.

Một số thiết bị thông minh hiện nay còn được tích hợp thêm khả năng "bắt chước" một vài hành vi hay cách tư duy của con người ở các mức độ khác nhau. Ví dụ, người máy có thể hiểu và giao tiếp bằng ngôn ngữ tự nhiên với con người; xe tự hành có thể dự đoán khả năng va chạm, từ đó giảm tốc độ và tránh để giữ an toàn,... Các khả năng "bắt chước" đó của thiết bị thông minh nói riêng và của máy móc nói chung, tuy còn hạn chế, được gọi chung là trí tuệ nhân tạo (AI - Artificial Intelligence).

b) Vai trò của thiết bị thông minh đối với xã hội trong cuộc cách mạng công nghiệp lần thứ tư



Hình 2.2. *Các thời kì của cách mạng công nghiệp (CMCN)*

Thế giới đã trải qua ba cuộc cách mạng công nghiệp và đang bước vào cuộc cách mạng công nghiệp lần thứ tư (gọi tắt là cách mạng công nghiệp 4.0). Đây là cuộc cách mạng công nghiệp dựa trên nền tảng công nghệ số và tích hợp với các công nghệ thông minh để tạo ra quy trình và phương thức sản xuất mới.

Việc áp dụng các thành tựu của công nghệ thông tin như trí tuệ nhân tạo; kết nối vạn vật (còn gọi là Internet vạn vật – Internet of Things, IoT); điện toán đám mây (Cloud Computing); dữ liệu lớn (BigData - cung cấp khả năng xử lí trong khoảng thời gian chấp nhận được những khối lượng dữ liệu cực lớn, không ngừng gia tăng, không ngừng biến đổi) cùng các tiến bộ mang tính đột phá của công nghệ sinh học, công nghệ vật liệu, tự động hoá và các công nghệ khác vào hệ thống sản xuất và sản phẩm đã làm mờ ranh giới giữa các hệ thống thực (thế giới vật lí, thế giới các thực thể) và các hệ thống ảo (thế giới số, không gian số). Điều đó tạo ra một nền sản xuất thông minh mà ở đó sản phẩm được sản xuất trong thế giới vật lí nhưng quá trình tính toán, thiết kế, tạo mẫu,... được thực hiện trên không gian số. Ví dụ, các xe tự hành sử dụng các cảm biến thông minh thu thập dữ liệu về biển báo, phương tiện tham gia giao thông, đường đi và chướng ngại vật để tính toán tốc độ và hướng đi tối ưu (trong thế giới ảo) và gửi quyết định cho xe thực hiện (trong thế giới thực).

Ngày nay, tri thức – yếu tố trí tuệ của con người – ngày càng chiếm tỉ trọng lớn hơn trong cơ cấu giá trị của sản phẩm và dịch vụ. Ví dụ, trong giá thành sản xuất một chiếc điện thoại thông minh thì giá trị nguyên vật liệu, nhân công chỉ chiếm chưa đến một nửa; phần còn lại là chi phí thiết kế, làm phần mềm và mua các bản quyền công nghệ – đó chính là giá trị của tri thức. Nền kinh tế tri thức là nền kinh tế chủ yếu dựa vào tri thức, trên cơ sở phát triển khoa học và công nghệ cao, lấy tri thức làm động lực

chủ yếu tăng trưởng kinh tế. Cuộc cách mạng công nghiệp 4.0 đã thúc đẩy mạnh mẽ nền kinh tế tri thức.

IoT là việc kết nối các thiết bị thông minh với nhau nhằm thu thập và xử lí thông tin một cách tự động, tức thời trên diện rộng như trong các ứng dụng giám sát giao thông, cảnh báo thiên tai, lái xe tự động, điều khiển quá trình sản xuất trong nhà máy và nhiều ứng dụng khác. IoT là một yếu tố cơ bản trong cách mạng công nghiệp 4.0, trong đó thiết bị thông minh là thành phần chủ chốt.

- Thiết bị thông minh là thiết bị điện tử có thể hoạt động tự chủ không cần sự can thiệp của con người, tự thích ứng với hoàn cảnh và có khả năng kết nối với các thiết bị khác để trao đổi dữ liệu.
- Thiết bị thông minh đóng vai trò chủ chốt trong các hệ thống IoT - một nội dung cơ bản của cuộc cách mạng công nghiệp 4.0.

1. Thiết bị nào trong Hình 2.3 là thiết bị thông minh? Tại sao?



a) *Cân điện tử*



b) *Đồng hồ kết nối với điện thoại qua bluetooth*



c) *Máy tính cầm tay*

Hình 2.3

2. Ngoài những thiết bị đã nêu trong bài, nhà em có những thiết bị thông minh nào?

2. CÁC THÀNH TỰU CỦA TIN HỌC

Các thành tựu của tin học cần được nhìn nhận trên hai phương diện:

– Các thành tựu về ứng dụng.

– Các thành tựu liên quan đến sự phát triển của chính ngành Tin học.

Hoạt động 2 **Vai trò của tin học đối với xã hội**

Cuộc sống sẽ thay đổi như thế nào nếu không có máy tính và các thiết bị thông minh?

a) Đóng góp của tin học với xã hội

Tin học được ứng dụng trong mọi lĩnh vực. Sau đây là một số lĩnh vực ứng dụng phổ biến nhất.

Quản lí. Trường học cần quản lí học tập của học sinh. Ngân hàng cần quản lí tài khoản và giao dịch của khách hàng. Doanh nghiệp cần quản lí sản xuất, mua bán hàng hoá, kế toán,... Dùng máy tính quản lí các quy trình nghiệp vụ giúp xử lí công việc nhanh chóng, chính xác, hiệu quả và tiện lợi, tiết kiệm chi phí.

Tự động hoá. Nhờ máy tính, tự động hoá đã thay đổi căn bản. Các thiết bị thông minh hoạt động theo chương trình có thể thực hiện tốt nhiều công việc thay cho con người. Robot là một ví dụ điển hình về thành công của tự động hoá. Không những hoạt động chính xác, thông minh, robot còn có thể làm việc ở những nơi có môi trường nguy hiểm như trong nhà máy điện hạt nhân hay dưới nước.

Giải quyết các bài toán khoa học kĩ thuật. Với khả năng tính toán nhanh, chính xác, máy tính có thể hỗ trợ cho công việc tính toán, mô phỏng, kiểm nghiệm trong nghiên cứu, thiết kế các công trình, dự báo thời tiết, giải mã gen, ứng dụng bản đồ số,...

Thay đổi cách thức làm việc của nhiều ngành nghề. Nhiều công việc có thể được thực hiện trực tuyến như dạy học, mua hàng. Các ứng dụng tin học văn phòng đã góp phần tăng chất lượng và hiệu quả hoạt động hành chính. Nhiều ngành nghề thay đổi hoàn toàn như công nghệ in theo kiểu sắp chữ được thay bằng chế bản trên máy tính hay chụp ảnh dùng phim được thay bằng chụp ảnh số với chất lượng và chi phí được cải thiện rõ rệt.

Giao tiếp cộng đồng. Tin học giúp trao đổi thông tin nhanh chóng và hiệu quả qua các ứng dụng như thư điện tử, các diễn đàn trên các trang web và các mạng xã hội như Youtube, Facebook, Twitter, Zalo,... Mạng xã hội đã trở thành môi trường giao tiếp tiện lợi mà trên đó mọi hoạt động đều dễ dàng, trực quan, tương tác nhanh.

b) Một số thành tựu phát triển của Tin học

Ở các lớp dưới ta đã biết sự phát triển mạnh mẽ của phần cứng bao gồm máy tính, các thiết bị ngoại vi và các thiết bị số nói chung, giúp cho tin học phát triển vượt bậc. Sau đây là một số thành tựu giúp tin học nói chung và máy tính nói riêng trở thành một phần không thể thiếu trong xã hội hiện đại.

Hệ điều hành. Hệ điều hành OS/360 (1964) trên dòng máy IBM/360 là hệ điều hành đầu tiên đã đưa ra hầu hết các nguyên lí của các hệ điều hành hiện đại. Hệ điều hành giúp quản lí thông tin, quản lí phần cứng, quản lí các tiến trình xử lí của máy tính và cung cấp giao diện làm việc với người dùng.

DOS và Windows là các hệ điều hành cho dòng máy PC; macOS là hệ điều hành cho dòng máy Mac. Android và iOS là các hệ điều hành thông dụng trên thiết bị di động. Còn trên các máy tính lớn, UNIX vẫn là hệ điều hành chủ đạo.

Mạng và Internet. Mạng máy tính cho phép kết nối các máy tính và thiết bị thông minh để trao đổi dữ liệu với nhau nhằm thực hiện các ứng dụng liên quan đến nhiều người hay nhiều thiết bị trong một phạm vi rộng.

Một thành tựu nổi bật là Internet cho phép kết nối toàn cầu nhờ thiết lập được các quy tắc trao đổi dữ liệu (được biết đến với tên gọi là giao thức TCP/IP) vào năm 1983.

Việc phát minh ra World Wide Web (WWW) vào năm 1992 đã tạo ra phương tiện truy cập Internet dễ dàng và nhất quán, giúp phổ cập Internet. Nhờ các công cụ tìm kiếm chúng ta có thể tìm kiếm thông tin trên không gian mạng một cách dễ dàng và nhanh chóng. Internet không chỉ giúp chia sẻ thông tin, mà còn cho phép chia sẻ cả phần cứng và phần mềm.

Rất nhiều hoạt động đã được thực hiện trực tuyến. Internet đã thay đổi cuộc sống của con người.

Các ngôn ngữ lập trình bậc cao. Các chương trình máy tính phải được viết trong một ngôn ngữ lập trình. Thời kì đầu, chương trình được viết trên ngôn ngữ máy hoặc hợp ngữ, về cơ bản là phụ thuộc vào loại máy tính cụ thể, đòi hỏi viết rất tỉ mỉ nên rất khó kiểm soát dẫn đến hiệu quả phát triển phần mềm thấp.

Người ta đã sáng tạo ra các ngôn ngữ bậc cao giúp người lập trình chỉ cần thể hiện cách giải quyết vấn đề (thuật toán) mà không cần biết đến các lệnh máy (mã máy). Các chương trình dịch sẽ chuyển chương trình sang dạng mã máy. FORTRAN là ngôn ngữ lập trình bậc cao đầu tiên được công bố chính thức vào năm 1957 và

vẫn được dùng tới ngày nay cho những người lập trình khoa học kĩ thuật. Sau này đã xuất hiện thêm nhiều ngôn ngữ lập trình khác như Cobol, C, Pascal, Python,... Ngôn ngữ lập trình bậc cao đã giúp làm tăng hiệu quả của việc lập trình.

Các hệ quản trị cơ sở dữ liệu cung cấp các công cụ để tổ chức, cập nhật, truy cập dữ liệu không phụ thuộc vào các bài toán cụ thể. Người phát triển phần mềm chỉ cần tập trung vào việc viết chương trình cho các hoạt động nghiệp vụ mà không phải tự viết các phần mềm khởi tạo, cập nhật, tìm kiếm, trích xuất dữ liệu. Nhờ có các hệ quản trị cơ sở dữ liệu mà các ứng dụng quản lí mới phát triển được như ngày nay.

Hệ quản trị cơ sở dữ liệu đầu tiên ra đời vào cuối những năm 60 của thế kỉ XX nhưng sau năm 1970 mới có các hệ quản trị cơ sở dữ liệu sử dụng dữ liệu dưới dạng bảng như ngày nay, ví dụ DB2, Microsoft SQL Server, Oracle, MySQL,...

Các thành quả nghiên cứu khoa học của tin học như trí tuệ nhân tạo, dữ liệu lớn,... là những thành tựu lớn của tin học, được ứng dụng trong nhiều lĩnh vực khác nhau.

- Ngày nay, Tin học đã đem lại nhiều thay đổi trong mọi lĩnh vực của xã hội, từ quản lí điều hành, tự động hoá các quy trình sản xuất, giải quyết các bài toán cụ thể trong khoa học và kĩ thuật cho tới việc thay đổi cách thức làm việc của nhiều ngành nghề cũng như thói quen giao tiếp cộng đồng,...
- Sự phát triển mạnh mẽ của các hệ thống phần cứng, các thiết bị số cùng các phần mềm hệ thống, phần mềm công cụ, phần mềm ứng dụng, các hệ quản trị cơ sở dữ liệu,... và sự phát triển mang tính bùng nổ của mạng máy tính và Internet là những yếu tố quyết định để máy tính trở thành một phần không thể thiếu trong xã hội hiện đại.



1. Tin học đã giúp gì cho em trong học tập?

2. Em hãy cho ví dụ về một số ứng dụng trực tuyến.



LUYỆN TẬP

1. Thiết bị thông minh nào có thể nhận dạng được hình ảnh?

2. Các phần mềm tin học văn phòng đã trở thành các phần mềm được dùng nhiều nhất. Em hãy nêu tác dụng của các phần mềm tin học văn phòng.



VẬN DỤNG

1. Các hệ thống giám sát giao thông nhờ các camera thông minh đang được triển khai ở các thành phố. Hãy truy cập Internet, tìm hiểu về cách kết nối các thiết bị thông minh trong các hệ thống đó. Nêu lợi ích của hệ thống.

2. Xe tự hành được xem là một thành tựu điển hình của cuộc cách mạng công nghiệp 4.0. Với xe tự hành, người dân không cần sở hữu xe cá nhân, muốn đi lại chỉ cần đặt xe qua Internet. Hãy tìm hiểu lợi ích của xe tự hành giúp hạn chế ô nhiễm, ùn tắc giao thông và giảm chi phí.

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

BÀI 3

MỘT SỐ KIỂU DỮ LIỆU VÀ DỮ LIỆU VĂN BẢN

SAU BÀI NÀY EM SẼ:

- Nêu được các loại thông tin và các kiểu dữ liệu sẽ gặp trong chương trình tin học phổ thông.
- Biết được các bảng mã thông dụng ASCII và Unicode.
- Giải thích được sơ lược về việc số hoá văn bản.

Thông tin đưa vào bộ nhớ máy tính dưới dạng các dãy bit. Như vậy khi đưa vào máy tính, phải mã hoá thông tin thành dữ liệu nhị phân. Tuỳ theo bản chất của thông tin được mã hoá mà dữ liệu tương ứng có cách biểu diễn riêng, từ đó hình thành nên các kiểu dữ liệu khác nhau. Vậy trong máy tính có các kiểu dữ liệu nào?

1. PHÂN LOẠI VÀ BIỂU DIỄN THÔNG TIN TRONG MÁY TÍNH

Hoạt động 1 Phân loại thông tin

Hình 3.1 minh hoạ một thẻ căn cước công dân. Trên đó có những thông tin gì?

Hãy chia những thông tin đó thành các nhóm, ví dụ nhóm các thông tin có thể tách ghép được hay so sánh được để tìm kiếm và nhóm các thông tin có thể thực hiện được với các phép tính số học.

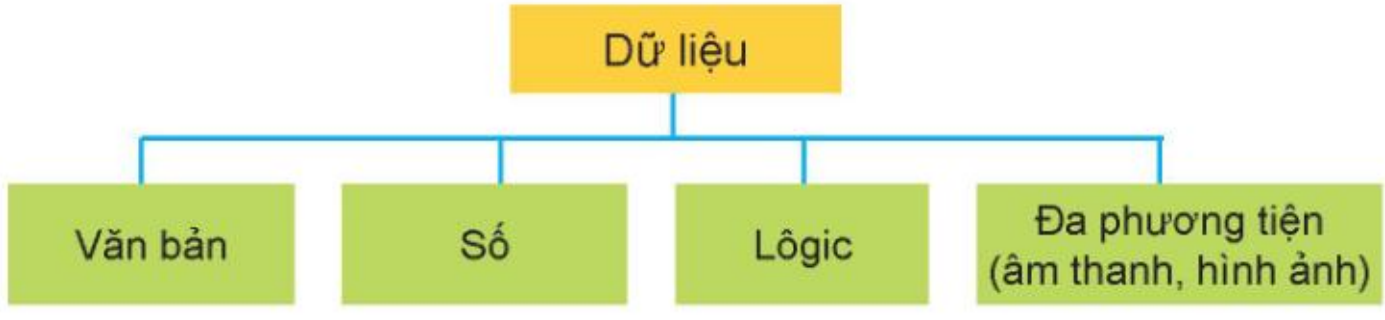


Hình 3.1. *Căn cước công dân*

Khi đưa vào máy tính thông tin được chuyển thành dữ liệu. Dữ liệu trên máy cũng cần được phân loại cho phù hợp với các phép xử lí trong máy tính. Ví dụ, đối với các dữ liệu là số có thể tính toán và so sánh. Còn đối với các dữ liệu dạng văn bản thì có thể tách, ghép, so sánh.

Việc mã hoá thông tin thành dữ liệu nhị phân được gọi là biểu diễn thông tin. Biểu diễn thông tin là bước đầu để có thể đưa thông tin vào máy tính.

Hình 3.2 là sơ đồ phân loại các kiểu dữ liệu được đề cập trong chương trình tin học phổ thông.



Hình 3.2. *Sơ đồ phân loại các kiểu dữ liệu*

- Biểu diễn thông tin trong máy tính là cách mã hoá thông tin.
- Các kiểu dữ liệu thường gặp là văn bản, số, hình ảnh, âm thanh và lôgic.
- Việc phân loại dữ liệu để có cách biểu diễn phù hợp nhằm tạo thuận lợi cho việc xử lí thông tin trong máy tính.



1. Theo em số căn cước công dân có kiểu số hay kiểu văn bản?
2. Kiểu số thực thường dùng để biểu diễn các số có phần thập phân (phần lẻ). Em hãy cho ví dụ một loại hồ sơ có dữ liệu kiểu số thực.

2. BIỂU DIỄN DỮ LIỆU VĂN BẢN

Việc đưa văn bản vào máy tính như thế nào không chỉ phụ thuộc vào kiểu dữ liệu là kí tự, xâu kí tự hay tệp văn bản mà còn phụ thuộc vào các kí tự ấy được mã hoá như thế nào? Cách mã hoá được quy định trong bảng kí tự.

Hoạt động 2 **Bảng chữ cái tiếng Anh và bảng chữ cái tiếng Việt**

1. Bảng chữ cái tiếng Anh có những kí tự nào?
2. Trong tin học, mỗi nguyên âm có dấu thanh của tiếng Việt là một kí tự. Hãy kể tên các kí tự tiếng Việt không có trong bảng chữ cái tiếng Anh. Có bao nhiêu kí tự như vậy?



a) Bảng mã ASCII

Bảng mã được dùng phổ biến nhất trong tin học là "bảng mã chuẩn của Mỹ để trao đổi thông tin" (American Standard Code for Information Interchange viết tắt là ASCII). Ban đầu bảng mã này dùng các mã 7 bit, với 128 (2^7) mã khác nhau nên chỉ thể hiện được đúng 128 kí tự. Bảng mã 7 bit chỉ đủ dùng cho tiếng Anh, trong khi đó nhiều quốc gia có các kí tự riêng, như tiếng Hy Lạp có các kí tự α, β,γ; tiếng Nga có các kí tự б, г, д. Do đó, người ta đã mở rộng bảng mã 7 bit thành bảng mã 8 bit gọi là bảng mã ASCII mở rộng, cho phép mã hoá 256 kí tự, trong đó giữ nguyên 128 kí tự cũ. 128 vị trí được thêm vào trong bảng mã 8 bit so với bảng mã 7 bit được gọi là phần mở rộng của bảng mã ASCII. Các quốc gia có thể sử dụng phần mở rộng này cho các kí tự riêng của mình. Bảng mã ASCII với phần mở rộng chưa được thay thế bởi các kí tự riêng của các quốc gia được nêu trong Bảng phụ lục trang 165. Trong bảng này, muốn lấy mã nhị phân của một kí tự thì chỉ cần ghép 4 bit ở chỉ số hàng với 4 bit ở chỉ số cột tương ứng với kí tự. Ví dụ mã nhị phân của "A" (có số thứ tự là 65) là 01000001.

b) Bảng mã Unicode và tiếng Việt trong Unicode

Ngoài các kí tự có trong bảng chữ cái tiếng Anh, tiếng Việt còn có 134 nguyên âm có dấu thanh và phụ âm "đ", kể cả chữ in hoa, đều không có sẵn trong bảng mã ASCII gốc, tuy nhiên phần mở rộng của bảng mã này lại chỉ có 128 vị trí.

Tình trạng thiếu vị trí còn trầm trọng hơn đối với các quốc gia dùng chữ tượng hình như Trung Quốc, Nhật Bản với vài chục nghìn kí tự. Chính vì thế, đầu những năm 1980, người ta đã đề xuất một chương trình quốc tế nhằm xây dựng một bảng mã hợp nhất, dùng chung cho mọi quốc gia, gọi là Unicode. Unicode thực tế là một bộ tiêu chuẩn biểu diễn kí tự văn bản trong máy tính, cho phép sử dụng nhiều hơn 8 bit để biểu diễn các kí tự thuộc nhiều ngôn ngữ khác nhau trên thế giới. Nhờ vậy, nếu bảng mã ASCII chỉ cho phép mã hoá 256 kí tự, thì Unicode hiện nay đã cho phép mã hoá hàng trăm nghìn kí tự khác nhau. Cùng với quy định, nếu quốc gia nào đã có một mặt chữ được xác định trước (bao gồm các kí tự gốc của bảng mã ASCII) thì quốc gia khác có thể dùng lại mà không phải định nghĩa một mã mới, Unicode tránh được tình trạng thiếu nhất quán do các quốc gia dùng các mặt chữ giống nhau nhưng mã khác nhau. Việc sử dụng Unicode tạo ra những ứng dụng đa ngôn ngữ, sử dụng đồng thời nhiều ngôn ngữ khác nhau như các trình duyệt web, ngôn ngữ lập trình, các phần mềm ứng dụng,...

Năm 2001 Việt Nam đã ban hành Tiêu chuẩn TCVN 6909:2001 về Bộ mã kí tự tiếng Việt 16-bit để sử dụng chung. Tiêu chuẩn này hoàn toàn phù hợp với tiêu chuẩn quốc tế về Unicode. Nó quy định mỗi kí tự đều được biểu diễn bằng 2 byte (các kí tự đã có trong bảng mã ASCII được bổ sung vào phía trước 8 bit giá trị 0). Năm 2017, Việt Nam cũng đã ban hành quy định bắt buộc sử dụng UTF-8 để biểu diễn bộ kí tự Unicode trong máy tính, trong đó sử dụng 1 byte để mã hoá các kí tự La tinh không dấu, sử dụng 2 byte để mã hoá các nguyên âm có dấu cùng các kí tự đ, Đ và chỉ dùng 3 byte để mã hoá một số rất ít các kí tự đặc biệt. UTF-8 (8-bit Unicode Transformation Format) là một trong các hệ thống định dạng chuyển đổi cho phép mã hoá kí tự với độ dài khác nhau (từ 1 tới 4 byte) dành cho Unicode.

Như vậy, hiểu một cách ngắn gọn, các bảng mã ASCII và Unicode quy định cách *biểu diễn kí tự*.

c) Số hoá văn bản

Tệp văn bản là định dạng lưu trữ ở bộ nhớ ngoài. Việc số hoá văn bản được thực hiện bằng các phần mềm soạn thảo văn bản như Word (của Microsoft) hay Writer (của Open Office). Gần đây người ta đã có thể nhập văn bản bằng nhận dạng tiếng nói. Chỉ cần đọc lời, máy tính có thể nhận dạng âm thanh và tạo ra văn bản.

- Bảng mã ASCII mở rộng sử dụng 8 bit để biểu diễn một kí tự.
- Unicode là bảng mã hợp nhất quốc tế, cho phép tạo ra các ứng dụng đa ngôn ngữ. Mỗi kí tự Unicode có thể được mã hoá bởi nhiều byte.



1. Sử dụng phụ lục Bảng mã ASCII mở rộng trang 165, hãy xác định mã nhị phân và mã thập phân (là số thứ tự của kí tự trong bảng) của các kí tự S, G, K.

2. Trong bảng mã UNICODE, mỗi kí tự Tiếng Việt theo UTF-8 được biểu diễn bởi bao nhiêu byte?

A. 1 byte. B. 2 byte. C. 4 byte. D. từ 1 đến 3 byte.

LUYỆN TẬP

1. Giấy phép lái xe có các thông tin nêu ở cột bên trái của bảng sau. Hãy ghép mỗi thông tin ở cột bên trái với kiểu dữ liệu thích hợp ở cột bên phải.

Thông tin	Kiểu dữ liệu
Ảnh	Số
Số	Văn bản
Họ tên	Hình ảnh
Ngày sinh	Âm thanh
Quốc tịch	
Nơi cư trú	



2. Câu trả lời nào đúng cho câu hỏi "Tại sao cần xây dựng bảng mã Unicode?"?

A. Để đảm bảo bình đẳng cho mọi quốc gia trong ứng dụng tin học.

B. Bảng mã ASCII mã hoá mỗi kí tự bởi 1 byte. Giá thành thiết bị lưu trữ ngày càng rẻ nên không cần phải sử dụng các bộ kí tự mã hoá bởi 1 byte.

C. Dùng một bảng mã chung cho mọi quốc gia, giải quyết vấn đề thiếu vị trí cho bộ kí tự của một số quốc gia, đáp ứng nhu cầu dùng nhiều ngôn ngữ đồng thời trong cùng một ứng dụng.

D. Dùng cho các quốc gia sử dụng chữ tượng hình.

VẬN DỤNG

1. Dựa trên bảng mã ASCII, Việt Nam xây dựng bảng mã VSCII (Vietnamese Standard Code for Information Interchange), còn gọi là TCVN 5712:1993. Hãy tìm hiểu bảng mã này trên Internet theo những gợi ý sau:

– Bảng mã có đủ cho tất cả các kí tự tiếng Việt không?

– Bảng mã có bảo toàn bảng mã ASCII 7 bit không?

2. Phông chữ là hình ảnh của kí tự ứng với mã của kí tự. Không phải phông chữ nào cũng được thiết kế đầy đủ cho tiếng Việt. Hãy sử dụng phần mềm soạn thảo gõ một câu tiếng Việt và định dạng với các phông chữ khác nhau để tìm hiểu ngoài phông Times New Roman còn những phông chữ nào đã thiết kế cho tiếng Việt Unicode.

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

BÀI 4

HỆ NHỊ PHÂN VÀ DỮ LIỆU SỐ NGUYÊN

SAU BÀI NÀY EM SẼ:

- Biết được hệ nhị phân và biểu diễn số nguyên trong máy tính.
- Giải thích được ứng dụng của hệ nhị phân trong tin học.

Trong hệ thập phân, mỗi số có thể được phân tích thành tổng các luỹ thừa của 10 với hệ số của mỗi số hạng chính là các chữ số tương ứng của số đó. Ví dụ số 513 có thể viết thành: $5 \times 10^2 + 1 \times 10^1 + 3 \times 10^0$.

Ta cũng có thể phân tích một số thành tổng các luỹ thừa của 2, chẳng hạn 13 có thể viết thành: $1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$ với các hệ số chỉ là 0 hay 1.

Khi đó, có thể thể hiện 13 bởi 1101 được không? Em hãy cho biết việc thể hiện giá trị của một số bằng dãy bit có lợi gì.

1. HỆ NHỊ PHÂN VÀ BIỂU DIỄN SỐ NGUYÊN

Hoạt động 1 Biểu diễn một số dưới dạng tổng các luỹ thừa của 2

Em hãy viết số 19 thành tổng các luỹ thừa của 2.

Gợi ý: Hãy lập danh sách các luỹ thừa của 2 như 16, 8, 4, 2, 1 và tách dần khỏi 19 cho đến hết.

a) Hệ nhị phân

Số 19 có thể được biểu diễn bằng tổng $2^4 + 2^1 + 2^0$ hoặc viết dưới dạng đầy đủ các luỹ thừa: $1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$.

Tương tự như hệ thập phân, 2 có thể được dùng làm cơ số cho một hệ đếm gọi là hệ đếm cơ số 2 hay hệ nhị phân với các đặc điểm sau:

- Chỉ dùng hai chữ số là 0 và 1, các chữ số 0 và 1 gọi là các chữ số nhị phân.
- Mỗi số có thể biểu diễn bởi một dãy các chữ số nhị phân.
- Trong biểu diễn số nhị phân, một chữ số ở một hàng sẽ có giá trị gấp 2 lần chính chữ số đó ở hàng liền kề bên phải. Vì vậy chữ số 1 ở vị trí thứ k kể từ phải sang trái sẽ mang giá trị là 2^{k-1}.

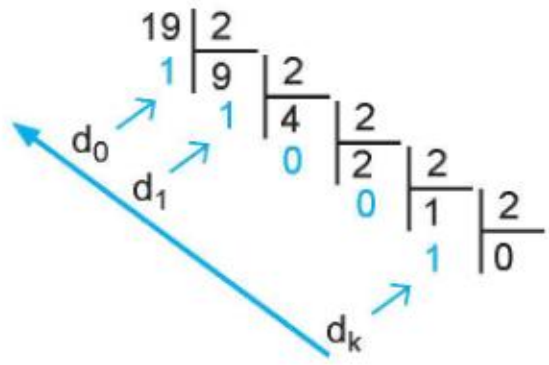
Trong hệ nhị phân, số 19 sẽ có biểu diễn là 10011. Khi cần phân biệt số được biểu diễn trong hệ đếm nào người ta viết cơ số làm chỉ số dưới như 19_{10} hay 10011_2.

b) Đổi biểu diễn số nguyên dương từ hệ thập phân sang hệ nhị phân

Giả sử cần đổi số tự nhiên N trong hệ thập phân sang số nhị phân có dạng $d_k d_{k-1} \ldots d_1 d_0$, nghĩa là cần tìm các số d_k, d_{k-1},..., d_1, d_0 có giá trị bằng 0 hoặc 1 sao cho

$$N = d_k \times 2^k + d_{k-1} \times 2^{k-1} + \ldots + d_1 \times 2 + d_0.$$

Để tìm các số d_k, d_{k-1},... , d_1, d_0, người ta chia liên tiếp N cho 2 để tìm số dư như minh hoạ việc đổi số 19 sang số nhị phân ở Hình 4.1.



Hình 4.1. *Đổi một số từ hệ thập phân sang hệ nhị phân*

Viết các số dư theo chiều từ dưới lên, ta được số nhị phân cần tìm:

$$19_{10} = 10011_2.$$

Việc đổi số nhị phân có dạng $d_k d_{k-1} ... d_1 d_0$ sang số thập phân thực chất chỉ là việc tính tổng $d_k \times 2^k + d_{k-1} \times 2^{k-1} + ... + d_1 \times 2 + d_0$. Ví dụ:

$$1101_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 13.$$

c) Biểu diễn số nguyên trong máy tính

Có hai phương pháp để biểu diễn số trong máy tính là dấu phẩy tĩnh và dấu phẩy động, trong đó phương pháp dấu phẩy động thường được dùng khi tính toán với các số quá lớn, quá nhỏ hoặc không nguyên (có phần thập phân). Dưới đây chúng ta chỉ đề cập tới cách biểu diễn dấu phẩy tĩnh cho số nguyên.

Biểu diễn số nguyên không dấu chính là thể hiện của số trong hệ đếm cơ số 2. Khi được đưa vào bộ nhớ, tuỳ theo số nhỏ hay lớn mà có thể phải dùng một hay nhiều byte. Ví dụ số 19 trong hệ đếm nhị phân có biểu diễn là 10011 chỉ cần một byte với ba bit 0 bổ sung thêm bên trái cho đủ 8 bit, nhưng số $620_{10} = 1001101100_2$ sẽ phải sử dụng 2 byte và cần bổ sung thêm 6 bit 0 vào phía trái cho đủ 16 bit.

Đối với số nguyên có dấu, có một số cách mã hoá như mã thuận (còn gọi là mã dấu – lượng), mã bù 1 (còn gọi là mã đảo) và mã bù 2. Cả ba cách mã hoá này đều dành ra một bit tận cùng bên trái để mã hoá dấu, dấu + được mã hoá bởi bit 0, dấu – được mã hoá bởi bit 1. Số dương trong cả ba cách mã hoá này đều giống nhau, sau bit dấu (bit 0) là biểu diễn nhị phân của số. Đối với số âm thì biểu diễn của ba cách mã hoá này khác nhau. Ví dụ số 19_{10} trong cả ba cách mã hoá đều có mã là 00010011. Trong khi đó số -19_{10} sẽ có mã thuận là 10010011, mã bù 1 là 11101100 và mã bù 2 là 11101101.

- Hệ nhị phân chỉ dùng hai chữ số 0 và 1. Mọi số đều có thể biểu diễn được trong hệ nhị phân. Nhờ vậy, có thể biểu diễn số trong máy tính.
- Biểu diễn số nguyên dương trong máy tính được thực hiện một cách tự nhiên bằng cách đổi biểu diễn số sang hệ nhị phân rồi đưa vào bộ nhớ máy tính. Đối với các số nguyên có dấu, có nhiều kiểu biểu diễn khác nhau.



1. Em hãy đổi các số sau từ hệ thập phân sang hệ nhị phân.

a) 13. b) 155. c) 76.

2. Em hãy đổi các số sau từ hệ nhị phân sang hệ thập phân.

a) 110011. b) 10011011. c) 1001110.

2. CÁC PHÉP TÍNH SỐ HỌC TRONG HỆ NHỊ PHÂN

Các phép tính trong hệ nhị phân sẽ được thực hiện như thế nào?

Trong phần này, chúng ta sẽ làm quen với các phép toán cộng và nhân.

Hoạt động 2 **Phép tính trong hệ nhị phân**

Hãy chuyển các toán hạng của hai phép tính sau ra hệ nhị phân để chuẩn bị kiểm tra kết quả thực hiện các phép toán trong hệ nhị phân. (Ví dụ 3 + 4 = 7 sẽ được chuyển dạng thành 11 + 100 = 111).

a) 27 + 26 = 53. b) 13 × 5 = 65.

a) Bảng cộng và nhân trong hệ nhị phân

Bảng 4.1 là bảng cộng và nhân trong hệ nhị phân tương tự như trong hệ thập phân. Lưu ý là 1 + 1 = 10.

Bảng 4.1. *Bảng cộng và nhân trong hệ nhị phân*

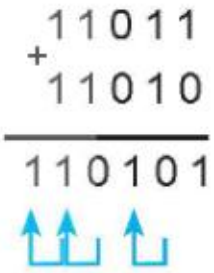
x	y	x + y	x × y
0	0	0	0
0	1	1	0
1	0	1	0
1	1	10	1

b) Cộng hai số nhị phân

Phép cộng cũng được thực hiện tương tự như trong hệ thập phân, thực hiện từ phải sang trái.

Khi phép cộng hai bit có kết quả là 10 thì ghi 0 ở hàng tương ứng dưới tổng và nhớ 1 sang hàng bên trái. Có thể xảy ra trường hợp cộng hai bit 1 mà phải nhớ 1 từ hàng trước chuyển sang thì kết quả sẽ là 11, khi đó chúng ta ghi 1 ở hàng tương ứng dưới tổng và nhớ 1 sang hàng tiếp theo bên trái.

Hình 4.2 minh hoạ phép cộng hai số nhị phân 11011 và 11010.

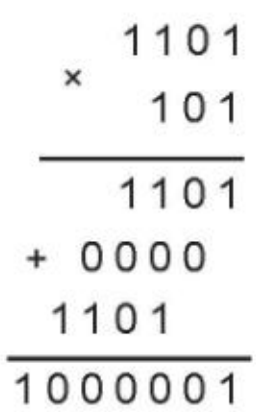


Hình 4.2. *Thực hiện phép cộng*

c) Nhân hai số nhị phân

Phép nhân trong hệ nhị phân cũng được thực hiện tương tự như trong hệ thập phân.

Ta sẽ nhân thừa số thứ nhất lần lượt với từng chữ số của thừa số thứ hai, theo thứ tự từ phải sang trái và đặt kết quả căn phải theo đúng vị trí chữ số của thừa số thứ hai, rồi cộng tất cả lại (Hình 4.3).



Hình 4.3. *Thực hiện phép nhân*

- Các phép tính số học trên hệ nhị phân cũng tương tự như thực hiện trên hệ thập phân.
- Do máy tính biểu diễn số trên hệ nhị phân nên máy tính cần thực hiện các phép tính số học trực tiếp trên hệ nhị phân. Vì vậy, có thể coi tính toán số học trong máy tính là ứng dụng của hệ nhị phân.

Hãy thực hiện các phép tính sau trong hệ nhị phân:

a) $101101 + 11001$. b) 100111×1011.

LUYỆN TẬP

Thực hiện tính toán trên máy tính luôn theo quy trình sau:



Hình 4.4. *Quy trình thực hiện phép tính trên máy tính*

1. Hãy thực hiện các phép tính sau đây theo quy trình Hình 4.4.

a) $125 + 17$. b) $250 + 175$. c) $75 + 112$.

2. Em hãy thực hiện các phép tính sau đây theo quy trình Hình 4.4.

a) 15×6. b) 11×9. c) 125×4.

VẬN DỤNG

1. Em hãy tìm hiểu trên Internet hoặc các tài liệu khác cách đổi phần thập phân của một số trong hệ thập phân sang hệ đếm nhị phân.

2. Em hãy tìm hiểu mã bù 2 với hai nội dung:

a) Mã bù 2 được lập như thế nào?

b) Mã bù 2 được dùng để làm gì?

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

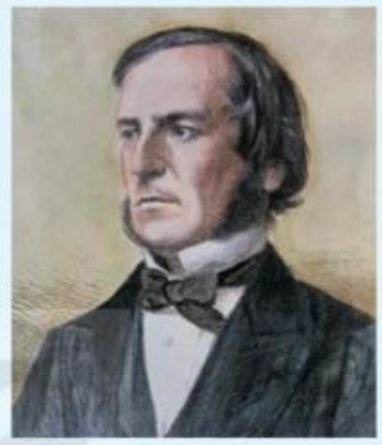
BÀI 5 DỮ LIỆU LÔGIC

SAU BÀI NÀY EM SẼ:

- Biết được giá trị chân lí và các phép toán lôgic AND, OR, XOR, NOT.
- Biết được biểu diễn dữ liệu lôgic.

Việc thiết kế các mạch điện tử của máy tính có liên quan đến lôgic toán mà người có đóng góp nhiều nhất cho ngành Toán học này là nhà toán học người Anh George Boole (1815 – 1864). Ông đã xây dựng nên đại số lôgic, trong đó có các phép toán liên quan đến các yếu tố "đúng", "sai".

Vậy phép toán trên các yếu tố "đúng", "sai" là các phép toán nào?



Hình 5.1. *George Boole*

1. CÁC GIÁ TRỊ CHÂN LÍ VÀ CÁC PHÉP TOÁN LÔGIC

Hoạt động 1 Đúng hay sai?

Dự báo thời tiết cho biết "Ngày mai trời lạnh và có mưa". Thực tế thì không phải khi nào dự báo thời tiết cũng đúng. Có bốn trường hợp có thể xảy ra như Bảng 5.1, trường hợp nào dự báo là đúng? Trường hợp nào dự báo là sai ?

Bảng 5.1. *Các trường hợp dự báo*

Ngày mai trời lạnh	Ngày mai trời có mưa	Dự báo
Đúng	Đúng	?
Đúng	Sai	?
Sai	Đúng	?
Sai	Sai	?

Nếu ngày mai trời lạnh là đúng và ngày mai trời có mưa là đúng thì dự báo thời tiết là đúng. Như vậy chỉ trường hợp thứ nhất là đúng, còn tất cả các trường hợp khác đều sai.

a) Lôgic mệnh đề

Mệnh đề là một khẳng định có tính chất hoặc đúng hoặc sai. Ví dụ "Hà Nội là Thủ đô của Việt Nam" là một mệnh đề đúng, còn "9 là số nguyên tố" là một mệnh đề sai.

Các giá trị "Đúng" hay "Sai" chính là *giá trị chân lí* (giá trị lôgic) của mệnh đề mà nó thể hiện. *Đại lượng lôgic* là đại lượng chỉ nhận giá trị là giá trị lôgic. Để ngắn gọn, người ta thường biểu diễn các giá trị lôgic "Đúng" và "Sai" tương ứng là 1 và 0.

Trong toán học, các biểu thức so sánh đều là các mệnh đề. Ví dụ "3 > 5" là mệnh đề sai; "2 × 3 = 6" là mệnh đề đúng.

Trong các ngôn ngữ lập trình, các biến hay các hàm cũng có thể mang giá trị lôgic.

b) Các phép toán lôgic cơ bản

Từ ví dụ về dự báo thời tiết trên ta thấy, nếu ghép hai mệnh đề bằng liên từ "VÀ" thì được một mệnh đề mới và có thể "tính" được giá trị Đúng/Sai của mệnh đề mới từ giá trị lôgic của hai mệnh đề thành phần. "VÀ" có thể coi là một phép toán lôgic.

Bốn phép toán lôgic quan trọng nhất là các phép toán AND (phép nhân lôgic), OR (phép cộng lôgic), XOR (viết tắt của eXclusive OR - cộng loại trừ lôgic) và NOT (phép phủ định). Giá trị lôgic của mệnh đề là kết quả của các phép toán được cho trong Bảng 5.2:

Bảng 5.2. *Các phép toán lôgic*

p	q	p AND q	p OR q	p XOR q	NOT p
1	1	1	1	0	0
1	0	0	1	1	0
0	1	0	1	1	1
0	0	0	0	0	1

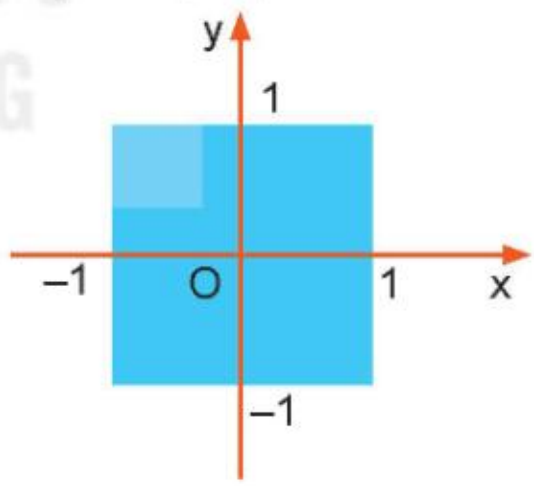
Như vậy, p AND q (đọc là p và q) là mệnh đề có giá trị đúng nếu cả p và q đều đúng; p OR q (đọc là p hoặc q) là mệnh đề có giá trị sai khi cả p và q đều sai; p XOR q là mệnh đề có giá trị sai khi p và q có giá trị như nhau; NOT p (đọc là phủ định của p) là mệnh đề có giá trị sai khi p đúng.

Biểu thức lôgic là một dãy các đại lượng lôgic được nối với nhau bằng các phép toán lôgic, có thể có dấu ngoặc để chỉ định thứ tự ưu tiên thực hiện các phép toán.

Ví dụ về các biểu thức lôgic:

- p AND (q OR r).
- Giả sử p là mệnh đề ($|x| \le 1$), q là mệnh đề ($|y| \le 1$). Khi đó p AND q chính là tập tất cả các điểm có toạ độ (x, y) thuộc hình vuông được biểu diễn trong Hình 5.2.

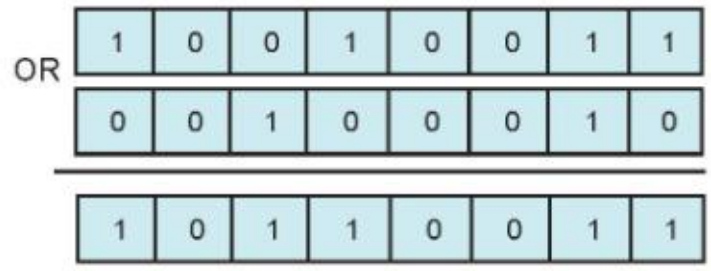
Trong một biểu thức lôgic, phép toán đặt trong dấu ngoặc có độ ưu tiên cao nhất. Nếu không có dấu ngoặc thì phép phủ định được thực hiện trước. Các phép toán lôgic AND và OR có độ ưu tiên ngang nhau, được thực hiện tuần tự từ trái sang phải.



Hình 5.2. *Hình vuông trong mặt phẳng toạ độ*

Kết quả của các phép toán trong Bảng 5.2 chỉ phụ thuộc vào giá trị lôgic (tương ứng với giá trị 1 hoặc 0) của các mệnh đề mà không phụ thuộc vào nội dung câu chữ cụ thể của chúng. Do vậy, các phép toán lôgic nêu trên có thể coi là các phép toán lôgic trên các bit (gọi tắt là các phép toán trên bit). Ví dụ, 1 AND 0 = 0; 1 OR 0 = 1,...

Các phép toán lôgic cũng được mở rộng cho các dãy bit. Ví dụ, phép cộng lôgic 2 byte sẽ cộng từng cặp bit tương ứng của 2 byte đó như trong ví dụ Hình 5.3.



Hình 5.3. *Cộng lôgic hai dãy bit*

- Các giá trị lôgic gồm "Đúng" và "Sai", được thể hiện tương ứng bởi 1 và 0 trong đại số lôgic.
- p AND q chỉ đúng khi cả p và q đều đúng.
- p OR q là đúng khi ít nhất một trong p hoặc q đúng.
- p XOR q chỉ đúng khi p và q có giá trị khác nhau.
- NOT p cho giá trị đúng nếu p sai và cho giá trị sai nếu p đúng.

1. Cho mệnh đề p là "Hùng khéo tay" , q là "Hùng chăm chỉ". Em hãy diễn giải bằng lời các mệnh đề "p AND NOT q"; "p OR q" và đề xuất một hoàn cảnh thích hợp để phát biểu các mệnh đề đó. Ví dụ, mệnh đề "NOT p" nghĩa là "Hùng không khéo tay".

2. Cho Bảng 5.3 như sau. Phương án nào có kết quả sai?

Bảng 5.3. *Giá trị của biểu thức lôgic p AND NOT q*

Phương án	p	q	p AND NOT q
A.	0	1	0
B.	1	0	1
C.	0	0	1
D.	1	1	0

2. BIỂU DIỄN DỮ LIỆU LÔGIC

Trong cuộc sống, những sự vật/hiện tượng có hai trạng thái đối lập như "sáng/tối", "bật/tắt","có/không"... đều có thể coi là thể hiện của hai đại lượng lôgic "Đúng/Sai".

Trong Tin học, chỉ cần 1 bit với các giá trị 1 hoặc 0 là đủ để biểu diễn dữ liệu lôgic, với quy ước 1 là "Đúng", 0 là "Sai". Tuy nhiên, một số ngôn ngữ lập trình có quy ước riêng, không mã hoá các đại lượng lôgic bởi một bit. Chẳng hạn, ngôn ngữ lập trình Python coi số 0 thể hiện giá trị "Sai" còn một số bất kì khác 0 thể hiện giá trị "Đúng". Trong tiếng Anh, đúng là True, sai là False nên có ngôn ngữ lập trình dùng ngay hai kí tự "T" và "F" để biểu diễn dữ liệu lôgic.

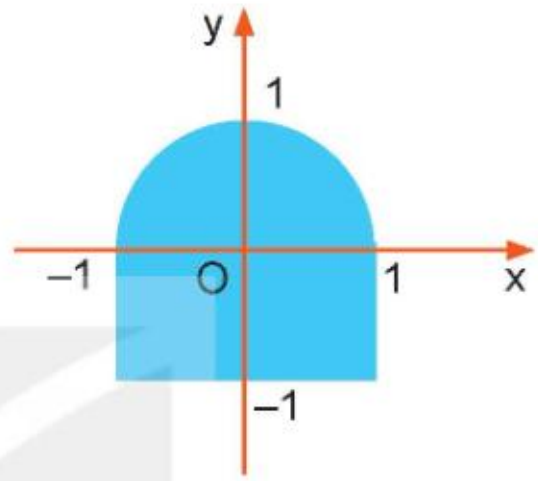
KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

- Chỉ cần 1 bit để biểu diễn dữ liệu lôgic, bit có giá trị bằng 1 cho giá trị đúng và bit có giá trị bằng 0 cho giá trị sai.
- Trên thực tế, có thể biểu diễn dữ liệu lôgic theo các cách khác miễn là tạo ra hai trạng thái đối lập.

Em hãy tìm một vài ví dụ về thông tin có hai giá trị đối lập, có thể quy về kiểu lôgic.

LUYỆN TẬP

1. Một hình tạo bởi nửa hình tròn đơn vị và một hình chữ nhật trong mặt phẳng toạ độ như minh hoạ trong Hình 5.4. Hãy viết biểu thức lôgic mô tả hình vẽ.



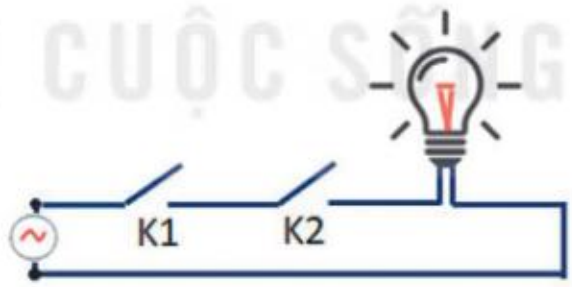
Hình 5.4

2. Tại sao p AND NOT p luôn luôn bằng 0, còn p OR NOT p luôn luôn bằng 1?

VẬN DỤNG

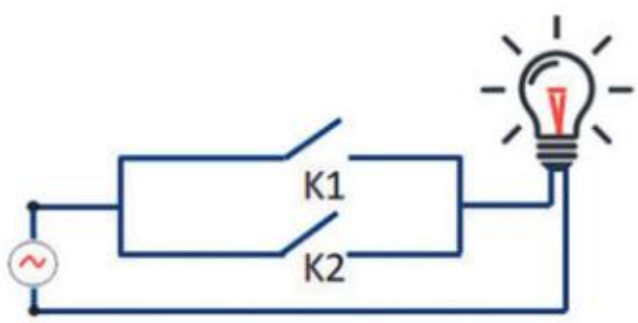
Trong mạch điện có các công tắc và bóng đèn, ta quy ước các công tắc đóng thể hiện giá trị lôgic 1 và công tắc mở thể hiện giá trị lôgic 0; đèn sáng thể hiện giá trị lôgic 1 còn đèn tắt thể hiện giá trị lôgic 0.

a) Cho một mạch điện có hai công tắc K1 và K2 nối với một bóng đèn như Hình 5.5. Giá trị lôgic của đèn được tính qua giá trị lôgic của các công tắc K1 và K2 như thế nào?



Hình 5.5

b) Cho mạch điện mắc như Hình 5.6. Giá trị lôgic của đèn được tính qua giá trị lôgic của các công tắc K1 và K2 như thế nào?



Hình 5.6

BÀI 6 DỮ LIỆU ÂM THANH VÀ HÌNH ẢNH

SAU BÀI NÀY EM SẼ:

- Giải thích được việc số hoá âm thanh.
- Giải thích được việc số hoá hình ảnh.

Trong tin học, âm thanh và hình ảnh là hai trong các dạng thông tin quan trọng của đa phương tiện (multimedia) mà con người có thể tiếp nhận qua các giác quan. Những thông tin này được lưu trong máy tính như thế nào?

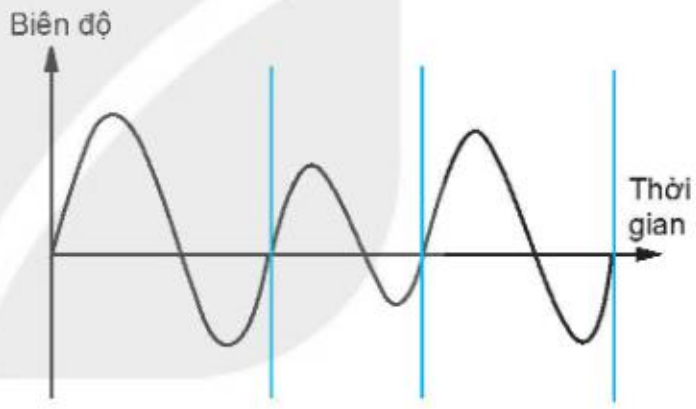
1. BIỂU DIỄN ÂM THANH

Hoạt động 1 Bản chất của âm thanh

Âm thanh được truyền đi bằng sóng âm.

Trên thực tế, sóng âm có dạng hình sin như Hình 6.1, trục hoành là trục thời gian, trục tung thể hiện biên độ của tín hiệu. Tín hiệu âm thanh có đồ thị liên tục như vậy được gọi là tín hiệu âm thanh tương tự (analog).

Để có thể xử lí một cách hiệu quả, âm thanh trong máy tính cần được lưu trữ dưới dạng số hoá (âm thanh số). Vậy âm thanh số được tạo ra như thế nào?



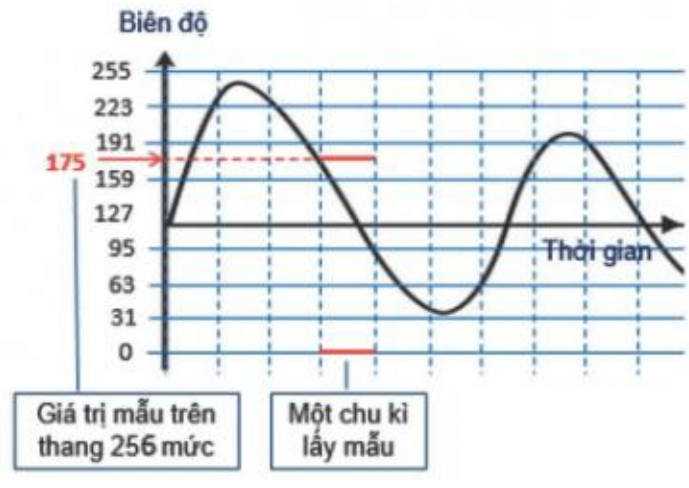
Hình 6.1. *Đồ thị của sóng âm*

a) Số hoá âm thanh

Phương pháp cơ bản số hoá âm thanh là điều chế mã xung (Pulse Code Modulation, viết tắt là PCM) được thực hiện theo các bước như sau:

Bước 1: Lấy mẫu. Lấy giá trị biên độ tín hiệu ở những thời điểm rời rạc, cách đều nhau. Khoảng thời gian giữa hai lần lấy mẫu gọi là chu kì lấy mẫu.

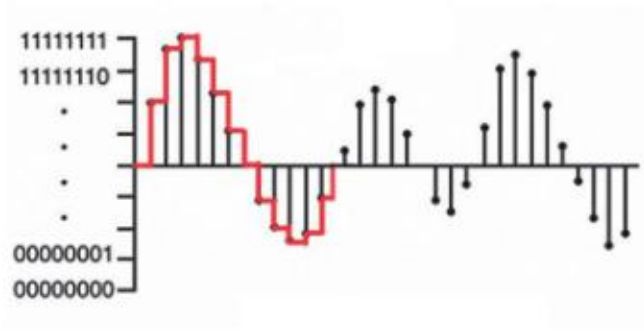
Bước 2: Biểu diễn giá trị mẫu. Chọn một thang biểu diễn giá trị mẫu, gồm một số mức đều nhau, ví dụ 256 mức. Biên độ tín hiệu được quy đổi theo tỉ lệ trên thang lấy mẫu và làm tròn. Ví dụ với thang 256 (2^8) mức thì giá trị mẫu sẽ nhận trong khoảng từ 0 tới 255, hay từ 00000000 đến 11111111 trong hệ nhị phân, có thể ghi trong một byte.



Hình 6.2. *Chu kì lấy mẫu và thang giá trị mẫu*

Bước 3: Biểu diễn âm thanh. Dãy giá trị biên độ đã quy đổi tại các điểm lấy mẫu được ghi lại làm biểu diễn âm thanh, ví dụ 128, 192, 242, 255, 235, 210,...

Như vậy, đồ thị liên tục dạng hình sin của sóng âm (Hình 6.2) được xấp xỉ bằng đồ thị hình bậc thang (đường màu đỏ trên Hình 6.3). Trong đó, giá trị biên độ tín hiệu được coi là không thay đổi trong một chu kì lấy mẫu.



Hình 6.3. *Xấp xỉ tín hiệu tương tự bằng tín hiệu số*

Để đồ thị đường bậc thang bám sát hơn với đồ thị của tín hiệu gốc, chu kì lấy mẫu cần phải nhỏ và dùng thang lấy mẫu chi tiết hơn. Khi đó, khối lượng dữ liệu âm thanh cho một đơn vị thời gian tăng thêm nhưng âm thanh số sẽ trung thực hơn. Để số hoá âm thanh, người ta dùng các thiết bị ghi âm cài đặt sẵn phần mềm số hoá, trong đó có các mạch điện tử chuyển tín hiệu tương tự sang tín hiệu số (Analog to Digital Converter – ADC).

Số bit cần thiết để biểu diễn được một giây âm thanh gọi là tốc độ bit (bit–rate).

Các thiết bị âm thanh số cần có mạch điện tử gọi là DAC (Digital to Analog Converter) có chức năng tạo lại tín hiệu tương tự từ tín hiệu số để phát ra loa hoặc tai nghe.

b) Các định dạng lưu trữ âm thanh

Cách số hoá âm thanh theo phương pháp PCM cho chất lượng âm thanh khá trung thực nhưng kích thước tệp lớn. Do đó, người ta đã tìm các phương pháp nhằm giảm kích thước tệp. Có hai phương pháp chính:

Phương pháp thứ nhất là nén dữ liệu nhưng không làm giảm chất lượng âm thanh, tạo nên định dạng âm thanh không mất mát (lossless).

Phương pháp thứ hai là bỏ bớt một phần thông tin âm thanh, nhưng vẫn đảm bảo chất lượng chấp nhận được. Một trong các định dạng thông dụng nhất là Mp3, có thể làm giảm kích thước tệp khoảng 10 lần so với định dạng wav của PCM (là định dạng thường được dùng trong các ứng dụng trên Windows) mà chất lượng âm thanh giảm không đáng kể.

- Âm thanh được số hoá bằng cách lấy mẫu biên độ tín hiệu của sóng âm theo chu kì lấy mẫu. Chu kì lấy mẫu càng nhỏ, thang lấy mẫu càng chi tiết, âm thanh càng trung thực nhưng cần nhiều không gian lưu trữ.
- Có nhiều định dạng âm thanh khác nhau giúp giảm bớt không gian lưu trữ trên cơ sở nén không mất mát (lossless) hoặc giảm chất lượng âm thanh ở mức chấp nhận được.



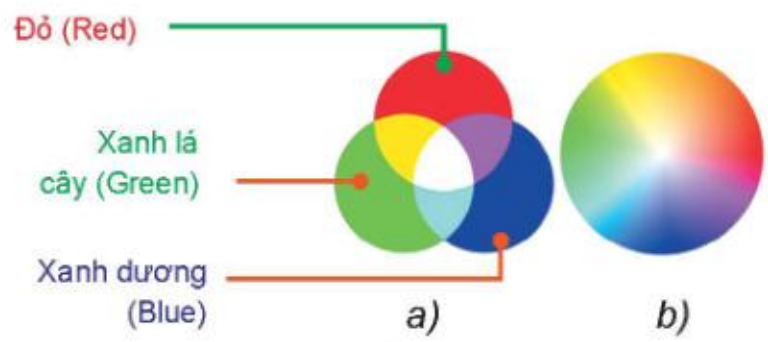
1. Khi số hoá âm thanh, chu kì lấy mẫu tăng thì lượng thông tin lưu trữ tăng hay giảm?
2. Tốc độ bit 128 Kb/s (còn được viết là Kbps) nghĩa là gì?

2. BIỂU DIỄN HÌNH ẢNH

Hoạt động 2 **Tạo màu như thế nào?**

Hãy đọc để biết màu trên màn hình máy tính hay ti vi được tạo như thế nào.

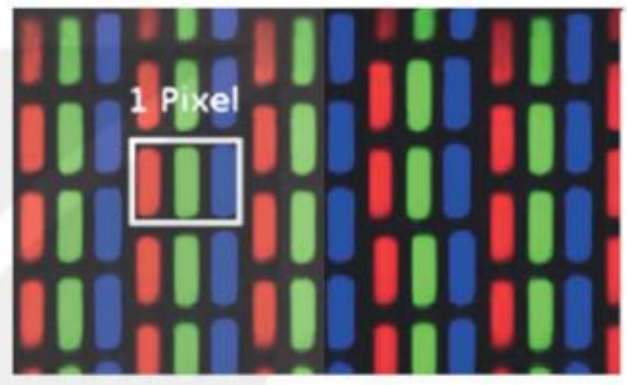
Hệ ba màu cơ bản đỏ, xanh lá cây – lục và xanh dương – lam (Hình 6.4.a) phối hợp theo các "liều lượng" khác nhau để tạo ra tất cả các màu (Hình 6.4.b) được gọi là hệ màu RGB (viết tắt từ Red - Green – Blue).



Hình 6.4. *Hệ màu RGB*

Màn hình LCD hay OLED của máy tính hay ti vi ngày nay dùng ba diode cạnh nhau phát ba màu theo hệ RGB để tạo thành một điểm ảnh (Hình 6.5).

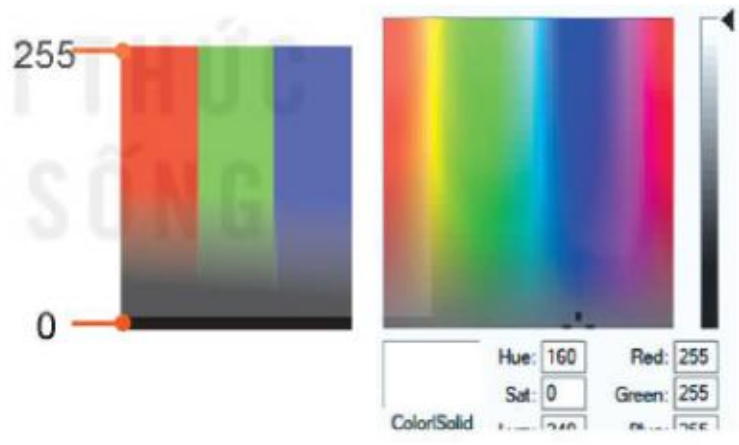
Biểu diễn tự nhiên nhất của hình ảnh số chính là tập hợp thông tin màu của các điểm ảnh. Điểm ảnh trong tiếng Anh gọi là pixel (picture element – phần tử ảnh). Ảnh lưu thông tin theo từng điểm ảnh gọi là ảnh bitmap.



Hình 6.5. *Một số điểm ảnh trên màn hình LCD*

Số bit cần thiết để mã hoá thông tin màu của một điểm ảnh trong tiếng Anh là "bit depth" được hiểu là độ sâu màu. Độ sâu màu càng lớn thì màu sắc của ảnh càng tinh tế.

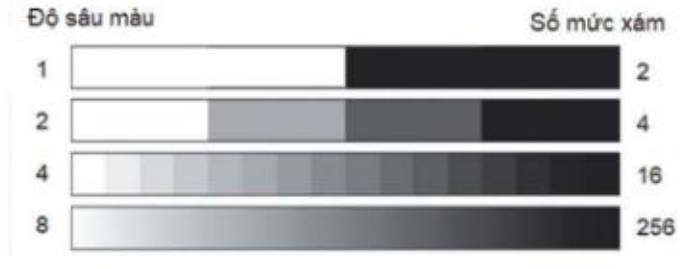
Ảnh màu. Ảnh màu thông dụng có độ sâu màu 24 bit, mỗi màu cơ bản được mã hoá bởi 8 bit, tương ứng với 256 sắc độ khác nhau. Mã màu $255_{10} = 11111111_2$ có sắc độ đậm nhất. Mã màu càng nhỏ thì độ màu giảm đi, đến $0_{10} = 00000000_2$ là mất màu, trở thành đen hoàn toàn.



Hình 6.6. *Mẫu màu ảnh màu*

Màu trắng có mã (255, 255, 255), màu đỏ có mã (255, 0, 0), màu xanh lá cây có mã (0, 255, 0), màu xanh dương có mã (0, 0, 255) còn màu đen có mã (0, 0, 0). Tổng cộng có 2563 tổ hợp tạo ra khoảng 16,7 triệu sắc độ màu khác nhau.

Ảnh xám và ảnh đen trắng. Ngoài ảnh màu, người ta cũng dùng ảnh xám, trong tiếng Anh gọi là grayscale, với nhiều mức đậm nhạt khác nhau, phổ biến là 256 mức.



Hình 6.7. *Số mức xám phụ thuộc vào độ sâu màu*

Ảnh đen trắng chỉ có hai sắc độ màu là đen và trắng, tương đương với độ sâu màu là 1.



Hình 6.8. *Ảnh màu và ảnh xám tương ứng*

Biểu diễn ảnh bitmap

Ảnh bitmap nguyên gốc được lưu vào các tệp có phần mở rộng là .bmp.

Lưu ảnh theo thông tin của từng điểm ảnh rất tốn bộ nhớ. Có hai cách giải quyết vấn đề này: hoặc nén tệp, lúc xem thì giải nén mà không gây mất mát chất lượng; hoặc giảm bớt một phần thông tin, chịu mất mát một phần chất lượng. Sau đây là một số định dạng ảnh phổ biến thường được dùng trong các ứng dụng trên web:

- ". jpeg": là ảnh đã được nén có mất mát chất lượng nhưng có tệp dung lượng khá nhỏ, tốn ít thời gian truyền và không gian lưu trữ.
- ".png": có độ nén tốt, không mất mát chất lượng, có thể có nền trong suốt để chồng ảnh mà không che ảnh dưới nền.

Việc số hoá hình ảnh có thể thực hiện bằng các thiết bị số như máy ảnh số, máy quét, điện thoại thông minh,...

- Ảnh màu thông dụng trong máy tính là ảnh theo hệ RGB. Mỗi điểm ảnh được mã hoá bởi 24 bit, mỗi màu cơ bản sử dụng 8 bit để mô tả sắc độ từ 0 (đen) đến 255 (màu đậm nhất).
- Ảnh xám thông dụng có độ sâu màu 8 bit, cho 256 sắc độ xám khác nhau.
- Có nhiều định dạng tệp hình ảnh khác nhau. Mỗi định dạng có mức lưu trữ và hiệu ứng thể hiện khác nhau.

1. Hình ảnh hiển thị trên màn hình máy tính sử dụng hệ màu nào?

A. Đỏ – Lam – Vàng (RBY) .

B. Đỏ – Lục – Lam (RGB).

C. Xanh lơ – Hồng sẫm – Vàng (CMY).

D. Xanh lơ – Hồng sẫm – Vàng – Đen (CMYK).

2. Điều nào sai khi nói về ảnh định dạng ".jpeg"?

A. Kích thước tệp nhỏ, giảm được chi phí lưu trữ.

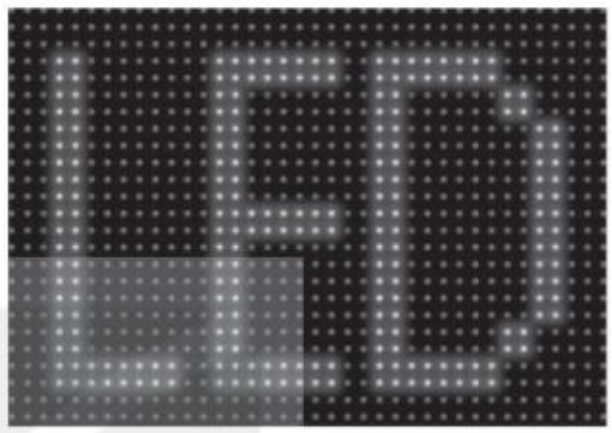
B. Kích thước tệp nhỏ nên khi dùng với web tải về nhanh hơn.

C. Tuy kích thước giảm đáng kể so với ảnh bitmap nhưng chất lượng ảnh đủ tốt.

D. Công nghệ web không dùng được với các định dạng ảnh khác với ".jpeg".

LUYỆN TẬP

1. Có một bảng quảng cáo LED như trong Hình 6.9. Nếu coi mỗi vị trí đặt bóng LED tương ứng với một điểm ảnh thì độ sâu màu của ảnh này là bao nhiêu?



Hình 6.9. *Bảng quảng cáo đèn Led*

2. Nhạc CD có tốc độ bit là 1411 Kb/s. Hãy ước tính một đĩa nhạc CD có dung lượng 650 MB có thể nghe được bao lâu.

VẬN DỤNG

1. Có nhiều website cung cấp dịch vụ nhạc số. Một số trang cho phép tải nhạc về máy. Khi tải nhạc thường có gợi ý lựa chọn 128 Kbps, 320 Kbps hay Lossless (Hình 6.10). Em hãy giải thích ý nghĩa của những lựa chọn đó.



Hình 6.10. *Lựa chọn tải nhạc*

2. Sử dụng phần mềm Paint có sẵn trong Windows mở một hình, sau đó chọn lệnh **Save As**. Phần mềm sẽ hỏi lưu ảnh dưới định dạng nào trong các định dạng ".png", ".jpeg" , ".bmp" và ".gif". Hãy lưu tệp với bốn định dạng trong cùng một thư mục và so sánh độ lớn của các tệp.

BÀI 7 THỰC HÀNH SỬ DỤNG THIẾT BỊ SỐ THÔNG DỤNG

SAU BÀI NÀY EM SẼ:

- Biết được thiết bị số cá nhân thông dụng thường có những gì.
- Biết được một số tính năng tiêu biểu của thiết bị số cá nhân thông dụng.
- Khai thác sử dụng một số ứng dụng và dữ liệu trên các thiết bị di động như máy tính bảng (tablet), điện thoại thông minh (smartphone).

Em có biết các thiết bị có trong hình dưới đây có tên gọi là gì không?



Hình 7.1. *Một số thiết bị số hỗ trợ cá nhân*

1. TRỢ THỦ SỐ CÁ NHÂN

Hoạt động 1 **Tìm hiểu thiết bị trợ thủ số cá nhân**

Sự phát triển của Công nghệ thông tin và vi điện tử đã dẫn tới sự ra đời của hàng loạt các thiết bị số hỗ trợ cá nhân, còn gọi là trợ thủ số cá nhân. Các em hãy liệt kê một số thiết bị có thể là trợ thủ số cá nhân và các ứng dụng tiêu biểu đi kèm.

Trợ thủ số cá nhân (Personal Digital Assistant – PDA) là các thiết bị số trong đó tích hợp một số chức năng hữu ích cho người dùng trong đời sống hằng ngày. Ban đầu, PDA thường có một số chức năng cơ bản đó là sổ lịch để ghi lịch công tác, đồng hồ để xem thời gian có thể đặt giờ báo thức, sổ danh bạ ghi địa chỉ, số điện thoại liên hệ, danh sách việc cần làm, sổ ghi nhớ (notes), máy tính bỏ túi,...



Hình 7.2. *Màn hình PDA*

Tuy nhiên, ngày nay PDA đã tích hợp thêm nhiều chức năng hữu ích khác như nghe nhạc, ghi âm, xem phim, gọi điện thoại, chụp ảnh, quay phim, tìm đường, điều khiển các thiết bị điện tử từ xa và có các cổng giao tiếp như USB, các loại thẻ nhớ, kết nối bluetooth, kết nối wifi. Trong số các PDA phổ biến có thể kể tới là điện thoại thông minh, máy tính bảng, đồng hồ thông minh, máy đọc sách,... Phần lớn các PDA dạng điện thoại di động và máy tính bảng hiện nay (Hình 7.2) đều chạy trên hai hệ điều hành phổ biến là iOS của hãng Apple và Android của hãng Google.

Trợ thủ số cá nhân hay PDA là thiết bị số tích hợp nhiều chức năng và phần mềm ứng dụng hữu ích cho người dùng với đặc điểm quan trọng là nhỏ gọn, có khả năng kết nối mạng.

Kết nối nào không phải là kết nối phổ biến trên các PDA hiện nay?

A. Wifi. B. Bluetooth. C. Hồng ngoại. D. USB.

2. THỰC HÀNH SỬ DỤNG THIẾT BỊ SỐ CÁ NHÂN

Về cơ bản điện thoại thông minh và máy tính bảng có giao diện và cách thức sử dụng tương tự như nhau. Trong phần này, các em sẽ tìm hiểu về cách thức sử dụng điện thoại thông minh.



iPhone *Điện thoại Android* *iPad* *Máy tính bảng Android*

Hình 7.3. *Các thiết bị thông minh*

Nhiệm vụ 1. Quan sát để nhận biết các nút bấm của điện thoại thông minh. Khởi động điện thoại thông minh, tìm hiểu hệ điều hành đang sử dụng và các chế độ của màn hình.

Hướng dẫn.

Bước 1: Quan sát điện thoại thông minh (Hình 7.4). Phía hai bên thân máy thường có một số nút bấm sau:

– Nút khoá: dùng để bật máy hoặc tắt màn hình.



Hình 7.4. *Các nút trên điện thoại thông minh*

– Nút tăng/giảm âm lượng. Một số máy có nút bật/tắt âm thanh.

Bước 2: Bấm nút khoá để khởi động điện thoại di động. Quan sát và nhận biết hệ điều hành trên điện thoại đang dùng.

Nhiệm vụ 2. Làm quen với màn hình làm việc và các chức năng trên màn hình của điện thoại thông minh.

Hướng dẫn.

Bước 1: Quan sát màn hình làm việc của điện thoại thông minh. Màn hình chính có một số thông tin như sau:

– Thanh trạng thái: hiển thị tình trạng kết nối, thời gian hiện tại, tỉ lệ % pin còn lại,...

– Các biểu tượng ứng dụng (application – gọi tắt app) cài trên máy. Các ứng dụng được nhà sản xuất cài đặt sẵn hoặc do người dùng cài đều được liệt kê ở đây. Với kích thước hữu hạn của màn hình chính, sau một thời gian, màn hình sẽ hết chỗ, khi đó sẽ có thêm các trang để chứa các biểu tượng của các ứng dụng mới.

– Thanh truy cập nhanh chứa các ứng dụng hay dùng, sẽ được lặp lại ở cuối tất cả các trang của màn hình chính.



Hình 7.5. *Màn hình chính*

– Thanh điều hướng (navigation bar): Hầu hết các thiết bị sử dụng hệ điều hành Android không trang bị nút Home vật lí, thay vào đó là thanh điều hướng với các nút ảo ở dưới màn hình cảm ứng, trong đó có hai nút cảm ứng rất quan trọng là nút **Quay lại** (Back) và nút hiển thị danh sách tất cả các ứng dụng đang chạy là nút **Tổng quan** (Overview).

Bước 2: Thực hiện các thao tác sau và nhận xét.

– Bấm nút **Home**.

– Vuốt màn hình cảm ứng theo các chiều trái, phải, lên.

– Bấm vào phím **Quay lại** và phím **Tổng quan** (nếu dùng điện thoại có hệ điều hành Android).

Nhiệm vụ 3. Quan sát các biểu tượng điện thoại thông minh. Tìm hiểu thêm về các chức năng và các ứng dụng trên điện thoại thông minh.

Hướng dẫn.

Bước 1: Quan sát các biểu tượng Hình 7.6 và cho biết những ứng dụng mà em biết.

Các chức năng và các ứng dụng có sẵn hoặc được cài đặt sau này đều được thể hiện bởi các biểu tượng trên màn hình.

Một số chức năng thiết yếu của điện thoại là: Gọi điện, Nhắn tin, Quản lí danh bạ.

Một số ứng dụng thường dùng có sẵn trên điện thoại là: Chụp ảnh và quản lí kho ảnh, Trình duyệt, Email, Máy tính, Lịch, Hẹn giờ, Báo thức, Chợ phần mềm,...

Người sử dụng có thể cài đặt thêm các ứng dụng khác lấy từ chợ phần mềm trên mạng xuống như các chương trình hỗ trợ học tập trực tuyến Zoom, MS Teams, Google Meets,..., các dịch vụ lưu trữ đám mây như OneDrive, Google Drive,...

Bước 2:

– Mở một ứng dụng hỗ trợ học tập trực tuyến như Zoom, Google Meets, tham gia buổi học trực tuyến do thầy/cô giáo thiết lập.

– Mở và đăng kí dịch vụ lưu trữ đám mây như OneDrive, Google Drive.



Hình 7.6. *Một số ứng dụng trên điện thoại*

Nhiệm vụ 4. Hãy tìm xem trên điện thoại của bạn một ứng dụng quản lí tệp. Mở một tệp ảnh bất kì để xem thông tin, xoá tệp trên máy.

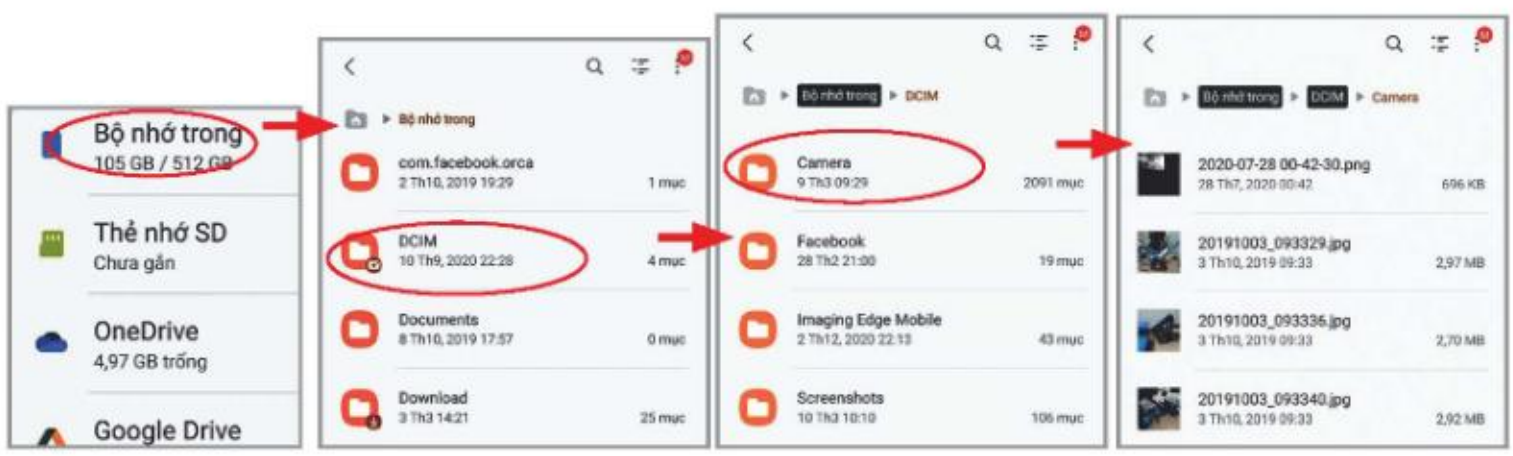
Hướng dẫn.

Bước 1: Mỗi dòng điện thoại có thể dùng ứng dụng quản lí tệp có giao diện riêng, hãy tìm xem trên điện thoại của bạn có một ứng dụng nào có tên như trong danh sách sau File Manager, My File, Files, File Master, FileApp, File Explorer, File Browser hoặc tương tự. Nếu có thì có nhiều khả năng đó là các ứng dụng quản lí tệp trên điện thoại.

Hệ thống tệp tổ chức theo cấu trúc phân cấp tương tự như trên máy tính. Các không gian nhớ của máy, của thẻ nhớ hay dịch vụ lưu trữ đám mây tương tự như các ổ đĩa trong máy tính. Cấu trúc thư mục cũng tương tự như trên máy tính, bắt đầu từ thư mục gốc, trong mỗi thư mục lại có thể mở nhiều thư mục, các tệp được đặt ở trong các thư mục.

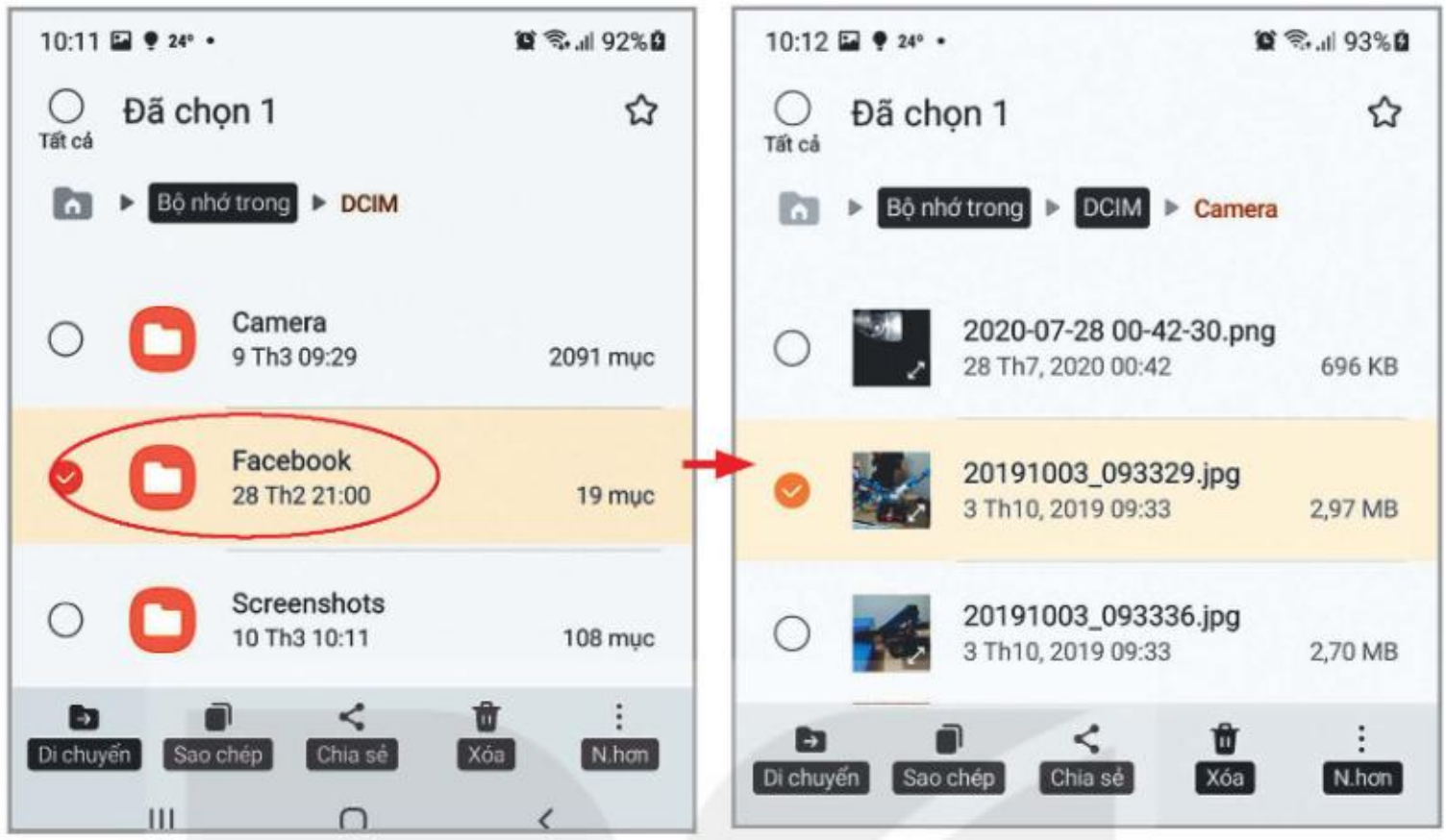
Bước 2: Thao tác mở, chọn, xem, sao chép, di chuyển các tệp tin trên điện thoại.

Ví dụ, truy cập vào thư mục ảnh chụp ở bộ nhớ trong để xem các tệp ảnh.



Hình 7.7. *Mở để xem các tệp ảnh chụp*

Nếu chọn thư mục hay tệp bằng cách chạm và giữ lâu một chút, sẽ xuất hiện các nút điều khiển để ta có thể di chuyển, sao chép, chia sẻ hoặc xoá thư mục hay tệp.



Hình 7.8. *Chọn tệp để di chuyển, sao chép, chia sẻ và xoá*

LUYỆN TẬP

1. Điện thoại thông minh khác với điện thoại di động thường ở điểm nào?

A. Có danh bạ.

B. Có thể nhắn tin.

C. Có thể kết nối Internet.

D. Có hệ điều hành và có thể chạy được một số ứng dụng.

2. Em hãy chụp một tấm ảnh bằng điện thoại thông minh. Sau đó vào hệ thống quản lí tệp để tìm đến thư mục ảnh chứa ảnh đã chụp. Em hãy mở xem ảnh đó, sau đó xoá đi.

VẬN DỤNG

1. Em hãy kết nối điện thoại thông minh với máy tính để sao chép ảnh chụp vào thư mục trên máy tính.

2. Hãy thực hành lưu trữ các ảnh đó trên dịch vụ lưu trữ đám mây.

3. Hãy thực hành gửi các ảnh này qua phần mềm hỗ trợ học trực tuyến như Zoom.

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

Chủ đề
2

BÀI 8

Mạng máy tính và Internet

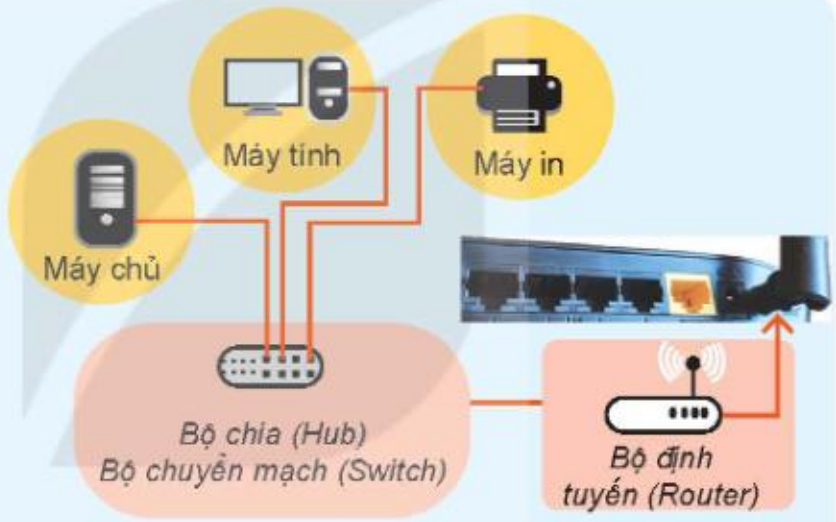
MẠNG MÁY TÍNH TRONG CUỘC SỐNG HIỆN ĐẠI

SAU BÀI NÀY EM SẼ:

- Hiểu được sự khác biệt giữa mạng LAN và Internet.
- Trình bày được những thay đổi về chất lượng cuộc sống, phương thức học tập và làm việc trong xã hội khi mạng máy tính được sử dụng rộng rãi.
- Nêu được một số công nghệ dựa trên Internet như dịch vụ điện toán đám mây hay kết nối vạn vật (IoT).

Hình 8.1 là mô hình của một mạng máy tính; máy chủ, máy tính để bàn và máy in là các thiết bị đầu cuối, còn bộ chia (HUB), bộ chuyển mạch (Switch), bộ định tuyến (Router) là các thiết bị kết nối.

Bộ định tuyến có một số cổng để cắm cáp mạng, có phân biệt các cổng LAN và cổng INTERNET. Tại sao phải phân biệt như vậy?



Hình 8.1. *Mô hình của mạng cục bộ*

1. MẠNG LAN VÀ INTERNET

Hoạt động 1 **Phân biệt Internet với mạng LAN**

1. Phạm vi sử dụng của Internet là:
A. Chỉ trong gia đình. B. Chỉ trong một cơ quan. C. Toàn cầu.

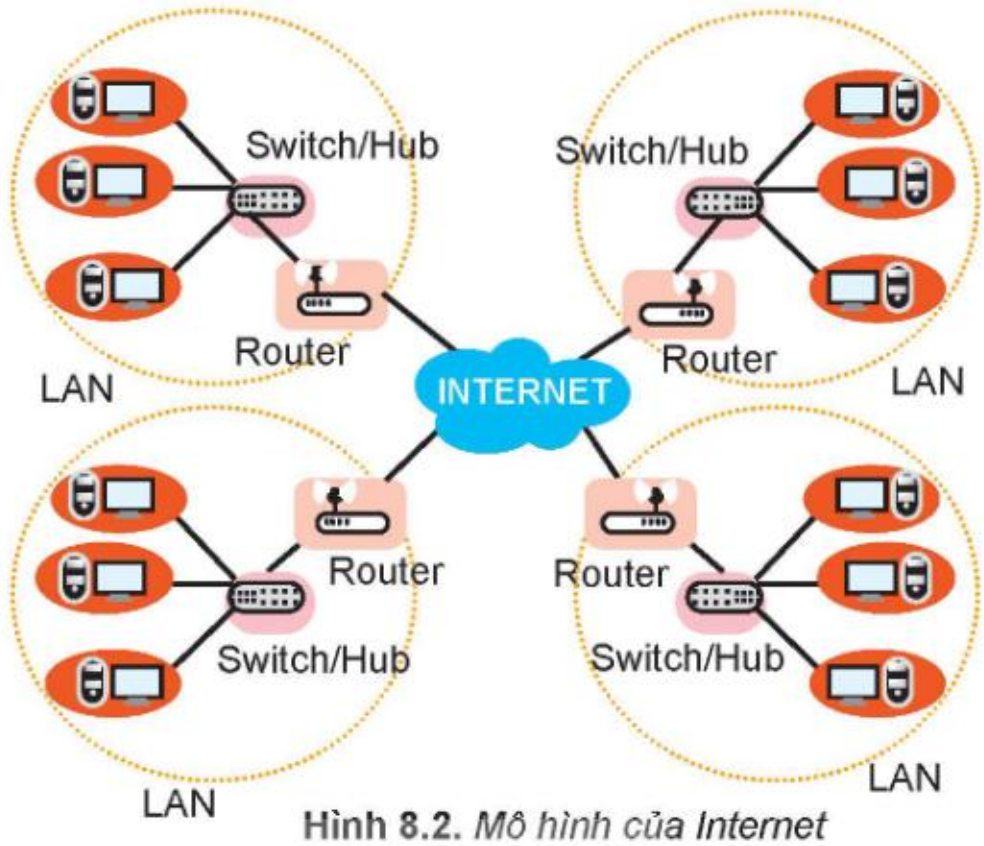
2. Điện thoại thông minh được kết nối với Internet bằng cách nào?
A. Qua dịch vụ 3G, 4G, 5G. B. Kết nối gián tiếp qua wifi. C. Cả A và B.

Theo phạm vi địa lí, các mạng máy tính có thể chia thành hai loại là mạng cục bộ (Local Area Network, viết tắt là LAN) và mạng diện rộng (Wide Area Network, viết tắt là WAN).

Mạng LAN có phạm vi địa lí nhỏ như gia đình, trường học hay công ty. Mạng diện rộng được hình thành bằng cách liên kết các LAN hay các máy tính đơn lẻ.

Internet là mạng diện rộng có quy mô toàn cầu. Các cơ quan, trường học, gia đình có thể lắp đặt mạng cục bộ, sau đó đăng kí với nhà cung cấp dịch vụ Internet để có thể truy cập Internet.

Trong mạng máy tính, bộ chia hay bộ chuyển mạch chỉ chuyển tiếp dữ liệu trong nội bộ LAN. Còn bộ định tuyến dùng để kết nối các LAN với nhau. Nguyên lí hoạt động của bộ định tuyến là nếu địa chỉ nơi nhận của dữ liệu không có trong LAN thì nó gửi theo cổng Internet.



Hình 8.2. *Mô hình của Internet*

LAN là sở hữu của một cá nhân, gia đình hay một cơ quan, tổ chức xác định; tuy nhiên Internet là không của riêng ai theo nghĩa nó hoạt động mà không có một cơ quan quản lí tập trung nào, nhưng có một vài tổ chức phi lợi nhuận quốc tế điều phối tài nguyên (như địa chỉ, tên miền) và tiêu chuẩn hoá các nền tảng kĩ thuật.

	Mạng cục bộ	Internet
Phạm vi, quy mô	Cơ quan, gia đình.	Toàn cầu.
Cách kết nối	Trực tiếp trong mạng qua thiết bị kết nối như Hub, Switch, Wifi.	Kết nối qua các Router thông qua các nhà cung cấp dịch vụ kết nối.
Sở hữu	Có chủ sở hữu.	Không có chủ sở hữu.

Để kết nối điện thoại, máy tính hay ti vi với Internet, phải đăng kí qua một nhà cung cấp dịch vụ Internet như Viettel, FPT, VNPT,.... Em có biết nhà cung cấp dịch vụ Internet nào khác không? Em dùng dịch vụ của nhà cung cấp nào?

2. VAI TRÒ CỦA INTERNET

Internet ngày càng được sử dụng rộng rãi trên toàn thế giới và có ảnh hưởng đến hầu hết các lĩnh vực hoạt động của con người.

Trong giao tiếp cộng đồng: Internet đã thay đổi cách mọi người tương tác với nhau. Ta có thể liên lạc, kết nối thông tin, chia sẻ kiến thức, kinh nghiệm, đưa ra ý tưởng và nhận phản hồi ngay lập tức vào bất cứ lúc nào, ở bất cứ đâu mà không cần tiếp xúc trực tiếp.

Trong giáo dục: Internet đã giúp hoạt động giáo dục hiệu quả hơn. Internet là một nguồn thông tin khổng lồ về mọi lĩnh vực. Ta có thể sử dụng Internet để truy cập thư viện các bài giảng, các bách khoa toàn thư,.... Sự xuất hiện của các nền tảng học tập trực tuyến giúp người học có thể học từ xa mọi lúc, mọi nơi, biến mong muốn không ngừng học tập nâng cao hiểu biết, học tập suốt đời trở thành hiện thực.

- Internet là một kho tri thức khổng lồ thường xuyên được cập nhật, có thể truy cập bất cứ ở đâu, bất cứ lúc nào.
- Internet đã giúp con người có thể kết nối và giao tiếp với nhau một cách dễ dàng và tiện lợi.
- Internet đã có ảnh hưởng sâu sắc tới mọi lĩnh vực của đời sống xã hội, làm thay đổi cách thức làm việc, học tập và giao tiếp với nhau.



1. Em hãy nêu một số ứng dụng của Internet đối với hoạt động giải trí.
2. Em hãy nêu một số ứng dụng của Internet với hoạt động bảo vệ sức khoẻ.

3. ĐIỆN TOÁN ĐÁM MÂY

a) Khái niệm về điện toán đám mây

Hoạt động 2 **Mua hay sử dụng thuê bao?**

Hãy đọc hai ví dụ và trả lời các câu hỏi:

Ví dụ 1. Bạn An có rất nhiều ảnh cần lưu nhưng ổ đĩa cứng sắp hết chỗ. Thay vì mua thêm một ổ đĩa cứng lớn, An đã đăng kí dịch vụ lưu trữ trên Internet như Google Drive, Dropbox, fShare. Khi cần, An chỉ cần kết nối Internet, đăng nhập và sử dụng giống như một ổ đĩa trên máy cá nhân. Nếu dùng ít thì không phải trả tiền, dùng nhiều tới một mức nào đó thì phải trả theo mức sử dụng.

Ví dụ 2. Công việc của cô Bình phải làm tài liệu rất nhiều và phải di chuyển thường xuyên. Thay vì mua phần mềm soạn thảo Word cài đặt trên máy tính ở nhà, cô đăng kí sử dụng phần mềm Google Docs chạy trên máy chủ của Google. Cô có thể soạn thảo bất cứ lúc nào, bất cứ ở đâu, dùng bất cứ máy tính nào miễn là có kết nối đến máy chủ Google Docs qua Internet. Văn bản cũng được lưu trên máy chủ của Google.

Câu hỏi:

1. Bạn An, cô Bình đã thuê bao dịch vụ cung cấp loại tài nguyên nào?
   A. Phần cứng. B. Phần mềm. C. Dữ liệu.
2. So với việc tự mua sắm thì việc thuê bao dịch vụ qua Internet có những ích lợi gì?



Sự phát triển mang tính bùng nổ của Internet, đặc biệt liên quan tới tốc độ truy cập ngày một tăng cao, cho phép người dùng có thể truy cập để sử dụng nhiều nguồn tài nguyên dùng chung (mạng, máy chủ, trung tâm lưu trữ, các ứng dụng, dịch vụ,...). Điều đó đã làm xuất hiện nhu cầu sử dụng và cung cấp nhiều dịch vụ công nghệ thông tin đa dạng thông qua khả năng kết nối Internet một cách dễ dàng theo yêu cầu mọi lúc, mọi nơi. Đặc biệt, người dùng không biết cụ thể những dịch vụ ấy được triển khai cụ thể ở đâu ngoài việc hình dung "đâu đó trên mạng", vốn thường được nói theo cách ẩn dụ trên "đám mây".

Việc chia sẻ các tài nguyên mạng theo nhu cầu qua Internet miễn phí hoặc trả phí theo hạn mức sử dụng được gọi là *dịch vụ điện toán đám mây* (gọi tắt là *dịch vụ đám mây*). Để sử dụng dịch vụ đám mây, người dùng phải đăng kí thuê bao, thoả thuận hạn mức sử dụng nếu phải trả phí và được cấp tài khoản truy cập.

Google Docs, Dropbox,... là những ví dụ điển hình của dịch vụ đám mây.

b) Các loại dịch vụ đám mây cơ bản

Các dịch vụ đám mây cơ bản nói chung đều chủ yếu liên quan tới việc cho thuê các tài nguyên phần mềm và phần cứng.

Người ta chia phần mềm thành hai nhóm, nhóm các *phần mềm ứng dụng* dùng trong các hoạt động nghiệp vụ thông thường và nhóm các *phần mềm nền tảng* (platform) là công cụ làm ra các sản phẩm khác trong đó có phần mềm ứng dụng.

Phần mềm soạn thảo Google Docs hay phần mềm tạo lớp học ảo Zoom là phần mềm ứng dụng.

Việc cho thuê phần mềm ứng dụng được viết tắt là SaaS (Software as a service – *phần mềm như là dịch vụ*).

Các tổ chức làm tin học thường thuê cả một hệ thống các phần mềm nền tảng để phát triển các phần mềm ứng dụng. Ví dụ, các công ty chuyên làm website thuê phần mềm quản trị cơ sở dữ liệu, phần mềm quản lí tin tức. Các tổ chức làm phần mềm đặt xe taxi, giao hàng, hướng dẫn du lịch đều cần bản đồ để định vị và dẫn đường. Việc tự đầu tư sẽ rất tốn kém nên họ trả tiền để thuê nền tảng bản đồ số của Google.

Việc cho thuê nền tảng được viết tắt là PaaS (Platform as a service – *nền tảng như là dịch vụ*).

Phần cứng như máy chủ, thiết bị lưu trữ,... – những cấu thành quan trọng của hạ tầng (infrastructure) công nghệ thông tin cũng có thể cho thuê qua Internet. Lưu trữ thông tin trên Internet thông qua Dropbox hay Google Drive là một trong các ví dụ thuê phần cứng đơn giản nhất.

Việc cho thuê hạ tầng được viết tắt là IaaS (Infrastructure as a service – *hạ tầng như là dịch vụ*).

SaaS, PaaS và IaaS là các loại dịch vụ chủ yếu của *điện toán đám mây*.

c) Lợi ích của dịch vụ đám mây

Tính mềm dẻo và độ sẵn sàng cao: Người dùng không bị lệ thuộc vào phương tiện cá nhân, thời gian và địa điểm làm việc miễn là có kết nối Internet.

Chất lượng cao: Các nhà cung cấp dịch vụ đám mây thường đầu tư chuyên nghiệp. Phần mềm được kiểm định nhờ số lượng người dùng lớn. Hạ tầng có công suất dự phòng lớn, ổn định và an toàn.

Kinh tế hơn: Do chia sẻ cho nhiều người, dịch vụ đám mây có thể phân tải các dịch vụ và người dùng để không bị lãng phí. Chính người dùng cũng chỉ trả tiền theo mức sử dụng. Rất nhiều dịch vụ đám mây miễn phí đối với người dùng cá nhân (chỉ thu phí với người dùng là tổ chức) như Gmail để gửi thư, Google maps để tìm đường.

- Điện toán đám mây được định nghĩa như là việc phân phối các tài nguyên Công nghệ thông tin theo nhu cầu qua Internet với chính sách thanh toán theo mức sử dụng. SaaS, PaaS, IaaS là các loại hình dịch vụ chủ yếu của điện toán đám mây.
- Sử dụng dịch vụ đám mây linh hoạt hơn, tin cậy hơn, chi phí nói chung rẻ hơn so với tự mua sắm phần cứng và phần mềm.



1. Báo điện tử, giúp mọi người có thể đọc tin tức hằng ngày có phải là dịch vụ đám mây hay không?

2. Thư điện tử Gmail có phải là dịch vụ đám mây không?

4. KẾT NỐI VẠN VẬT

Hoạt động 3 **Công tơ điện tử làm việc như thế nào?**

Trước đây, hằng tháng các nhân viên điện lực phải đi đọc các công tơ điện, ghi lại rồi nhập vào máy tính để lập hoá đơn và thống kê tình hình sử dụng điện. Hiện nay, công tơ truyền thống đang được thay thế bằng công tơ điện tử. Công tơ điện tử được gắn vi xử lí để đọc các chỉ số điện và đều đặn gửi về một đầu mối, từ đó chuyển về trung tâm dữ liệu qua Internet. Công tơ điện tử chính là một thiết bị thông minh.

Hãy thảo luận, ích lợi của việc dùng công tơ điện tử.



Hình 8.3. *Một công tơ điện tử*

Ý tưởng liên kết thiết bị thông minh là nguồn gốc của kết nối vạn vật (Internet of Things, viết tắt là IoT). IoT được định nghĩa là việc liên kết các thiết bị thông minh để tự động thu thập, trao đổi và xử lí dữ liệu phục vụ cho các mục đích khác nhau.

Một số lợi ích của IoT:

– Có thể thu thập dữ liệu trên diện rộng nhờ mạng máy tính; Có thể làm việc ở những nơi có điều kiện bất lợi mà con người không làm được, như ghi dữ liệu giám sát trong lò phản ứng hạt nhân.

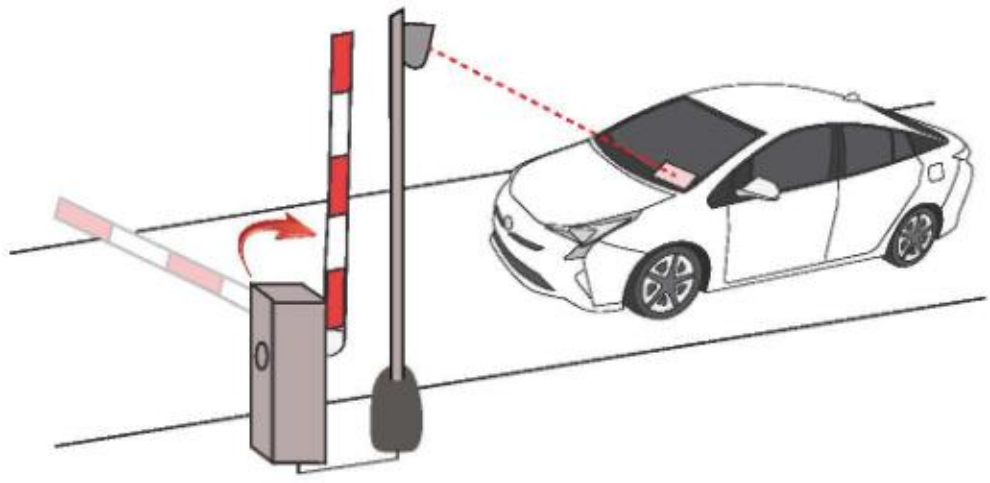
– Có thể hoạt động liên tục, tự động, cung cấp dữ liệu tức thời – điều này đặc biệt quan trọng đối với các hệ thống thời gian thực (real time) mà một quyết định chậm trễ có thể gây thảm hoạ, ví dụ điều khiển lò phản ứng hạt nhân hay lái xe tự động.

– Tiết kiệm chi phí do giảm bớt lao động thu thập và xử lí thông tin mang tính thủ công.

Một vài ví dụ về IoT:

Ví dụ 1. Thu phí không dừng trên các đường cao tốc.

Các xe ô tô phải dán thẻ định danh giao tiếp qua sóng radio (Radio Frequency Identification, RFID) ở kính lái. Khi vào hay ra khỏi cao tốc, thiết bị đọc thẻ RFID sẽ thu thông tin trên thẻ rồi truyền về hệ thống kiểm soát. Nếu tài khoản còn đủ tiền thì hệ thống trừ phí và truyền lệnh về trạm thu phí mở thanh chắn.



Hình 8.4. *Thẻ RFID gắn trên kính ô tô giúp tự động thu phí*

Ví dụ 2. Nhà thông minh (Smart home).

Các cảm biến có thể tự động bật tắt điều hoà không khí, máy đun nước nóng và hệ thống chiếu sáng khi cần thiết để tiết kiệm năng lượng. Các khoá cửa có khả năng nhận dạng khuôn mặt. Hệ thống kiểm soát chống đột nhập có khả năng đánh giá nguy cơ và tự động nhắn tin cho chủ nhà khi cần. Đồ gia dụng thông minh cũng được tích hợp vào hệ thống cho phép bật máy giặt, robot quét nhà qua Internet,...Toàn bộ hệ thống như vậy có thể được giám sát và điều phối qua Internet bằng điện thoại thông minh.



Hình 8.5. *Nhà thông minh*

Nhà thông minh đang là một nhu cầu lớn hiện nay.

- IoT là hệ thống liên mạng bao gồm các phương tiện và vật dụng, các thiết bị thông minh được cài đặt các cảm biến, phần mềm chuyên dụng giúp chúng có thể tự động kết nối, thu thập và trao đổi dữ liệu qua mạng mà không nhất thiết có sự tương tác trực tiếp giữa con người với con người, hay con người với máy tính.
- Với khả năng thu thập dữ liệu tự động trên diện rộng, phát hiện và xử lí kịp thời các vụ việc phát sinh IoT mang lại nhiều lợi ích trong các hoạt động nghiệp vụ và đem lại nhiều tiện nghi cho cuộc sống. Vì vậy, IoT được xem là một nội dung chủ chốt của cuộc cách mạng công nghiệp lần thứ tư.

Trong một mạng IoT, có nhất thiết là thiết bị thông minh chỉ nối với nhau qua Internet hay không?

LUYỆN TẬP

1. Phân tích ích lợi của giải pháp thu phí không dừng trên đường cao tốc.
2. Các mạng xã hội như facebook, youtube cho mọi người sử dụng miễn phí, nhưng nếu ai sử dụng để bán hàng hay quảng cáo thì phải trả tiền. Đây có phải là dịch vụ đám mây không?

VẬN DỤNG

1. Tìm qua Internet một ứng dụng điện toán đám mây của một doanh nghiệp Việt Nam.
2. Bộ giám sát hành trình trên xe tải hoặc xe khách hiện nay, cứ 30 giây một lần lại gửi dữ liệu tốc độ, toạ độ cùng thời điểm lấy toạ độ của xe về máy chủ giám sát. Với dữ liệu đó, có thể biết được những vi phạm giao thông nào của lái xe?

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

BÀI 9

AN TOÀN TRÊN KHÔNG GIAN MẠNG

SAU BÀI NÀY EM SẼ:

- Nêu được những nguy cơ và tác hại khi tham gia các hoạt động trên Internet một cách thiếu hiểu biết và bất cẩn. Trình bày được một số cách đề phòng những tác hại đó.
- Nêu được một vài cách phòng vệ khi bị bắt nạt trên mạng. Biết cách bảo vệ dữ liệu cá nhân.
- Trình bày được sơ lược về các phần mềm xấu (mã độc). Biết sử dụng một số công cụ để phòng chống phần mềm xấu.

Không gian mạng – (trong một số hoàn cảnh cụ thể được gọi vắn tắt là "mạng") chính là Internet, là một môi trường rất mở. Trên mạng mọi người có thể liên lạc, chia sẻ thông tin với nhau một cách dễ dàng nhưng chính điều đó lại bị những kẻ xấu lợi dụng khiến mạng cũng là nơi đầy rẫy những cạm bẫy. Cần tự bảo vệ mình như thế nào?

1. MỘT SỐ NGUY CƠ TRÊN MẠNG

Hoạt động 1 Nguy cơ trên mạng

Hãy thảo luận và cho ví dụ minh hoạ về những nguy cơ có thể khi lên Internet để:

a) Kết bạn.

b) Xem tin tức.

c) Tải các phần mềm.

Tin giả và tin phản văn hoá. Ai cũng có thể đưa tin bài lên mạng kể cả các tin bài xấu. Có nhiều tin giả, tin bài phản cảm như những tình tiết có tính bạo lực trong một vụ án hoặc có nội dung dẫn đến các nhận thức lệch lạc.

Lừa đảo trên mạng. Nhiều người lợi dụng mạng để lừa đảo. Một ví dụ là kẻ xấu lấy các ảnh, tin tức trên trang facebook của một người để lập một trang giống hệt rồi kết bạn với những người bạn của nạn nhân. Cuối cùng chúng mạo danh vay mượn để chiếm đoạt tài sản.

Lộ thông tin cá nhân. Thông tin cá nhân bao gồm tên tuổi, số điện thoại, tài khoản email, tài khoản ngân hàng,... và các tài khoản của các ứng dụng trên mạng. Khi lộ tài khoản các ứng dụng phần mềm, ta có thể bị mạo danh. Còn khi lộ tài khoản của ngân hàng, ta có thể mất tiền.

Các biện pháp bảo vệ thông tin cá nhân:

- Không ghi chép thông tin cá nhân ở những nơi mà người khác có thể đọc;
- Giữ cho máy tính không bị nhiễm các phần mềm gián điệp;
- Cẩn trọng khi truy cập mạng qua wifi công cộng vì hầu hết những trạm wifi công cộng không mã hoá thông tin khi truyền.

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

Bắt nạt trên không gian mạng. Một trong những vấn nạn trên mạng là bắt nạt. Mức thấp là xỉ vả, lăng nhục. Mức cao là đe doạ tung thông tin cá nhân, đưa tin bịa đặt, vu khống, thậm chí tống tiền hoặc ép buộc làm điều xấu. Hành vi bắt nạt trên mạng ảnh hưởng nghiêm trọng tới tâm lí của nạn nhân vì:

- Việc bắt nạt có thể xảy ra dai dẳng, bất cứ lúc nào;
- Người bắt nạt có thể ẩn danh, không biết là ai để đối phó;
- Số người theo dõi, bình luận có thể rất đông gây áp lực nặng nề, khiến nạn nhân có nguy cơ tự cô lập;
- Nhiều người không tự giải quyết được nhưng không dám nói ra, dẫn đến trầm cảm và có các hành vi tiêu cực. Bắt nạt là một kiểu khủng bố trên không gian mạng.

Một số biện pháp phòng chống hành vi bắt nạt:

- Không nên kết bạn dễ dãi qua mạng.
- Không trả lời thư từ hay tin nhắn, không tranh luận với kẻ bắt nạt trên diễn đàn.
- Hãy lưu giữ tất cả các bằng chứng.
- Hãy chia sẻ với bố mẹ hoặc thầy cô.
- Khi sự việc nghiêm trọng hãy báo cho cơ quan công an kèm theo bằng chứng.

Nghiện mạng. Có một số người dành rất nhiều thời gian cho mạng, đặc biệt là chơi game đến mức nghiện, ảnh hưởng nghiêm trọng tới sức khoẻ. Đã từng có những thanh niên chơi trò chơi điện tử liên tục nhiều ngày và đột quỵ ngay trên bàn máy tính.

- Mạng là môi trường giao tiếp nhanh chóng, thuận tiện nhưng ẩn chứa nhiều nguy cơ gây mất an toàn thông tin.
- Chỉ truy cập các trang web tin cậy, hãy cảnh giác với các thông tin giả, lừa đảo.
- Hãy giữ bí mật thông tin cá nhân.
- Chỉ nên kết bạn với những người quen biết trong mạng xã hội. Khi bị bắt nạt, hãy chia sẻ với người thân hoặc thầy cô.
- Không nên sử dụng Internet quá nhiều.



1. Em hãy đưa ra một số tình huống có thể làm lộ mật khẩu tài khoản.
2. Em có biết một hành vi lừa đảo nào trên mạng không? Nếu có, em hãy kể cách thức lừa đảo.

2. PHẦN MỀM ĐỘC HẠI

Hoạt động 2 **Có những loại phần mềm độc hại nào?**

Em hiểu gì về virus máy tính? Có phải tất cả phần mềm độc hại đều là virus?



Một đối tượng gây mất an toàn là phần mềm độc hại (malicious software, viết tắt là malware), những phần mềm được viết ra với ý đồ xấu, gây hại cho người dùng.

Những phần mềm độc hại có mục đích gây ảnh hưởng trên quy mô lớn thường có cơ chế lây nhiễm. Theo cơ chế lây nhiễm, có hai loại phần mềm độc hại là virus và worm. Còn một loại phần mềm độc hại khác là trojan chỉ nhằm chiếm đoạt thông tin hay chiếm quyền sử dụng máy tính sẽ ít chú trọng đến tính năng lây nhiễm.

a) Tìm hiểu về virus, trojan, worm và cơ chế hoạt động

Virus. Virus không phải là các phần mềm hoàn chỉnh, mà chỉ là các đoạn mã độc và phải gắn với một phần mềm mới phát tác và lây lan được. Khi chạy một phần mềm đã nhiễm virus, đoạn mã độc sẽ được đưa vào bộ nhớ, chờ khi thi hành một phần mềm khác sẽ chèn vào để hoàn thành một chu kì lây lan.

Worm, sâu máy tính. Khác với virus, worm là một phần mềm hoàn chỉnh. Để lây, worm lợi dụng những lỗ hổng bảo mật của hệ điều hành hoặc dẫn dụ, lừa người dùng chạy để cài đặt vào máy của nạn nhân. Cách lừa thông thường là để một liên kết ngầm trong email hoặc tin nhắn với vỏ bọc là một nội dung lành mạnh, ví dụ "bấm vào đây để nhận tin" nhưng khi bấm vào, ngoài bản tin thì chính phần mềm độc hại cũng được tải vào máy.

Trojan. Phần mềm nội gián, gọi là trojan, theo truyền thuyết "Con ngựa thành Troa" (Trojan Horse) trong truyện thần thoại Hy Lạp. Tuỳ hành vi, trojan có thể mang những tên khác nhau như:

- Spyware: (phần mềm gián điệp) có mục đích ăn trộm thông tin để chuyển ra ngoài.
- Keylogger: là một loại spyware ngầm ghi hoạt động của bàn phím và chuột để tìm hiểu người sử dụng máy làm gì.
- Backdoor: tạo một tài khoản bí mật, giống như cửa sau, để có thể truy cập ngầm vào máy tính.
- Rootkit: chiếm quyền cao nhất của máy, có thể thực hiện được mọi hoạt động kể cả xoá các dấu vết. Rootkit cũng có tài khoản truy nhập ngầm.

b) Tác hại của phần mềm độc hại

Như vậy, hai đặc trưng chính của virus hay worm là lây lan và gây ra các tác động không mong muốn, còn trojan thì thực hiện các hoạt động nội gián.

Tác động không mong muốn có khi chỉ gây khó chịu, nhưng các virus hay worm "dữ" có thể làm hỏng các phần mềm khác trong máy, xoá dữ liệu hay làm tê liệt hệ thống máy tính.

Virus có thể bị phát hiện theo hành vi, nhưng các worm (sâu) thường do chính nạn nhân bị lừa cài đặt nên rất khó phát hiện. Nhiều sâu đã gây ra những thảm hoạ, ví dụ:

- Sâu Melissa (1999) có cơ chế lừa để lây rất hiệu quả đã từng gây thiệt hại hơn 1 tỉ đô la.
- Sâu Code Red (2001) lợi dụng một khiếm khuyết bảo mật của Windows, chiếm quyền các máy chủ Windows, trong 10 ngày đã gây thiệt hại khoảng 2 tỉ đô la.
- Sâu WannaCry (2017) tống tiền bằng cách mã hoá toàn bộ thông tin có trên đĩa cứng và đòi tiền chuộc mới cho phần mềm hoá giải.
- Một số loại virus hay worm được phát tán rộng rãi, trở thành các đội quân ngầm, mỗi khi nhận được lệnh là truy cập đồng thời vào một máy chủ định trước, gây quá tải, làm tê liệt máy chủ. Hình thức tấn công này gọi là tấn công từ chối dịch vụ (Denial of Service – DOS) rất khó chống.

c) Phòng chống phần mềm độc hại

Để phòng ngừa phần mềm độc hại, điều quan trọng đầu tiên là cần thận trọng khi chép các tệp chương trình hay dữ liệu vào máy từ ổ cứng rời, thẻ nhớ hoặc tải về từ mạng. Rất nhiều phần mềm bẻ khoá cho lấy tự do trên mạng là các phần mềm bị gắn mã độc một cách cố ý.

Để tránh bị lừa, không mở các liên kết trong email hay tin nhắn mà không biết rõ có an toàn hay không. Ngay cả khi nhận thư từ hộp thư của bạn bè, nếu thấy có yêu cầu nháy vào liên kết thì tốt nhất hãy liên lạc qua một phương tiện khác, ví dụ gọi điện cho bạn để xác minh trước khi tải xuống.

Đừng để lộ mật khẩu các tài khoản của mình để tránh bị kẻ xấu chiếm quyền, mạo danh.

Ngoài ra, hãy sử dụng các phần mềm phòng chống các phần mềm độc hại.

- Phần mềm độc hại là phần mềm viết ra với ý đồ xấu, gây ra các tác động không mong muốn.
- Virus và worm là các phần mềm độc hại có khả năng lây nhiễm; Trojan là phần mềm nội gián để ăn cắp thông tin và chiếm đoạt quyền trên máy.
- Để phòng ngừa phần mềm độc hại, không lấy từ trên mạng hoặc sao chép qua các thiết bị nhớ những phần mềm mà mình không biết rõ. Khi nhận được email hay tin nhắn có liên kết, nếu không rõ về nguồn gốc thì không nên mở.
- Hãy sử dụng các phần mềm chống phần mềm độc hại để bảo vệ máy tính.

Em hãy tổng kết về ba loại phần mềm độc hại theo bảng sau:

	Tính hoàn chỉnh	Cơ chế lây nhiễm	Tác hại
Virus	?	?	?
Trojan	?	?	?
Worm	?	?	?

THỰC HÀNH

Dùng phần mềm phòng chống virus Windows Defender.

Có hàng trăm phần mềm phòng chống virus như Kaspersky, AVG, Avast, McAfee, Norton Antivirus, Panda hay một phần mềm của Việt Nam là BKAV. Thực tế, các phần mềm này không chỉ phát hiện và diệt virus mà còn phòng chống nhiều nguy cơ làm mất an toàn khác như kiểm tra các tệp tải về, cảnh báo truy cập các website có mã độc,... Chính vì thế, các phần mềm này còn được gọi là Firewall (bức tường lửa) với ý nghĩa ngăn chặn các nguy cơ từ ngoài.

Bài thực hành này giới thiệu sử dụng phần mềm Windows Defender được tích hợp sẵn trong hệ điều hành Windows phiên bản 10, tự động chạy ngầm để bảo vệ các máy tính dùng hệ điều hành Windows. Defender tự động cập nhật các mẫu virus mới mỗi khi hệ điều hành được cập nhật (theo tiện ích Windows Update).

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

Nhiệm vụ: Thiết lập các lựa chọn và quét virus với Windows Defender.

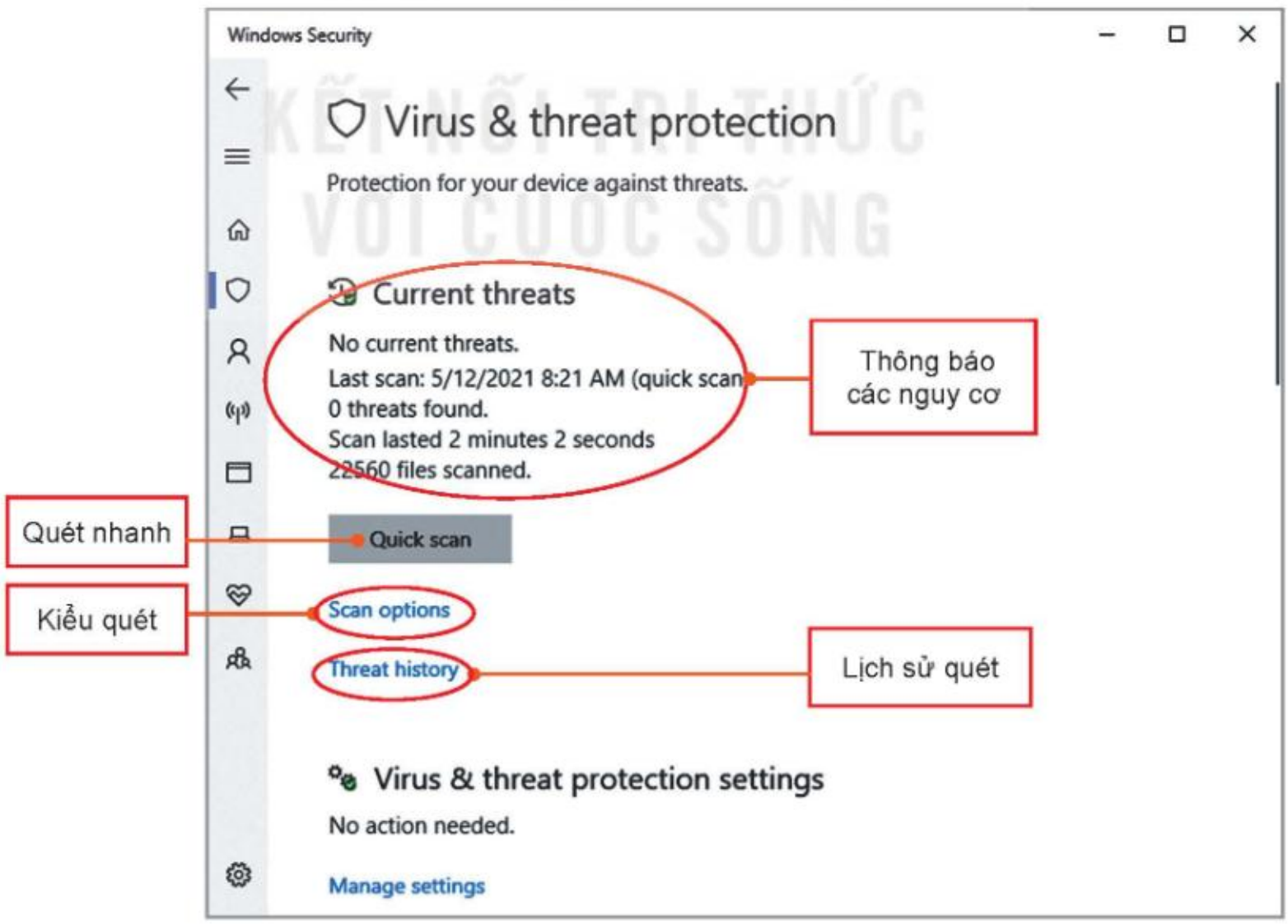
Hướng dẫn.

Bước 1: Từ nút **Start** chọn **Settings** (có thể dùng cách nhanh hơn là gõ chữ "Defender" vào hộp tìm kiếm nằm ở thanh trạng thái và nhấn Enter), màn hình xuất hiện tương tự như sau:



Hình 9.1. *Truy cập chức năng bảo vệ chống virus*

Bước 2: Thực hiện các thao tác như hướng dẫn ở Hình 9.1 sẽ xuất hiện cửa sổ như Hình 9.2.



Hình 9.2. *Chức năng bảo vệ chống virus và các nguy cơ*

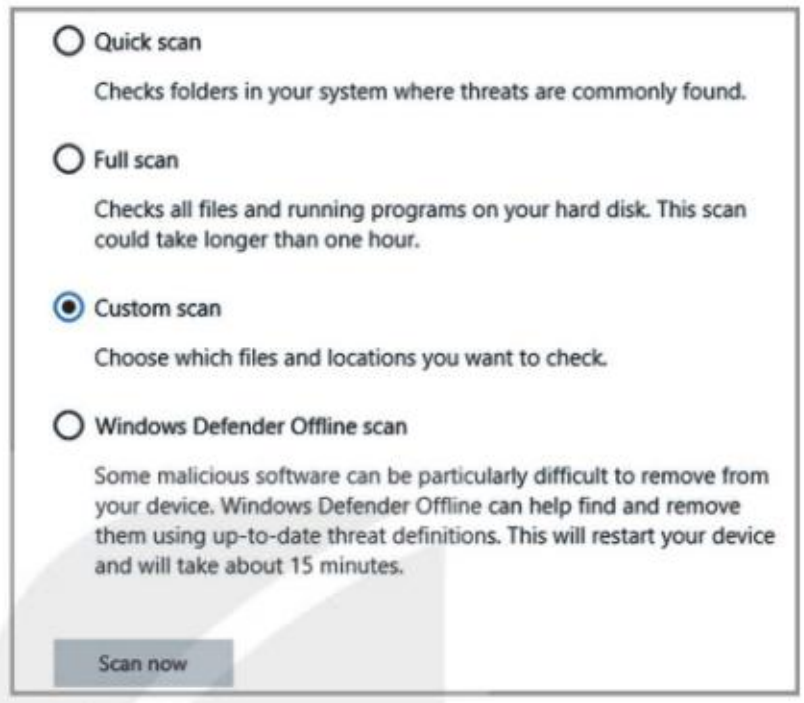
Current threats: thống kê những nguy cơ tìm thấy trong thời gian gần nhất khi các tệp được quét kiểm tra.

Quick scan: nếu nháy vào nút này phần mềm sẽ quét tất cả các tệp chương trình ở các thư mục mà virus thường lây nhiễm.

Bước 3: Quét virus. Ta có thể nháy vào nút Quick scan hoặc vào lựa chọn Scan options để lựa chọn kiểu quét và quét.

Trong Scan options, ta có thể lựa chọn các kiểu quét, có bốn lựa chọn:

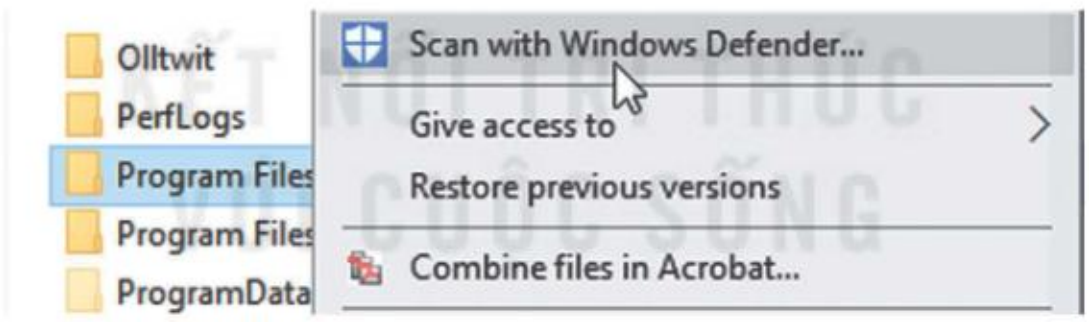
1. Quét nhanh (Quick scan): quét các thư mục có nguy cơ cao.
2. Quét hết (Full scan): quét tất cả các ổ đĩa.
3. Quét theo yêu cầu (Custom scan), chỉ quét trên một thư mục nào đó. Khi đó, Defender sẽ yêu cầu chỉ ra thư mục em muốn quét.
4. Quét ngoại tuyến (Windows Defender Offline scan). Chúng ta sẽ không bàn đến lựa chọn này vì nó là trường hợp đòi hỏi những hiểu biết rất sâu.



Hình 9.3. *Các lựa chọn trong scan options*

Sau khi chọn một lựa chọn, nháy nút Scan now và đợi kết quả.

Nếu đang làm việc ở thư mục mà muốn quét thư mục đó thì không cần truy cập vào Defender, ta có thể nháy nút phải chuột vào tên thư mục để xuất hiện bảng chọn tắt, chọn lệnh Scan with Windows Defender (Hình 9.4).



Hình 9.4. *Truy cập nhanh lệnh quét trên thư mục*

LUYỆN TẬP

1. Em hãy kể ra các nguy cơ mất an toàn khi tham gia các mạng xã hội.
2. Em hãy kể ra những trường hợp có thể bị nhiễm phần mềm độc hại và biện pháp phòng, chống tương ứng.

VẬN DỤNG

1. Em hãy tìm hiểu qua Internet các cách thức tấn công từ chối dịch vụ.
2. Em hãy tìm trên mạng thông tin về worm, kể một worm với tác hại của nó.

BÀI 10 THỰC HÀNH KHAI THÁC TÀI NGUYÊN TRÊN INTERNET

SAU BÀI NÀY EM SẼ:

- Khai thác được một số dịch vụ và tài nguyên trên Internet phục vụ học tập gồm: phần mềm dịch, kho học liệu mở.

Trên Internet có rất nhiều tài nguyên và phần mềm hỗ trợ cho học tập. Em có biết phần mềm nào hỗ trợ kĩ năng nghe, nói, đọc, viết khi học ngoại ngữ không? Các bài giảng số *Tin học 10* có thể tìm ở đâu?

Nhiệm vụ 1. Sử dụng phần mềm dịch đa ngữ của Google Translate để học ngoại ngữ.

Hướng dẫn.

Bước 1: Truy cập vào trang web có địa chỉ https://translate.google.com/ sẽ xuất hiện trang màn hình tương tự như sau:

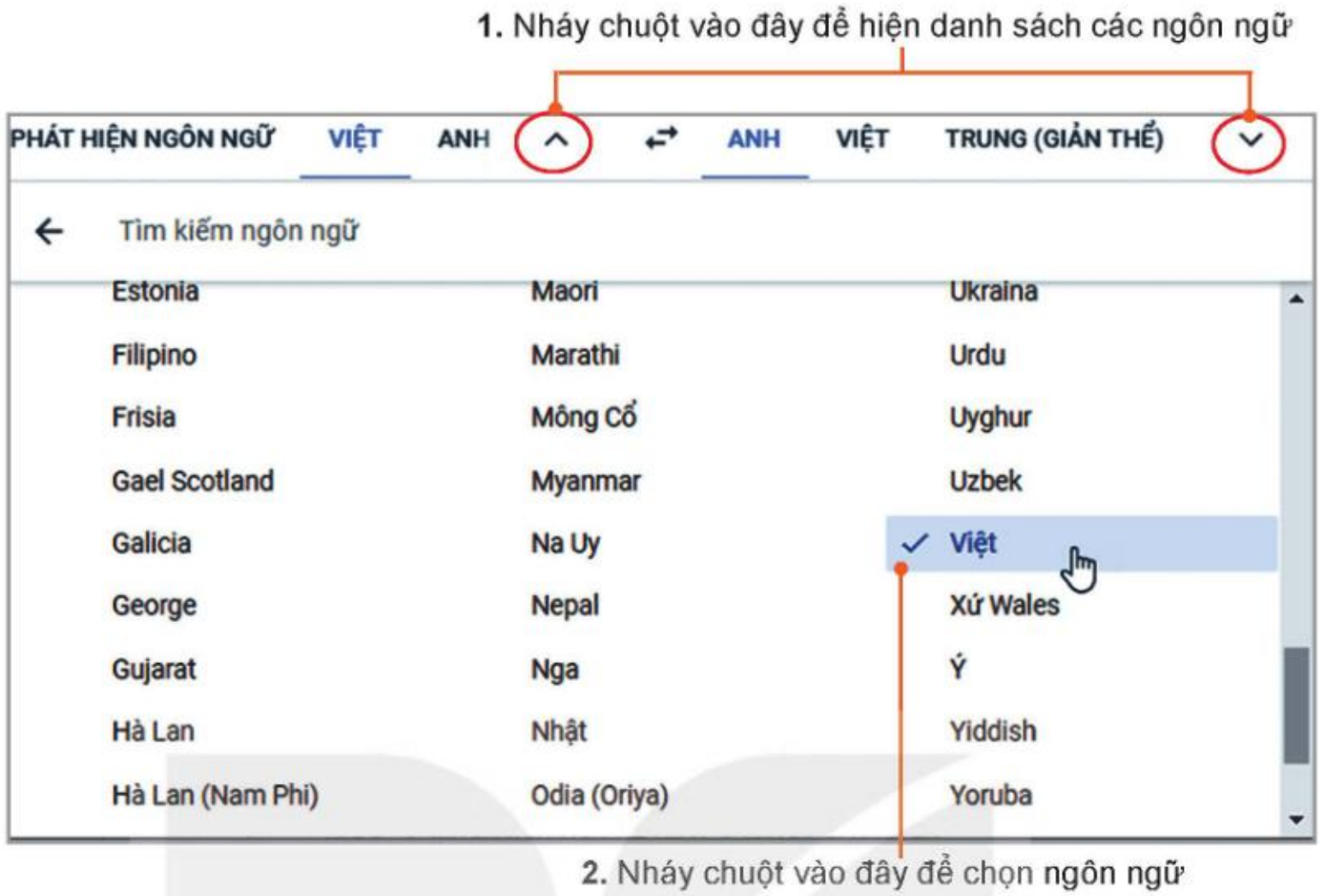


Hình 10.1. *Giao diện dịch đa ngữ của Google Translate*

Bên trái là khung của ngôn ngữ nguồn, nơi nhập văn bản cần dịch. Bên phải là khung chứa kết quả dịch của ngôn ngữ đích.

Bước 2: Xác định ngôn ngữ nguồn và ngôn ngữ đích.

Ngôn ngữ nguồn là ngôn ngữ đầu vào cần dịch và ngôn ngữ đích là ngôn ngữ đầu ra thể hiện kết quả của việc dịch. Để chọn ngôn ngữ hãy nháy chuột vào biểu tượng ⌄ sẽ mở ra danh sách các ngôn ngữ được phần mềm hỗ trợ. Chọn một ngôn ngữ mà mình muốn theo các bước minh hoạ sau đây:

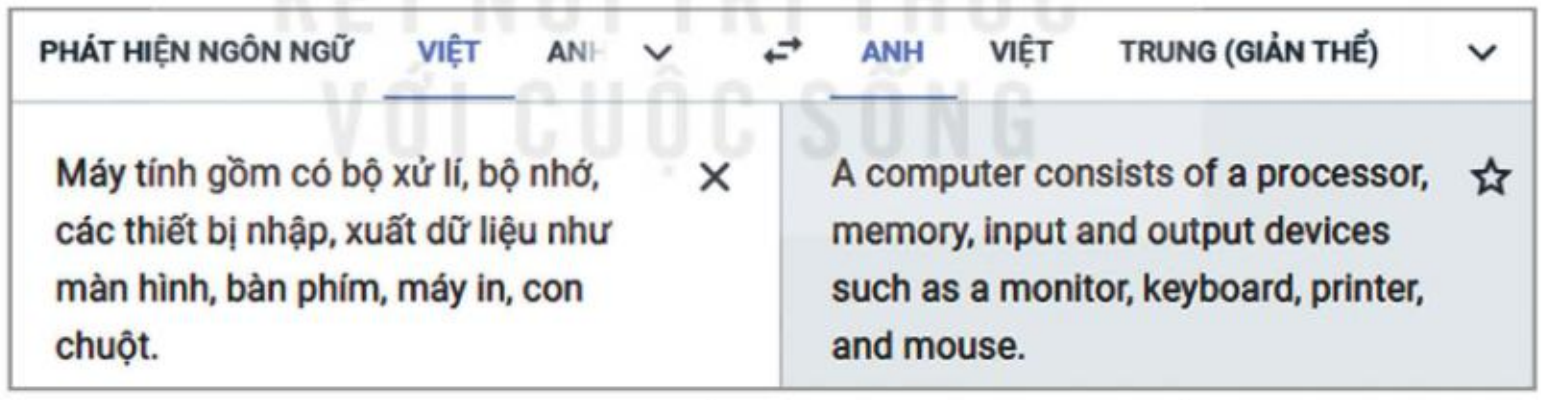


Hình 10.2. *Chọn ngôn ngữ làm việc*

Bước 3: Nhập văn bản để dịch.

Có ba cách nhập:

1. Nhập trực tiếp văn bản vào khung ngôn ngữ nguồn. Đây là chế độ mặc định, khi đó biểu tượng Văn bản sẽ có màu xanh. Ta chỉ cần gõ trực tiếp văn bản vào khung ngôn ngữ nguồn, bản dịch sẽ xuất hiện bên khung của ngôn ngữ đích (Hình 10.3).



Hình 10.3. *Nhập văn bản trực tiếp vào khung ngôn ngữ nguồn*

2. Nhập bằng giọng nói. Trong trường hợp này, máy tính phải có micro để thu âm. Trước khi nói, phải chọn biểu tượng micro . Khi biểu tượng micro đổi thành màu xanh thì em hãy đọc đoạn văn bản cần dịch. Nếu nháy chuột vào biểu tượng micro một lần nữa thì chế độ nhập bằng lời dừng, chuyển sang chế độ gõ trực tiếp văn bản cần dịch.

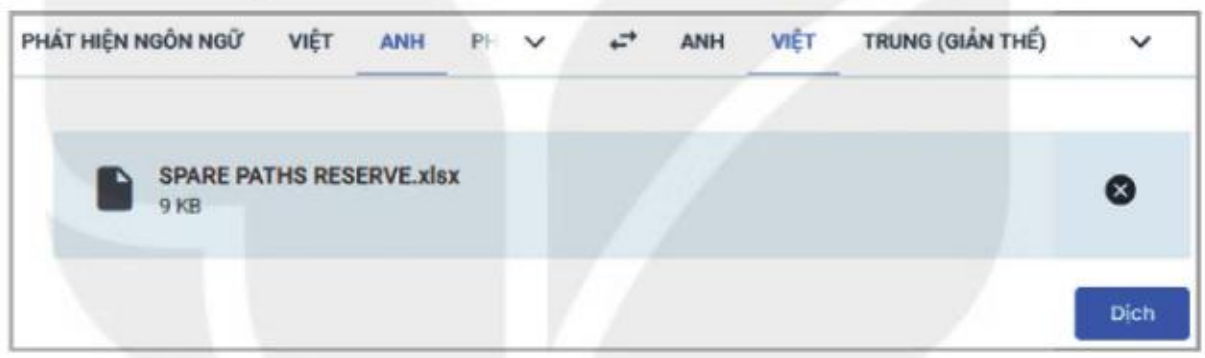
Hình 10.4 là giao diện dịch từ tiếng Việt sang tiếng Nga một câu trong một đoạn văn nổi tiếng của nhà văn Ostrovski trong tiểu thuyết "Thép đã tôi thế đấy". Kết quả dịch gần chính xác so với bản gốc trong tác phẩm.



Hình 10.4. *Nhập văn bản bằng giọng nói*

Google Translate không chỉ "nghe" được mà còn "nói" được. Để nghe máy đọc, ta chỉ cần nháy vào biểu tượng loa . Nháy chuột lần thứ nhất, máy sẽ đọc tốc độ bình thường, nháy chuột lần thứ hai thì máy sẽ đọc chậm hơn.

3. Nhập từ một tệp. Nháy chuột vào Tài liệu, phần mềm yêu cầu em chọn tệp sẽ dịch. Hãy nháy chuột vào nút lệnh Tìm tệp trên máy tính của bạn để chọn tệp. Tệp được chọn có thể là một tệp văn bản Word, tệp bảng tính Excel, tệp trình chiếu Powerpoint hay tệp PDF. Khi đó tên tệp sẽ xuất hiện như Hình 10.5.



Hình 10.5. *Nhập văn bản từ tệp*

Sau đó nháy vào nút lệnh Dịch để dịch.

Bước 4. Sao chép kết quả dịch vào tệp văn bản.

Kết quả được thể hiện dưới một định dạng văn bản trung gian trong một cửa sổ riêng. Muốn lấy kết quả dịch, ta chọn phần văn bản ở khung ngôn ngữ đích, rồi nhấn tổ hợp phím **Ctrl** + C để sao, sau đó mở tệp văn bản và nhấn tổ hợp phím **Ctrl** + **V** để dán.

Bản dịch có thể chưa thực sự trau chuốt và vẫn còn có thể nhầm nhưng có thể dùng để hỗ trợ hoàn thiện bản dịch.

Hình 10.6 là ví dụ về việc dịch một bảng tính từ tiếng Anh sang tiếng Việt.



	A	B	C
1	RESERVE SUPPLIES		
2	No	SPARE PART	QUANTITY
3	1	Color Monitor	2
4	2	Printer	3
5	3	Mouse	4
6	4	Keyboard	5
7	5	Hard disk	6

DỰ TRỮ CUNG ỨNG		
Không	PHỤ TÙNG	SỐ LƯỢNG
1	Màn hình màu	2
2	Máy in	3
3	Chuột	4
4	Bàn phím	5
5	Ổ đĩa cứng	6

Hình 10.6. *Kết quả dịch bảng tính*

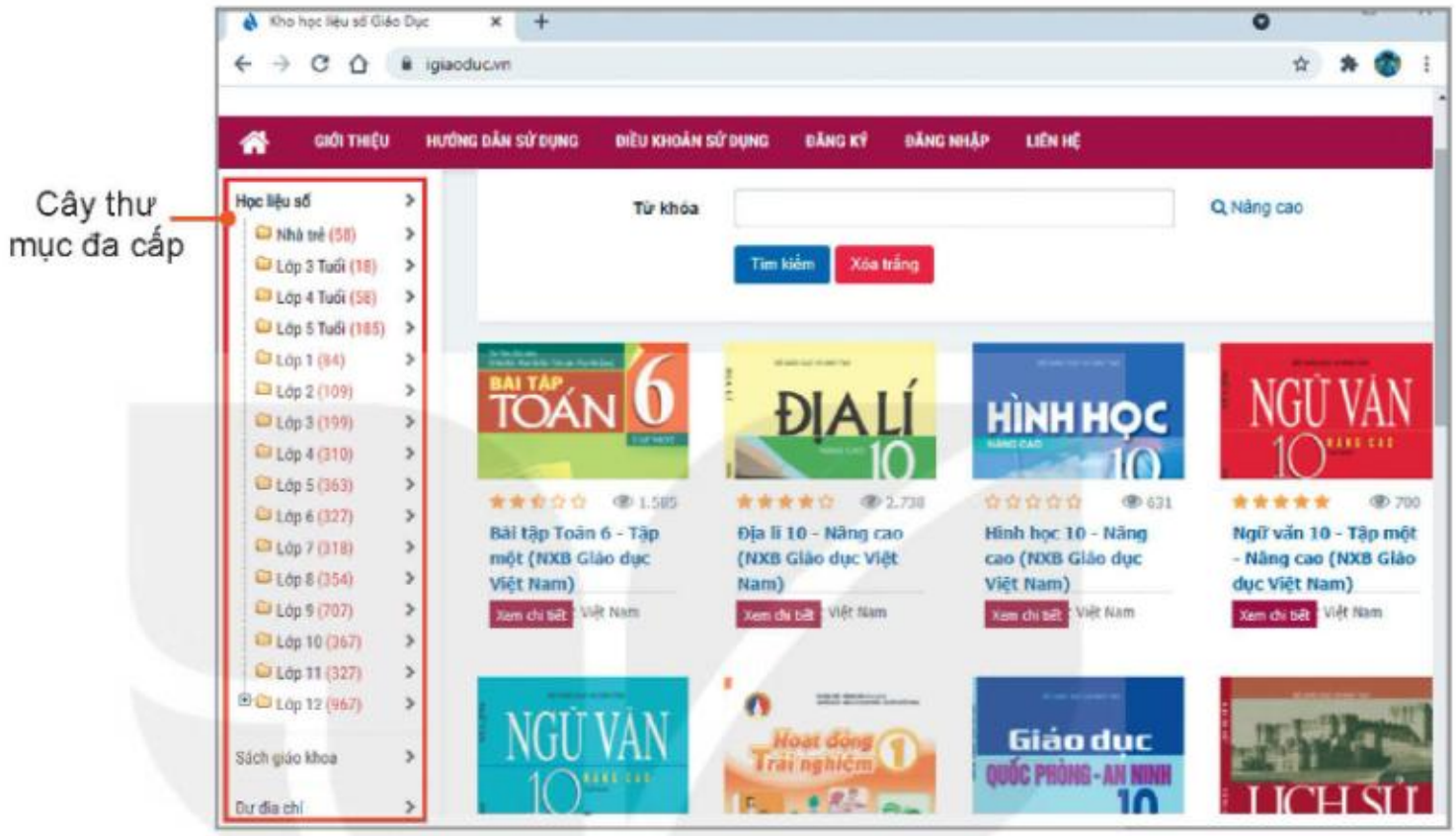
Nhiệm vụ 2. Khai thác một nguồn học liệu mở trên Internet để tìm các nguồn tài liệu phục vụ học tập.

Hướng dẫn.

Khái niệm học liệu mở được dùng rộng rãi khi Học viện Công nghệ Massachusetts của Mỹ năm 2002 đưa toàn bộ nội dung giảng dạy của mình lên Internet để sử dụng miễn phí. Nhiều cơ sở giáo dục sau đó theo gương, mở ra cơ hội cho mọi người có thể tiếp cận các tri thức mới với chi phí thấp. Ở Việt Nam, Bộ Giáo dục và Đào tạo cũng có một kho học liệu với hàng vạn học liệu mở đặt tại địa chỉ https://igiaoduc.vn.

Sau đây, chúng ta sẽ làm quen và khai thác kho học liệu đó.

Bước 1: Truy cập địa chỉ https://igiaoduc.vn/ trang chủ tương tự Hình 10.7:



Hình 10.7. *Trang chủ kho học liệu số*

Trên trang chủ, các loại học liệu được xếp theo các chủ đề trên cây thư mục đa cấp phía bên trái.

Bước 2: Tìm kiếm, truy cập các học liệu

– Thư mục cấp một gồm ba loại là Học liệu số, Sách giáo khoa và Dư địa chí. Muốn xem loại nào nháy chuột vào biểu tượng > ở mục tương ứng để mở ra thư mục cấp hai.

– Nháy chuột vào một mục của thư mục cấp hai, ví dụ Học liệu số để mở ra thư mục cấp ba.

– Thư mục cấp ba của Học liệu số và Sách giáo khoa trải ra theo các môn học. Chỉ cần chọn môn để xem tất cả các học liệu của môn đó.



Hình 10.8. *Các bài giảng môn Đại số lớp 10*

Bước 3: Xem bài giảng.

Để xem học liệu nào, chỉ cần nháy chuột vào ảnh học liệu tương ứng. Khi đó sẽ xuất hiện mô tả của học liệu đó với tên bài, chủ đề, tác giả (Hình 10.9).



Hình 10.9. *Giới thiệu học liệu*

Nháy chuột vào Học trực tuyến để xem bài giảng. Có nhiều dạng bài giảng như video quay trực tiếp bài giảng hoặc được chuyển thể từ các bản trình chiếu bài giảng của giáo viên.

LUYỆN TẬP

1. Chọn ngôn ngữ nguồn là tiếng Việt và ngôn ngữ đích là một ngoại ngữ mà em được học, sau đó gõ vào một số câu tiếng Việt để dịch.

   Em hãy nghe phần mềm đọc câu đã dịch ra tiếng nước ngoài.

   Nếu máy tính có micro, em có thể đọc để nhập văn bản thay vì gõ bàn phím.
2. Sử dụng Google Translate để dịch từ một ngoại ngữ sang tiếng Việt.

   Đảo lại vai trò của ngôn ngữ nguồn và đích bằng cách nháy chuột vào biểu tượng ⇄ sau đó gõ một vài câu trong tiếng nước ngoài để dịch sang tiếng Việt.
3. Em hãy mở trang học liệu mở https://igiaoduc.vn/ và chọn một bài học trực tuyến để nghe bài giảng.

VẬN DỤNG

1. Em hãy sử dụng một tệp văn bản sẵn có hoặc tự soạn một tệp văn bản trong tiếng Việt rồi sử dụng Google Translate để dịch ra ngôn ngữ mà em đã được học.
2. Em hãy tìm thêm một số kho học liệu để xem các bài giảng, tài liệu học tập.

Chủ đề 3

Đạo đức, pháp luật và văn hoá trong môi trường số

BÀI 11

ỨNG XỬ TRÊN MÔI TRƯỜNG SỐ. NGHĨA VỤ TÔN TRỌNG BẢN QUYỀN

SAU BÀI NÀY EM SẼ:

- Nêu được những vấn đề nảy sinh về đạo đức, pháp luật và văn hoá khi giao tiếp qua mạng trở nên phổ biến.
- Giải thích được một số nội dung pháp lí liên quan tới việc đưa tin lên mạng và tôn trọng bản quyền thông tin, sản phẩm số.



Trong cuộc sống, ai cũng có thể có những xung đột lợi ích với cộng đồng và đôi khi có những hành vi đi ngược lại lợi ích chung. Pháp luật quy định rõ những hành vi nào là vi phạm pháp luật mà Nhà nước sẽ cưỡng chế. Những hành vi khác không phù hợp với lợi ích chung của cộng đồng hay xã hội sẽ được coi là thuộc hành vi vi phạm đạo đức.

Đạo đức là hệ thống các quy tắc, chuẩn mực xã hội mà con người phải tự giác thực hiện cho phù hợp với lợi ích của cộng đồng, của xã hội. Dư luận xã hội và giáo dục là các biện pháp điều chỉnh đạo đức.

Theo em, những vấn đề về đạo đức và pháp luật nảy sinh khi giao tiếp trên mạng đã và đang trở thành phổ biến là gì?

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

1. NHỮNG VẤN ĐỀ ĐẠO ĐỨC, PHÁP LUẬT VÀ VĂN HOÁ

Hoạt động 1 **Hành vi vi phạm pháp luật hay đạo đức?**

Xem xét tình huống sau và trả lời câu hỏi:

Do mâu thuẫn ở một diễn đàn trên mạng, một nhóm nữ sinh đánh một bạn nữ khác. Các bạn ở xung quanh đã không can ngăn mà còn quay phim rồi đưa lên mạng xã hội. Do có nhiều bình luận thiếu thiện ý trên mạng xã hội dẫn đến xấu hổ với bạn bè, nạn nhân đã bỏ nhà ra đi không để lại lời nhắn.

Câu hỏi:

1. Trong tình huống trên, hành vi nào vi phạm pháp luật, hành vi nào vi phạm đạo đức?

2. Theo em, yếu tố nào của Internet đã khiến sự việc trở nên trầm trọng?

Việc đánh bạn ít nhất là vi phạm đạo đức. Nếu người đánh bạn đủ tuổi chịu trách nhiệm hình sự (16 tuổi) và gây thương tích nặng sẽ bị coi là vi phạm pháp luật và sẽ bị xử lí hình sự về tội cố ý gây thương tích, làm nhục người khác. Việc đưa video có nội dung như trên lên mạng, gián tiếp cổ vũ bạo lực học đường là hành vi vi phạm đạo đức.

Có thể kể đến một số loại hành vi xấu khi giao tiếp trên mạng như:

Đưa tin không phù hợp lên mạng (bao gồm cả đăng và chia sẻ tin bài). Tuỳ theo nội dung thông tin và hậu quả của việc đăng tin, mà những hành vi đó có thể là vi phạm pháp luật hay vi phạm đạo đức.

Công bố thông tin cá nhân hay tổ chức mà không được phép, gây ảnh hưởng đến uy tín và danh dự của cá nhân hay tổ chức là hành vi vi phạm pháp luật.

Gửi thư rác hay tin nhắn rác. Những thư hay tin nhắn nhằm mục đích quảng cáo mà người nhận không muốn nhận hoặc không bắt buộc phải nhận theo quy định của pháp luật được gọi là thư hay tin nhắn rác. Về bản chất, quảng cáo bằng tin nhắn không phải là một hành vi xấu và không vi phạm pháp luật, nhưng nếu gửi nhiều mà người nhận đã có phản ứng không muốn tiếp nhận thì lại trở thành hành vi quấy nhiễu.

Vi phạm bản quyền khi sử dụng dữ liệu và phần mềm. Vấn đề bản quyền không chỉ đặt ra khi giao dịch trên mạng mà trong bất cứ hoàn cảnh nào, những sản phẩm được số hoá và đưa lên mạng rất dễ bị lấy, phát tán, sửa đổi gây thiệt hại cho chủ sở hữu.

Bắt nạt qua mạng. Chúng ta đã tìm hiểu hành vi này trong Bài 9.

Lừa đảo qua mạng. Các hình thức lừa đảo trên mạng khá phổ biến và tinh vi. Nhiều trường hợp lừa đảo qua mạng gây ra những thiệt hại rất lớn.

Ứng xử thiếu văn hoá. Trên các diễn đàn mạng hiện nay, có nhiều người tranh luận thiếu văn hoá, không tôn trọng người đối thoại, thậm chí chửi tục hay công kích, sỉ nhục lẫn nhau.

Trong xã hội thực, phần lớn giao tiếp của con người là trực tiếp, còn trên không gian mạng, giao tiếp không chỉ gián tiếp mà còn có thể ẩn danh. Điều này làm cho kẻ xấu mạnh bạo hơn và hành vi xấu trên mạng khó kiểm soát hơn. Bên cạnh đó, trên không gian mạng người ta có thể gặp nhau bất cứ lúc nào, bất cứ ở đâu miễn là kết nối được với Internet; tin bài đã được lưu trên mạng, lúc nào cũng xem được. Do đó, tin tức đưa lên mạng được phát tán nhanh, rộng và lưu trữ lâu dài. Chính vì thế ảnh hưởng của một hành vi xấu trên mạng có xu hướng trầm trọng hơn.

Cần có những quy định pháp luật và các chuẩn mực đạo đức để đảm bảo lợi ích chung của cộng đồng người dùng mạng.

Những hành vi vi phạm về đạo đức, pháp luật và văn hoá khi sử dụng mạng:

- Đưa tin không phù hợp lên mạng.
- Công bố thông tin cá nhân không được phép.
- Phát tán thư điện tử, tin nhắn rác.
- Vi phạm bản quyền.
- Bắt nạt qua mạng.
- Ứng xử thiếu văn hoá.

Em hãy lấy ví dụ về các vấn đề tiêu cực có thể nảy sinh khi tham gia các hoạt động sau trên mạng.

a) Tranh luận trên facebook.

b) Gửi thư điện tử.

2. MỘT SỐ QUY ĐỊNH PHÁP LÍ ĐỐI VỚI NGƯỜI DÙNG TRÊN MẠNG

a) Các văn bản quy phạm pháp luật

Các văn bản quy phạm pháp luật gồm nhiều loại trong đó có: các bộ luật do Quốc hội ban hành và các nghị định do Chính phủ ban hành để cụ thể hoá các điều khoản của luật.

Quốc hội Việt Nam đã ban hành nhiều bộ luật liên quan đến Công nghệ Thông tin (CNTT) như: Luật Giao dịch điện tử (2005), Luật Công nghệ Thông tin (2006), Luật An ninh mạng (2018) và Luật Hình sự (2017).

Cụ thể hoá các bộ luật trên, có những nghị định đề cập đến các hành vi giao dịch trên mạng như: Nghị định 90/2008/NĐ-CP ban hành ngày 13/8/2008 về chống thư rác; Nghị định 72/2013/NĐ-CP ban hành ngày 15/7/2013 về quản lí, cung cấp và sử dụng dịch vụ Internet; Nghị định 15/2020/NĐ-CP ban hành ngày 3/2/2020 quy định xử phạt vi phạm hành chính trong lĩnh vực bưu chính, viễn thông, tần số vô tuyến điện, CNTT và giao dịch điện tử. Tháng 6/2021, Bộ Thông tin và Truyền thông đã ban hành Bộ Quy tắc ứng xử trên mạng xã hội theo Quyết định số 874/QĐ-BTTTT ban hành ngày 17/6/2021.

Các luật và nghị định trên đều quy định rõ trách nhiệm trong hoạt động cung cấp các sản phẩm và dịch vụ của các tổ chức hay cá nhân và trách nhiệm quản lí của các cơ quan nhà nước. Riêng đối với người dùng cá nhân, các văn bản pháp lí trên tập trung vào các quy định về sử dụng không gian mạng và bảo vệ bản quyền của các sản phẩm số.

Hoạt động 2 **Hành vi đưa tin lên mạng**

1. Em hãy đưa ra một số ví dụ đưa tin lên mạng không đúng.

2. Một cá nhân đưa tin bịa đặt về bệnh dịch xảy ra làm nhiều người hết sức lo lắng. Hành vi chia sẻ lại tin đó có vi phạm pháp luật hay không?

b) Các quy định của pháp luật đối với người dùng trên không gian mạng

Một trong những trách nhiệm quan trọng nhất của người đưa tin lên Internet là đảm bảo tính đúng đắn của thông tin.

Điều 12 khoản 2 của Luật công nghệ thông tin quy định cấm "Cung cấp, trao đổi, truyền đưa, lưu trữ, sử dụng thông tin số" nhằm các mục đích sau đây:

a) Chống Nhà nước Cộng hoà xã hội chủ nghĩa Việt Nam, phá hoại khối đoàn kết toàn dân.

b) Kích động bạo lực, tuyên truyền chiến tranh xâm lược, gây hận thù giữa các dân tộc và nhân dân các nước, kích động dâm ô, đồi trụy, tội ác, tệ nạn xã hội, mê tín dị đoan, phá hoại thuần phong mĩ tục của dân tộc.

c) Tiết lộ bí mật nhà nước, bí mật quân sự, an ninh, kinh tế, đối ngoại và những bí mật khác đã được pháp luật quy định.

d) Xuyên tạc, vu khống, xúc phạm uy tín của tổ chức, danh dự, nhân phẩm, uy tín của công dân.

e) Quảng cáo, tuyên truyền hàng hoá, dịch vụ thuộc danh mục cấm đã được pháp luật quy định.

Điều 8 khoản 1 trong Luật an ninh mạng cấm sử dụng không gian mạng để thực hiện hành vi sau đây:

a) Hành vi quy định tại khoản 1 Điều 18 của Luật này.

b) Tổ chức, hoạt động, câu kết, xúi giục, mua chuộc, lừa gạt, lôi kéo, đào tạo, huấn luyện người chống Nhà nước Cộng hoà xã hội chủ nghĩa Việt Nam.

c) Xuyên tạc lịch sử, phủ nhận thành tựu cách mạng, phá hoại khối đại đoàn kết toàn dân tộc, xúc phạm tôn giáo, phân biệt đối xử về giới, phân biệt chủng tộc.

d) Thông tin sai sự thật gây hoang mang trong Nhân dân, gây thiệt hại cho hoạt động kinh tế – xã hội, gây khó khăn cho hoạt động của cơ quan nhà nước hoặc người thi hành công vụ, xâm phạm quyền và lợi ích hợp pháp của cơ quan, tổ chức, cá nhân khác.

e) Hoạt động mại dâm, tệ nạn xã hội, mua bán người; đăng tải thông tin dâm ô, đồi truỵ, tội ác; phá hoại thuần phong, mĩ tục của dân tộc, đạo đức xã hội, sức khoẻ của cộng đồng.

f) Xúi giục, lôi kéo, kích động người khác phạm tội.

Một số hành vi vi phạm pháp luật về đưa tin trên mạng xã hội được cụ thể hoá kèm theo mức phạt trong Điều 101, khoản 1 của Nghị định 15/2020/NĐ-CP bao gồm:

a) Cung cấp, chia sẻ thông tin giả mạo, thông tin sai sự thật, xuyên tạc, vu khống, xúc phạm uy tín của cơ quan, tổ chức, danh dự, nhân phẩm của cá nhân.

b) Cung cấp, chia sẻ thông tin cổ suý các hủ tục, mê tín, dị đoan, dâm ô, đồi truỵ, không phù hợp với thuần phong, mĩ tục của dân tộc.

c) Cung cấp, chia sẻ thông tin miêu tả tỉ mỉ hành động chém, giết, tai nạn, kinh dị, rùng rợn.

d) Cung cấp, chia sẻ thông tin bịa đặt, gây hoang mang trong Nhân dân, kích động bạo lực, tội ác, tệ nạn xã hội, đánh bạc hoặc phục vụ đánh bạc.

e) Cung cấp, chia sẻ các tác phẩm báo chí, văn học, nghệ thuật, xuất bản phẩm mà không được sự đồng ý của chủ thể quyền sở hữu trí tuệ hoặc chưa được phép lưu hành hoặc đã có quyết định cấm lưu hành hoặc tịch thu.

f) Quảng cáo, tuyên truyền, chia sẻ thông tin về hàng hoá, dịch vụ bị cấm.

g) Cung cấp, chia sẻ hình ảnh bản đồ Việt Nam nhưng không thể hiện hoặc thể hiện không đúng chủ quyền quốc gia.

h) Cung cấp, chia sẻ đường dẫn đến thông tin trên mạng có nội dung bị cấm.

Bộ Thông tin và Truyền thông đã ban hành Quyết định 874/QĐ-BTTTT (2021) về bộ quy tắc ứng xử trên mạng xã hội. Ngoài nội dung yêu cầu các tổ chức cá nhân tuân thủ những quy định về đưa thông tin lên mạng đã được các văn bản luật quy định, có thêm các khuyến cáo về việc tuân thủ các quy định của nhà cung cấp dịch vụ mạng xã hội, khuyến khích dùng tên thật và khuyến khích đăng tin quảng bá hình ảnh đất nước, con người và văn hoá Việt Nam; chia sẻ thông tin tích cực, những tấm gương người tốt, việc tốt.

Các nguyên tắc để nâng cao tính an toàn khi chia sẻ thông tin trên môi trường số:

- Trước khi đăng tin, hãy kiểm tra tin bài có vi phạm pháp luật hay không.
- Không chia sẻ tin bài vi phạm pháp luật. Ngoài ra cần phải biết tin tức có chính xác không. Ngày nay trên mạng rất nhiều tin giả, việc chia sẻ một tin giả chính là tiếp tay cho hành vi tung tin giả.
- Ngay cả trong trường hợp việc đưa tin không vi phạm pháp luật thì cũng phải cân nhắc hậu quả, nhất là khía cạnh đạo đức. Một tin vô hại với người này có thể mang lại tai hoạ cho một người khác.

- Khi đưa tin lên mạng, hãy xem xét nội dung các tin bài có vi phạm các quy định của pháp luật hay không. Đừng quên rằng, việc chia sẻ một tin vi phạm luật cũng là vi phạm luật.
- Ngay khi tin đưa không phạm luật, vẫn phải tính đến các hậu quả của nó khi nó vi phạm những chuẩn mực đạo đức.

1. Trong đợt bùng phát dịch Covid−19 vào đầu năm 2021, một cá nhân đã đăng tin sai về hành trình đi lại của một bệnh nhân bị dương tính với virus Covid−19. Sự việc này đã gây hoang mang cho cả một khu dân cư.

Theo em, cá nhân trên đã vi phạm điều nào trong bộ các Luật liên quan đến Công nghệ thông tin?

2. Trên mạng hiện nay có rất nhiều quảng cáo sai sự thật. Quảng cáo sai về tác dụng của một loại thuốc sẽ bị phạt theo mục nào của điều 101 khoản 1 của Nghị định 15/2020/NĐ-CP?

3. QUYỀN TÁC GIẢ VÀ BẢN QUYỀN

a) Quyền tác giả

Hoạt động 3 Tìm hiểu về quyền tác giả

Em hiểu thế nào là quyền tác giả? Tác giả của một tác phẩm (bức tranh, chương trình máy tính) có những quyền gì đối với tác phẩm của mình?

Quyền tác giả là quyền của tổ chức, cá nhân đối với tác phẩm do mình sáng tạo ra hoặc sở hữu.

Vấn đề này liên quan tới cả các biến thể của tác phẩm (tác phẩm/bản phái sinh), khi từ tác phẩm gốc người ta tạo ra các biến thể, ví dụ một cuốn sách được dịch sang một ngôn ngữ khác hay một phần mềm máy tính được cải biên để dùng trên điện thoại di động,...

Trong Luật Sở hữu trí tuệ được Quốc Hội ban hành ngày 25 tháng 6 năm 2019, quy định quyền tác giả đối với tác phẩm (bài thơ, bài báo, bức tranh, hình vẽ, chương trình máy tính, sưu tập dữ liệu,...) bao gồm quyền nhân thân và quyền tài sản. Trong đó:

Quyền nhân thân bao gồm các quyền: Đặt tên cho tác phẩm; Đứng tên thật hoặc bút danh trên tác phẩm; được nêu tên thật hoặc bút danh khi tác phẩm được công bố, sử dụng; Công bố tác phẩm hoặc cho phép người khác công bố tác phẩm; Bảo vệ sự toàn vẹn của tác phẩm, không cho người khác sửa chữa, cắt xén hoặc xuyên tạc tác phẩm dưới bất kì hình thức nào gây phương hại đến danh dự và uy tín của tác giả,...

Quyền tài sản bao gồm các quyền: Làm tác phẩm phái sinh; Sao chép tác phẩm; Truyền đạt tác phẩm đến công chúng; Cho thuê bản gốc hoặc bản sao, chương trình máy tính,...

Trong thực tế, bản quyền và quyền tác giả thường được dùng chung, coi như không có khác biệt. Tuy nhiên, về mặt bản chất, hai thuật ngữ đó không hoàn toàn tương đồng. *Quyền tác giả* coi người sáng tạo ra tác phẩm (tác giả) là trung tâm và bảo hộ các quyền bao gồm quyền nhân thân và quyền tài sản còn *bản quyền* ưu tiên bảo vệ quyền lợi đầu tư về kinh tế của người sở hữu quyền tác giả hơn là chính tác giả.

Do quá trình hội nhập quốc tế, những khác biệt này cũng dần được thu hẹp. Các văn bản pháp luật của Việt Nam hiện tại sử dụng thuật ngữ chính thức đó là quyền tác giả.

b) Vi phạm bản quyền đối với các sản phẩm tin học

Hoạt động 4 **Tìm hiểu về vi phạm bản quyền**

Ai vi phạm bản quyền trong những tình huống sau?

1. Hoàng mua với giá rất rẻ một thẻ nhớ USB chứa các video âm nhạc mà người bán đã sưu tầm từ trên Internet không có thoả thuận gì với tác giả hay ca sĩ biểu diễn.

2. Lan mua một phần mềm có bản quyền trên đĩa CD. Sau khi cài đặt lên máy tính của mình, Lan cài thêm trên máy của một bạn thân.

Vi phạm quyền tài sản hay quyền nhân thân đều là vi phạm bản quyền. Hành vi vi phạm bản quyền rất đa dạng. Sau đây là một số các hành vi vi phạm bản quyền đối với các tác phẩm số.

– Mạo danh tác giả.

– Công bố mà không được phép. Quyền công bố vốn là quyền nhân thân, nhưng việc công bố không đúng lúc, đúng chỗ có thể ảnh hưởng đến việc kinh doanh nên các tổ chức làm phần mềm không những giữ toàn bộ các quyền tài sản mà còn giữ cả quyền công bố.

– Sửa chữa, chuyển thể phần mềm, dữ liệu mà không được phép của tác giả. Hành vi này phá hoại tính toàn vẹn của sản phẩm, làm ảnh hưởng tới tác giả.

– Sử dụng phần mềm lậu, không mua quyền sử dụng phần mềm đối với các phần mềm phải trả tiền.

– Phá khoá phần mềm, vô hiệu hoá các biện pháp kĩ thuật do chủ sở hữu quyền tác giả thiết lập. Mặc dù phá khoá phần mềm tức là thay đổi phần mềm, một hành vi vi phạm tính toàn vẹn của tác phẩm, nhưng việc phá khoá nhằm mục đích sử dụng không trả tiền thì hành vi này lại vi phạm quyền tài sản. Rất nhiều các phần mềm nổi tiếng bị phá khoá và đưa lên mạng công khai để mọi người có thể lấy về dùng.

– Làm bản phái sinh, phân phối dữ liệu hay phần mềm, kể cả bản phái sinh mà không được phép.

– Chiếm đoạt mã phần mềm. Thường thì nhân viên phát triển phần mềm cho một công ty là tác giả chứ không sở hữu phần mềm ấy. Có nhiều trường hợp, các nhân viên làm phần mềm ra khỏi công ty đã sử dụng các mã nguồn phần mềm mình đã viết để làm những công việc khác, trong đó có cả việc làm ra các phần mềm cạnh tranh với chính phần mềm gốc.

– Đăng tải các tác phẩm, kể cả bản phái sinh mà không được phép của chủ sở hữu.

c) Tôn trọng bản quyền trong tin học

Trong tin học, khi mua phần mềm, cần phân biệt rõ việc mua bản quyền với mua quyền sử dụng (licence). Khi mua bản quyền thì người mua có quyền thực hiện các hoạt động kinh doanh đối với tác phẩm đó giống như cách Google mua Youtube. Còn nếu chỉ mua quyền sử dụng thì chỉ được sử dụng. Chúng ta mua hệ điều hành Windows, các phần mềm văn phòng như Word, Excel, PowerPoint đều chỉ là mua quyền sử dụng. Bản quyền của tất cả các phần mềm đó vẫn thuộc công ty Microsoft.

Quyền sử dụng phần mềm tính theo số máy được cài đặt. Nếu mua quyền sử dụng cho một máy, rồi lại cài đặt cho máy thứ hai là vi phạm bản quyền.

Vi phạm quyền tài sản sẽ gây thiệt hại trực tiếp đến hoạt động kinh doanh của các chủ sở hữu và gián tiếp đến toàn bộ ngành trong lĩnh vực đó. Hãy tưởng tượng, nếu các phần mềm tin học cứ làm ra là bị ăn cắp mà không bị trừng phạt thì không ai muốn đầu tư để làm phần mềm nữa. Trong lĩnh vực tin học, vi phạm bản quyền có thể gây thiệt hại rất lớn cho người đầu tư vì sản phẩm số có đặc tính:

– Dễ sao chép với chi phí rất thấp. Một công ty mất rất nhiều thời gian, tiền bạc, công sức để làm ra một phần mềm nhưng kẻ cắp thì chỉ mất một vài phút để sao chép.

– Dễ phát tán trên quy mô lớn. Nếu phần mềm bị phát tán thì công ty không thể bán được và có thể mất hết đầu tư.

Nhà nước Việt Nam đã có các quy định rất rõ ràng về những hành vi vi phạm quyền tác giả. Ví dụ, Nghị định số 131/2013/NĐ-CP quy định những hành vi vi phạm bản quyền như sau:

– "Truyền đạt tác phẩm đến công chúng mà không được phép của chủ sở hữu quyền tác giả". Việc đưa dữ liệu hay phần mềm của người khác lên mạng không được phép sẽ bị phạt.

– "Sao chép tác phẩm mà không được phép của chủ sở hữu quyền tác giả". Việc sao chép lậu dữ liệu hay phần mềm sẽ bị phạt.

– "Cố ý huỷ bỏ hoặc làm vô hiệu các biện pháp kĩ thuật, công nghệ do chủ sở hữu quyền tác giả thực hiện để bảo vệ quyền tác giả đối với tác phẩm của mình". Những ai bẻ khoá phần mềm (crack) để dùng sẽ bị phạt.

- Quyền tác giả là quyền của tổ chức, cá nhân đối với tác phẩm do mình sáng tạo ra hoặc sở hữu. Khái niệm quyền tác giả và bản quyền không hoàn toàn tương đồng, tuy nhiên, trong thực tế chúng thường được dùng chung. Trong các văn bản pháp luật Việt Nam sử dụng thuật ngữ chính thức là quyền tác giả.
- Phần mềm và dữ liệu số rất dễ bị xâm phạm bản quyền do dễ sao chép, dễ phát tán.
- Việc vi phạm bản quyền là hành vi phạm pháp, làm tổn hại đến việc kinh doanh của các chủ sở hữu, ảnh hưởng xấu đến sự phát triển của những ngành tạo ra sản phẩm trí tuệ, trong đó có tin học.
- Nhà nước đã ban hành nhiều quy định xử lí các hành vi vi phạm bản quyền. Hãy tôn trọng bản quyền để phát triển các ngành tạo ra sản phẩm trí tuệ, trong đó có tin học.

1. Trong các hành vi sau, những hành vi nào là vi phạm bản quyền?

A. Sao chép các đĩa cài đặt phần mềm.

B. Một người bạn của em mua tài khoản học một khoá tiếng Anh trực tuyến. Em mượn tài khoản để cùng học.

C. Phá khoá phần mềm chỉ để thử khả năng phá khoá chứ không dùng.

D. Em dùng nhờ một phần mềm trên máy tính của bạn.

2. Em hãy nêu ví dụ về một hành vi vi phạm bản quyền đối với sản phẩm tin học.

LUYỆN TẬP

1. Trong các ý kiến sau, em đồng ý, không đồng ý hay chỉ đồng ý một phần với các ý kiến nào? Tại sao?

a) Chúng ta có quyền đưa lên mạng xã hội tất cả các tin không phải là tin giả.

b) Chúng ta có quyền đưa lên mạng xã hội tất cả các tin miễn là không có hại đến cá nhân ai.

c) Chúng ta có quyền đưa lên mạng xã hội tất cả các tin miễn là không vi phạm pháp luật.

2. Trong đại dịch Covid-19, một người dùng Facebook đã chia sẻ tin "Bắt đầu từ ngày 28/3/2020, toàn thành phố Hồ Chí Minh sẽ bị phong toả trong 14 ngày,...". Khi bị triệu tập để xử phạt, người này đã chứng minh rằng anh ta chỉ đưa lại một tin chứ không bịa.

Người này sai ở đâu?

VẬN DỤNG

1. Nếu đăng trên mạng xã hội nhận xét có tính xúc phạm đến một người khác thì hành vi này là:

A. Vi phạm pháp luật.

B. Vi phạm đạo đức.

C. Tuỳ theo mức độ, có thể vi phạm đạo đức hay pháp luật.

D. Không vi phạm gì.

2. An nhắc Bình về việc Bình dùng phần mềm lậu và giảng giải cho Bình biết các quy định về quyền tác giả. Nghe xong Bình bảo "Trước đây mình không biết, mà không biết là không có lỗi". Quan niệm của Bình như vậy có đúng không?

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

Chủ đề
4

BÀI 12

Ứng dụng tin học

PHẦN MỀM THIẾT KẾ ĐỒ HOẠ

SAU BÀI NÀY EM SẼ:

- Biết được khái niệm về thiết kế đồ hoạ, phân biệt được đồ hoạ vectơ và đồ hoạ điểm ảnh.
- Sử dụng được các chức năng cơ bản của phần mềm thiết kế đồ hoạ Inkscape để vẽ hình đơn giản.

Em hãy quan sát hai hình sau và đưa ra nhận xét về màu sắc, độ nét và sự đa dạng các chi tiết của mỗi hình.



Hình 12.1a



Hình 12.1b

1. THIẾT KẾ ĐỒ HOẠ

Hoạt động 1 **So sánh giữa ảnh chụp và hình vẽ**

Thảo luận về sự khác nhau giữa ảnh chụp và hình vẽ bằng phần mềm.

Thiết kế đồ hoạ (graphic design) là quá trình thiết kế các thông điệp truyền thông bằng hình ảnh; giải quyết vấn đề thông qua sự kết hợp giữa hình ảnh, kiểu chữ với ý tưởng để truyền tải thông tin đến người xem.

Có hai loại đồ hoạ cơ bản: đồ hoạ điểm ảnh (bitmap) và đồ hoạ vectơ (vector). Trong đồ hoạ điểm ảnh, hình ảnh được tạo thành từ các điểm ảnh (pixel), mỗi điểm ảnh có màu riêng. Trong đồ hoạ vectơ, hình ảnh được xác định theo đường nét, mỗi một đường có điểm đầu và điểm cuối, được tính bằng một phương trình toán học. Sự khác biệt trong hai loại đồ hoạ này thể hiện rõ khi phóng to hình ảnh. Trong ví dụ ở phần khởi động, Hình 12.1a là ảnh chụp có độ chi tiết cao, đa dạng về màu sắc nhưng có thể bị nhoè, vỡ hình khi phóng to – hình bitmap, còn Hình 12.1b là hình vẽ bằng phần mềm thường là tổ hợp từ các hình khối đơn giản, số màu ít và giữ nguyên độ nét khi phóng to hay thu nhỏ – hình vectơ.



Ảnh chụp khi phóng to một phần



Hình vẽ khi phóng to một phần

Hình 12.2. *Phóng to một phần ảnh gốc*

So sánh giữa đồ hoạ điểm ảnh và đồ hoạ vectơ:

Đồ hoạ điểm ảnh	Đồ hoạ vectơ
Định nghĩa bằng tập điểm.	Định nghĩa bằng phương trình toán học.
Phù hợp chỉnh sửa ảnh, nhiều chi tiết, màu sắc liền mạch và chân thực.	Phù hợp tạo logo, minh hoạ và bản vẽ kĩ thuật,...
Phóng to có ảnh hưởng chất lượng hình.	Có thể co dãn mà không bị vỡ hình.
Ảnh lớn, độ chi tiết cao tương ứng tệp có kích thước lớn.	Tạo bản in với kích thước tuỳ ý, độ lớn của tệp không thay đổi.
Không thể chuyển sang đồ hoạ vectơ mà giữ nguyên chất lượng.	Dễ dàng chuyển sang đồ hoạ điểm ảnh.

- Thiết kế đồ hoạ là tạo ra sản phẩm bằng hình ảnh, chữ để truyền tải thông tin đến người xem.
- Hai loại đồ hoạ là đồ hoạ điểm ảnh (bitmap) và đồ hoạ vectơ.

1. Ảnh chụp là loại đồ hoạ nào?
2. Tại sao dùng đồ hoạ vectơ phù hợp hơn dùng đồ hoạ điểm ảnh khi thiết kế logo?

2. PHẦN MỀM ĐỒ HOẠ

Hoạt động 2 Phần mềm đồ hoạ

Minh muốn thiết kế logo cho câu lạc bộ bóng đá của trường nhưng bạn ấy không biết bắt đầu từ đâu. Theo em, Minh cần những gì?

Để tạo ra một logo, trước tiên bạn Minh cần có phần mềm thiết kế đồ hoạ cài trên máy tính. Có hai loại phần mềm đồ hoạ:

– Phần mềm tạo, chỉnh sửa hình vectơ: Adobe Illustrator, CorelDRAW, Inkscape,...

– Phần mềm xử lí ảnh bitmap: Adobe Photoshop, GIMP,...

Inkscape, GIMP là các phần mềm được sử dụng miễn phí. Tuỳ vào tính chất của công việc cần chọn loại phần mềm phù hợp. Ví dụ để thiết kế các sản phẩm cần in với kích thước đa dạng như logo, biển quảng cáo,... thì chọn phần mềm Inkscape, để xử lí ảnh dùng phần mềm GIMP.

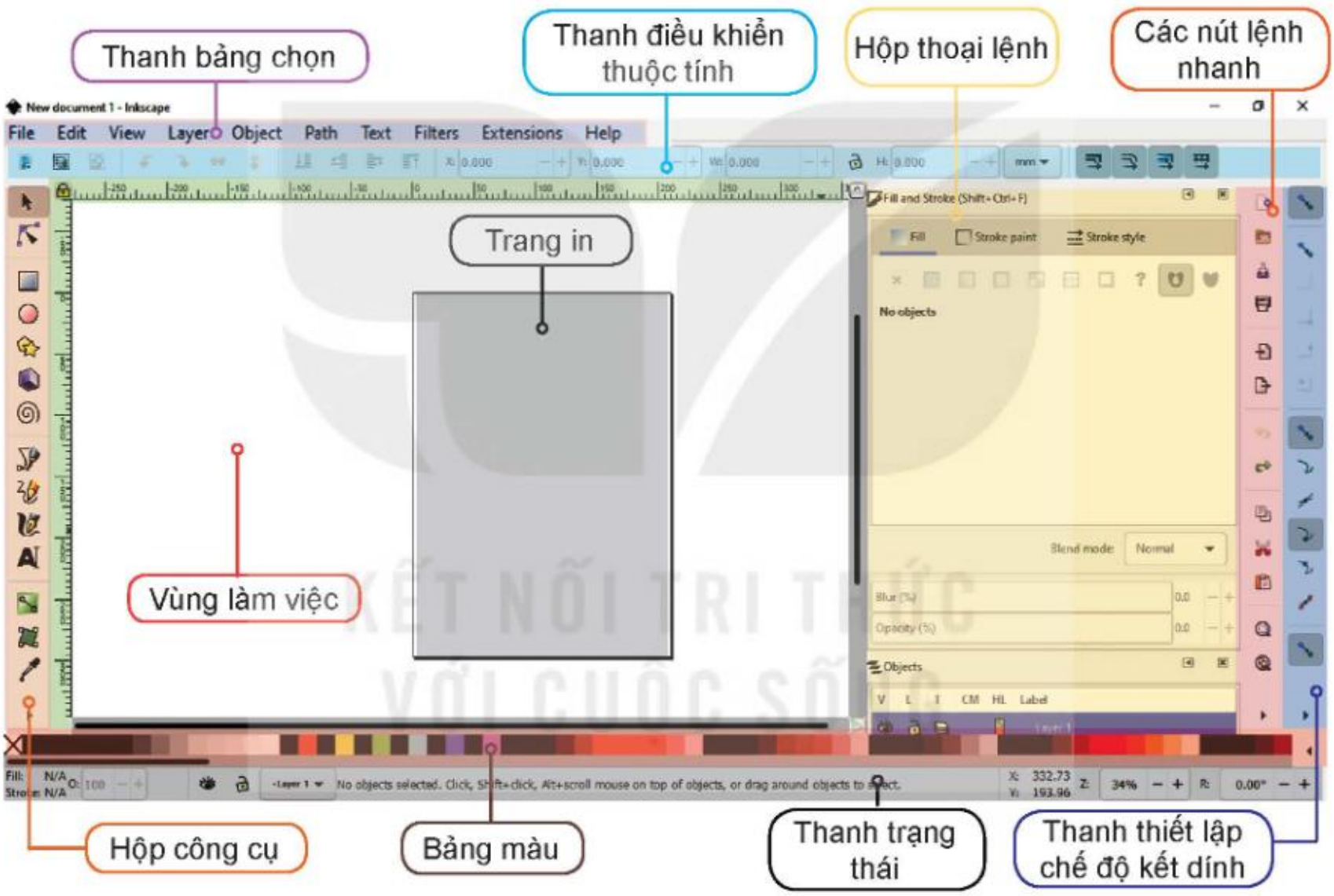
a) Tải và cài đặt phần mềm

Tải phần mềm tại địa chỉ: https://inkscape.org/release/inkscape-1.0/. Chọn phiên bản tương ứng với hệ điều hành đang sử dụng (Windows (32/64 bits), MacOS,...) để tải về máy. Sau khi tải xong, cài đặt theo hướng dẫn.

b) Giao diện của Inkscape

Nháy đúp chuột vào biểu tượng 

để khởi động phần mềm Inkscape. Màn hình làm việc của Inkscape tương tự như sau:



Hình 12.3. *Màn hình làm việc của Inkscape 1.0.1*

Thanh bảng chọn (Menu bar): Chứa các lệnh thường dùng liên quan đến tệp tin, các lệnh tạo và biến đổi đối tượng.

Hộp công cụ (Tool box): chứa các công cụ để khởi tạo, vẽ, điều chỉnh các đối tượng đồ hoạ. Đây là các công cụ làm việc chính, cơ bản của phần mềm.

Thanh điều khiển thuộc tính (Tool control bar): chứa thuộc tính của đối tượng đang được lựa chọn, các thuộc tính thay đổi tuỳ theo đối tượng đang chọn.

Vùng làm việc (Canvas): là toàn bộ phần nền trắng, ta sẽ thực hiện việc thêm các đối tượng vào để thu được hình vẽ. Phần trang in ứng với kích thước khi in của sản phẩm.

Bảng màu (Color Palette): chứa các màu có sẵn để thiết lập màu tô và màu vẽ của đối tượng.

Thanh trạng thái (Status bar): chứa chỉ báo kiểu, lớp vẽ hiện tại, vị trí con trỏ, mức thu phóng hiện tại, thay đổi kích thước cửa sổ,...

- Có hai loại phần mềm đồ hoạ: Phần mềm tạo, chỉnh sửa hình vectơ và phần mềm xử lí ảnh bitmap.
- Inkscape là phần mềm miễn phí để tạo, chỉnh sửa sản phẩm đồ hoạ vectơ.

1. Cần thiết kế một bộ các sản phẩm bút, sổ, danh thiếp, bì thư, túi giấy. Theo em nên dùng phần mềm nào?

A. Photoshop. B. Inkscape.

2. Em hãy cho biết có thể vẽ hình vào đâu trên màn hình làm việc của Inkscape.

A. Toàn bộ vùng làm việc. B. Trong khu vực trang in.

3. CÁC ĐỐI TƯỢNG ĐỒ HOẠ CỦA HÌNH VẼ

Hoạt động 3 Sắp xếp các đối tượng

Nếu chúng ta có một quả dưa hấu và một quả cam ở trên bàn. Xác định xem cần xếp hai quả này như thế nào để:

a) Nhìn thấy đủ cả hai quả, không quả nào bị che khuất.

b) Chỉ nhìn thấy quả dưa hấu.

c) Nhìn thấy đủ quả cam, quả dưa hấu bị che một phần.

d) Nhìn thấy đủ quả dưa hấu, quả cam bị che một phần.

Khi em khởi động Inkscape, phần mềm sẽ mở một tệp trống và sẵn sàng để cho em vẽ hình. Em cũng có thể tạo tệp mới bằng lệnh **File/New...** hoặc nhấn tổ hợp phím **Ctrl + N**.

Các đối tượng được thêm vào hình bằng cách tạo ra các bản sao của các đối tượng đã có trong vùng làm việc hoặc chọn đối tượng từ hộp công cụ và vẽ vào vùng làm việc. Các đối tượng sẽ xuất hiện theo thứ tự lớp, đối tượng vẽ trước sẽ ở lớp dưới, đối tượng này có thể bị che khuất toàn bộ hay một phần bởi các đối tượng vẽ sau ở lớp trên.

Để thêm các đối tượng có sẵn trên hộp công cụ như hình chữ nhật, hình vuông, hình tròn, hình elip, cung tròn, ngôi sao, đa giác, hình xoắn ốc hoặc vẽ đối tượng tự do hay chèn chữ,... ta thực hiện theo ba bước:

Bước 1: Chọn công cụ tương ứng trên hộp công cụ.



Hình 12.4. *Một số công cụ có sẵn trong hộp công cụ*

Bước 2: Chỉnh tuỳ chọn trên thanh điều khiển thuộc tính nếu cần.

Bước 3: Xác định vị trí hình vẽ trong vùng làm việc, kéo thả chuột để vẽ hình.

Ta có thể chọn đối tượng trên hình vẽ bằng cách chọn công cụ Select trong hộp công cụ rồi nháy chuột vào đối tượng. Đối tượng được chọn có thể được di chuyển, phóng to, thu nhỏ, xoay, thay đổi thứ tự lớp hay tô màu,...

Để chọn màu cho một đối tượng, ta cần xác định màu tô (**Fill Color**) và màu vẽ (**Stroke Color**). Ta chọn màu tô và màu vẽ trên bảng màu (lưu ý để chọn màu vẽ cần nhấn giữ phím **Shift** khi chọn màu).

Sau khi hoàn thành việc vẽ hình, em lưu tệp bằng lệnh **File/Save** hoặc nhấn tổ hợp phím **Ctrl + S**. Phần mềm Inkscape lưu lại sản phẩm đã tạo dưới dạng tệp hình vectơ có phần mở rộng là **svg**.

Mỗi hình vẽ bao gồm các đối tượng đồ hoạ. Các đối tượng này sẽ xuất hiện theo thứ tự lớp, các đối tượng vẽ trước sẽ ở lớp dưới, đối tượng vẽ sau sẽ ở lớp trên. Ta có thể thay đổi thứ tự lớp của đối tượng.

Theo em, thanh công cụ nào được sử dụng nhiều nhất trong Inkscape?

A. Bảng màu.
B. Thanh thiết lập chế độ kết dính.
C. Thanh điều khiển thuộc tính.
D. Hộp công cụ.

THỰC HÀNH

Nhiệm vụ 1. Vẽ một bông hoa (Hình 12.5).

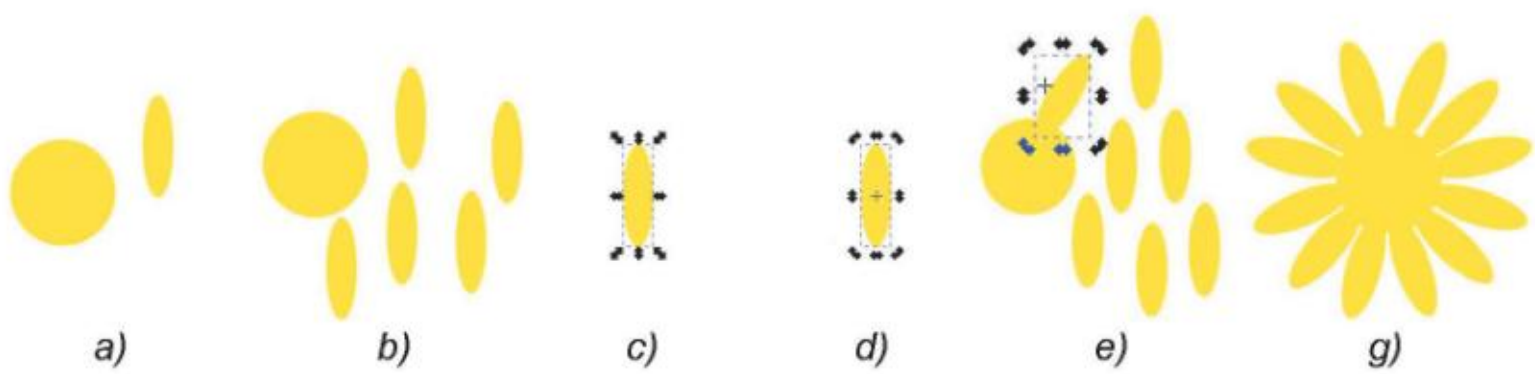
Hướng dẫn.

Bước 1: Chọn công cụ trên hộp công cụ, vẽ hình tròn và hình elip là nhuỵ hoa và cánh hoa, chọn màu bông hoa ở bảng màu (Hình 12.5a).

Bước 2: Chọn hình elip rồi nháy nút phải chuột, chọn **Duplicate** để có các bản sao của cánh hoa (Hình 12.5b).

Bước 3: Quay cánh hoa và di chuyển đến vị trí phù hợp.

- Chọn công cụ trên hộp công cụ. Nháy chuột lên một cánh hoa, nháy lần một để hiển thị chế độ phóng to thu nhỏ (Hình 12.5c), nháy lần hai để hiển thị chế độ quay (Hình 12.5d).
- Nháy chuột vào các mũi tên, kéo thả chuột để điều chỉnh cánh hoa quay hướng phù hợp (Hình 12.5e).
- Nháy chuột vào cánh hoa và kéo thả đến vị trí phù hợp (Hình 12.5g).



Hình 12.5. *Vẽ bông hoa từ hình tròn và những hình elip*

Nhiệm vụ 2. Vẽ Quốc kì Việt Nam (Hình 12.6).

Hướng dẫn.

Bước 1: Vẽ ngôi sao 5 cánh.

- Chọn công cụ trên hộp công cụ.

– Trên thanh điều khiển thuộc tính, chọn nút lệnh , đặt giá trị **Corners** là 5 và **Spoke ratio** là 0.4.

– Vẽ hình và tô màu vàng cho ngôi sao (quay cho thẳng nếu cần) (Hình 12.6a).

Bước 2: Vẽ hình chữ nhật tỉ lệ rộng và dài là 2 : 3, tô nền màu đỏ (Hình 12.6b). Vì hình chữ nhật vẽ sau nên khi di chuyển ngôi sao vào giữa hình chữ nhật thì ngôi sao sẽ bị che khuất.

Bước 3: Đưa hình chữ nhật xuống lớp dưới (Hình 12.6c). Chọn công cụ rồi chọn hình chữ nhật, nháy vào nút trên thanh điều khiển thuộc tính để đưa hình chữ nhật xuống lớp dưới.



a) *b)* *c)*

Hình 12.6. *Vẽ Quốc kì Việt Nam*

LUYỆN TẬP

1. Sau khi vẽ một hình tròn trong Inkscape. Ta thực hiện liên tục các thao tác sau:

Nháy vào công cụ trên hộp công cụ, nháy chuột lên hình tròn, nháy chuột chọn lần lượt các màu , , trên bảng màu, giữ phím **Shift** và nháy chuột vào màu trên bảng màu.

Em hãy xác định xem kết quả hình tròn sẽ có màu sắc như thế nào.

2. Để chuyển một ngôi sao thành một đa giác. Em sẽ tìm công cụ ở thanh công cụ nào?

A. Bảng màu. B. Thanh điều khiển thuộc tính.

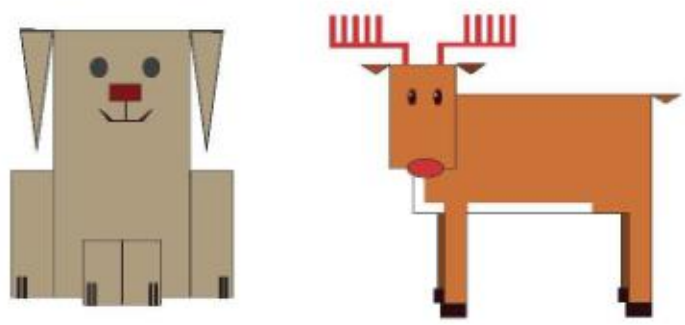
C. Hộp công cụ. D. Hộp thoại lệnh.

VẬN DỤNG

1. Hãy vẽ quốc kì các nước: Thái Lan, Lào, Myanmar và Anh.



2. Hãy vẽ các hình sau bằng các hình cơ bản.



BÀI 13 BỔ SUNG CÁC ĐỐI TƯỢNG ĐỒ HOẠ

SAU BÀI NÀY EM SẼ:

- Biết và sử dụng được một số chức năng của các lệnh tạo, điều chỉnh các đối tượng đồ hoạ đơn giản.
- Biết và thực hiện được các phép ghép đối tượng đồ hoạ.

Quan sát hình vẽ miếng dưa hấu ở Hình 13.1 và kể các đối tượng có trong hình vẽ. Xác định thứ tự các lớp của các đối tượng trong hình vẽ.



Hình 13.1. *Miếng dưa hấu*

1. CÁC ĐỐI TƯỢNG HÌNH KHỐI

Hoạt động 1 Các đối tượng hình khối

Các hình sau được vẽ từ một công cụ có sẵn của Inkscape. Theo em đó là công cụ nào? Thảo luận để tìm cách vẽ được các hình đó.



Hình 13.2

Inkscape cung cấp một số đối tượng đã được định nghĩa sẵn trong hộp công cụ như hình chữ nhật, hình vuông, hình tròn, hình elip, vòng cung, ngôi sao, đa giác,... Mỗi hình khối được đặc trưng bởi các thuộc tính khác nhau. Ta có thể thay đổi giá trị các thuộc tính trên thanh điều khiển thuộc tính để chỉnh hình theo ý muốn. Bảng sau đây mô tả thuộc tính cơ bản của một số đối tượng:

Bảng các thuộc tính cơ bản của một số hình có sẵn

Đối tượng	Thuộc tính	Ý nghĩa
Hình vuông, hình chữ nhật ( )	W, H,	Chiều rộng, chiều dài.
	Rx, Ry	Bán kính của góc bo (bằng 0 nếu góc của hình là góc vuông).
Hình tròn, hình elip ( )	Rx, Ry	Bán kính theo phương ngang và thẳng đứng.
	Start, End	Góc của điểm đầu và điểm cuối khi sử dụng chế độ vẽ (đơn vị: độ).

Hình đa giác, hình sao ( )	Corners	Số đỉnh của đa giác hoặc số cánh của hình sao.
	Rounded	Độ cong tại các đỉnh của hình.
	Spoke ratio	Tỉ lệ bán kính từ tâm đến góc trong và từ tâm tới đỉnh đầu hình sao.
	Randomized	Tham số làm méo hình ngẫu nhiên.

Ta có thể thay đổi thuộc tính của mỗi đối tượng được tạo từ công cụ có sẵn trên hình vẽ.

Nếu trên hình vẽ có sẵn một hình sao 5 cánh nhọn, em cần thay đổi tham số nào để các đỉnh ngôi sao trở nên cong?

2. THIẾT LẬP MÀU TÔ, MÀU VẼ VÀ TÔ MÀU CHO ĐỐI TƯỢNG

Hoạt động 2 **So sánh điểm khác nhau**

Quan sát các hình sau và nhận xét các hình có điểm gì khác nhau:



Hình 13.3

Trong Inkscape có một số tuỳ chọn khác nhau cho màu tô và màu vẽ của một đối tượng. Có thể thiết lập màu tô và màu vẽ độc lập với nhau. Để tuỳ chỉnh màu tô và màu vẽ ta sử dụng hộp thoại **Fill and Stroke** sau:

✕ Không màu (trong suốt).

Màu đồng nhất.

Màu chuyển giữa hai hoặc nhiều màu.

Màu chuyển giữa hai hoặc nhiều màu từ tâm của đối tượng.

Hoa văn (đối tượng được lấp đầy bởi một mẫu hoa văn).



Hình 13.4. *Hộp thoại Fill and Stroke*

? Huỷ đặt (loại bỏ tất cả các thiết lập liên quan đến màu sắc của đối tượng, điều này cần thiết khi ta sao chép đối tượng và thay đổi thuộc tính cho đối tượng).

Các bước thực hiện việc chỉnh sửa nền và đường nét:

Bước 1: Chọn đối tượng cần chỉnh, chọn lệnh **Objects/Fill and Stroke** (hoặc nháy nút phải chuột chọn **Fill and Stroke**) xuất hiện hộp thoại **Fill and Stroke**.

Bước 2: Chọn **Fill** để chọn kiểu tô cho màu tô, chọn **Stroke paint** để chọn kiểu tô cho màu vẽ, chọn **Stroke style** để thay đổi thiết lập kiểu nét vẽ và độ dày mỏng của nét.

Bước 3: Tuỳ chỉnh màu sắc bằng cách chọn kiểu tô và thiết lập màu.

Lưu ý: Khi tô màu chuyển thì màu của đối tượng được tô sẽ chuyển dần từ màu này sang màu khác. Để tô màu chuyển ta sử dụng hai kiểu tô là và . Nháy vào biểu tượng cây bút phía dưới của hộp thoại **Fill and Stroke** để thay đổi thông số của màu chuyển. Ta có thể thay đổi thông số của màu chuyển bằng bảng điều khiển hoặc kéo thả chuột tại các vị trí điều khiển.

Khi sử dụng kiểu tô ta có thể sao chép màu tại một vị trí nào đó trong vùng làm việc bằng cách sử dụng công cụ phía dưới của hộp thoại **Fill and Stroke**.

Giá trị **Opacity** thể hiện độ trong suốt của màu khi tô lên đối tượng, độ trong suốt của màu cũng được xác định bởi giá trị A (alpha) khi sử dụng bảng màu.

Có thể thiết lập màu tô, màu vẽ và các thuộc tính về màu tô và màu vẽ cho đối tượng.

1. Để xác định đường viền của đối tượng dạng nét đứt, em cần chọn trang nào trong hộp thoại **Fill and Stroke**?

A. Fill. B. Stroke paint. C. Stroke style. D. Cả A và B.

2. Để chỉnh thông số của màu chuyển, em cần chọn biểu tượng nào?

A. B. C. D.

3. CÁC PHÉP GHÉP ĐỐI TƯỢNG ĐỒ HOẠ

Hoạt động 3 Ghép các đối tượng hình khối

Tìm cách xếp ba mẩu giấy như Hình 13.5 thành một hình trái tim.

Gợi ý. Em có thể xoay hình và xếp các hình lên nhau.



Hình 13.5

Các hình phức tạp có thể thu được bằng cách ghép từ các hình đơn giản.

Các phép ghép được sử dụng để ghép và cắt hình trong Inkscape gồm: hợp, hiệu, giao, hiệu đối xứng, chia, cắt của hai hay nhiều đối tượng đơn. Các phép ghép này được thực hiện bằng cách chọn lệnh trong bảng chọn **Path**.

Phép hợp (Union, nhấn tổ hợp phím **Ctrl** + **+**): là tất cả các phần thuộc một trong các hình đơn.

Phép hiệu (Difference, nhấn tổ hợp phím **Ctrl** + **–**): là phần thuộc hình lớp dưới nhưng không thuộc hình lớp trên.

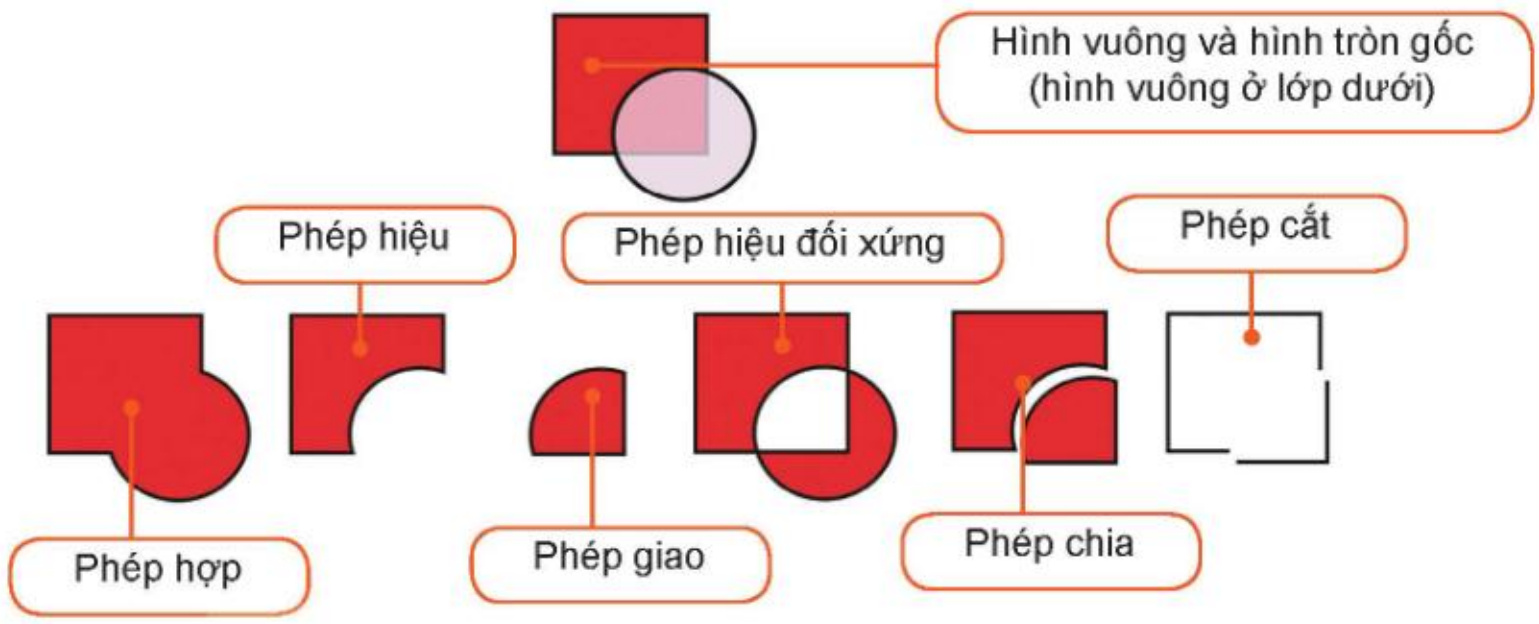
Phép giao (Intersection, nhấn tổ hợp phím **Ctrl** + ***): là phần thuộc cả hai hình được chọn.

Phép hiệu đối xứng (Exclusion, nhấn tổ hợp phím **Ctrl** + **^**): là phần hình thuộc các hình trừ phần giao nhau.

Phép chia (Division, nhấn tổ hợp phím **Ctrl** + */*): Hình lớp dưới được chia thành các phần bởi đường nét của hình lớp trên.

Phép cắt (Cut Path, nhấn tổ hợp phím **Ctrl** + **Alt** + *I*): Cắt hình lớp dưới thành các phần bởi các điểm giao ở viền với hình lớp trên. Kết quả là các hình mới không có màu.

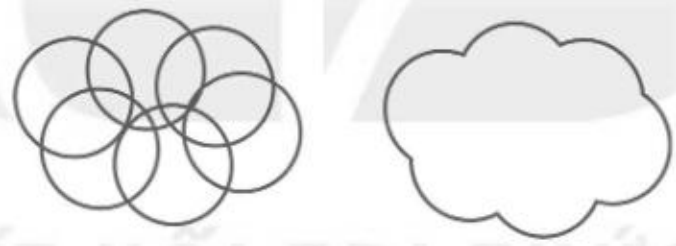
Hình 13.6 thể hiện kết quả của các phép ghép trên hai đối tượng:



Hình 13.6. *Các phép ghép hai hay nhiều hình*

Trong Inkscape, ta có thể làm các phép ghép với các hình để thu được hình mới.

Em hãy nêu phép ghép hình và các bước thực hiện để vẽ đám mây như Hình 13.7.



Hình 13.7. *Vẽ đám mây*

THỰC HÀNH

Nhiệm vụ 1. Vẽ logo tương tự Hình 13.8.



Hình 13.8. *Vẽ logo*

Hướng dẫn.

Bước 1: Chọn công cụ trên hộp công cụ, vẽ hình tròn (H.13.9a).

Bước 2: Chọn công cụ trên hộp công cụ, kéo thả chuột ở nút tròn trên hình để thay đổi độ mở (H.13.9b).

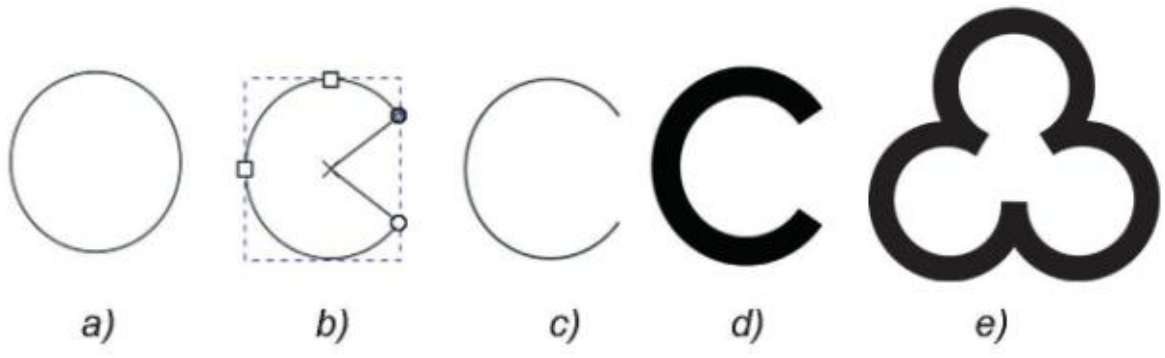
Bước 3: Trên thanh điều khiển thuộc tính, chọn dạng của hình là (H.13.9c).

Bước 4: Nháy nút phải chuột vào hình chọn **Fill and Stroke**, trong hộp thoại **Fill and Stroke** chọn **Stroke style** và thay đổi giá trị trong ô **Width** để có độ dày phù hợp (H.13.9d).

Bước 5: Tạo bản sao của hình (nhấn tổ hợp phím **Ctrl** + **D**), quay hình và di chuyển vào vị trí phù hợp (H.13.9e).

Bước 6: Thiết lập màu vẽ như Hình 13.8.

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG



a) b) c) d) e)

Hình 13.9. *Các bước vẽ logo*

Nhiệm vụ 2. Vẽ hình cây như Hình 13.10.

Hướng dẫn.



Hình 13.10. *Hình cây*

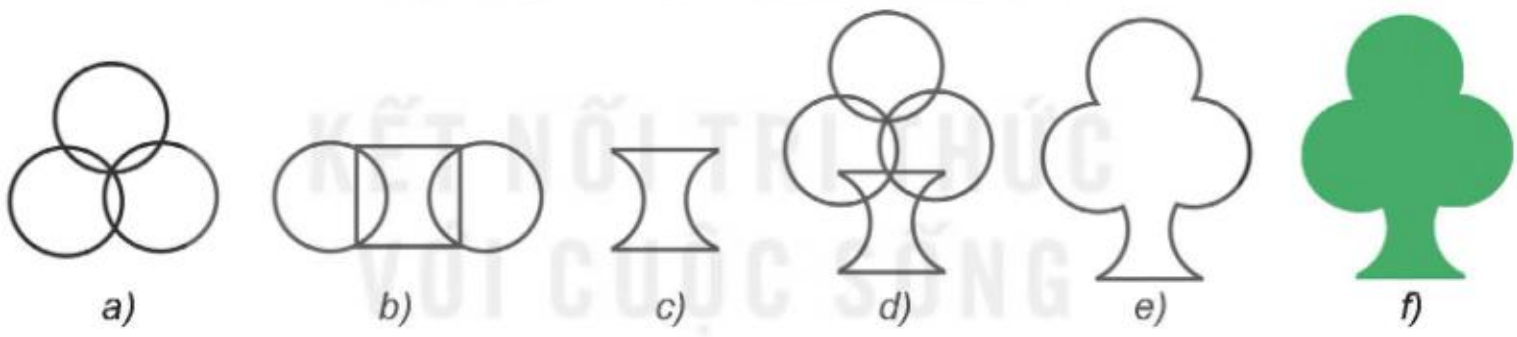
Bước 1: Vẽ một vòng tròn và sao chép thành ba hình, di chuyển về vị trí thích hợp (Hình 13.11a).

Bước 2: Vẽ hình chữ nhật và hai hình tròn như (Hình 13.11b). Chú ý: hình chữ nhật vẽ trước hoặc chuyển xuống lớp dưới hai hình tròn.

Bước 3: Chọn hình chữ nhật và một hình tròn, rồi chọn lệnh **Path/Difference**. Tiếp tục chọn hình vừa được tạo với hình tròn còn lại, rồi chọn lệnh **Path/Difference** (Hình 13.11c).

Bước 4: Di chuyển kết quả của bước 3 vào vị trí của hình trong bước 1 (Hình 13.11d).

Bước 5: Chọn cả bốn phần trong Hình 13.11d và chọn lệnh **Path/Union** (Hình 13.11e), tô màu xanh cho hình (Hình 13.11f).



a) b) c) d) e) f)

Hình 13.11. *Các bước vẽ hình cây*

Nhiệm vụ 3. Vẽ hình cầu như Hình 13.12.

Hướng dẫn.



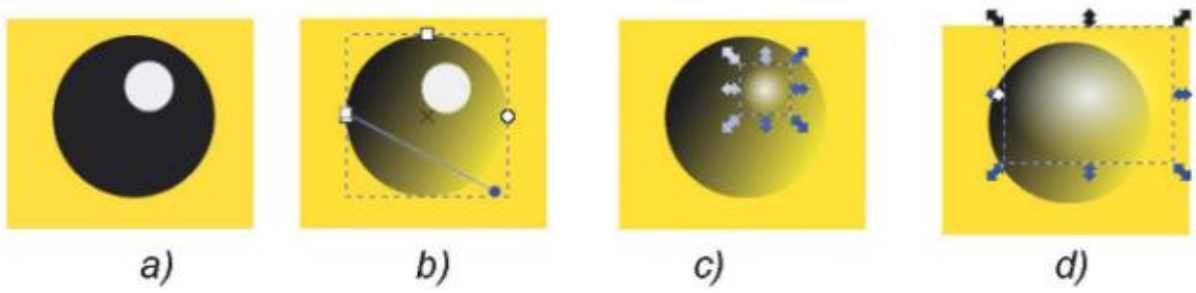
Hình 13.12. *Hình cầu*

Bước 1: Vẽ các hình có màu vẽ trong suốt, màu tô của hình chữ nhật là màu vàng, màu tô của hình tròn to là màu đen và màu tô của hình tròn nhỏ là màu trắng (Hình 13.13a).

Bước 2: Chọn hình tròn màu đen. Chọn biểu tượng , sau đó chọn biểu tượng  trong hộp thoại **Fill and Stroke**, di chuyển các điểm điều khiển để hình tròn đen chuyển màu từ trái qua phải, từ trên xuống dưới (Hình 13.13b).

Bước 3: Chọn hình tròn màu trắng, chọn biểu tượng  (Hình 13.13c).

Bước 4: Tăng cỡ hình của hình tròn trắng và di chuyển các hình cho phù hợp (Hình 13.13.d). Ta được sản phẩm hình cầu.



a) b) c) d)

Hình 13.13. *Vẽ hình cầu*

LUYỆN TẬP

1. Hãy vẽ một hình sao rồi thay đổi giá trị **Rounded** và quan sát tác động của thuộc tính này. Vẽ bông hoa Hình 13.14 bằng hình sao và hình tròn.

Gợi ý: Tuỳ chỉnh thuộc tính **Corners**, **Rounded**, **Spoke ratio** của ngôi sao.

2. Hãy vẽ hình như Hình 13.15.

3. Hãy vẽ chùm bóng (Hình 13.16).

Gợi ý: Sử dụng hình tròn, cắt hình để tạo hiệu ứng ánh sáng, sử dụng công cụ để vẽ dây.



Hình 13.14. *Vẽ bông hoa*



Hình 13.15



Hình 13.16. *Chùm bóng*

VẬN DỤNG

1. Hãy vẽ miếng dưa hấu Hình 13.1.

2. Sử dụng kiến thức học trong bài, hãy vẽ hình trong không gian.

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

BÀI 14 LÀM VIỆC VỚI ĐỐI TƯỢNG ĐƯỜNG VÀ VĂN BẢN

SAU BÀI NÀY EM SẼ:

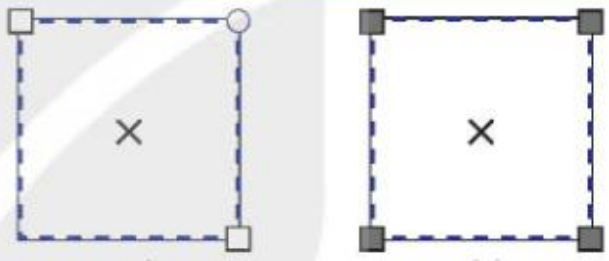
- Biết các thao tác chỉnh sửa hình.
- Thực hiện được việc vẽ, chỉnh sửa hình để được đường cong mong muốn.
- Biết tạo và định dạng văn bản.

Cần ít nhất bao nhiêu điểm để xác định một đường thẳng? Đường parabol? Đường elip?

1. LÀM QUEN VỚI ĐỐI TƯỢNG DẠNG ĐƯỜNG

Hoạt động 1 So sánh hai hình chữ nhật

Quan sát hai hình chữ nhật ở Hình 14.1 và tìm ra điểm khác nhau giữa hai hình.

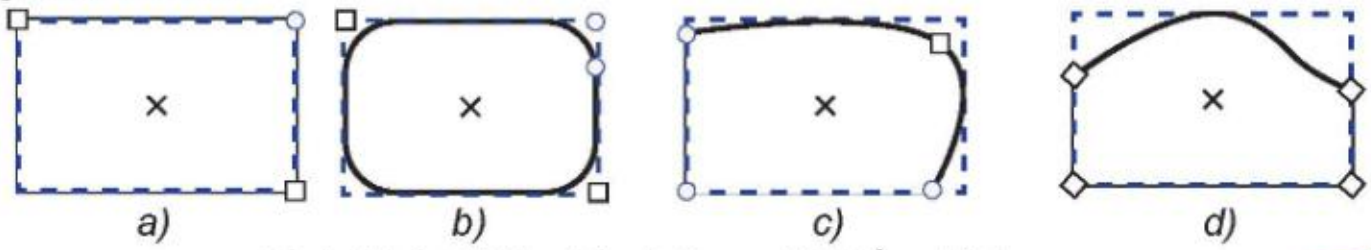


Hình 14.1

Hình khối là đối tượng được định nghĩa sẵn trong Inkscape và xác định bởi các tính chất toán học chặt chẽ. Đối tượng tự do dạng đường do người dùng tạo ra, là tổ hợp của một hay nhiều đoạn cong hoặc thẳng nối lại với nhau.

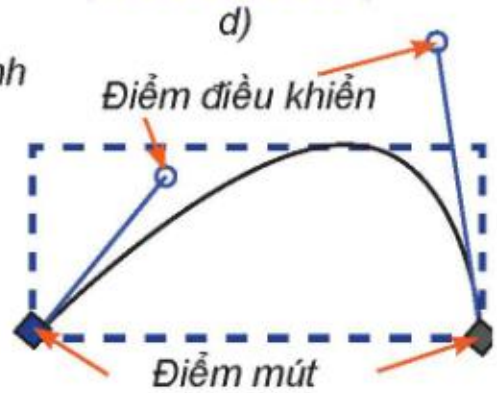
Khi điều chỉnh hình khối ta thu được hình khối mới với đặc trưng không đổi, trong khi đó với các đối tượng tự do dạng đường có thể điều chỉnh các đoạn độc lập với nhau để tạo ra hình dạng khác.

Ví dụ, đối tượng hình khối chữ nhật (Hình 14.2a) cho phép thay đổi thuộc tính là độ cong tại bốn góc (Hình 14.2b). Trong khi đó Hình 14.2c và Hình 14.2d là các đối tượng tự do có hình dạng ban đầu là hình chữ nhật có thể điều chỉnh thành các hình dạng khác nhau.



Hình 14.2. *Hình chữ nhật sau khi điều chỉnh*

Đối tượng đường thường được biểu diễn bởi một chuỗi các đoạn thẳng hoặc đoạn cong nối với nhau. Mỗi đoạn cong biểu diễn bởi bốn điểm, hai điểm đầu mút và hai điểm điều khiển. Mỗi điểm điều khiển kết hợp với một điểm đầu mút tạo ra tiếp tuyến của đường cong tại điểm mút tương ứng (Hình 14.3).



Hình 14.3. *Một đoạn cong*

Các bước vẽ đối tượng đường:

Bước 1: Chọn công cụ Pen  trên hộp công cụ.

Bước 2: Chọn kiểu  trên thanh điều khiển thuộc tính để tạo đường cong.

Bước 3: Nháy chuột để đặt các điểm neo trên hình vẽ (có thể kết hợp nháy chuột và kéo thả).

Bước 4: Kết thúc đường bằng cách nhấn phím **Enter** hoặc nháy đúp chuột tại vị trí neo cuối cùng.

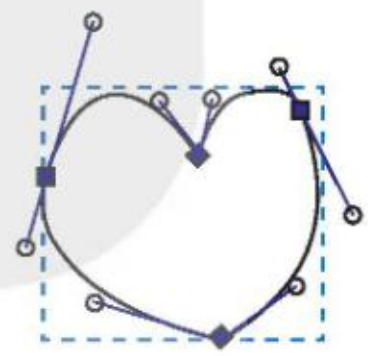
Đường cong thường được biểu diễn bởi một chuỗi các đoạn cong ghép với nhau.

Để vẽ một hình chữ nhật tròn góc em nên dùng công cụ nào? Giải thích tại sao.

2. SỬ DỤNG CÔNG CỤ TINH CHỈNH ĐƯỜNG

Hoạt động 2 Nhận dạng điểm neo

Quan sát hình trái tim, xác định xem các điểm được đánh dấu nằm trên Hình 14.4 có những đặc điểm gì?

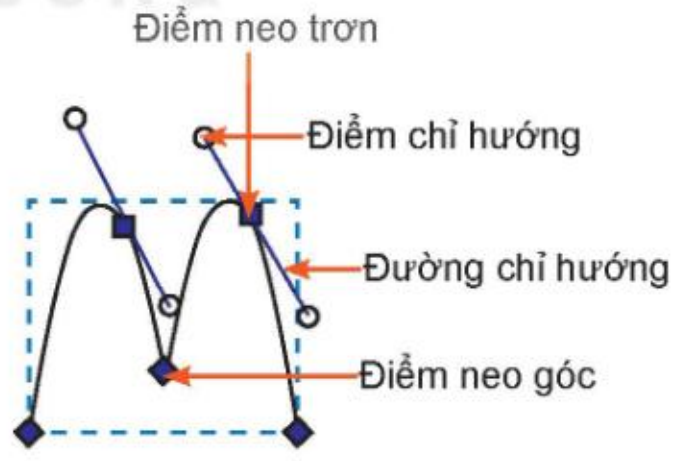


Hình 14.4. *Điểm neo góc và điểm neo trơn*

Khi nối các đoạn thẳng hoặc đoạn cong với nhau ta thu được đường cong phức tạp hơn.

Các điểm nối giữa các đoạn có thể là điểm **neo trơn** (smooth nodes – thể hiện bởi hình vuông hay hình tròn) hoặc điểm **neo góc** (corner nodes – thể hiện bởi một hình thoi) (Hình 14.4).

Khi xác định điểm neo trên bản vẽ, các đoạn ở giữa sẽ được tự động tạo ra để nối các điểm neo đã có. Hình vẽ ban đầu có thể chưa đúng với ý tưởng hoàn chỉnh, điều này sẽ được chỉnh sửa bằng công cụ tinh chỉnh điểm .



Hình 14.5. *Điểm và đường*

Độ cong tại mỗi điểm neo phụ thuộc vào **điểm chỉ hướng** và **đường chỉ hướng** tại điểm đó (Hình 14.5). Độ cong tại một điểm neo thay đổi bằng cách kéo thả điểm chỉ hướng.

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

Điểm neo trơn có hai đường chỉ hướng luôn cùng phương với nhau.

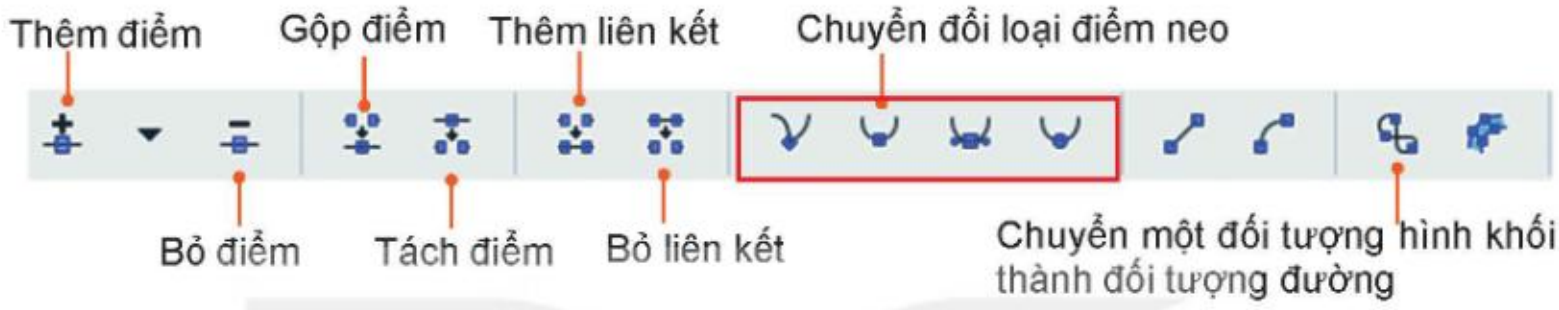
Các bước thực hiện việc chỉnh sửa điểm neo:

Bước 1: Chọn công cụ trên hộp công cụ.

Bước 2: Nháy chuột vào hình muốn chỉnh sửa.

Bước 3: Nháy vào điểm neo cần sửa, chọn điểm neo hoặc điểm chỉ hướng rồi kéo thả chuột sang vị trí mới.

Nếu muốn loại bỏ, thêm mới hay chuyển đổi loại điểm neo,... ta chọn vào biểu tượng tương ứng trên thanh điều khiển thuộc tính (Hình 14.6).



Hình 14.6. *Các biểu tượng trên thanh điều khiển thuộc tính*

Cụ thể:

– Để chuyển loại điểm neo, ta nháy chuột vào điểm neo muốn chuyển và nháy chuột vào biểu tượng điểm neo tương ứng (là neo góc, còn lại là neo trơn).

– Muốn thêm một điểm neo vào giữa hai điểm neo, ta nháy chuột vào đoạn đó trên đường cong và chọn biểu tượng thêm điểm.

– Muốn bỏ một hay nhiều điểm neo, ta chọn các điểm này và chọn biểu tượng bỏ điểm.

– Muốn gộp hai hay nhiều điểm ta chọn các điểm đó và chọn biểu tượng gộp điểm.

– Biểu tượng tách điểm sẽ tách điểm được chọn thành hai điểm rời nhau, sau khi thực hiện tách điểm, ta nháy chuột vào vị trí điểm neo và kéo ra chỗ khác sẽ thấy được hai điểm tách rời – không có đường nối liên kết ở giữa.

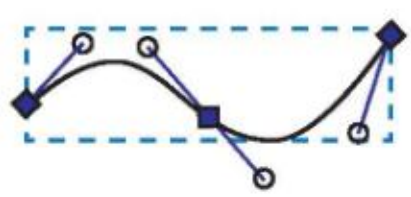
– Giữa hai điểm neo không liên kết và vẫn còn dư liên kết (mỗi điểm mới liên kết với không quá một điểm khác) ta có thể thêm đường nối bằng cách chọn biểu tượng thêm liên kết.

– Với hai điểm neo đã có sẵn liên kết, đoạn liên kết này có thể bỏ đi bằng cách chọn biểu tượng bỏ liên kết.

Ta có thể tinh chỉnh đối tượng đường dựa vào điểm neo và các điểm, đường chỉ hướng.



Hình 14.7 có mấy đoạn cong? Xác định điểm neo trơn và neo góc của hình.



Hình 14.7

3. ĐỐI TƯỢNG VĂN BẢN

Hoạt động 3 **Tìm hiểu đối tượng văn bản**

Trong các cách trình bày văn bản sau, em thấy cách nào đẹp hơn? Các phần mềm em đã học có thể thiết kế văn bản như vậy không?

TIẾT KIỆM ĐIỆN, NƯỚC LÀ GÓP PHẦN BẢO VỆ MÔI TRƯỜNG



a)



b)

Hình 14.8. *Trình bày văn bản trên hình*

Inkscape có thể tạo các đối tượng văn bản như tiêu đề và chú thích,... một cách đơn giản. Đồng thời có thể điều chỉnh các đối tượng này để cho sinh động và phù hợp với nội dung.

Để bổ sung đối tượng văn bản, ta chọn biểu tượng  trên hộp công cụ. Inkscape cung cấp các tính năng chuẩn để điều chỉnh một đoạn văn bản như thay đổi phông chữ, kiểu chữ, màu chữ, kích thước, khoảng cách giữa các dòng và căn chỉnh lề đoạn văn bản. Văn bản có thể nằm ngang hoặc dọc. Mỗi kí tự trong văn bản có thể được chọn và tuỳ chỉnh riêng lẻ. Văn bản có thể được đặt trong một khuôn dạng hoặc uốn lượn theo một đường có sẵn hay theo một đường có hình dạng tuỳ ý.

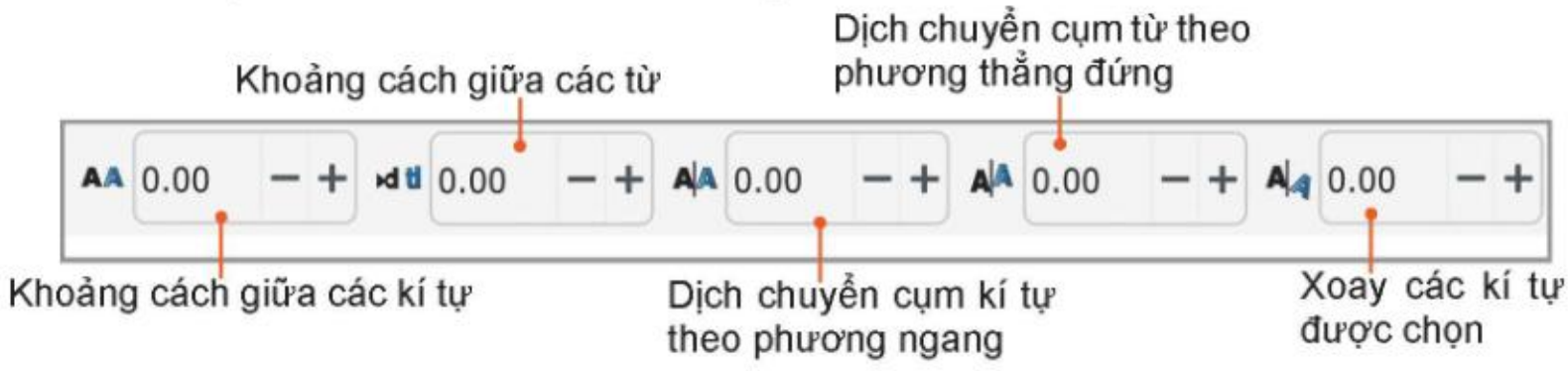
Ta có thể tuỳ chỉnh *từng phần* trong một đối tượng văn bản bằng cách chọn đối tượng văn bản và bôi đen phần văn bản muốn căn chỉnh, sau đó điều chỉnh tham số trên thanh điều khiển thuộc tính hoặc lệnh trong bảng chọn **Text**.

Bốn ô đầu tiên trên thanh điều khiển thuộc tính (Hình 14.9) cho phép ta thay đổi phông chữ, kiểu chữ (bình thường/ in nghiêng/ in đậm/ in đậm và nghiêng), cỡ chữ và khoảng cách giữa các dòng,...



Hình 14.9. *Điều chỉnh phông chữ, cỡ chữ, kiểu chữ, khoảng cách dòng*

Năm ô phía sau điều chỉnh các thông tin như trên Hình 14.10.



Hình 14.10. *Điều chỉnh thuộc tính từng cụm kí tự*

Ví dụ: Với đoạn văn bản:

TIẾT KIỆM ĐIỆN, NƯỚC LÀ GÓP PHẦN BẢO VỆ MÔI TRƯỜNG

Trong đó ĐIỆN, NƯỚC là cụm kí tự được bôi đen, kết quả thu được khi ta thay đổi giá trị tham số như sau:

Bảng các lệnh điều chỉnh đối tượng văn bản

Ý nghĩa	Lệnh	Kết quả
Thay đổi khoảng cách giữa các kí tự	AA 20.00 − +	TIẾT KIỆM Đ I Ệ N , N Ư Ớ C LÀ GÓP PHẦN BẢO VỆ MÔI TRƯỜNG
Thay đổi khoảng cách giữa các từ	20.00 − +	TIẾT KIỆM ĐIỆN, NƯỚC LÀ GÓP PHẦN BẢO VỆ MÔI TRƯỜNG
Dịch cụm kí tự theo chiều ngang	A|A 20.00 − +	TIẾT KIỆM ĐIỆN LÀ NƯỚC PHẦN BẢO VỆ MÔI TRƯỜNG
Dịch cụm kí tự theo chiều thẳng đứng	A|A 20.00 − +	TIẾT KIỆM LÀ GÓP PHẦN BẢO VỆ MÔI TRƯỜNG ĐIỆN, NƯỚC
Quay cụm kí tự được chọn một góc (20 độ)	A|A 20.00 − +	TIẾT KIỆM ĐIỆN, NƯỚC LÀ GÓP PHẦN BẢO VỆ MÔI TRƯỜNG

Muốn bỏ các tuỳ chỉnh đã đặt, ta chọn đối tượng văn bản rồi chọn lệnh **Text/ Remove Manual Kerns**.

Muốn đặt văn bản theo đường đã có, ta chọn đối tượng văn bản và đối tượng đường, sau đó chọn lệnh **Text/Put on Path**.

Muốn bỏ đặt văn bản theo đường, ta chọn đối tượng và chọn lệnh **Text/ Remove from Path**.

Trong Inkscape, văn bản được bổ sung có thể tuỳ chỉnh theo từng kí tự và có thể đặt đoạn văn bản theo đường hoặc vào một khuôn dạng.

Hãy cho biết phát biểu nào sau đây là **sai** khi làm việc với đoạn văn bản trong Inkscape.

A. Trong một đoạn văn có nhiều chữ ta có thể tô mỗi chữ bằng một màu khác nhau.

B. Nếu đặt đoạn văn uốn lượn theo một đường, ta không thể thay đổi định dạng đó.

C. Trong một đoạn văn có nhiều chữ, ta có thể tuỳ chỉnh để mỗi chữ độ cao thấp khác nhau.

D. Ta có thể đặt đoạn văn bản theo một khuôn dạng nhất định.

THỰC HÀNH

Nhiệm vụ 1. Vẽ chiếc lá và tô màu cho chiếc lá (Hình 14.11).

Hướng dẫn.

Bước 1: Vẽ hình chữ nhật ta được hình chữ nhật với các điểm neo như Hình 14.12a.



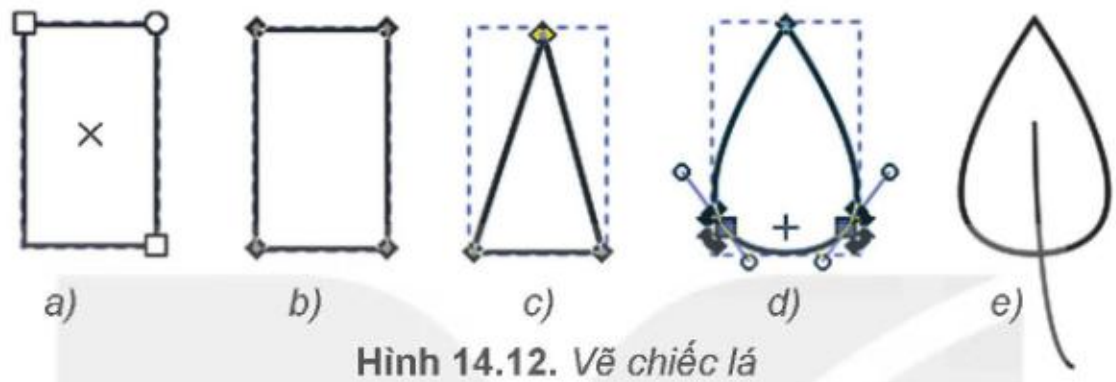
Hình 14.11. *Chiếc lá*

Bước 2: Chọn hình chữ nhật và chọn lệnh **Path/Object to Path** để đưa hình chữ nhật thành bốn cạnh với các điểm neo rời nhau (Hình 14.12b).

Bước 3: Chọn hai điểm neo cạnh trên của hình chữ nhật để nhập thành một điểm bằng cách chọn biểu tượng trên thanh điều khiển thuộc tính. Ta được hình tam giác (Hình 14.12c).

Bước 4: Chọn hai điểm neo ở cạnh dưới của hình tam giác để chuyển thành điểm neo trơn bằng chọn nút lệnh (Hình 14.12d).

Bước 5: Thêm cuống lá bằng công cụ Pen trên thanh công cụ (Hình 14.12e).



Hình 14.12. *Vẽ chiếc lá*

Bước 6: Tô màu cho lá như Hình 14.11.

Nhiệm vụ 2. Viết chữ theo khuôn dạng của hình (Hình 14.13).



Hình 14.13. *Viết chữ theo hình*

Hướng dẫn.

Bước 1: Chọn công cụ Pen , chọn kiểu vẽ trên thanh thuộc tính điều khiển.

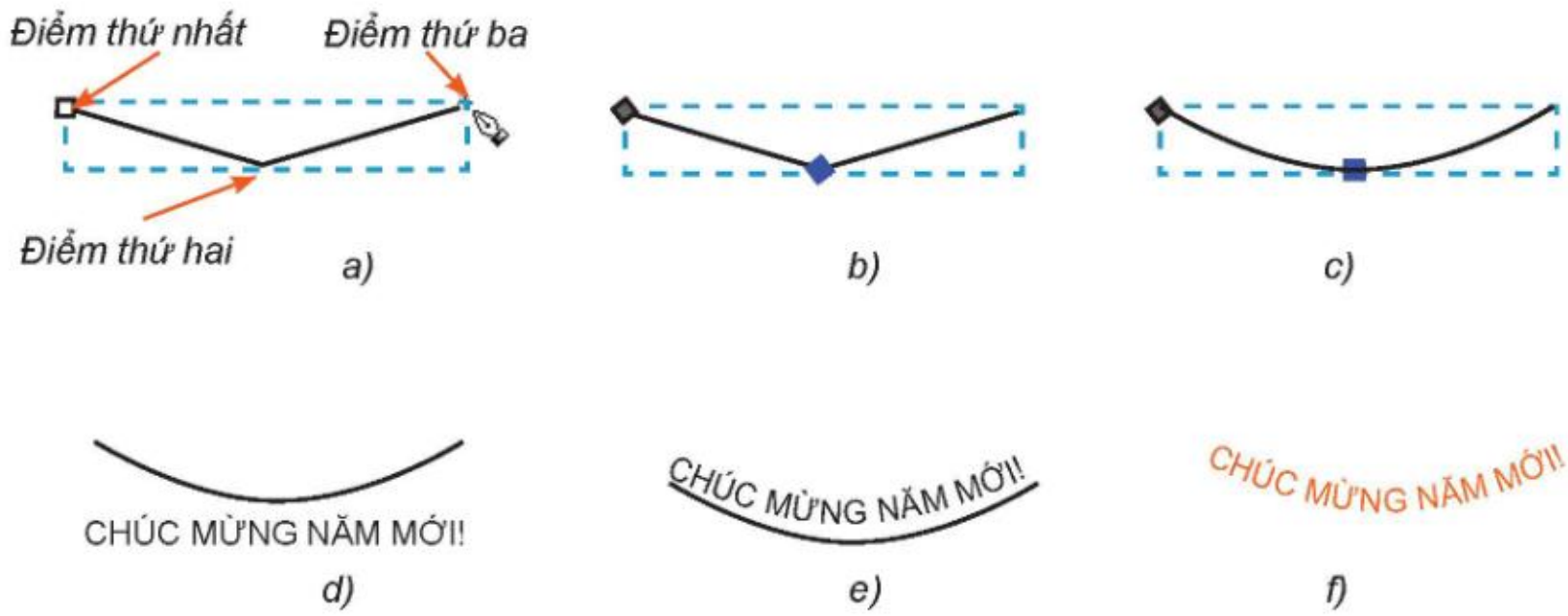
Bước 2: Nháy chuột vào các điểm để xác định điểm neo thứ nhất và thứ hai, cuối cùng nháy đúp chuột tại điểm neo thứ ba, ta nhận được hình tương tự Hình 14.14a.

Bước 3: Chọn công cụ trên hộp công cụ, ta xác định được các điểm neo (Hình 14.14b). Chọn điểm neo thứ hai, rồi chọn biểu tượng để chuyển điểm neo góc thành neo trơn (Hình 14.14c).

Bước 4: Nháy vào biểu tượng trong hộp công cụ và gõ dòng chữ "Chúc mừng năm mới!" (Hình14.14d).

Bước 5: Chọn hai đối tượng đường cong và dòng chữ. Chọn lệnh **Text/Put on Path** để đặt dòng chữ uốn theo đường cong (Hình14.14e). Tô màu cho chữ và chọn nét đường cong không màu (Hình14.14f).

Bước 6: Vẽ hình tam giác tương tự nhiệm vụ 1. Sao chép, xoay, tô màu và di chuyển đến vị trí phù hợp (Hình 14.13).



Hình 14.14. *Các bước vẽ đường cong và tạo chữ theo đường cong*

LUYỆN TẬP

1. Hãy vẽ hình con chuột (Hình 14.15).

2. Hãy vẽ chiếc lá và tô màu (Hình 14.16).



Hình 14.15



Hình 14.16

3. Em hãy mở lại hình miếng dưa hấu đã vẽ ở bài trước và thêm chi tiết để được hình như Hình 14.17.



Hình 14.17. *Vẽ dưa hấu*

VẬN DỤNG

Em hãy sưu tầm các mẫu logo đơn giản và vẽ lại mẫu.

BÀI 15

HOÀN THIỆN HÌNH ẢNH ĐỒ HOẠ

SAU BÀI NÀY EM SẼ:

- Tạo được sản phẩm số đơn giản, hữu ích và thực tế như thiết kế logo, tạo banner, topic quảng cáo, băng-rôn, áp phích, poster và thiệp chúc mừng,...

Em được giao nhiệm vụ thiết kế tờ rơi quảng bá cho Hội chợ sách.
Thảo luận theo nhóm để trả lời các câu hỏi sau:

– Tờ rơi cho Hội chợ sách cần cung cấp cho người xem những thông tin gì?

– Xếp thứ tự độ quan trọng của các thông tin mà nhóm đã chọn.

– Để phù hợp với nội dung là Hội chợ sách, nên có những hình ảnh minh hoạ gì?

– Phân loại các nhóm đối tượng tạo nên tờ rơi.

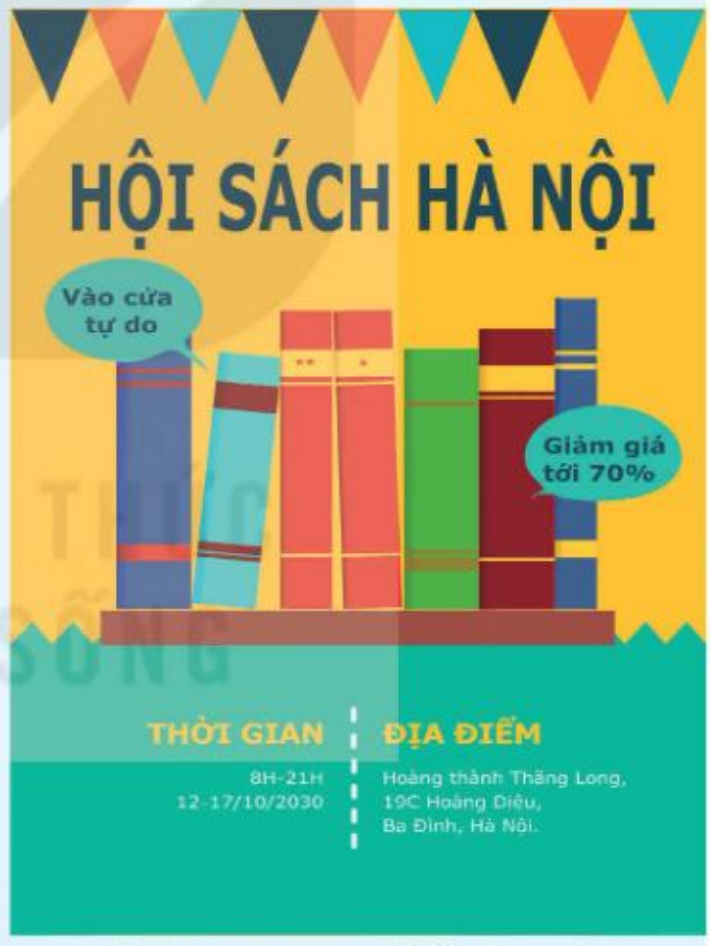
Tờ rơi cần được thiết kế sáng sủa, dễ nắm bắt thông tin. Trong bài thực hành này, chúng ta sẽ tiến hành thiết kế tờ rơi cho Hội chợ sách theo mẫu Hình 15.1.

Hướng dẫn thực hiện: Hình vẽ gồm một số thành phần, ta sẽ chia nhỏ thành các nhóm đối tượng để vẽ, sau đó sắp xếp các đối tượng vào vị trí phù hợp và chỉnh sửa màu sắc (nếu cần).

Các nhóm đối tượng cần vẽ:

1. Nền.
2. Dây cờ tam giác.
3. Sách và giá sách.
4. Các đoạn văn bản.
5. Hai bóng hộp thoại.
6. Đoạn nét đứt nằm giữa thời gian và địa điểm.

Ta sẽ tiến hành vẽ từng đối tượng.



Hình 15.1. *Tờ rơi Hội chợ sách*

Nhiệm vụ 1. Vẽ nền.

Hướng dẫn.

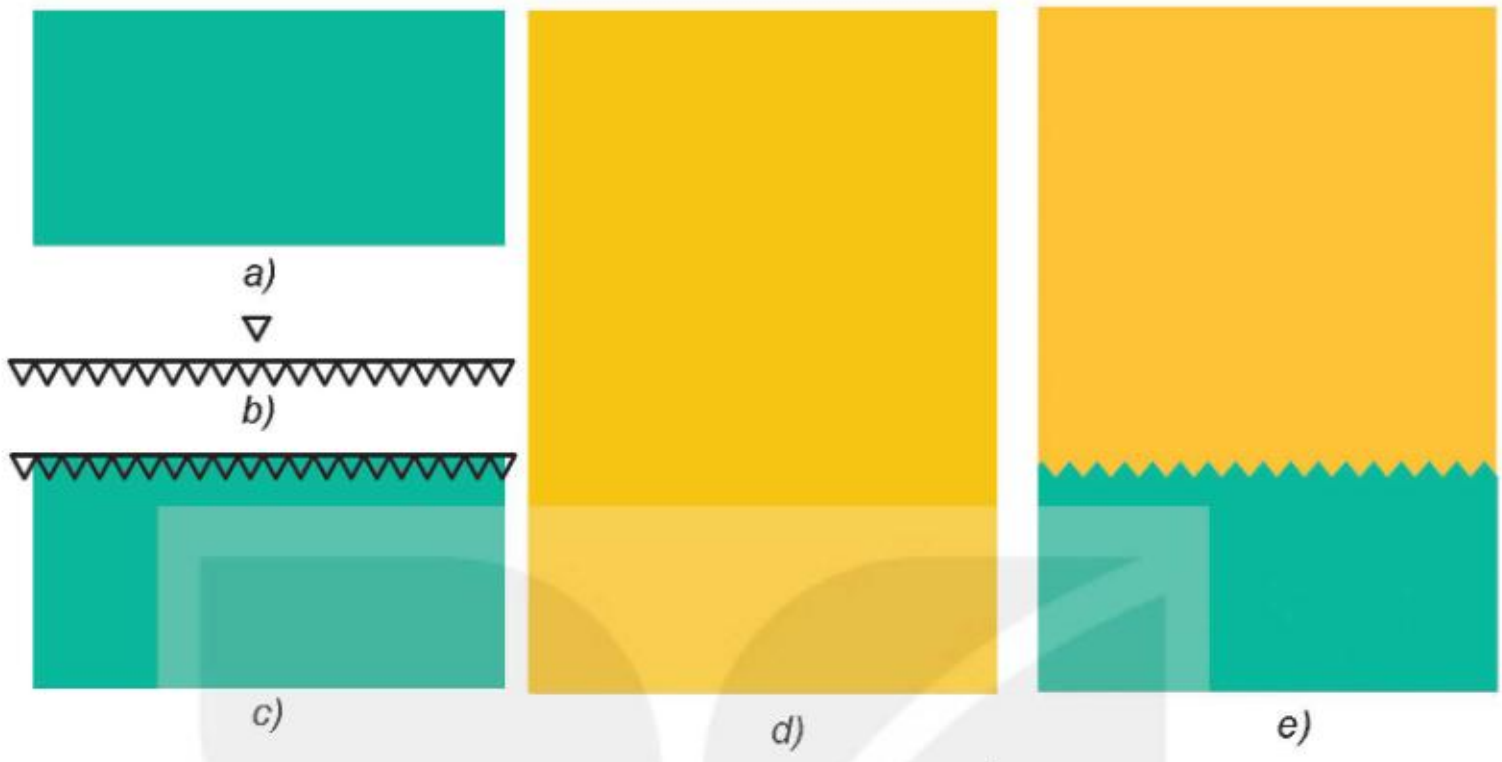
Bước 1: Vẽ một hình chữ nhật màu xanh (Hình 15.2a).

Bước 2: Vẽ một hình tam giác rồi **Duplicate** và dịch chuyển sang bên cạnh để tạo ra một dây tam giác (Hình 15.2b). **Duplicate** cả dây để thu được hai dây tam giác, để riêng một dây cho phần sau.

Bước 3: Chọn tất cả các tam giác trong dây ban đầu và thực hiện **Union**.

Bước 4: Xếp dây tam giác đã **Union** vào trên đỉnh của hình chữ nhật và thực hiện **Difference** (Hình 15.2c).

Bước 5: Vẽ hình chữ nhật màu vàng có chiều rộng bằng chiều rộng của hình chữ nhật xanh (Hình 15.2d) và đưa kết quả của bước 4 vào phía dưới (Hình 15.2e).



Hình 15.2. *Các bước vẽ nền*

Nhiệm vụ 2. Vẽ dây cờ tam giác.

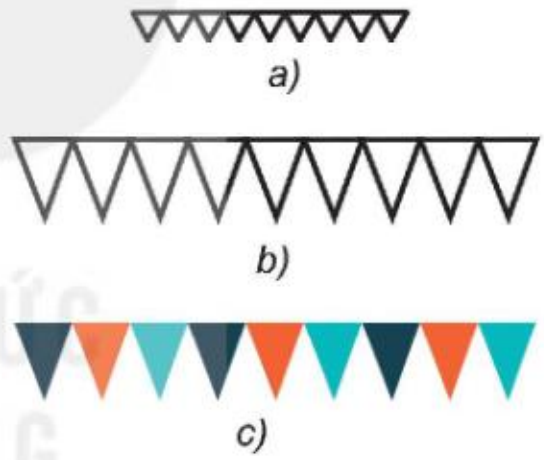
Hướng dẫn.

Bước 1: Cắt ra một số tam giác từ dây tam giác đã để riêng trong nhiệm vụ 1 (Hình 15.3a).

Bước 2: Nhóm các tam giác lại bằng lệnh **Group**, sau đó thực hiện co dãn để được dây tam giác lớn hơn và có kích thước vừa với khung đã vẽ (Hình 15.3b).

Bước 3: Bỏ nhóm dây tam giác bằng lệnh **Ungroup** và tô màu phù hợp (Hình 15.3c).

Xếp dây tam giác lên trên cùng của cụm nền tờ rơi.

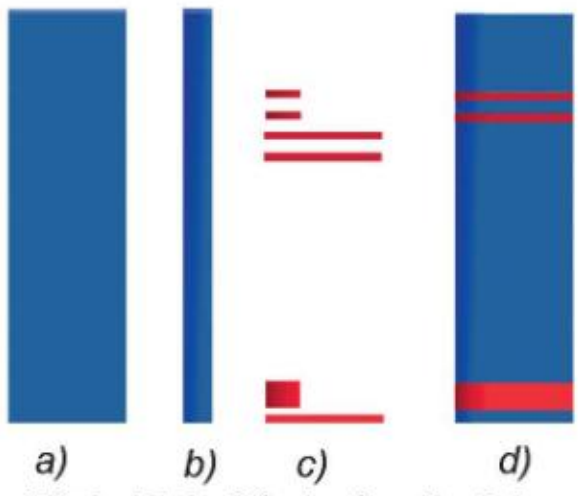


Hình 15.3. *Vẽ dây cờ tam giác*

Nhiệm vụ 3. Vẽ sách và giá sách.

Hướng dẫn.

Bước 1: Vẽ một hình chữ nhật để làm gáy sách (chiều dài tương ứng với độ cao của quyển sách, chiều rộng tương ứng với độ dày) (Hình 15.4a).



Hình 15.4. *Các bước vẽ sách*

Bước 2: Để gáy sách có độ cong ở mép ta sẽ thêm một hình chữ nhật có chiều cao bằng hình trên, chiều rộng tương ứng với đoạn rìa và tô màu chuyển sắc sao cho màu ở rìa ngoài đậm hơn và màu phía trong bằng với màu tô gáy sách phía trên (Hình 15.4b).

Bước 3: Để trang trí cho gáy sách (Hình 15.4c) ta có thể :

+ Thêm các vạch phía trên và dưới (lưu ý làm hai khoảng màu đậm nhạt như bước 1 và 2).

+ Thêm dấu sao hoặc tròn cho những sách cùng bộ.

+ Thêm tên sách và quay dọc theo chiều gáy sách.

Bước 4: Sắp xếp và tô màu cho phù hợp để hoàn thiện một quyển sách (Hình 15.4d).

Sau đó sao chép các quyển sách và chỉnh sửa kích cỡ các hình chữ nhật, màu sắc và chi tiết trang trí để thu được các quyển sách khác nhau.

Bước 5: Vẽ giá sách bằng hình chữ nhật nằm ngang và tô màu phù hợp.

Xếp sách lên giá, quay một vài quyển để tạo cảm giác tự nhiên. Gom cụm cả phần giá sách và đặt lên trên phần nền.

Nhiệm vụ 4. Nhập các đoạn văn bản.

Hướng dẫn.

Bước 1: Lần lượt nhập từng đoạn văn bản – mỗi nội dung là đối tượng khác nhau (Hình 15.5).

Bước 2: Định dạng cỡ chữ và màu sắc phù hợp, sau đó sắp xếp vào vị trí tương ứng.

HỘI SÁCH HÀ NỘI

THỜI GIAN | ĐỊA ĐIỂM

8H - 21H | 12-17/10/ 2030

Hoàng thành Thăng Long

19C Hoàng Diệu, Ba Đình, Hà Nội

Vào cửa tự do

Giảm giá tới 70%

Hình 15.5. *Các đoạn văn bản*

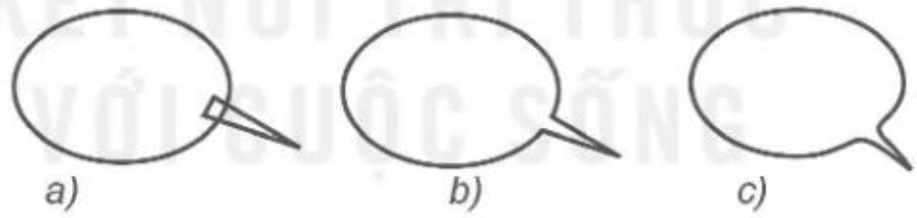
Nhiệm vụ 5. Vẽ bóng hội thoại.

Hướng dẫn.

Bước 1: Vẽ hình elip và hình tam giác (Hình 15.6a).

Bước 2: Xếp hai hình chồng nhau và chọn Union (Hình 15.6b).

Bước 3: Có thể chỉnh lại như Hình 15.6c, tô màu phù hợp.



Hình 15.6. *Cách vẽ bong bóng hội thoại*

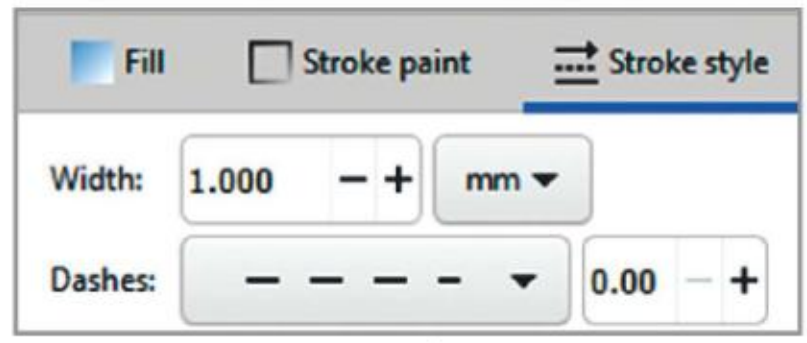
Đặt bóng hội thoại vào vị trí và đặt các cụm chữ tương ứng vào bóng hội thoại và chỉnh lớp (layer) để hiển thị phù hợp.

Nhiệm vụ 6. Vẽ đoạn nét đứt nằm giữa thời gian và địa điểm.

Hướng dẫn.

Bước 1: Chọn công cụ Pen trên hộp công cụ. Vẽ đoạn thẳng đứng để tạo vách ngăn.

Bước 2: Mở hộp thoại **Fill and Stroke** điều chỉnh lại độ dày **Width** là 1 và **Dashes** là kiểu nét đứt (Hình 15.7).



Hình 15.7. *Điều chỉnh nét*

Bước 3: Điều chỉnh lại bố cục, đổi màu vẽ để thu được hình ảnh hoàn thiện như mẫu.

Nhiệm vụ 7. Xuất sản phẩm hoàn thiện ra định dạng đồ hoạ điểm ảnh.

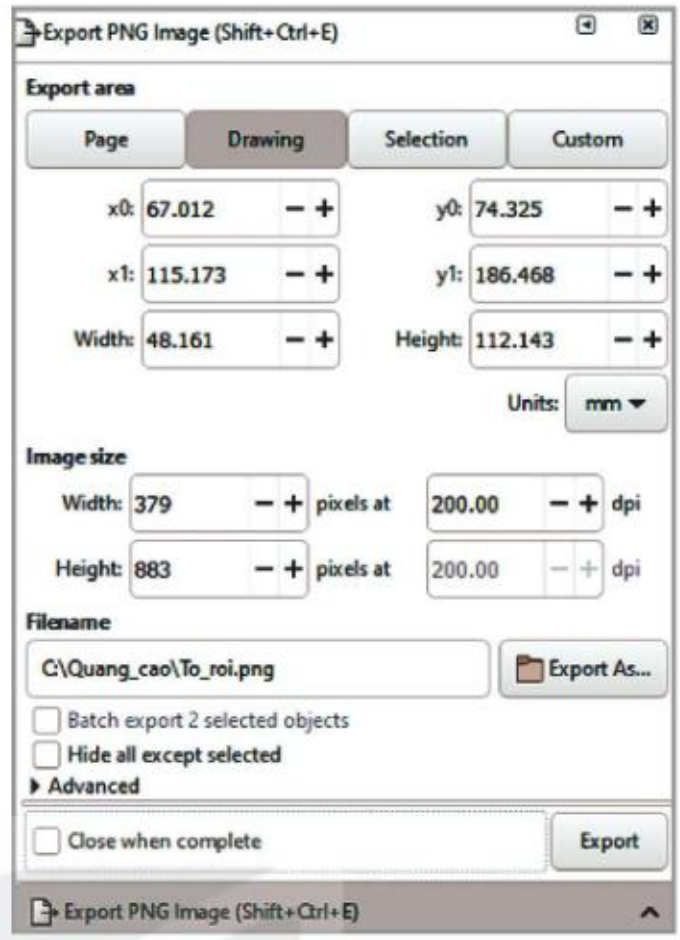
Hướng dẫn.

Sau khi hoàn tất sản phẩm, ta có thể xuất ra tệp dạng đồ hoạ điểm ảnh để dễ dàng sử dụng cho các mục đích khác bằng cách:

Chọn lệnh **File/Export PNG Image**. Cửa sổ thiết lập thông số sẽ xuất hiện như Hình 15.8. Các nội dung chính cần chú ý là:

+ **Export area** – Vùng xuất ảnh: thường dùng nhất là **Page** – trang giấy in được đóng khung và **Selection** – chỉ xuất tệp ảnh cho các đối tượng đang được chọn.
+ **Image size** – Kích thước và độ phân giải của ảnh.
+ **Filename** – Tên tệp và đường dẫn tới tệp.

Sau khi thiết lập các thông số, chọn lệnh **Export** để xuất ra tệp PNG.



Hình 15.8. *Thiết lập thông số để xuất ra tệp đồ hoạ điểm ảnh*

LUYỆN TẬP

1. Thiết kế một thiệp đơn giản (sinh nhật, tiệc Noel...) (Hình 15.9).

2. Vẽ lại logo tiết kiệm điện nước như Hình 14.8b.

VẬN DỤNG

Vẽ minh hoạ bãi biển (Hình 15.10).



Hình 15.9. *Thiệp mời sinh nhật*



Hình 15.10. *Bãi biển*

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

Chủ đề 5

Giải quyết vấn đề với sự trợ giúp của máy tính

BÀI 16

NGÔN NGỮ LẬP TRÌNH BẬC CAO VÀ PYTHON

SAU BÀI NÀY EM SẼ:

- Biết khái niệm ngôn ngữ lập trình bậc cao và ngôn ngữ lập trình bậc cao Python.
- Phân biệt được chế độ gõ lệnh trực tiếp và chế độ soạn thảo chương trình trong môi trường lập trình Python.
- Biết cách tạo và thực hiện một chương trình Python.

Em hãy quan sát các đoạn chương trình được viết bằng các ngôn ngữ lập trình khác nhau trong Hình 16.1 và cho biết câu lệnh trong ngôn ngữ nào dễ hiểu nhất.

```
01101101  01101001  01101110  00100000
01101110  00100000  01110100  01110010
11100001  10111010  10100001  01101110
10100101  01110100  00100000  11000100
00100000  01110100  01110010  01100001
```

Ngôn ngữ máy

```
0013            RESETA  EQU     %00010011
0011            CTLREG  EQU     %00010001

C003  86  13    INITA   LDA  A  #RESETA
C005  B7  80  04        STA  A  ACIA
C008  86  11            LDA  A  #CTLREG
C00D  7E  80  01        STA  A  ACIA

C00D  7E  C0  F1        JMP     SIGNON
```

Hợp ngữ (Assembly)

```
# Nhập từ bàn phím dãy n số và tính tổng.
n = int(input("Nhập số tự nhiên n: "))
s = 0
for i in range(n):
  a = int(input("Nhập số thứ "+str(i+1)+":"))
  s = s + a
print("Tổng các số đã nhập là:",s)
```

Ngôn ngữ Python

Hình 16.1

1. NGÔN NGỮ LẬP TRÌNH BẬC CAO

Hoạt động 1 **Tìm hiểu ngôn ngữ lập trình bậc cao**

1. Ngôn ngữ lập trình là gì? Có những loại ngôn ngữ lập trình nào?
2. Hãy kể tên một số ngôn ngữ lập trình bậc cao mà em biết.

Các lệnh viết bằng ngôn ngữ máy ở dạng mã nhị phân hay hợp ngữ sử dụng một số từ viết tắt (thường là tiếng Anh) không thuận tiện cho việc viết hoặc hiểu chương trình. Khó khăn đó đã được giảm bớt khi lập trình bằng ngôn ngữ bậc cao với các câu lệnh được viết gần với ngôn ngữ tự nhiên. Tuy nhiên, để máy tính có thể hiểu và thực hiện, các chương trình đó cần được dịch sang ngôn ngữ máy nhờ một chương trình chuyên dụng được gọi là chương trình dịch.

Hiện nay đã có nhiều ngôn ngữ lập trình bậc cao khác nhau, trong số đó Java, C/C++, Python,... là những ngôn ngữ lập trình thông dụng nhất.

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

Python là ngôn ngữ lập trình bậc cao do Guido van Rossum, người Hà Lan tạo ra và ra mắt lần đầu năm 1991. Các câu lệnh của Python có cú pháp đơn giản. Môi trường lập trình Python dễ sử dụng, không phụ thuộc vào hệ điều hành, chạy trên nhiều loại máy tính, điện thoại thông minh, robot giáo dục,... Python có mã nguồn mở nên thu hút nhiều nhà khoa học cùng phát triển. Nhờ có các thư viện chương trình phong phú về trí tuệ nhân tạo, phân tích dữ liệu, kĩ thuật robot,... Python là ngôn ngữ lập trình được dùng phổ biến trong nghiên cứu và giáo dục.

- *Ngôn ngữ lập trình bậc cao* có các câu lệnh được viết gần với ngôn ngữ tự nhiên giúp cho việc đọc, hiểu chương trình dễ dàng hơn.
- Python là một ngôn ngữ lập trình bậc cao phổ biến trong nghiên cứu và giáo dục.

Theo em, viết chương trình bằng loại ngôn ngữ lập trình nào dễ nhất?

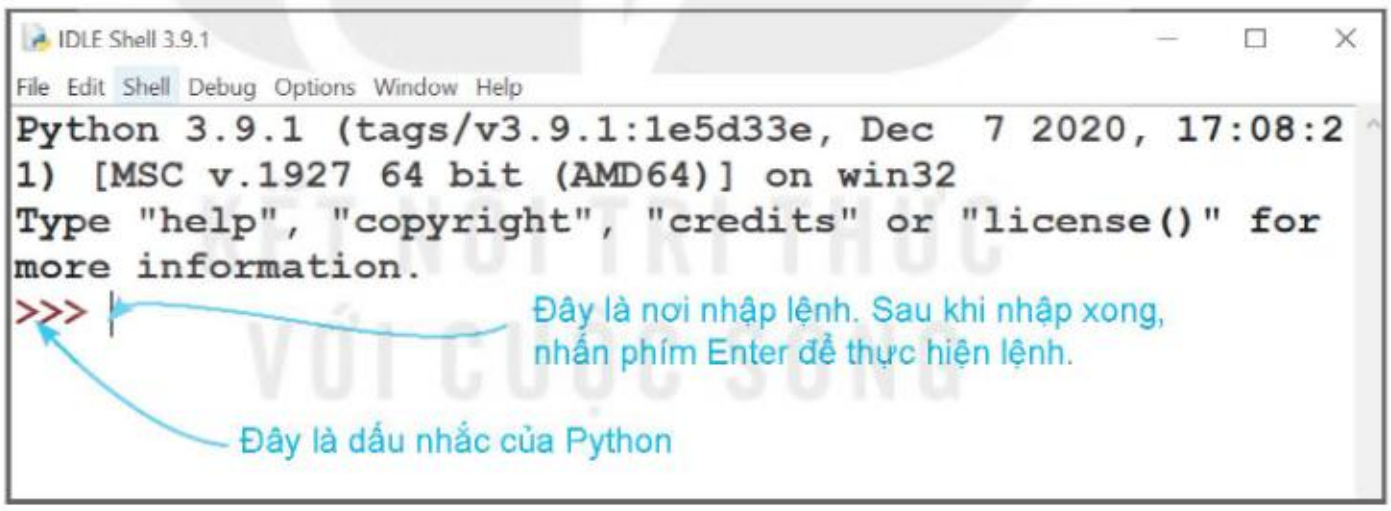
A. Ngôn ngữ máy. B. Hợp ngữ. C. Ngôn ngữ lập trình bậc cao.

2. MÔI TRƯỜNG LẬP TRÌNH PYTHON

Hoạt động 2 **Làm quen với môi trường lập trình Python**

1. Tìm hiểu cách viết và thực hiện các lệnh trong môi trường lập trình Python.
2. Phân biệt chế độ gõ lệnh trực tiếp và chế độ soạn thảo chương trình của Python.

Sau khi khởi động, màn hình làm việc của Python có dạng tương tự như sau:



Hình 16.2. *Màn hình làm việc của Python*

Phần mềm Python là một môi trường lập trình cho phép soạn thảo chương trình bằng ngôn ngữ Python, hỗ trợ gỡ lỗi, phân tích cú pháp dòng lệnh và thực hiện các chương trình Python (chương trình hoàn chỉnh hoặc từng câu lệnh). Môi trường lập trình Python có hai chế độ:

– Chế độ gõ lệnh trực tiếp thường được dùng để tính toán và kiểm tra nhanh các lệnh.

– Chế độ soạn thảo dùng để viết các chương trình có nhiều dòng lệnh.

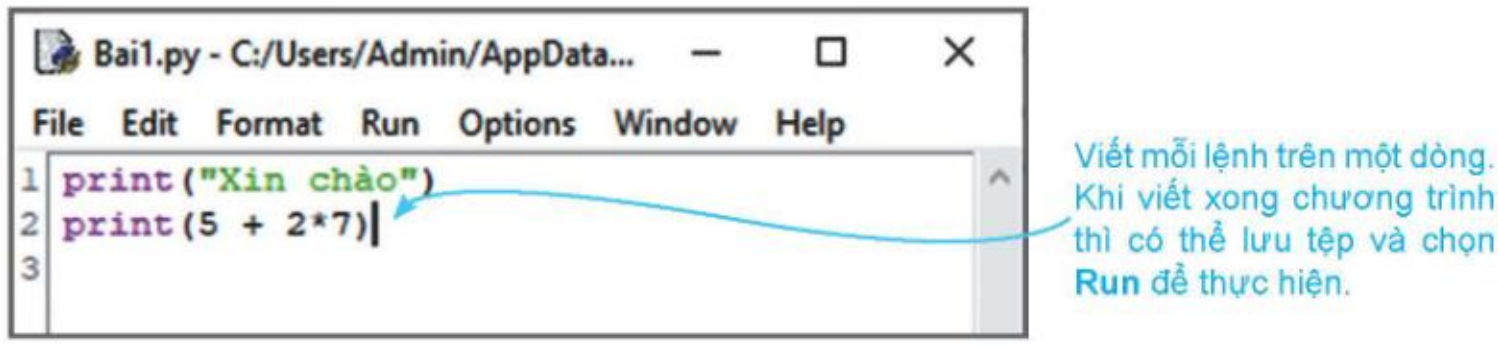
a) Chế độ gõ lệnh trực tiếp

Trong một phiên làm việc với Python, em có thể *gõ lệnh trực tiếp* sau dấu nhắc >>> và nhấn phím Enter để thực hiện lệnh như sau:

```
>>> <lệnh Python>
```

b) Chế độ soạn thảo

Trong môi trường lập trình Python, chúng ta cũng có thể soạn thảo chương trình hoàn chỉnh bằng cách chọn **File/New File** để mở ra màn hình soạn thảo chương trình tương tự như sau:



Hình 16.3. *Màn hình soạn thảo trong môi trường Python*

Chú ý. Người ta có thể soạn thảo chương trình Python bằng phần mềm soạn thảo văn bản hoặc phần mềm lập trình Python như Wingware, PyCharm, Thonny, Visual Studio,...

Môi trường lập trình của Python có hai chế độ: *chế độ gõ lệnh trực tiếp* và *chế độ soạn thảo*.

1. Dấu nhắc chính là con trỏ soạn thảo chương trình Python. Đúng hay sai?
2. Việc thực hiện câu lệnh ở chế độ gõ lệnh trực tiếp và chế độ soạn thảo có điểm gì giống nhau, khác nhau?

3. MỘT SỐ LỆNH PYTHON ĐẦU TIÊN

Hoạt động 3 **Làm quen với câu lệnh của Python**

Quan sát một số lệnh trong chế độ gõ lệnh trực tiếp để biết chức năng của các lệnh này.

Ví dụ 1. Các lệnh đầu tiên.

```
>>> 5
5
>>> 2.6
2.6
>>> "học sinh lớp 10"
'học sinh lớp 10'
```

Khi nhập 5, Python hiểu rằng đó là số nguyên 5. Nếu nhập 2.6 thì Python tự động nhận ra đó là số thực 2.6. Còn nếu gõ các chữ giữa cặp dấu nháy kép " " thì Python hiểu là xâu kí tự.

Python tự nhận biết kiểu dữ liệu và thực hiện các phép toán ngay trên dòng lệnh.

Ví dụ 2. Các lệnh với phép toán.

```
>>> 3 + 7
10
>>> 12*5
60
```

Các phép toán thông thường với số bao gồm phép cộng (+), trừ (–), nhân (*) và chia (/).

Ví dụ 3. Lệnh `print()`.

```
>>> print(12)
12
>>> print(10,3.4 + 4.1, "hoà bình")
10 7.5 hoà bình
>>> print("Dãy ba số chẵn:",2,4,6)
Dãy ba số chẵn: 2 4 6
>>> print("3 + 7 =",3+7)
3 + 7 = 10
```

Chú ý: nếu lệnh `print()` nhiều giá trị thì các dữ liệu này sẽ được đưa ra trên một dòng, giữa các dữ liệu sẽ có dấu cách.

Lệnh `print()` có thể tính toán và đưa ra kết quả của biểu thức.

Trong Python, lệnh `print()` có chức năng đưa dữ liệu ra (xuất dữ liệu). Mặc định dữ liệu sẽ được in ra màn hình. Lệnh `print()` cho phép in một hoặc nhiều giá trị ra màn hình. Cú pháp lệnh `print()` như sau:

```
print(v1, v2,..., vn)
```

trong đó `v1, v2,..., vn` là các giá trị cần đưa ra màn hình.

- Khi nhập giá trị số hoặc xâu kí tự từ dòng lệnh, Python tự nhận biết kiểu dữ liệu.
- Python có thể thực hiện các phép toán thông thường với số, phân biệt số thực và số nguyên.
- Lệnh `print()` có chức năng in dữ liệu ra màn hình, có thể in ra một hoặc nhiều giá trị đồng thời.

1. Kết quả của mỗi lệnh sau là gì? Kết quả đó có kiểu dữ liệu nào?

```
>>> 5/2
>>> 12 + 1.5
>>> "Bạn là học sinh lớp 10"
>>> 10 + 7/2
```

2. Lệnh sau sẽ in ra kết quả gì?

```
>>> print("13 + 10*3/2 - 3*2 = ", 13 + 10*3/2 - 3*2)
```

THỰC HÀNH

Nhiệm vụ: Sử dụng chế độ soạn thảo chương trình của Python để tạo, nhập và chạy chương trình đầu tiên có tên Bai1.py như sau:

Bai1.py

```
# Chương trình đầu tiên
print("Xin chào!")
```

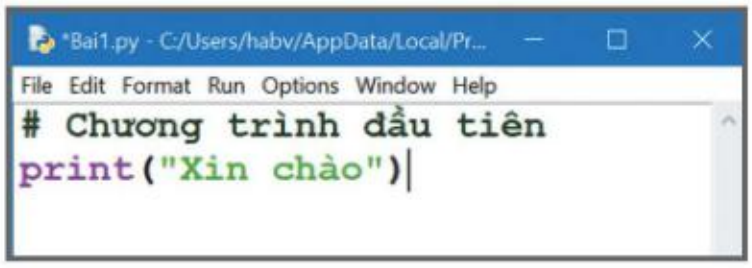
Kí hiệu # là vị trí bắt đầu dòng chú thích lệnh của Python.

Hướng dẫn.

Bước 1: Nháy đúp chuột vào biểu tượng  của Python để khởi động.

Bước 2: Chọn chế độ soạn thảo chương trình của môi trường lập trình Python: Trong môi trường lập trình Python, chọn **File/New File**.

Bước 3: Nhập nội dung chương trình như Hình 16.4.



Hình 16.4

Bước 4: Chọn **File/Save** hoặc nhấn tổ hợp phím **Ctrl** + **S** để lưu tệp.

Bước 5: Chọn **Run/Run module** hoặc nhấn phím **F5** để thực hiện chương trình.

Bước 6: Để kết thúc một phiên làm việc, nháy nút ☒ ở góc trên bên phải màn hình hoặc gõ lệnh `quit()` hoặc lệnh `exit()` rồi nhấn phím ENTER. Ví dụ:

```
>>> quit()
```

LUYỆN TẬP

1. Hãy viết lệnh để tính giá trị các biểu thức sau trong chế độ gõ lệnh trực tiếp của Python:

a) 10 + 13 b) 20 – 7 c) 3 × 10 – 16 d) 12/5 + 13/6

2. Các lệnh sau có lỗi không? Vì sao?

```
>>> 3 + * 5
>>> "Bạn là học sinh, bạn tên là "Nguyễn Việt Anh""
```

3. Viết các lệnh in ra màn hình thông tin như sau:

a) 1 × 3 × 5 × 7 = 105. b) Bạn Hoa năm nay 16 tuổi.

VẬN DỤNG

1. Ngoài cách viết xâu kí tự giữa cặp dấu nháy đơn hoặc nháy kép còn có thể viết giữa cặp ba dấu nháy kép. Nếu một xâu được viết giữa cặp ba dấu nháy kép thì chúng ta có thể dùng phím **Enter** để xuống dòng ở giữa xâu. Hãy thực hiện lệnh sau và quan sát kết quả:

```
>>> print("""Không có việc gì khó
Chỉ sợ lòng không bền
Đào núi và lấp biển
Quyết chí ắt làm nên""")
```

2. Viết chương trình Python in ra màn hình bảng nhân trong phạm vi 10.

BÀI 17

BIẾN VÀ LỆNH GÁN

SAU BÀI NÀY EM SẼ:

- Biết cách thiết lập biến. Phân biệt được biến và từ khoá.
- Biết sử dụng lệnh gán và thực hiện một số phép toán trên kiểu số nguyên, số thực và xâu kí tự.

Trong Đại số, người ta dùng chữ để thay thế cho số cụ thể, ví dụ hằng đẳng thức $(a + b)^2 = a^2 + 2ab + b^2$ đúng cho mọi giá trị a, b. Trong các ngôn ngữ lập trình, người ta cũng dùng các kí tự hoặc nhóm các kí tự được gọi là *biến* (variable) để thay cho việc phải chỉ ra các giá trị dữ liệu cụ thể.

Theo em, sử dụng biến có những lợi ích gì?

1. BIẾN VÀ LỆNH GÁN

Hoạt động 1 Tìm hiểu khái niệm biến và lệnh gán

Quan sát các lệnh sau, n ở đây được hiểu là gì?

```
>>> n = 5
>>> n
5
>>> n + 3
8
```

Sau khi gán n = 5, n sẽ được hiểu là đối tượng số nguyên có giá trị 5.

Biến là tên (định danh) của một vùng nhớ dùng để lưu trữ giá trị (dữ liệu) và giá trị đó có thể được thay đổi khi thực hiện chương trình. Biến trong Python được tạo ra khi thực hiện lệnh gán. Cú pháp của lệnh gán như sau:

```
<biến> = <giá trị>
```

Khi thực hiện lệnh gán, <giá trị> bên phải sẽ được gán cho <biến>. Nếu biến chưa được khai báo thì nó sẽ được khởi tạo khi thực hiện câu lệnh gán. Biến trong Python được xác định kiểu dữ liệu tại thời điểm gán giá trị nên không cần khai báo trước kiểu dữ liệu cho biến như một số ngôn ngữ khác. Ví dụ:

```
>>> x = 5
>>> x        # x được gán 5 và có kiểu số nguyên tại đây.
5
>>> y = "Tin học 10"
>>> y        # y được gán "Tin học 10" có kiểu xâu kí tự tại đây.
'Tin học 10'
```

Sau câu lệnh gán, kiểu dữ liệu của biến được xác định. Cần ghi nhớ điều này để tránh thực hiện các phép toán giữa các biến có kiểu dữ liệu khác nhau. Chẳng hạn, nếu sau các lệnh trên, ta thực hiện lệnh z = x + y thì Python sẽ thông báo lỗi do không thể thực hiện phép cộng giữa dữ liệu kiểu số (biến x) với dữ liệu kiểu xâu kí tự (biến y).

Có thể thực hiện tất cả các phép toán thông thường như: +, -, *, /,... trên các biến có cùng kiểu dữ liệu. Ví dụ:

```
>>> x = y = 1
>>> x = y + 1
>>> z = (x+y)**x
>>> z
9
```

Có thể gán đồng thời nhiều biến với một giá trị.

Phép tính luỹ thừa: $(x + y)^x$

Ví dụ trên cho thấy có thể gán giá trị biểu thức cho biến. Câu lệnh gán có cú pháp tổng quát như sau:

```
<biến> = <biểu thức>
```

Khi thực hiện lệnh này, Python sẽ tính giá trị của <biểu thức> và gán kết quả cho <biến>. Do đó, mọi biến có trong <biểu thức> đều cần được xác định giá trị trước đó.

Ngoài việc gán giá trị trực tiếp cho biến, ta có thể gán giá trị cho biến thông qua tính toán giá trị của biểu thức với các biến đã được xác định trước. Ví dụ:

```
>>> x = 5
>>> y = 2
>>> z = x/y
>>> z
2.5
```

x là biến kiểu số nguyên có giá trị bằng 5.

y là biến kiểu số nguyên có giá trị bằng 2.

z là biến kiểu số thực có giá trị bằng 2.5.

Tên biến thường được đặt sao cho dễ nhớ và có ý nghĩa.

```
>>> ten = "Hoài Nam"
>>> print("Xin chào",ten)
Xin chào Hoài Nam
```

Có thể gán nhiều giá trị đồng thời cho nhiều biến.

```
>>> x,y,z = 10,5,1
>>> x+y+z
16
```

Cú pháp của lệnh gán đồng thời như sau:

```
<var1>,<var2>,..., <varn> = <gt1>, <gt2>,..., <gtn>
```

Chú ý trong lệnh trên, số các biến bên trái bằng số các giá trị bên phải dấu "=".

- *Biến* là tên của một vùng nhớ dùng để lưu trữ giá trị (dữ liệu) và giá trị đó có thể được thay đổi khi thực hiện chương trình.
- Cú pháp *lệnh gán*:

```
<biến> = <biểu thức>
```

- Quy tắc đặt tên biến:
  - Chỉ gồm các chữ cái tiếng Anh, các chữ số từ 0 đến 9 và kí tự gạch dưới "_".
  - Không bắt đầu bằng chữ số.
  - Phân biệt chữ hoa và chữ thường.



1. Các tên biến nào dưới đây là hợp lệ trong Python?

A. _name B. 12abc C. My country D. m123&b E. xyzABC

2. Sau các lệnh dưới đây, các biến x, y nhận giá trị bao nhiêu?

```
>>> x = 10
>>> y = x**2 - 1
>>> x = x/2 + y
```

3. a, b nhận giá trị gì sau các lệnh sau?

```
>>> a,b = 2,3
>>> a,b = a+b, a-b
```

2. CÁC PHÉP TOÁN TRÊN MỘT SỐ KIỂU DỮ LIỆU CƠ BẢN

Hoạt động 2 **Các phép toán trên dữ liệu kiểu số và kiểu xâu kí tự**

Tìm hiểu các phép toán trên dữ liệu kiểu số và kiểu xâu kí tự.



Ví dụ 1. Các phép toán trên dữ liệu kiểu số.

```
>>> a,b = 10,3
>>> (a+b)**2 + (a-b)*10
239
>>> (a//b)*b + a%b
10
>>> c = b/2
>>> c
1.5
```

Sau lệnh này thì a = 10, b = 3.

Phép toán // lấy thương nguyên của phép chia nguyên.
Phép toán % lấy số dư của phép chia nguyên.

Sau lệnh này, c là biến kiểu thực có giá trị 1.5.

Tất cả các phép toán đều được thực hiện từ trái sang phải, riêng phép luỹ thừa (**) thì thực hiện từ phải sang trái. Ví dụ, biểu thức 4**2**3 được thực hiện như sau 4**(2**3) = 4**8 = 65536.

```
>>> 4**2**3
65536
>>> (4**2)**3
4096
```

Các phép toán cơ bản với dữ liệu kiểu số (số thực và số nguyên) trong Python là phép cộng "+", trừ "–", nhân "*", chia "/" , lấy thương nguyên "//", lấy số dư "%" và phép luỹ thừa "**".

Thứ tự thực hiện các phép tính như sau: phép luỹ thừa ** có ưu tiên cao nhất, sau đó là các phép toán /, *, //, %, cuối cùng là các phép toán +, –. Ví dụ, lệnh sau:

```
>>> 3/2+4*2**4-5//2**2
```

tương đương với lệnh:

```
>>> 3/2+4*(2**4) - 5//(2**2)
```

Chú ý. Nếu có ngoặc thì biểu thức trong ngoặc được ưu tiên thực hiện trước.

Ví dụ 2. Các phép toán với dữ liệu kiểu xâu kí tự.

```
>>> s1 = "Hà Nội"
>>> s2 = "Việt Nam"
>>> s1 + s2          <-- Phép + nối hai xâu kí tự.
'Hà NộiViệt Nam'
>>> "123"*5          <-- Phép * n lặp n lần xâu gốc.
'123123123123123'
>>> s*0              <-- Phép * n với số n ≤ 0 thì được kết quả là xâu rỗng.
''
```

Trong biểu thức có cả số thực và số nguyên thì kết quả sẽ có kiểu số thực.

– Các phép toán trên dữ liệu kiểu số: +, –, *, /, //, %, **.
– Các phép toán trên dữ liệu kiểu xâu: + (nối xâu) và * (lặp).



1. Mỗi lệnh sau là đúng hay sai? Nếu đúng thì cho kết quả là bao nhiêu?

```
>>> (12 - 10//2)**2 - 1
>>> (13 + 45**2)(30//12 - 5/2)
```

2. Mỗi lệnh sau cho kết quả là xâu kí tự như thế nào?

```
>>> ""*20 + "010"
>>> "10"+"0"*5
```

3. TỪ KHOÁ

Hoạt động 3 **Phân biệt biến và từ khoá**

Quan sát các lệnh sau, tìm hiểu vì sao Python báo lỗi.

```
>>> if = 12
SyntaxError: invalid syntax
>>> with = "Độ rộng"
SyntaxError: invalid syntax
```

Trong Python, cũng như mọi ngôn ngữ lập trình bậc cao, luôn có một tập hợp các từ tiếng Anh đặc biệt được sử dụng vào mục đích riêng của ngôn ngữ lập trình, được gọi là các từ khoá (keyword) của ngôn ngữ lập trình. Khi viết chương trình không được đặt tên biến hay các định danh trùng với từ khoá. Các lệnh trên, do đặt tên biến trùng với các từ khoá `if` và `with` nên bị báo lỗi.

Một số từ khoá trong Python phiên bản 3.x.

False	break	else	if	not	as	from	while	return
None	class	except	import	or	assert	global	with	
True	continue	finally	in	pass	del	lambda	yield	
and	def	for	is	raise	elif	nonlocal	try	

Từ khoá là các từ đặc biệt tham gia vào cấu trúc của ngôn ngữ lập trình. Không được phép đặt tên biến hay các định danh trùng với từ khoá.

Các tên biến sau có hợp lệ không?

a) _if b) global c) nolocal d) return e) true

THỰC HÀNH

Tạo và làm việc với biến, tính toán với các kiểu dữ liệu cơ bản trong Python.

Nhiệm vụ 1. Thực hiện các phép tính sau trong môi trường lập trình Python, so sánh kết quả với việc tính biểu thức toán học.

a) $(1 + 2 + 3 + ... + 10)^3$.

b) $1/2 + 1/3 + 1/4 + 1/5$.

c) Thực hiện lệnh gán $x = 2$, $y = 5$ rồi tính giá trị biểu thức $(x + y)(x^2 + y^2 - 1)$.

d) Thực hiện lệnh gán $a = 2$, $b = 3$, $c = 4$ rồi tính giá trị biểu thức $(a + b + c)(a + b - c)$.

Hướng dẫn. Các phép tính trên có thể thực hiện trong môi trường lập trình Python như sau:

```
>>> (1+2+3+4+5+6+7+8+9+10)**3
>>> 1/2+1/3+1/4+1/5
>>> x, y = 2, 5
>>> (x+y)*(x**2+y**2-1)
>>> a,b,c = 2,3,4
>>> (a+b+c)*(a+b-c)
```

Nhiệm vụ 2. Gán giá trị cho biến R là bán kính hình tròn rồi viết chương trình tính và in ra kết quả theo mẫu:

```
Chu vi hình tròn là: .....
Diện tích hình tròn là: ....
```

Hướng dẫn. Soạn thảo chương trình sau trong môi trường lập trình Python:

```
R = 4.5
pi = 3.14
print("Chu vi hình tròn là:",2*R*pi)
print("Diện tích hình tròn là:",pi*R*R)
```

Thực hiện chương trình và kiểm tra kết quả, so sánh với chế độ gõ lệnh trực tiếp.

LUYỆN TẬP

1. Lệnh sau có lỗi gì?

```
>>> x = 1
>>> 123a = x + 1
SyntaxError: invalid syntax
```

2. Lệnh sau sẽ in ra kết quả gì?

```
>>> print("đồ rê mi "*3 + "pha son la si đô "*2)
```

VẬN DỤNG

1. Viết các lệnh để thực hiện việc đổi số giây ss cho trước sang số ngày, giờ, phút, giây, in kết quả ra màn hình.

Ví dụ, nếu ss = 684 500 thì kết quả in ra như sau:

684 500 giây = 7 ngày 22 giờ 8 phút 20 giây.

Gợi ý: Sử dụng các phép toán lấy thương nguyên, lấy số dư và các cách đổi sau:

1 ngày = 86 400 giây; 1 giờ = 3 600 giây; 1 phút = 60 giây.

2. Hãy cho biết trước và sau khi thực hiện các lệnh sau, giá trị các biến x, y là bao nhiêu. Em có nhận xét gì về kết quả nhận được?

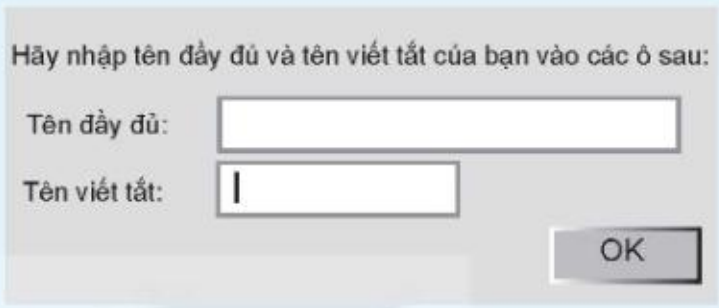
```
>>> x, y = 10, 7
>>> x, y = y, x
```

BÀI 18 CÁC LỆNH VÀO RA ĐƠN GIẢN

SAU BÀI NÀY EM SẼ:

- Biết và thực hiện được một số lệnh vào, ra đơn giản.
- Thực hiện được một số chuyển đổi dữ liệu giữa các kiểu dữ liệu cơ bản.

Để tương tác với người sử dụng trong khi thực hiện chương trình, các ngôn ngữ lập trình có các câu lệnh để đưa dữ liệu ra màn hình hay nhập dữ liệu vào từ bàn phím. Em đã biết Python có lệnh `print()` dùng để đưa dữ liệu ra màn hình. Để nhập dữ liệu từ bàn phím khi thực hiện chương trình, Python sử dụng câu lệnh `input()`.



Em dự đoán lệnh nhập dữ liệu `input()` có cú pháp và chức năng như thế nào?

1. CÁC LỆNH VÀO RA ĐƠN GIẢN

Trong Bài 1 chúng ta đã biết quá trình xử lí thông tin trong máy tính thực chất cũng là xử lí dữ liệu. Do vậy, trong các ngôn ngữ lập trình bậc cao cần có các câu lệnh tương ứng để nhập, xuất dữ liệu. Trong Python, lệnh `print()` có chức năng đưa dữ liệu ra, còn lệnh `input()` đưa dữ liệu vào.

Hoạt động 1 Tìm hiểu chức năng của lệnh input()

Quan sát lệnh sau và trả lời các câu hỏi: Lệnh `input()` cho phép nhập dữ liệu từ đâu? Giá trị được nhập sẽ là số hay xâu?

```
>>> input("Nhập một số: ")
Nhập một số: 12
'12'
```

Lệnh `input()` có chức năng nhập dữ liệu từ thiết bị vào chuẩn, thường là bàn phím. Nội dung nhập có thể nhập số, biểu thức hay xâu và cho kết quả là một xâu kí tự. Cú pháp của lệnh `input()` như sau:

```
<biến> = input(<Dòng thông báo>)
```

Cần nhập một xâu kí tự thì có thể dùng lệnh `input()` tương tự như sau:

```
name = input("Nhập họ tên em: ")
print("Xin chào ",name)
```

Lệnh `print()` có chức năng đưa dữ liệu ra thiết bị ra chuẩn, thường là màn hình. Thông tin cần đưa ra có thể bao gồm một hay nhiều dữ liệu với kiểu khác nhau, cho phép cả biểu thức tính toán.

Các lệnh vào ra đơn giản của Python bao gồm lệnh `input()` và lệnh `print()`.

2. CHUYỂN ĐỔI KIỂU DỮ LIỆU CƠ BẢN CỦA PYTHON

Hoạt động 2 Nhận biết kiểu dữ liệu của biến

Chúng ta đã biết một số kiểu dữ liệu cơ bản như số nguyên, số thực và xâu kí tự. Trong Python có cách nào để nhận biết được kiểu dữ liệu của biến không?

Quan sát các lệnh sau để biết kiểu dữ liệu của mỗi biến.

```
>>> n,x,s = 10,1.8,"One"
>>> n
10
>>> type(n)
<class 'int'>
>>> type(x)
<class 'float'>
>>> type(s)
<class 'str'>
```

Gán n = 10, x = 1.8, s = "One" đồng thời.

Biến n thuộc kiểu **int** – số nguyên.

Biến x thuộc kiểu **float** – số thực.

Biến s thuộc kiểu **str** – xâu kí tự.

Kiểu dữ liệu lôgic cũng là kiểu dữ liệu cơ bản và dữ liệu kiểu này chỉ có hai giá trị là True (đúng) và False (sai). Ví dụ dữ liệu kiểu lôgic là kết quả phép so sánh:

```
>>> 3 > 2
True
>>> 5 < 0
False
>>> b = 10 > 3
>>> type(b)
<class 'bool'>
```

Các biểu thức so sánh chỉ nhận giá trị **True** hoặc **False**, và có giá trị thuộc kiểu lôgic.

Tên kiểu dữ liệu lôgic là **bool**.

- Một số kiểu dữ liệu cơ bản của Python bao gồm: `int` (số nguyên), `float` (số thực), `str` (xâu kí tự), `bool` (lôgic).
- Lệnh `type()` dùng để nhận biết kiểu dữ liệu của biến trong Python.

Xác định kiểu và giá trị của các biểu thức sau:

a) "15 + 20 – 7" b) 32 > 45 c) 13 != 8 + 5 d) 1 == 2

Hoạt động 3 Tìm hiểu cách chuyển đổi giữa các kiểu dữ liệu

1. Có chuyển đổi dữ liệu kiểu này sang kiểu khác được không?

2. Giả sử có biến s với giá trị "123". Nếu muốn biến s có giá trị là số nguyên 123 chứ không phải là xâu "123" thì em phải làm gì?

Lệnh `int()` có chức năng chuyển đổi số thực hoặc xâu chứa số nguyên thành số nguyên. Quan sát các lệnh sau:

```
>> int(12.6)
12
>>> int("123")
123
>>> int("10.35")
Traceback (most recent call last):
  File "<pyshell#21>", line 1, in <module>
    int("10.35")
ValueError: invalid literal for int() with base 10: '10.35'
```

Lệnh `int()` không chuyển đổi được xâu chứa số thực.

Lệnh `float()` dùng để chuyển đổi số nguyên và xâu kí tự thành số thực.

```
>>> float(8)
8.0
>>> float("10.23")
10.23
```

Lệnh `str()` dùng để chuyển đổi các kiểu dữ liệu khác thành xâu kí tự.

```
>>> str(12+34)
'46'
>>> str(12.567)
'12.567'
>>> str(2>3)
'False'
```

Chú ý: Các lệnh `int()`, `float()` chỉ có thể chuyển đổi các xâu ghi giá trị số trực tiếp, không chuyển đổi xâu có công thức, ví dụ:

```
>>> int("12+45")
Traceback (most recent call last):
  File "<pyshell#27>", line 1, in <module>
    int("12+45")
ValueError: invalid literal for int() with base 10: '12+45'
```

- Các lệnh `int()`, `float()`, `str()` có chức năng chuyển đổi dữ liệu từ các kiểu khác tương ứng về kiểu số nguyên, số thực và xâu kí tự.
- Các lệnh `int()`, `float()` không thực hiện xâu là biểu thức toán.

1. Mỗi lệnh sau sẽ trả lại các giá trị nào?

a) `str(150)` b) `int("1110")` c) `float("15.0")`

2. Lệnh nào sau đây sẽ báo lỗi?

A. `int("12.0")` B. `float(13+1)` C. `str(17.001)`

Hoạt động 4 Nhập dữ liệu kiểu số nguyên hoặc số thực từ bàn phím

Dữ liệu nhập từ bàn phím bằng lệnh `input()` luôn là xâu kí tự nên muốn nhập dữ liệu đầu vào là số nguyên hay số thực thì phải làm thế nào?

Nếu cần nhập số nguyên thì sau khi nhập giá trị số cần dùng lệnh `int()` để chuyển đổi sang kiểu số nguyên như sau:

```
>>> n = int(input("Nhập số tự nhiên: "))
Nhập số tự nhiên: 13
```

Nếu cần nhập số thực thì sau khi nhập giá trị số cần dùng lệnh `float()` để chuyển đổi sang kiểu số thực như sau:

```
>>> x = float(input("Nhập số thực x: "))
```

Dùng lệnh `x = input("Nhập số x:")` để nhập số cho biến x là đúng hay sai?

THỰC HÀNH. Nhập dữ liệu từ bàn phím bằng lệnh `input()`.

Nhiệm vụ 1. Viết chương trình nhập lần lượt ba số tự nhiên `m`, `n`, `p`, sau đó in ra tổng của ba số này.

Hướng dẫn. Cần thực hiện lần lượt ba lệnh nhập các số m, n, p. Chú ý cách nhập số nguyên cần dùng lệnh `int()` để chuyển đổi dữ liệu nhập từ bàn phím. Chương trình có thể viết như sau:

```
m = int(input("Nhập số nguyên m: "))
n = int(input("Nhập số nguyên n: "))
p = int(input("Nhập số nguyên p: "))
print("Tổng ba số đã nhập là",m+n+p)
```

Nhiệm vụ 2. Viết chương trình nhập họ tên, sau đó nhập tuổi của học sinh. Chương trình đưa ra thông báo, ví dụ: Bạn Nguyễn Hoà Bình 15 tuổi.

Hướng dẫn. Cần thực hiện hai lệnh nhập dữ liệu, một lệnh nhập tên học sinh, lệnh thứ hai nhập tuổi, sau đó thông báo ra màn hình. Chú ý khi nhập tuổi cần chuyển đổi dữ liệu.

```
ten = input("Nhập họ tên học sinh: "))
tuoi = int(input("Nhập tuổi: "))
print("Bạn", ten, tuoi, "tuổi")
```

LUYỆN TẬP

1. Những lệnh nào trong các lệnh sau sẽ báo lỗi?

a) `int("12+45")` b) `float(123.56)` c) `float("123,5.5")`

2. Vì sao khi nhập một số thực cần viết lệnh `float(input())`?

VẬN DỤNG

1. Viết chương trình nhập giá trị ss là số giây từ bàn phím. Thông báo ra màn hình thời gian ss giây này sau khi đổi thành thời gian tính bằng ngày, giờ, phút, giây.

2. Viết chương trình nhập ba số thực dương a, b, c và tính chu vi, diện tích của tam giác có độ dài các cạnh là a, b, c (a, b, c > 0 và thoả mãn bất đẳng thức tam giác).

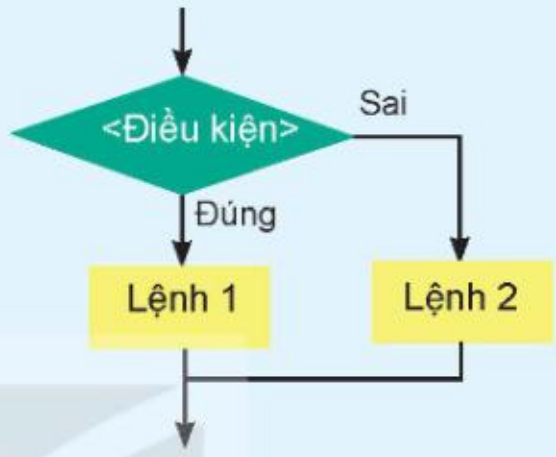
Gợi ý: Công thức Heron tính diện tích tam giác: $S = \sqrt{p(p-a)(p-b)(p-c)}$ với p là nửa chu vi tam giác.

BÀI 19 CÂU LỆNH RẼ NHÁNH IF

SAU BÀI NÀY EM SẼ:

- Biết và trình bày được các phép toán với kiểu dữ liệu lôgic.
- Sử dụng được lệnh rẽ nhánh if trong lập trình.

Trong cuộc sống, chúng ta vẫn thường gặp các tình huống một việc được thực hiện hay không phụ thuộc vào một điều kiện. Ví dụ, em dự định, nếu ngày mai trời không mưa em sẽ đi chơi cùng bạn, ngược lại nếu trời mưa em sẽ ở nhà làm bài tập. Các tình huống như vậy trong lập trình được gọi là rẽ nhánh. Em hãy điền thông tin ở tình huống trên vào vị trí <Điều kiện> và lệnh tương ứng trong sơ đồ cấu trúc rẽ nhánh ở Hình 19.1.



Hình 19.1

1. BIỂU THỨC LÔGIC

Hoạt động 1 **Khái niệm biểu thức lôgic**

Biểu thức nào sau đây có thể đưa vào vị trí `<điều kiện>` trong lệnh:

Nếu `<điều kiện>` thì `<lệnh>` của các ngôn ngữ lập trình bậc cao?

A. m, n = 1, 2. B. a + b > 1. C. a * b < a + b. D. 12 + 15 > 2 * 13.

Trong Python, biểu thức lôgic là biểu thức chỉ nhận giá trị `True` (đúng) hoặc `False` (sai). Biểu thức lôgic đơn giản nhất là các biểu thức so sánh số hoặc xâu kí tự. Trong Hoạt động 1, các phương án B, C, D là biểu thức lôgic.

Quan sát các lệnh sau để nhận biết kiểu dữ liệu lôgic.

```
>>> a,b,s = 10,2,"Number"
>>> a > 10
False
>>> b < 3
True
>>> s == "number"
False
```

Gán a = 10, b = 2, s = "Number"

a > 10 là sai, b < 3 là đúng.

s và "number" là hai xâu có giá trị khác nhau.

Các phép so sánh các giá trị số trong Python:

<	nhỏ hơn	>	lớn hơn	==	bằng nhau
<=	nhỏ hơn hoặc bằng	>=	lớn hơn hoặc bằng	!=	khác nhau

Chú ý: Với xâu kí tự cũng có đầy đủ các phép so sánh (sẽ học sau).

Các phép toán trên kiểu dữ liệu lôgic bao gồm phép `and` (và), `or` (hoặc) và `not` (phủ định). Bảng các phép toán lôgic như sau:

Phép toán and		
X	Y	X and Y
True	True	True
True	False	False
False	True	False
False	False	False

Phép toán or		
X	Y	X or Y
True	True	True
True	False	True
False	True	True
False	False	False

Phép toán not	
X	not X
True	False
False	True

Ví dụ. Cho các lệnh sau và dự đoán giá trị của các biến lôgic a, b, c.

```
>>> x, y, z = 10, 5, 9
>>> b = x < 11 and z > 5
>>> c = x > 15 or y < 9
>>> a = not b
```

Giải thích. Ta có x = 10, z = 9 do đó x < 11 là đúng, z > 5 đúng. Theo bảng phép toán **and** ta có `b = x < 11 and z > 5` nhận giá trị đúng.

Ta lại có: x > 15 sai (vì x = 10) nhưng y < 9 đúng (vì y = 5). Theo bảng phép toán **or** suy ra `c = x > 15 or y < 9` nhận giá trị đúng.

Cuối cùng, vì b là đúng nên `a = not b` sẽ nhận giá trị sai.

- Biểu thức lôgic là biểu thức chỉ nhận giá trị `True` hoặc `False`. Giá trị các biểu thức lôgic thuộc kiểu `bool`.
- Các phép toán trên kiểu dữ liệu lôgic là and (và), or (hoặc) và not (phủ định).

Mỗi biểu thức sau có giá trị True hay False?

a) `100%4 == 0` b) `111//5 != 20 or 20%3 != 0`

2. LỆNH IF

Hoạt động 2 **Cấu trúc lệnh if trong Python**

Cho trước số tự nhiên n (được gán hoặc nhập từ bàn phím). Đoạn chương trình như sau kiểm tra n > 0 thì thông báo "n là số lớn hơn 0".

```
if n > 0:
    print("n là số lớn hơn 0")
```

Em có nhận xét gì về cấu trúc lệnh `if`? Sau `<điều kiện>` lệnh `if` có kí tự gì? Lệnh `print()` được viết như thế nào?

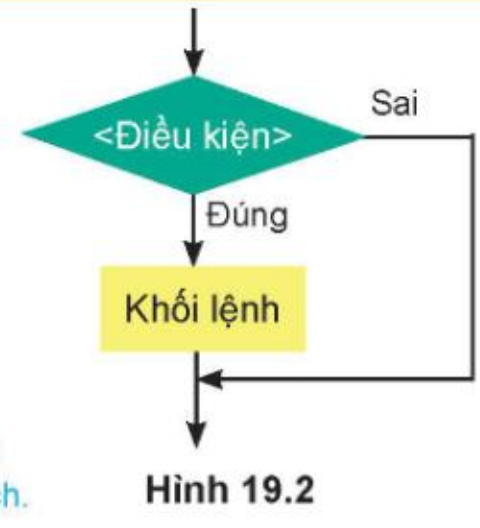
Để xử lí các tình huống rẽ nhánh, giống như các ngôn ngữ lập trình khác, Python cũng có các câu lệnh để mô tả cấu trúc rẽ nhánh:

Câu lệnh rẽ nhánh dạng thiếu:

```
if <điều kiện>:
    <khối lệnh>
```

Sau điều kiện cần có dấu hai chấm ":".

Nhóm, khối lệnh tiếp theo cần viết lùi vào và thẳng hàng, mặc định là 1 tab hay 4 dấu cách.



Hình 19.2

Khi thực hiện lệnh, Python sẽ kiểm tra <điều kiện> nếu đúng thì thực hiện <khối lệnh>, ngược lại thì bỏ qua chuyển sang lệnh tiếp theo sau lệnh if.

Câu lệnh rẽ nhánh dạng đủ:

```
if <điều kiện>:
    <khối lệnh 1>
else:
    <khối lệnh 2>
```

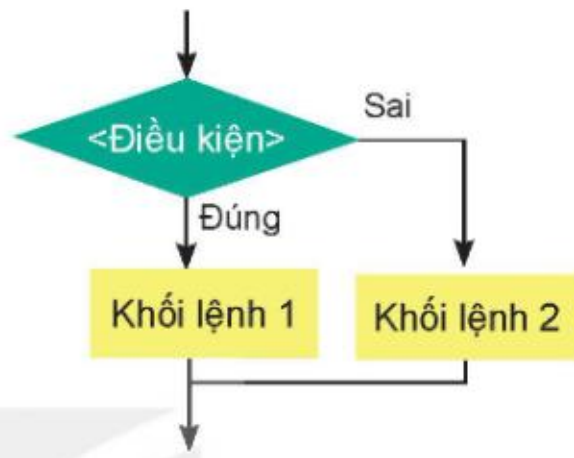
– Từ khoá if và else cần viết thẳng lề trái.

– Các khối lệnh 1 và khối lệnh 2 cần viết lùi vào và thẳng hàng, mặc định là 1 tab hay 4 dấu cách.

Khi thực hiện lệnh, Python sẽ kiểm tra <điều kiện> nếu đúng thì thực hiện <khối lệnh 1>, ngược lại thì thực hiện <khối lệnh 2>.

Ví dụ, nếu a, b là hai số đã được tạo thì lệnh sau sẽ in ra giá trị tuyệt đối của hiệu hai số:

```
if a > b:
    print(a - b)
else:
    print(b - a)
```



Hình 19.3

Chú ý: Các khối lệnh trong Python đều cần viết sau dấu ":" và lùi vào, thẳng hàng. Đây là điểm khác biệt của Python với các ngôn ngữ lập trình khác.

Câu lệnh rẽ nhánh if thể hiện cấu trúc rẽ nhánh trong Python. Khối lệnh rẽ nhánh của if được viết sau dấu ":", cần viết lùi vào và thẳng hàng.

Đoạn chương trình sau thực hiện công việc gì?

```
k = int(input("Nhập một số nguyên dương: "))
if k <= 0:
    print("Bạn nhập sai rồi!")
```

THỰC HÀNH

Các bài tập liên quan đến kiểu dữ liệu bool và lệnh if.

Nhiệm vụ 1. Viết chương trình nhập số tự nhiên n từ bàn phím. Sau đó thông báo số em đã nhập là số chẵn hay số lẻ phụ thuộc vào n là chẵn hay lẻ.

Hướng dẫn. Để kiểm tra một số tự nhiên n là chẵn hay lẻ, ta dùng phép toán lấy số dư n%2. Nếu số dư bằng 0 thì n là số chẵn, ngược lại n là số lẻ. Chương trình có thể như sau:

```
n = int(input("Nhập số tự nhiên n: "))
if n%2 == 0:
    print("Số đã nhập là số chẵn.")
else:
    print("Số đã nhập là số lẻ.")
```

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

Nhiệm vụ 2. Giả sử giá điện sinh hoạt trong khu vực gia đình em ở được tính luỹ kế theo từng tháng như sau (giá tính theo từng kWh điện tiêu thụ).

– Với mức điện tiêu thụ từ 0 đến 50 kWh, giá thành mỗi kWh là 1,678 nghìn đồng.

– Với mức từ 51 đến 100, giá thành mỗi kWh là 1,734 nghìn đồng.

– Từ mức 101 trở lên, giá thành mỗi kWh là 2,014 nghìn đồng.

Viết chương trình nhập số điện tiêu thụ trong tháng của gia đình em và tính số tiền điện phải trả.

Hướng dẫn. Gọi k là số kWh điện tiêu thụ của gia đình em. Khi đó theo cách tính luỹ kế trên chúng ta cần tính dựa trên các điều kiện sau:

– Nếu $k \le 50$ thì số tiền cần trả là $k \times 1{,}678$ nghìn đồng.

– Nếu $50 < k \le 100$ thì số tiền cần trả là $50 \times 1{,}678 + (k - 50) \times 1{,}734$ nghìn đồng.

– Nếu $k > 100$ thì số tiền cần trả là: $50 \times 1{,}678 + 50 \times 1{,}734 + (k - 100) \times 2{,}014$ nghìn đồng.

Chúng ta sử dụng lệnh `round(t)` để làm tròn số thực t. Chú ý trong máy tính dùng dấu "." để viết các số thập phân. Chương trình có thể như sau:

```
k = float(input("Nhập số kWh tiêu thụ điện nhà em: "))
if k<= 50:
    t = k*1.678
else:
    if k <= 100:
        t = 50*1.678 + (k-50)*1.734
    else:
        t = 50*1.678 + 50*1.734 + (k-100)*2.014
print("Số tiền điện phải trả là:",round(t), "nghìn đồng")
```

LUYỆN TẬP

1. Viết biểu thức lôgic ứng với mỗi câu sau:

a) Số x nằm trong khoảng (0; 10).

b) Số y nằm ngoài đoạn [1; 2].

c) Số z nằm trong đoạn [0; 1] hoặc [5; 10].

2. Tìm một vài giá trị m, n thoả mãn các biểu thức sau:

a) `100%m == 0 and n%5 != 0`

b) `m%100 == 0 and m%400 != 0`

c) `n%3 == 0 or (n%3 != 0 and n%4 == 0)`

VẬN DỤNG

1. Giá bán cam tại siêu thị tính như sau: nếu khối lượng cam mua dưới 5 kg thì giá bán là 12 000 đồng/kg, nếu khối lượng mua lớn hơn hoặc bằng 5 kg thì giá bán là 10 000 đồng/kg. Viết chương trình nhập số lượng mua (tính theo kg) sau đó tính số tiền phải trả.

2. Năm n là năm nhuận nếu giá trị n thoả mãn điều kiện: n chia hết cho 400 hoặc n chia hết cho 4 đồng thời không chia hết cho 100. Viết chương trình nhập số năm n và cho biết năm n có phải là nhuận hay không.

BÀI 20

CÂU LỆNH LẶP FOR

SAU BÀI NÀY EM SẼ:

- Biết được ý nghĩa của vùng giá trị tạo bởi lệnh range( ).
- Biết được chức năng của lệnh lặp for và cách dùng trong Python.

Em có thể đã gặp những trường hợp cần thực hiện một số công việc lặp đi lặp lại nhiều lần. Ví dụ, để kể tên tất cả các bạn trong lớp có 30 học sinh, em cần lần lượt đọc tên từng bạn; để đếm số lượng các số chia hết cho 3 trong khoảng từ 1 đến 50, em có thể kiểm tra lần lượt các số từ 1 đến 50 và ghi ra các số chia hết cho 3 (chẳng hạn, 3, 6, 9,...) rồi đếm các số đó. Ngôn ngữ lập trình bậc cao có các câu lệnh cho phép viết một cách ngắn gọn các bước cần thực hiện lặp đi lặp lại để tạo thành một cấu trúc lập trình được gọi là cấu trúc lặp.

Em có thể xác định được trong mỗi ví dụ trên công việc nào cần phải lặp và được lặp lại bao nhiêu lần không?

1. LỆNH LẶP FOR

Hoạt động 1 **Làm quen với lệnh lặp for**

Thực hiện đoạn chương trình sau trong chế độ gõ lệnh trực tiếp của Python để tính tổng 0 + 1 + ... + 9. Tổng này có giá trị bao nhiêu? Giải thích kết quả.

```
>>> S = 0
>>> for k in range(10):
        S = S + k
>>> print(S)
45
```

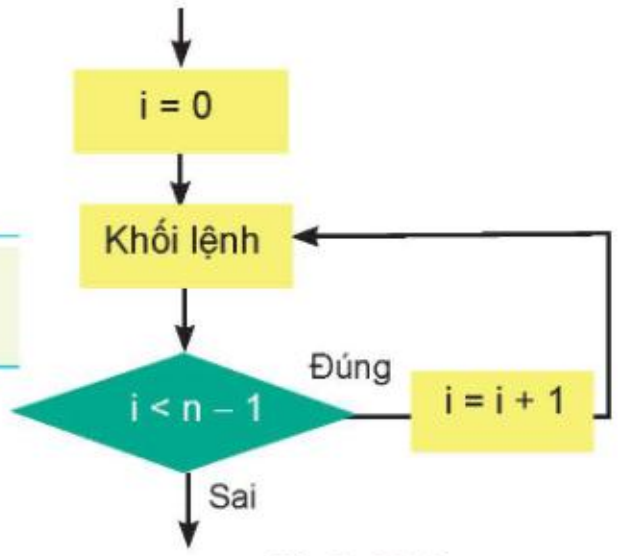
KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

Trong đoạn chương trình trên, lệnh `range(10)` trả lại một *vùng giá trị* gồm 10 số 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Lệnh **for** sẽ thực hiện 10 lần lặp, mỗi lần lặp ứng với một giá trị k trong vùng giá trị trên. Sau lệnh lặp for trên, biến S sẽ có giá trị là tổng 0 + 1 + ... + 9 = 45.

Lệnh **range(n)** trả lại *vùng giá trị* gồm n số từ 0 đến n – 1. Cú pháp của lệnh lặp với số lần biết trước **for** trong Python như sau:

```
for <i> in range(n):
    <khối lệnh>
```

Khi thực hiện, ở mỗi vòng lặp biến i sẽ được gán lần lượt các giá trị trong vùng giá trị của lệnh range() và thực hiện <khối lệnh>.



Hình 20.1

Ví dụ 1. Tính tổng các số tự nhiên chẵn nhỏ hơn n, với n cho trước (n = 10).

```
n = 10
S = 0
for k in range(n):
    if k%2 == 0:
        S = S + k
print(S)
```

Điều kiện k là số chẵn là k%2 = 0

Ví dụ 2. Đếm số các số nguyên nhỏ hơn n (n = 20) và là bội của 3.

```
n = 20
C = 0
for k in range(n):
    if k%3 == 0:
        C = C + 1
print(C)
```

Điều kiện k là bội của 3 là k%3 = 0

> **for** là lệnh lặp với số lần biết trước. Số lần lặp thường được xác định bởi vùng giá trị của lệnh range().

Với giá trị n cho trước, so sánh giá trị S trong đoạn chương trình sau với tổng 1 + 2 + ... + n.

```
S = 0
for k in range(1,n+1):
    S = S + k
```

2. LỆNH RANGE

Hoạt động 2 Tìm hiểu vùng giá trị xác định bởi lệnh range()

Quan sát các lệnh for sau và so sánh kết quả in ra để biết vùng giá trị được xác định bởi lệnh range(). Lưu ý, lệnh print() có thêm tham số để in bộ dữ liệu theo hàng ngang.

```
>>> for k in range(3,10):
        print(k, end = " ")
3 4 5 6 7 8 9
>>> for k in range(0,15):
        print(k, end = " ")
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
```

Đây là vùng **range(3,10)**

Đây là vùng **range(0,15)**

Lệnh tạo vùng giá trị range() có các dạng sau:

range(stop) trả lại vùng giá trị từ 0 đến stop - 1.

range(start,stop) trả lại vùng giá trị từ start đến stop - 1.

Ví dụ, range(n) cho vùng gồm các số 0, 1,..., n – 1.

range(1,n+1) cho vùng gồm các số 1, 2,..., n.

range(0,99) cho vùng giá trị gồm các số 0, 1, 2,..., 98.

range(100,1) cho vùng rỗng.

Lệnh tạo vùng giá trị có cú pháp `range(start,stop)` trả lại vùng giá trị gồm các số nguyên liên tiếp từ `start` đến `stop - 1`.

Hãy biểu diễn các dãy sau đây bằng lệnh `range()`.

a) 1, 2, 3,..., 50. b) 5, 6, 7, 8, 9, 10. c) 0, 1. d) 10.

THỰC HÀNH. Lệnh lặp for và lệnh range()

Nhiệm vụ 1. Nhập số tự nhiên n từ bàn phím và in ra màn hình dãy các ước số của n theo chiều ngang màn hình. Ví dụ nếu n = 10 thì chương trình sẽ in ra dãy số 1, 2, 5, 10.

Hướng dẫn. Các ước số của n là các số tự nhiên k thoả mãn: `n%k = 0`. Muốn in các số trên một hàng ngang cần dùng thêm tham số `end = " "` trong lệnh `print()`.

Chương trình có thể như sau:

```
n = int(input("Nhập số tự nhiên n: "))
for k in range(1, n+1):
    if n%k == 0:
        print(k, end = " ")
```

Nhiệm vụ 2. Nhập số tự nhiên n từ bàn phím và đếm số các ước số thực sự của n. Ước số thực sự của n là số tự nhiên k < n và là ước của n.

Hướng dẫn. Tương tự như chương trình ở nhiệm vụ 1, điểm khác là cần đếm số các ước số này và không tính n. Tạo một biến có tên `count` để đếm số các ước số thực sự của n.

```
n = int(input("Nhập số tự nhiên n: "))
count = 0
for k in range(1, n):
    if n%k == 0:
        count = count + 1
print(count)
```

LUYỆN TẬP

1. Đoạn chương trình sau in ra kết quả gì?

```
n = int(input("Nhập số tự nhiên n:"))
S = 0
for k in range(n+1):
    S = S + k
print(S*S)
```

2. Viết đoạn chương trình tính tích 1 × 2 × 3 ×...× n với n được nhập vào từ bàn phím.

VẬN DỤNG

1. Viết chương trình nhập từ bàn phím số tự nhiên n và in ra kết quả $S = 1 + \frac{1}{2} + ... + \frac{1}{n}$.

2. Viết chương trình nhập từ bàn phím số tự nhiên n và in ra kết quả là tổng sau:

$$S = 1^3 + 2^3 + ... + n^3.$$

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

BÀI 21 CÂU LỆNH LẶP WHILE

SAU BÀI NÀY EM SẼ:

- Biết và thực hành giải các bài toán sử dụng lệnh lặp while với số lần không biết trước.
- Biết ba cấu trúc lập trình cơ bản: tuần tự, rẽ nhánh, lặp.

Cho các việc được ghi trong cột A và cột B của bảng sau:

A	B
Vận động viên chạy 20 vòng xung quanh sân vận động.	Vận động viên chạy nhiều vòng xung quanh sân vận động trong thời gian 2 tiếng.
Em làm 5 bài tập thầy cô giao về nhà.	Em làm các bài tập về nhà đến giờ ăn cơm thì dừng lại.
Em đi lấy 15 xô nước giúp mẹ.	Em xách các xô nước giúp mẹ cho đến khi đầy thùng nước.

Đối với mỗi hàng, em hãy cho biết công việc được lặp đi lặp lại là gì? Điều kiện để dừng công việc là gì? Số lần thực hiện việc lặp giữa hai cột có gì khác nhau?

1. LỆNH LẶP WHILE

Hoạt động 1 **Làm quen với lệnh lặp while**

Quan sát đoạn chương trình sau, giải thích kết quả in ra.

```
>>> S = 0
>>> k = 1
>>> while k < 100:
        S = S + k
        k = k + 7

>>> print(S)
750
```

Điều kiện lặp: nếu <điều kiện> là **False** thì dừng lặp.

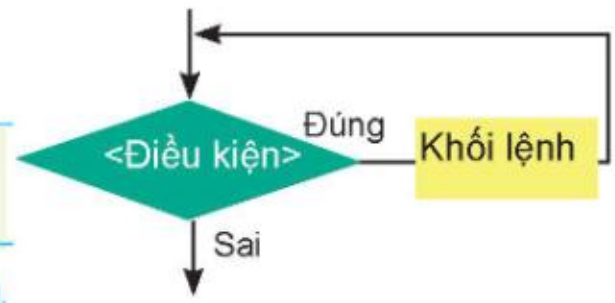
Khối các lệnh lặp được viết lùi vào và thẳng hàng. Sau mỗi vòng lặp, k tăng lên 7.

Lệnh lặp **while** thực hiện khối lệnh với số lần lặp không biết trước. Khối lệnh lặp được thực hiện cho đến khi <điều kiện> = False.

Cú pháp của lệnh **while** như sau:

```
while <điều kiện>:
    <khối lệnh>
```

Sau dấu ":", khối lệnh lặp cần được viết lùi vào và thẳng hàng. Mặc định các lệnh sẽ lùi vào 1 tab hoặc 4 dấu cách.



Hình 21.1

Trong đó <điều kiện> là biểu thức lôgic. Khi thực hiện lệnh, Python sẽ kiểm tra <điều kiện>, nếu đúng thì thực hiện <khối lệnh>, nếu sai thì kết thúc lệnh **while**.

Trong đoạn chương trình ở Hoạt động 1, lệnh lặp sẽ dừng khi k ≥ 100 và giá trị S nhận được là tổng 1 + 8 + 15 + ... + 99.

Ví dụ 1. Quan sát đoạn chương trình sau và cho biết S là giá trị của biểu thức toán học nào?

```
S = 0
k = 1
while k*k < 100:
    S = S + k*k
    k = k + 1
```

Giải thích: Đoạn chương trình tính tổng $1^2 + 2^2 + ... + k^2$ với điều kiện $k^2 < 100$. Vậy S chính là tổng bình phương các số tự nhiên nhỏ hơn 10.

Ví dụ 2. Thực hiện các lệnh sau. Kết quả sẽ in ra những số nào?

```
>>> k = 2
>>> while k < 50:
        print(k,end = " ")
        k = k + 3
```

Giải thích: Vòng lặp `while` sẽ dừng khi k vượt quá 50. Bắt đầu vòng lặp, k = 2. Sau mỗi bước lặp k tăng lên 3 đơn vị. Do vậy, kết quả sẽ phải in ra dãy sau:

```
2 5 8 11 14 17 20 23 26 29 32 35 38 41 44 47
```

while là lệnh lặp với số lần không biết trước. Số lần lặp của lệnh **while** phụ thuộc vào điều kiện của lệnh.

1. Lệnh **while** kiểm tra điều kiện trước hay sau khi thực hiện khối lệnh lặp?

2. Viết đoạn chương trình tính tổng 2 + 4 + ... + 100 sử dụng lệnh **while**.

Lưu ý:

1. Vì lệnh `while` không biết trước số lần lặp, mà phụ thuộc vào điều kiện. Do đó, cần chú ý đến điều kiện của lệnh `while` để tránh bị lặp vô hạn.

2. Trong trường hợp nếu muốn dừng và thoát ngay khỏi vòng lặp `while` hoặc `for` có thể dùng lệnh `break`.

```
>>> for k in range(10):
        print(k,end = " ")
        if k == 5:
            break
0 1 2 3 4 5
```

2. CẤU TRÚC LẬP TRÌNH

Hoạt động 2 Các cấu trúc lập trình cơ bản

Đọc, thảo luận để hiểu các cấu trúc lập trình cơ bản trong ngôn ngữ lập trình bậc cao.

Với việc sử dụng câu lệnh rẽ nhánh `if` và các câu lệnh lặp `for`, `while` ta có thể thấy một chương trình trên Python nói chung có thể được chia thành các khối lệnh sau:

– Khối gồm các lệnh được thực hiện theo trình tự từ trên xuống dưới. Khối này tương ứng với cấu trúc tuần tự trong chương trình và được thể hiện bằng các câu lệnh như gán giá trị, nhập/xuất dữ liệu,...

- Khối các câu lệnh chỉ được thực hiện tuỳ thuộc vào điều kiện nào đó là đúng hay sai. Khối lệnh này tương ứng với cấu trúc rẽ nhánh và được thể hiện bằng câu lệnh rẽ nhánh `if`.
- Khối các câu lệnh được thực hiện lặp đi lặp lại tuỳ theo điều kiện nào đó vẫn còn đúng hay sai. Khối lệnh này tương ứng với cấu trúc lặp và được thể hiện bằng các câu lệnh lặp `for`, `while`.

Ba cấu trúc chương trình trên được gọi là các cấu trúc lập trình cơ bản của các ngôn ngữ lập trình bậc cao.

> Ba cấu trúc lập trình cơ bản của các ngôn ngữ lập trình bậc cao gồm: cấu trúc tuần tự, cấu trúc rẽ nhánh, cấu trúc lặp.

THỰC HÀNH. Sử dụng lệnh lặp `while` và các lệnh đã học

Nhiệm vụ 1. Viết chương trình in toàn bộ dãy các số tự nhiên từ 1 đến 100 trên một hàng ngang.

Hướng dẫn. Mở phần mềm Python và nhập chương trình sau:

```
k = 0
while k < 100:
    k = k + 1
    print(k, end = " ")
```

Nhiệm vụ 2. Viết chương trình in ra màn hình dãy các chữ cái tiếng Anh từ "A" đến "Z" theo ba hàng ngang trên màn hình, hai hàng ngang đầu có 10 chữ cái, hàng thứ ba có 6 chữ cái.

Hướng dẫn. Chúng ta đã biết các chữ cái tiếng Anh từ A đến Z chiếm các vị trí từ 65 đến 90 trong bảng mã ASCII. Với số thứ tự k của bảng mã ASCII, chúng ta sử dụng lệnh `chr(k)` trả lại kí tự tương ứng trong bảng mã này.

```
i = 0
k = 65 # bắt đầu từ số thứ tự của chữ cái A.
while k <= 90:
    i = i + 1
    if i%10 == 0:
        print(chr(k))      <- Với các chữ cái ở cuối hàng sẽ in ra và xuống dòng.
    else:
        print(chr(k), end = " ")   <- Với các chữ cái khác thì in ra trên một hàng ngang.
    k = k + 1
```

LUYỆN TẬP

1. Cho dãy số 1, 4, 7, 10,.... Viết chương trình in ra phần tử lớn nhất của dãy nhưng nhỏ hơn 100.
2. Viết chương trình đếm trong dãy 100 số tự nhiên đầu tiên có bao nhiêu số thoả mãn điều kiện: hoặc chia hết cho 5 hoặc chia cho 3 dư 1.

VẬN DỤNG

Viết chương trình in các số tự nhiên từ 1 đến 100 ra màn hình thành 10 hàng, mỗi hàng 10 số, có dạng như sau:

```
1 2 3 ...  10
11 12 ....  20
.....
91 92 ..... 100
```

BÀI 22 KIỂU DỮ LIỆU DANH SÁCH

SAU BÀI NÀY EM SẼ:

- Biết được kiểu dữ liệu danh sách (list), cách khởi tạo và truy cập từng phần tử của danh sách.
- Biết và thực hiện được cách duyệt các phần tử của danh sách bằng lệnh for.
- Thực hành được một số phương thức đơn giản trên dữ liệu danh sách.

Em đã được học những kiểu dữ liệu cơ bản của Python như số nguyên, số thực, xâu kí tự, kiểu dữ liệu lôgic. Tuy nhiên, khi em cần lưu một dãy các số hay một danh sách học sinh thì cần kiểu dữ liệu dạng *danh sách* (còn gọi là *dãy* hay *mảng*). Kiểu dữ liệu danh sách được dùng nhiều nhất trong Python là kiểu list.

Em hãy tìm một số dữ liệu kiểu danh sách thường gặp trên thực tế.

1. KIỂU DỮ LIỆU DANH SÁCH

Hoạt động 1 **Khởi tạo và tìm hiểu dữ liệu kiểu danh sách**

Khởi tạo dữ liệu danh sách như thế nào? Cách truy cập, thay đổi giá trị và xoá một phần tử trong danh sách như thế nào?

Ví dụ 1. Quan sát các lệnh sau để tìm hiểu kiểu dữ liệu danh sách.

```
>>> A = [1,2,3,4,5]
>>> B = [1.5, 2, "Python", "List", 0]
>>> A[0]
1
>>> B[2]
'Python'
```

Có thể truy cập từng phần tử của danh sách thông qua chỉ số. Chỉ số của list đánh số từ 0.

Kiểu dữ liệu danh sách trong Python được khởi tạo như sau:

```
<tên list> = [<v1>, <v2>,..., <vn>]
```

Trong đó các giá trị $<v_k>$ có thể có kiểu dữ liệu khác nhau (số nguyên, số thực, xâu kí tự,...). Ta có thể truy cập từng phần tử của danh sách thông qua chỉ số. Chỉ số bắt đầu từ 0.

Việc chỉ số hoá từng phần tử của danh sách cho thấy, có thể dùng danh sách để biểu diễn dữ liệu tương tự như kiểu mảng trong nhiều ngôn ngữ lập trình bậc cao khác. Tuy nhiên, *danh sách của Python có một khác biệt quan trọng, đó là nó có thể gồm các phần tử có kiểu dữ liệu khác nhau.*

Ví dụ 2. Quan sát các lệnh sau để biết cách thay đổi hoặc xoá phần tử của danh sách.

```
>>> A = [1,2,3,4,5]
>>> len(A)
5
```

Lệnh len() tính độ dài của danh sách.

```
>>> A[1] = "One"
>>> A
[1, 'One', 3, 4, 5]
>>> del A[4]
>>> A
[1, 'One', 3, 4]
```

Thay đổi giá trị từng phần tử bằng lệnh gán.

Lệnh **del** để xoá một phần tử của danh sách.

Sau khi đã khởi tạo danh sách, chúng ta có thể thay đổi các giá trị của từng phần tử bằng lệnh gán hoặc xoá phần tử bằng lệnh **del**.

Ví dụ 3. Quan sát các lệnh sau để biết cách tạo danh sách rỗng (có độ dài 0) và các phép toán ghép danh sách (phép +).

```
>>> a = []
>>> len(a)
0
>>> [1,2] + [3,4,5,6]
[1, 2, 3, 4, 5, 6]
```

Lệnh tạo một danh sách rỗng có độ dài bằng 0.

Các phép ghép hai danh sách.

- List là kiểu dữ liệu danh sách (dãy, mảng) trong Python. Tạo list bằng lệnh gán với các phần tử trong cặp dấu ngoặc []. Các phần tử của danh sách có thể có các kiểu dữ liệu khác nhau. Truy cập hoặc thay đổi giá trị của từng phần tử thông qua chỉ số:
  <danh sách>[<chỉ số>]
- Chỉ số của danh sách bắt đầu từ 0 đến len() – 1, trong đó len() là lệnh tính độ dài danh sách.

1. Cho danh sách A = [1, 0, "One", 9, 15, "Two", True, False]. Hãy cho biết giá trị các phần tử:

a) A[0] b) A[2] c) A[7] d) A[len(A)]

2. Giả sử A là một danh sách các số, mỗi lệnh sau thực hiện công việc gì?

a) A = A + [10] b) del A[0]

c) A = [100] + A d) A = A[1]*25

2. DUYỆT CÁC PHẦN TỬ CỦA DANH SÁCH

Hoạt động 2 Dùng lệnh for để duyệt danh sách

Quan sát các lệnh sau để biết cách dùng lệnh for duyệt lần lượt các phần tử của một danh sách.

Ví dụ 1. Duyệt và in ra từng phần tử của danh sách.

```
>>> A = [1,2,3,4,5]
>>> for i in range(len(A)):
        print(A[i],end = " ")
1 2 3 4 5
```

Biến i chạy trên vùng chỉ số từ 0 đến len(A) – 1.

Ví dụ 2. Duyệt và in một phần của danh sách.

```
>>> A = [3, 2, 1, 5, 6, 10, 7, 12, 18]
>>> for i in range(2,5):
        print(A[i],end = " ")
1 5 6
```

Hai ví dụ trên cho thấy dùng lệnh **for** kết hợp với lệnh range() để duyệt từng phần tử của danh sách.

Có thể duyệt lần lượt các phần tử của danh sách bằng lệnh for kết hợp với vùng giá trị của lệnh range().

1. Giải thích các lệnh ở mỗi câu sau thực hiện công việc gì?

a)
```
>>> S = 0
>>> for i in range(len(A)):
        if A[i] > 0:
            S = S + A[i]
>>> print(S)
```

b)
```
>>> C = 0
>>> for i in range(len(A)):
        if A[i] > 0:
            C = C + 1
>>> print(C)
```

2. Cho dãy các số nguyên A, viết chương trình in ra các số chẵn của A.

3. THÊM PHẦN TỬ VÀO DANH SÁCH

Python có những lệnh đặc biệt để thêm phần tử vào một danh sách. Các lệnh này được thiết kế riêng cho kiểu dữ liệu danh sách và còn được gọi là phương thức (method) của danh sách.

Hoạt động 3 **Tìm hiểu lệnh thêm phần tử cho danh sách**

Quan sát các lệnh sau đây để biết cách thêm phần tử vào một danh sách bằng phương thức append().

Ví dụ. Thêm phần tử vào cuối danh sách.

```
>>> A = [1,2]
>>> A.append(10)
>>> A
[1, 2, 10]
```

Chú ý cách dùng phương thức append(): gõ tên biến danh sách, dấu ".", sau đó gõ append.

- Python có một số lệnh dành riêng (phương thức) cho dữ liệu kiểu danh sách. Cú pháp các lệnh đó như sau:

  <danh sách>.<phương thức>

- Lệnh thêm phần tử vào cuối danh sách là <danh sách>.append().

1. Sau khi thêm một phần tử vào danh sách A bằng lệnh append() thì độ dài danh sách A thay đổi như thế nào?

2. Danh sách A sẽ như thế nào sau các lệnh sau?

```
>>> A = [2,4,10,1,0]
>>> A.append(100)
>>> del A[1]
```

THỰC HÀNH

Khởi tạo, nhập dữ liệu, thêm phần từ cho danh sách

Nhiệm vụ 1. Nhập số n từ bàn phím, sau đó nhập danh sách n tên các bạn lớp em và in ra danh sách các tên đó, mỗi tên trên một dòng.

Hướng dẫn. Chương trình yêu cầu nhập số tự nhiên n, sau đó nhập từng tên trong danh sách, dùng phương thức `append()` để đưa dần vào danh sách.

Chú ý: Vì vùng giá trị của lệnh `range(n)` bắt đầu từ 0 nên trong thông báo nhập cần viết là `str(i+1)` để bắt đầu từ 1.

Chương trình có thể như sau:

```
dsLop = []
n = int(input("Nhập số học sinh trong lớp: "))
for i in range(n):
    name = input("Nhập họ tên học sinh thứ " + str(i+1)+": ")
    dsLop.append(name)
print("Danh sách học sinh đã nhập:")
for i in range(len(dsLop)):
    print(dsLop[i])
```

Nhiệm vụ 2. Nhập một dãy số từ bàn phím. Tính tổng, trung bình của dãy và in dãy số trên một hàng ngang.

Hướng dẫn. Tương tự nhiệm vụ 1, chỉ khác là nhập số nguyên nên dùng lệnh `int()` để chuyển đổi dữ liệu.

```
A = []
T = 0
n = int(input("Nhập số tự nhiên n: "))
for i in range(n):
    num = int(input("Nhập số thứ " + str(i+1)+": "))
    A.append(num)
    T = T + num
print("Dãy số đã nhập:")
for i in range(n):
    print(A[i],end = " ")
print()
print("Tổng:",T)
print("Trung bình:",T/n)
```

LUYỆN TẬP

1. Viết lệnh xoá phần tử cuối cùng của danh sách A bằng lệnh **del**.
2. Có thể thêm một phần tử vào đầu danh sách được không? Nếu có thì nêu cách thực hiện.

VẬN DỤNG

Cho dãy số A. Viết chương trình tìm giá trị và chỉ số của phần tử lớn nhất của A. Tương tự với bài toán tìm phần tử nhỏ nhất.

BÀI 23 MỘT SỐ LỆNH LÀM VIỆC VỚI DỮ LIỆU DANH SÁCH

SAU BÀI NÀY EM SẼ:

- Biết cách duyệt danh sách bằng toán tử **in**.
- Biết và thực hiện được một số phương thức thường dùng với danh sách.

Trong bài trước chúng ta đã biết cách dùng lệnh append để thêm phần tử vào cuối một danh sách. Vậy Python có lệnh nào dùng để:

- Xoá nhanh một danh sách?
- Chèn thêm phần tử vào đầu hay giữa danh sách?
- Kiểm tra một phần tử có nằm trong một danh sách không?

1. DUYỆT DANH SÁCH VỚI TOÁN TỬ IN

Hoạt động 1 **Sử dụng toán tử `in` với danh sách**

Quan sát ví dụ sau để biết cách dùng toán tử **in** để duyệt một danh sách.

Ví dụ 1. Dùng toán tử in để kiểm tra một giá trị có nằm trong danh sách hay không.

```
>>> A = [1,2,3,4,5]
>>> 2 in A
True
>>> 10 in A
False
```

`2 in A` ← Số nguyên 2 nằm trong dãy A, kết quả trả lại True.

`10 in A` ← Số 10 không nằm trong dãy A, kết quả trả lại False.

Câu lệnh dùng toán tử **in** để kiểm tra <giá trị> có trong <danh sách> không, nếu có thì trả lại True nếu không thì trả về False như sau:

```
<giá trị> in <danh sách>
```

Ví dụ 2. Sử dụng toán tử **in** để duyệt từng phần tử của danh sách.

```
>>> A = [10,11,12,13,14,15]
>>> for k in A:
        print(k, end = " ")
10 11 12 13 14 15
```

`for k in A:` ← Khi thực hiện lệnh này, biến k sẽ lần lượt nhận các giá trị từ dãy A.

- Toán tử in dùng để kiểm tra một phần tử có nằm trong danh sách đã cho không. Kết quả trả lại `True` (Đúng) hoặc `False` (Sai).

  `<giá trị> in <danh sách>`

- Có thể duyệt nhanh từng phần tử của danh sách bằng toán tử `in` và lệnh `for` mà không cần sử dụng lệnh `range()`.

1. Giả sử A = ["0","1","01","10"]. Các biểu thức sau trả về giá trị đúng hay sai?

a) `1 in A`

b) `"01" in A`

2. Hãy giải thích ý nghĩa từ khoá **in** trong câu lệnh sau:

```
for i in range(10):
    <các lệnh>
```

2. MỘT SỐ LỆNH LÀM VIỆC VỚI DANH SÁCH

Hoạt động 2 Tìm hiểu một số lệnh làm việc với danh sách

Quan sát ví dụ sau để tìm hiểu một số lệnh làm việc với dữ liệu kiểu danh sách.

Ví dụ 1. Lệnh `clear()` xoá toàn bộ một danh sách.

```
>>> A = [1,2,3,4,5]
>>> A.clear()
>>> A
[]
```

← Sau khi thực hiện lệnh clear(), danh sách gốc trở thành rỗng.

Ví dụ 2. Lệnh `remove(value)` sẽ xoá phần tử đầu tiên của danh sách có giá trị value. Nếu không có phần tử nào như vậy thì sẽ báo lỗi.

```
>>> A = [1,2,3,4,5]
>>> A.remove(1)
>>> A
[2, 3, 4, 5]
>>> A.remove(10)
Traceback (most recent call last):
  File "<pyshell#25>", line 1, in <module>
    A.remove(10)
ValueError: list.remove(x): x not in list
```

← Lệnh remove() có chức năng xoá một phần tử có giá trị cho trước.

← Lệnh báo lỗi nếu giá trị không có trong danh sách.

Ví dụ 3. Lệnh `insert` có chức năng chèn phần tử vào danh sách tại chỉ số cho trước.

```
>>> A = [1,2,6,10]
>>> A.insert(2,5)
>>> A
[1, 2, 5, 6, 10]
```

← Lệnh insert() có hai tham số cần nhập: vị trí cần chèn và giá trị được chèn. Lệnh `insert(2,5)` sẽ chèn số 5 tại chỉ số 2.

Lệnh `insert(index,value)` sẽ chèn giá trị value vào danh sách tại vị trí `index` và đẩy các phần tử từ vị trí này sang phải.

Chú ý nếu k nằm ngoài phạm vi chỉ số của danh sách thì lệnh vẫn có tác dụng: nếu k < 0 thì chèn vào đầu danh sách, nếu k > `len()` thì chèn vào cuối danh sách.

```
>>> A = []
>>> A.insert(-10,1)
```

← Lệnh này chèn số 1 vào đầu danh sách A.

```
>>> A.insert(100,2)
>>> A
[1, 2]
```

Lệnh này chèn số 2 vào cuối danh sách A.

Một số lệnh làm việc với dữ liệu danh sách:

`A.append(x)`	Bổ sung phần tử **x** vào cuối danh sách A.
`A.insert(k,x)`	Chèn phần tử **x** vào vị trí **k** của danh sách A.
`A.clear()`	Xoá toàn bộ dữ liệu của danh sách A.
`A.remove(x)`	Xoá phần tử **x** từ danh sách A.



1. Khi nào thì lệnh `A.append(1)` và `A.insert(0,1)` có tác dụng giống nhau?

2. Danh sách A trước và sau lệnh `insert()` là `[1,4,10,0]` và `[1,4,10,5,0]`. Lệnh đã dùng là gì?



THỰC HÀNH

Các lệnh làm việc với dữ liệu kiểu danh sách

Nhiệm vụ 1. Nhập số n từ bàn phím, sau đó nhập danh sách n tên học sinh trong lớp và in ra danh sách học sinh này, mỗi tên học sinh trên một dòng. Yêu cầu danh sách được in ra theo thứ tự ngược lại với thứ tự đã nhập.

Hướng dẫn. Chương trình sẽ yêu cầu nhập số tự nhiên n, sau đó sẽ lần lượt yêu cầu nhập n tên học sinh. Tuy nhiên do yêu cầu in danh sách học sinh theo thứ tự ngược lại so với thứ tự nhập nên cần dùng lệnh `insert()` để chèn tên học sinh được nhập vào đầu danh sách. Chương trình có thể như sau:

```
dsLop = []
n = int(input("Nhập số học sinh trong lớp: "))
for i in range(n):
    name = input("Nhập họ tên học sinh thứ " + str(i+1) + ": ")
    dsLop.insert(0,name)
print("Danh sách học sinh đã nhập:")
for name in dsLop:
    print(name)
```

Nhiệm vụ 2. Cho trước dãy số A. Viết chương trình xoá đi các phần tử có giá trị nhỏ hơn 0 từ A.

Hướng dẫn. Duyệt từng phần tử của dãy số A, kiểm tra nếu phần tử này nhỏ hơn 0 thì xoá đi.

Chương trình có thể như sau:

```
A = [0,1,-3,-10,5,9,-20,55]
i = 0
while i < len(A):
    if A[i] < 0:
```

```
        A.remove(A[i])
    else:
        i = i + 1
print(A)
```

Nhiệm vụ 3. Cho trước dãy số A. Viết chương trình tìm và chỉ ra vị trí đầu tiên của dãy số A mà ba số hạng liên tiếp có giá trị là 1, 2, 3. Nếu tìm thấy thì thông báo vị trí tìm thấy, nếu không thì thông báo "Không tìm thấy mẫu".

Hướng dẫn. Soạn thảo chương trình sau rồi thực hiện và kiểm tra tính đúng đắn của chương trình.

```
A = [0,4,0,1,2,3,8,9,0,1,2,3,17,-16,0,1,2]
p = [1,2,3]
pkq = -1
i = 0
while i < len(A) - 3 and pkq == -1:
    if A[i] == p[0] and A[i+1] == p[1] and A[i+2] == p[2]:
        pkq = i
    else:
        i = i + 1
if pkq >= 0:
    print("Tìm thấy mẫu ",p," tại vị trí", pkq)
else:
    print("Không tìm thấy mẫu",p)
```

LUYỆN TẬP

1. Cho dãy số [1,2,2,3,4,5,5]. Viết lệnh thực hiện:

a) Chèn số 1 vào ngay sau giá trị 1 của dãy.

b) Chèn số 3 và số 4 vào danh sách để dãy có số 3 và số 4 liền nhau hai lần.

2. Cho trước dãy số A. Viết chương trình thực hiện công việc sau:

– Xoá đi một phần tử ở chính giữa dãy nếu số phần tử của dãy là số lẻ.

– Xoá đi hai phần tử ở chính giữa của dãy nếu số phần tử của dãy là số chẵn.

VẬN DỤNG

1. Viết chương trình nhập n từ bàn phím, tạo và in ra màn hình dãy số A bao gồm n số tự nhiên chẵn đầu tiên.

2. Dãy số Fibonacci được xác định như sau:

$F_0 = 0$

$F_1 = 1$

$F_n = F_{n-1} + F_{n-2}$ (với $n \geq 2$).

Viết chương trình nhập n từ bàn phím, tạo và in ra màn hình dãy số A bao gồm n số hạng đầu của dãy Fibonacci.

BÀI 24 XÂU KÍ TỰ

SAU BÀI NÀY EM SẼ:

- Hiểu được xâu kí tự là kiểu dữ liệu cơ bản của Python.
- Biết và thực hiện được lệnh for để xử lí xâu kí tự.

Em đã biết dữ liệu xâu kí tự (gọi tắt là xâu) từ Bài 16 và chúng ta có thể tạo các biến kiểu xâu kí tự theo nhiều cách như sau:

```
>>> s = "Thời khoá biểu"
>>> xau = 'Hoa học trò'
>>> Cau_tho = """ Mình về mình có nhớ ta
Mười lăm năm ấy thiết tha mặn nồng"""
```

Liệu có lệnh nào trích ra từng kí tự của một xâu kí tự? Đếm số kí tự của một xâu?

1. XÂU LÀ MỘT DÃY CÁC KÍ TỰ

Hoạt động 1 Tìm hiểu cấu trúc của xâu kí tự

Quan sát các ví dụ sau để biết cấu trúc xâu kí tự, so sánh với danh sách để biết sự khác nhau giữa xâu (`string`) và danh sách (`list`).

Ví dụ 1. Xâu kí tự và cách truy cập đến từng kí tự của xâu.

```
>>> s = "Thời khoá biểu"
>>> len(s)
14
>>> s[0]
'T'
>>> s[10]
'b'
```

Lệnh **`len()`** sẽ tính độ dài của xâu hay số lượng các kí tự có trong xâu.

Có thể truy cập từng kí tự của xâu thông qua chỉ số. Chỉ số bắt đầu từ 0.

Một xâu kí tự được hiểu là một dãy các kí tự. Tương tự danh sách, ta có thể truy cập từng kí tự của xâu thông qua chỉ số, chỉ số bắt đầu từ 0.

Ví dụ 2. Quan sát các lệnh sau để thấy sự khác nhau giữa xâu và danh sách.

```
>>> d = ["a","b","c"]
>>> d[0] = "A"
>>> s = "abc"
>>> s[0] = "A"
Traceback (most recent call last):
  File "<pyshell#15>", line 1, in <module>
    s[0] = "A"
TypeError: 'str' object does not support item assignment
```

Python không cho phép thay đổi từng kí tự của một xâu. Điều này khác với danh sách.

Python không có kiểu dữ liệu kí tự. Kí tự chính là xâu có độ dài 1. Xâu rỗng được định nghĩa như sau:

```
empty = ""
```

> Xâu kí tự trong Python là dãy các kí tự Unicode. Xâu có thể được coi là danh sách các kí tự nhưng không thay đổi từng kí tự của xâu. Truy cập từng kí tự của xâu qua chỉ số, chỉ số từ 0 đến độ dài len() – 1.

1. Các xâu kí tự sau có hợp lệ không?

a) `"123&*()+-ABC"` b) `"1010110&0101001"`

c) `"Tây Nguyên"` d) `11111111 = 256`

2. Mỗi xâu hợp lệ ở Câu 1 có độ dài bằng bao nhiêu?

2. LỆNH DUYỆT KÍ TỰ CỦA XÂU

Hoạt động 2 Tìm hiểu lệnh duyệt từng kí tự của xâu

Quan sát các lệnh sau để biết cách duyệt từng kí tự của xâu kí tự bằng lệnh `for`. Có hai cách duyệt, theo chỉ số và theo phần tử của xâu kí tự.

```
>>> s = "Thời khoá biểu"
>>> for i in range(len(s)):
        print(s[i],end = " ")

T h ờ i   k h o á   b i ể u
>>> for ch in s:
        print(ch, end = " ")

T h ờ i   k h o á   b i ể u
```

Duyệt theo chỉ số với lệnh `range()`.

Duyệt theo kí tự của xâu kí tự.

– Cách thứ nhất, biến i lần lượt chạy theo chỉ số của xâu kí tự s, từ 0 đến `len(s) - 1`. Kí tự tại chỉ số i là s[i].

– Cách thứ hai duyệt theo từng kí tự của xâu s. Biến ch sẽ được gán lần lượt các kí tự của xâu s từ đầu đến cuối.

Chú ý: Từ khoá **in**, tuỳ trường hợp cụ thể, hoặc là toán tử lôgic dùng để kiểm tra một giá trị có mặt hay không trong một vùng giá trị/danh sách/xâu, hoặc để chọn lần lượt từng phần tử trong một vùng giá trị/danh sách/xâu.

```
>>> "a" in "abcd"
True
>>> "abc" in "abcd"
True
```

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

Có thể duyệt các kí tự của xâu bằng lệnh **for** tương tự như với danh sách.

s_1 **in** s_2 trả lại giá trị **True** nếu s_1 là xâu con của s_2.

1. Sau khi thực hiện các lệnh sau, biến skq sẽ có giá trị bao nhiêu?

```
>>> s = "81723"
>>> skq = ""
>>> for ch in s:
        if int(ch) % 2 != 0:
            skq = skq + ch
```

2. Cho s1 = "abc", s2 = "ababcabca". Các biểu thức lôgic sau cho kết quả là đúng hay sai?

a) `s1 in s2`

b) `s1 + s1 in s2`

c) `"abcabca" in s2`

d) `"abc123" in s2`

THỰC HÀNH

Các lệnh cơ bản làm việc với xâu kí tự

Nhiệm vụ 1. Viết chương trình nhập số tự nhiên n là số học sinh, sau đó nhập họ và tên học sinh. Lưu họ và tên học sinh vào một danh sách. In danh sách ra màn hình, mỗi họ tên trên một dòng.

Hướng dẫn. Chương trình có thể như sau:

```
n = int(input("Nhập số học sinh trong lớp: "))
ds_lop = []
for i in range(n):
    hoten = input("Nhập họ tên học sinh thứ " + str(i+1)+": ")
    ds_lop.append(hoten)
print("Danh sách lớp học:")
for i in range(n):
    print(ds_lop[i])
```

Nhiệm vụ 2. Nhập một xâu kí tự S từ bàn phím rồi kiểm tra xem xâu S có chứa xâu con "10" không.

Hướng dẫn. Cách 1. Nếu xâu S chứa xâu con "10" thì sẽ có chỉ số k mà `S[k] = "1"` và `S[k+1] = "0"`. Cách 2. Dùng toán tử **in** để kiểm tra xâu "10" có là xâu con của S.

Cách 1: Duyệt kí tự của xâu theo chỉ số.

```
S = input("Nhập xâu kí tự bất kì: ")
kq = False
```

```
for i in range(len(S)-1):
    if S[i] == "1" and S[i+1] == "0":
        kq = True
        break
if kq:
    print("Xâu gốc có chứa xâu '10'")
else:
    print("Xâu gốc không chứa xâu '10'")
```

Cách 2: Sử dụng toán tử **in**.

```
S = input("Nhập xâu kí tự bất kì: ")
s10 = "10"
if s10 in S:
    print("Xâu gốc có chứa xâu '10'")
else:
    print("Xâu gốc không chứa xâu '10'")
```



LUYỆN TẬP

1. Cho xâu S, viết đoạn lệnh trích ra xâu con của S bao gồm ba kí tự đầu tiên của S.

2. Viết chương trình kiểm tra xâu S có chứa chữ số không. Thông báo "S có chứa chữ số" hoặc "S không chứa chữ số nào".



VẬN DỤNG

1. Cho hai xâu s_1, s_2. Viết đoạn chương trình chèn xâu s_1 vào giữa s_2, tại vị trí $len(s_2)//2$. In kết quả ra màn hình.

2. Viết chương trình nhập số học sinh và họ tên học sinh. Sau đó đếm xem trong danh sách có bao nhiêu bạn tên là "Hương".

Gợi ý: Sử dụng toán tử **in** để kiểm tra một xâu có là xâu con của một xâu khác.

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

BÀI 25

MỘT SỐ LỆNH LÀM VIỆC VỚI XÂU KÍ TỰ

SAU BÀI NÀY EM SẼ:

- Biết và thực hiện được một số lệnh thường dùng với xâu kí tự.

Bài toán tìm kiếm xâu con trong một xâu là một trong những bài toán tin học được ứng dụng nhiều trong thực tế. Công cụ tìm kiếm thông tin trên Internet hay lệnh tìm kiếm trong soạn thảo văn bản được xây dựng trên cơ sở bài toán tìm xâu con.

Cho xâu c = "Trường Sơn" và xâu m = "Bước chân trên dải Trường Sơn". Em hãy cho biết xâu c có là xâu con của xâu m không. Nếu có thì tìm vị trí của xâu c trong xâu m.

1. XÂU CON VÀ LỆNH TÌM VỊ TRÍ XÂU CON

Hoạt động 1 Một số lệnh tìm kiếm xâu con trong xâu kí tự

Quan sát các ví dụ sau để tìm hiểu cách kiểm tra xâu con và tìm kiếm vị trí xâu con trong xâu kí tự.

Ví dụ 1. Dùng toán tử `in` để kiểm tra một xâu có là xâu con của xâu khác không.

```
>>> "abc" in "123abc"
True
>>> "010" in "1101"
False
```

Biểu thức kiểm tra <xâu 1> nằm trong <xâu 2> là:

```
<xâu 1> in <xâu 2>
```

Nếu đúng thì trả lại giá trị True, nếu sai trả lại giá trị False.

Ví dụ 2. Lệnh find() tìm vị trí xuất hiện của một xâu trong xâu khác.

```
>>> s = "ab bc cd 123 456 00"
>>> s.find("b")
1      <-- Vị trí xuất hiện đầu tiên của "b" trong xâu s là chỉ số 1.
>>> s.find("12")
9      <-- Vị trí tìm thấy đầu tiên của "12" trong xâu s là chỉ số 9.
>>> s.find("AB")
-1     <-- Không tìm thấy xâu "AB" trong xâu s nên trả về -1.
```

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

Tương tự danh sách, Python cũng có một số lệnh đặc biệt dành riêng cho xâu kí tự (phương thức). Cách thực hiện phương thức là:

```
<xâu>.<phương thức>
```

Cú pháp đơn của lệnh find():

```
<xâu mẹ>.find(<xâu con>)
```

Lệnh sẽ tìm vị trí đầu tiên của xâu con trong xâu mẹ và trả về vị trí đó. Nếu không tìm thấy thì trả về -1.

Cú pháp đầy đủ của lệnh find():

```
<xâu mẹ>.find(<xâu con>,start)
```

Lệnh sẽ tìm vị trí xâu con bắt đầu từ vị trí start.

Ví dụ 3

```
>>> sub = "Đà Nẵng"
>>> s = "Hà Nội - Đà Nẵng - Hồ Chí Minh"
>>> s.find(sub)
9
>>> s.find(sub,10)
-1
```

Để tìm một xâu trong một xâu khác có thể dùng toán tử in hoặc lệnh find(). Lệnh find() trả về vị trí của xâu con trong xâu mẹ.



1. Biểu thức lôgic sau là đúng hay sai?

```
>>> "010" in "001100"
```

2. Lệnh sau trả lại giá trị gì?

```
>>> "abababababab".find("ab",4)
```

2. MỘT SỐ LỆNH THƯỜNG DÙNG VỚI XÂU KÍ TỰ

Hoạt động 2 **Một số lệnh thường dùng với xâu kí tự**

Quan sát các ví dụ sau để biết cách sử dụng một số lệnh thường dùng với xâu kí tự như: split(), join().



Ví dụ 1. Lệnh split() tách một xâu thành danh sách các từ.

```
>>> s = "Tiên học lễ hậu học văn"
>>> s.split()
['Tiên', 'học', 'lễ', 'hậu', 'học', 'văn']
>>> st = "0,1,2,3,4,5,6,10"
>>> st.split(",")
['0', '1', '2', '3', '4', '5', '6', '10']
```

Tách xâu dùng dấu cách để phân biệt tách.

Tách xâu dùng dấu "," để phân biệt tách.

Lệnh `split()` tách một xâu thành các từ và đưa vào một danh sách. Kí tự tách dùng để phân tách các từ mặc định là dấu cách, tuy nhiên có thể thay thế kí tự tách bằng kí tự khác.

Cú pháp của lệnh `split()`.

```
<xâu mẹ>.split(<kí tự tách>)
```

Ví dụ 2. Lệnh `join()` nối danh sách gồm các từ thành một xâu.

```
>>> A = ['Tiên', 'học', 'lễ', 'hậu', 'học', 'văn']
>>> " ".join(A)
'Tiên học lễ hậu học văn'
>>> B = ['0', '1', '2', '3', '4', '5', '6', '10']
>>> ",".join(B)
'0,1,2,3,4,5,6,10'
```

Lệnh `join()` này sẽ nối các phần tử của danh sách A bởi dấu cách.

Lệnh `join()` này sẽ nối các phần tử của danh sách B bởi dấu ",".

Lệnh `join()` có tác dụng ngược với lệnh `split()`, có chức năng nối các phần tử (là xâu) của một danh sách thành một xâu. Cú pháp của lệnh `join()` là:

```
"kí tự nối".join(<danh sách>)
```

Python có các lệnh đặc biệt để xử lí xâu là `split()` dùng để tách xâu thành danh sách và lệnh `join()` dùng để nối danh sách các xâu thành một xâu.

Cho xâu kí tự: "gà,vịt,chó,lợn,ngựa,cá". Em hãy trình bày cách làm để xoá các dấu "," và thay thế bằng dấu " " trong xâu này.

THỰC HÀNH

Một số bài toán liên quan đến xâu kí tự.

Nhiệm vụ 1. Viết chương trình nhập nhiều số nguyên từ bàn phím, các số cách nhau bởi dấu cách. Khi nhập xong thông báo số lượng các số đã nhập và in các số này thành hàng ngang.

Hướng dẫn. Dữ liệu nhập vào là một xâu. Dùng lệnh `split()` để tách thành danh sách. Chuyển các phần tử của danh sách này thành số và in ra màn hình.

```
s = input("Nhập các số nguyên cách nhau bởi dấu cách: ")
sline = s.split()
n = len(sline)
nline = []
for x in sline:
    nline.append(int(x))
print("Bạn đã nhập",n,"số.")
for k in nline:
    print(k, end = " ")
```

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

Nhiệm vụ 2. Viết chương trình nhập một xâu kí tự có thể có nhiều dấu cách giữa các từ. Sau đó chỉnh sửa xâu kí tự đó sao cho giữa các từ chỉ có một dấu cách. In xâu kết quả ra màn hình.

Hướng dẫn. Chuyển xâu kí tự ban đầu thành danh sách các từ đơn bằng lệnh `split()`, sau đó nối các từ đơn này bằng lệnh `join()`.

```
s = input("Nhập đoạn văn bản: \n")
sline = s.split()
skq = " ".join(sline)
print(skq)
```

Nhiệm vụ 3. Viết chương trình nhập số tự nhiên n, rồi nhập họ tên của n học sinh. Sau đó in ra danh sách tên học sinh theo hai cột, cột 1 là tên, cột 2 là họ đệm.

Hướng dẫn. Họ tên ban đầu tách ra thành tên và họ đệm bằng lệnh `split()`. Các tên được đưa vào danh sách ten, các họ đệm được đưa vào danh sách hodem. Sau đó in ra danh sách theo yêu cầu.

```
n = int(input("Nhập số học sinh trong lớp: "))
ten = []
hodem = []
for i in range(n):
    s = input("Nhập họ tên học sinh thứ "+str(i+1)+": ")
    sline = s.split()
    m = len(sline)-1
    ten.append(sline[m])
    del sline[m]
    hodem.append(" ".join(sline))
print("Danh sách học sinh:")
for i in range(n):
    print(ten[i],hodem[i])
```

LUYỆN TẬP

1. Viết chương trình nhập nhiều số (số nguyên hoặc số thực) từ bàn phím, các số cách nhau bởi dấu cách. Sau đó in ra màn hình tổng các số đã nhập.
2. Viết chương trình nhập họ tên đầy đủ của người dùng, sau đó in thông báo tên và họ đệm của người đó.

VẬN DỤNG

1. Viết chương trình nhập hai số tự nhiên từ bàn phím, cách nhau bởi dấu cách và đưa ra kết quả là ƯCLN của hai số này.
2. Viết chương trình nhập số tự nhiên n rồi nhập n họ tên học sinh. Sau đó yêu cầu nhập một tên và thông báo số bạn có cùng tên đó trong lớp.

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

BÀI 26

HÀM TRONG PYTHON

SAU BÀI NÀY EM SẼ:

- Biết được chương trình con là hàm.
- Biết cách tạo hàm.

Các chương trình giải những bài toán thực tế phức tạp thường có rất nhiều dòng lệnh, trong đó có không ít những khối lệnh tương ứng với một số thao tác được lặp đi lặp lại nhiều lần ở những vị trí khác nhau. Để đỡ công viết đi viết lại các khối lệnh đó, trong tổ chức chương trình viết bằng ngôn ngữ lập trình bậc cao, người ta thường gom các khối lệnh như vậy thành những chương trình con. Khi đó, trong chương trình người ta chỉ cần thay cả khối lệnh bằng một lệnh gọi chương trình con tương ứng. Trong Python, các hàm chính là các chương trình con.

Em có thể kể tên một số hàm trong số các lệnh đã học hay không? Các hàm đó có những đặc điểm chung gì?

1. MỘT SỐ HÀM THIẾT KẾ SẴN CỦA PYTHON

Hoạt động 1 **Tìm hiểu một số hàm của Python**

Quan sát một số câu lệnh trong Bảng 26.1 và cho biết những câu lệnh này có điểm chung gì.

Bảng 26.1. *Một số lệnh trong Python*

`abs()`	`len()`	`range()`	`bool()`	`float()`
`list()`	`round()`	`chr()`	`input()`	`ord()`
`str()`	`divmod()`	`int()`	`print()`	`type()`

Trước tiên, về hình thức, em có thể thấy các lệnh trên đều có các dấu mở đóng ngoặc đi sau tên lệnh. Tiếp đó, khi viết trong chương trình, bên trong các dấu ngoặc, nói chung, em có thể cần ghi thêm các tham số là các đại lượng, các biến hoặc thậm chí, trong một số trường hợp, cả các biểu thức. Ví dụ:

– Lệnh `print("Thời khoá biểu")` thực hiện việc in xâu kí tự `"Thời khoá biểu"` trong dấu ngoặc ra màn hình.

– Lệnh `x = int("52")` chuyển xâu `"52"` thành số nguyên 52.

– Lệnh `type(y)` trả lại kiểu dữ liệu của biến y.

Trong một số trường hợp bên trong dấu ngoặc có thể bỏ trống. Ví dụ, lệnh `x = input()` thực hiện yêu cầu nhập vào một xâu kí tự bất kì và gán cho biến x.

Các lệnh trong Bảng 26.1 chính là các chương trình con được thiết kế sẵn của Python, cho phép người dùng tuỳ ý sử dụng trong các chương trình của riêng mình.

Trong các ví dụ trên, xâu kí tự bên trong ngoặc của các hàm `int()` và `print()` là tham số của hàm. Cú pháp câu lệnh gọi hàm trong Python có dạng chung như sau:

```
<tên hàm>(<danh sách tham số hàm>)
```

Python cung cấp sẵn nhiều hàm thực hiện những công việc khác nhau cho phép người dùng được tuỳ ý sử dụng khi viết chương trình bằng các câu lệnh gọi hàm tương ứng.

Mô tả tham số và giá trị trả lại của mỗi hàm sau: `float()`, `str()`, `len()`, `list()`.

2. THIẾT LẬP CÁC HÀM TỰ ĐỊNH NGHĨA

Ngoài các hàm thiết kế sẵn, Python còn cho phép người dùng tự thiết lập các hàm của riêng mình (còn gọi là các hàm tự định nghĩa).

Hoạt động 2 Cách thiết lập hàm trong Python

Quan sát các ví dụ sau để biết cách viết hàm.

Ví dụ 1. Cách viết hàm có trả lại giá trị.

```
>>> def inc(n):
        return n+1
>>> inc(3)
4
```

Tên hàm: **inc**.
Tham số hàm: số n.
Giá trị trả lại: số n + 1.

Ví dụ 2. Cách viết hàm không trả lại giá trị.

```
>>> def thong_bao(msg):
        print("Xin chào bạn", msg)
        return
>>> thong_bao("Trần Quang Minh")
Xin chào bạn Trần Quang Minh
```

Tên hàm: **thong_bao**.
Tham số hàm: xâu kí tự msg.
Giá trị trả lại: không có.

Hàm trong Python được định nghĩa bằng từ khoá `def`, theo sau là tên hàm (tên hàm sẽ theo quy tắc đặt tên định danh). Hàm có thể có hoặc không có tham số. Khối lệnh mô tả hàm được viết sau dấu " : " và viết lùi vào, thẳng hàng. Hàm có thể có hoặc không có giá trị trả lại sau từ khoá `return`.

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

Cú pháp thiết lập hàm có trả lại giá trị.

```
def <tên hàm>(<tham số>):
    <khối lệnh>
    return <giá trị>
```

Cần có lệnh **return** <giá trị>. Hàm sẽ kết thúc khi gặp lệnh **return** và trả lại <giá trị>.

Cú pháp thiết lập hàm không trả lại giá trị.

```
def <tên hàm>(<tham số>):
    <khối lệnh>
    return
```

Lệnh **return** không có giá trị trả lại. Hàm kết thúc khi gặp lệnh **return**. Nếu hàm không trả lại giá trị thì có thể không cần lệnh return.

Để thiết lập hàm trả lại giá trị, câu lệnh **return** trong khai báo hàm cần có <giá trị> đi kèm. Để thiết lập hàm không trả lại giá trị có thể dùng lệnh **return** không có <giá trị> hoặc không cần có **return**.

Quan sát các hàm sau, giải thích cách thiết lập và chức năng của mỗi hàm.

a)

```
def Nhap_xau():
    msg = input("Nhập một xâu: ")
    return msg
```

b)

```
def Inday(n):
    for k in range(n):
        print(k, end = " ")
```

THỰC HÀNH

Thiết lập hàm trong Python.

Nhiệm vụ 1. Viết hàm yêu cầu người dùng nhập họ tên rồi đưa lời chào ra màn hình.

Hướng dẫn. Chương trình có thể như sau:

```
def meeting():
    ten = input("Nhập họ tên của em: ")
    print("Xin chào ",ten)
meeting()
```

Nhiệm vụ 2. Viết hàm `prime(n)` với tham số là số tự nhiên n và trả lại `True` nếu n là số nguyên tố, trả lại `False` nếu n không phải là số nguyên tố.

Hướng dẫn. Số nguyên tố là số tự nhiên lớn hơn 1, không có ước nào ngoài 1 và chính nó. Để thiết lập hàm `prime(n)` chúng ta cần tính số ước thực sự của n (từ 1 đến n – 1). Biến C dùng để đếm số các ước thực sự của n. Khi đó, n sẽ là số nguyên tố khi và chỉ khi C = 1.

Hàm `prime(n)` và chương trình có thể được thiết lập như sau:

```
def prime(n):
    C = 0
    k = 1
```

Ban đầu đặt k = 1. Vòng lặp sẽ tăng k lên 1 đơn vị cho đến khi k = n thì dừng. Với mỗi k, kiểm tra nếu k là ước của n thì tăng C lên 1.

```
while k < n:
    if n%k == 0:
        C = C + 1
    k = k + 1
if C == 1:
    return True
else:
    return False
```

LUYỆN TẬP

1. Viết hàm với tham số là số tự nhiên n in ra các số là ước nguyên tố của n.

Gợi ý: Sử dụng hàm `prime()` trong phần thực hành.

2. Viết hàm `numbers(s)` đếm số các chữ số có trong xâu s.

Ví dụ `numbers("0101abc") = 4`.

VẬN DỤNG

1. Trong khi viết hàm có thể có nhiều lệnh **return**. Quan sát hàm sau và giải thích ý nghĩa của những lệnh **return**. Hàm này có điểm gì khác so với hàm `prime()` đã được mô tả trong phần thực hành.

```
def prime(n):
    if n < 2:
        return False
    C = 0
    k = 2
    while k < n:
        if n%k == 0:
            return False
        k = k + 1
    return True
```

2. Viết chương trình yêu cầu nhập từ bàn phím một xâu kí tự, sau đó thông báo:

– Tổng số các kí tự là chữ số của xâu.

– Tổng số các kí tự là chữ cái tiếng Anh trong xâu.

Viết hàm cho mỗi yêu cầu trên.

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

BÀI 27

THAM SỐ CỦA HÀM

SAU BÀI NÀY EM SẼ:

- Biết cách thiết lập các tham số của hàm. Hiểu được cách truyền giá trị thông qua đối số hàm.
- Biết viết chương trình có sử dụng chương trình con.

Quan sát các lệnh sau và cho biết sự khác nhau giữa tham số (parameter) và đối số (argument).

Tham số

```
>>> def f_sum(a,b,c):
        return a+b+c
```

Đối số

```
>>> x,y =5,3
>>> f_sum(10,x,y):
18
```

1. THAM SỐ VÀ ĐỐI SỐ CỦA HÀM

Hoạt động 1 **Phân biệt tham số và đối số**

Quan sát ví dụ sau, tìm hiểu cách dữ liệu được truyền qua tham số vào hàm. Thảo luận để giải thích kết quả.

Ví dụ. Cách truyền dữ liệu qua tham số.

```
>>> def f(a,b,c):
        return a+b+c
>>> f(1,2,3)
6
>>> x,y,z = 10,20,5
>>> f(x,y,z)
35
>>> f(a,b,c)
Traceback (most recent call last):
  File "<pyshell#6>", line 1, in <module>
    f(a,b,c)
NameError: name 'a' is not defined
>>>
```

Hàm f() có ba tham số a, b, c.

Hàm f() được gọi với ba giá trị cụ thể.

Hàm f() được gọi với ba biến đã có giá trị.

Lời gọi hàm bị lỗi nếu các tham số được truyền vào chưa có giá trị.

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

Giải thích:

– Dòng 1: Hàm f() đã được định nghĩa với ba tham số a, b, c. Hàm có trả lại giá trị là a + b + c. Chú ý trong định nghĩa hàm, các tham số được coi như biến.

– Dòng 3: Hàm f() được gọi với ba giá trị cụ thể là 1, 2, 3. Các giá trị được truyền qua tham số được gọi là đối số. Đối số tại dòng 3 là các số cụ thể.

– Dòng 6: Hàm f() được gọi với ba biến x, y, z, đã được gán giá trị (dòng 5). Các biến được truyền qua tham số được gọi là đối số, kết quả trả lại là x + y + z.

– Dòng 10: Hàm f() được gọi với ba biến a, b, c không xác định giá trị nên lời gọi hàm f(a, b, c) báo lỗi do không xác định được giá trị của a, b, c.

Tham số của hàm được định nghĩa khi khai báo hàm và được dùng như biến trong định nghĩa hàm. **Đối số** là giá trị được truyền vào khi gọi hàm. Khi gọi hàm, các **tham số (parameter)** sẽ được truyền bằng giá trị thông qua **đối số (argument)** của hàm, số lượng giá trị được truyền vào hàm bằng với số tham số trong khai báo của hàm.

1. Một hàm khi khai báo có một tham số, nhưng khi gọi hàm có thể có hai đối số được không?

2. Giả sử hàm f có hai tham số x, y khi khai báo, hàm sẽ trả lại giá trị x + 2y. Lời gọi hàm f(10,a) có lỗi hay không?

2. CÁCH SỬ DỤNG CHƯƠNG TRÌNH CON

Hoạt động 2 Khi nào nên sử dụng chương trình con?

Bài toán đưa ra là viết chương trình chính yêu cầu nhập số tự nhiên n từ bàn phím và in các số nguyên tố nhỏ hơn hoặc bằng n ra màn hình. Trong phần thực hành của Bài 26 em đã biết hàm `prime(n)` kiểm tra số n có là số nguyên tố.

Em sẽ viết chương trình giải bài toán này như thế nào?

Ví dụ 1. Việc kiểm tra một số có là số nguyên tố được lặp đi lặp lại từ 1 đến n và do đó nên sử dụng hàm `prime(n)` để kiểm tra sẽ giúp chương trình cấu trúc rõ ràng và dễ hiểu hơn.

Chương trình hoàn chỉnh giải bài toán trên có thể được viết như sau:

```
def prime(n):
    C = 0
    k = 1
    while k < n:
        if n%k == 0:
            C = C + 1
        k = k + 1
```

```
    if C == 1:
        return True
    else:
        return False
# Chương trình chính
n = int(input("Nhập số tự nhiên n: "))
for k in range(1,n+1):
    if prime(k):
        print(k, end = " ")
```

Sử dụng hàm prime đã định nghĩa ở trên để kiểm tra số k: Hàm prime(k) sẽ trả lại True nếu k là số nguyên tố.

Ví dụ 2. Chương trình sử dụng chương trình con.

Cho trước hai dãy số B, C, chương trình chính cần tính tổng các số hạng dương của mỗi dãy này. Chúng ta sẽ thiết lập hàm `tongduong(A)` để tính tổng các số hạng lớn hơn 0 của một dãy A. Chương trình chính sẽ gọi hàm `tongduong(A)`.

Chương trình có thể như sau:

```
>>> def tongduong(A):
    S = 0
    for k in A:
        if k > 0:
            S = S + k
    return S
# Chương trình chính
>>> A = [0,2,-1,5,10,-3]
>>> B = [1,-10,-11,8,2,0,-5]
>>> tongduong(A)
17
>>> tongduong(B)
>>> print(B)
11
```

Sử dụng hàm tongduong tính tổng các số dương của dãy A

Sử dụng hàm tongduong tính tổng các số dương của dãy B

Sử dụng chương trình con có thể giúp phân chia việc giải một bài toán lớn thành giải quyết các bài toán nhỏ và phát huy được tinh thần làm việc nhóm; Chương trình chính có cấu trúc rõ ràng, dễ hiểu hơn; Nếu cần hiệu chỉnh, phát triển và nâng cấp cũng thuận tiện hơn.



1. Sử dụng hàm prime, em hãy viết chương trình in ra các số nguyên tố trong khoảng từ m đến n, với m, n là hai số tự nhiên và 1< m < n.

2. Em hãy nêu một công việc/bài toán nào đó mà có thể sử dụng hàm để giải.

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

THỰC HÀNH

Truyền giá trị cho đối số của hàm.

Nhiệm vụ 1. Thiết lập hàm `f_sum(A,b)` có chức năng tính tổng các số của danh sách A theo quy định sau:

– Nếu b = 0 thì tính tổng các số của danh sách A.

– Nếu b khác 0 thì chỉ tính tổng các số dương của A.

Hướng dẫn. Chương trình luôn kiểm tra giá trị của đối số b khi tính tổng các số của danh sách A.

Chương trình có thể như sau:

```
def f_sum(A,b):
    S = 0
    for x in A:
        if b == 0:
            S = S + x
        else:
            if x > 0:
                S = S + x
    return S
```

Nhiệm vụ 2. Thiết lập hàm `f_dem(msg, sep)` có chức năng đếm số các từ của một xâu msg với kí tự tách từ là sep.

Ví dụ:

```
f_dem("Mùa thu lịch sử"," ")  <-- Trả lại giá trị 4.
f_dem("Mùa thu lịch sử","-")  <-- Trả lại giá trị 1.
```

Hướng dẫn. Để tách xâu msg thành các từ, chúng ta dùng lệnh `split()`. Tham số sep chính là tham số của lệnh `split()`.

Chương trình có thể như sau:

```
def f_dem(msg,sep):
    xlist = msg.split(sep)
    return len(xlist)
```

Nhiệm vụ 3. Thiết lập hàm `merge_str(s1,s2)` với s1, s2 là hai xâu cần gộp.

Hàm này sẽ gộp hai xâu s1, s2 theo cách, lấy lần lượt từng kí tự của s1, s2 đưa vào xâu kết quả. Nếu có một xâu hết kí tự thì đưa phần còn lại của xâu dài hơn vào xâu kết quả. Ví dụ nếu `s1 = "1111"`, `s2 = "0000"` thì xâu kết quả là `"10101010"`.

Hướng dẫn. Gọi S là xâu kết quả sau khi gộp hai xâu s1 và s2, chương trình có thể như sau:

```
def merge_str(s1, s2):
    S = ""
    l1 = len(s1)
    l2 = len(s2)
    l = min(l1, l2)
    for i in range(l):
        S = S + s1[i] + s2[i]
    if l1 < l2:
        for i in range(l,l2):
            S = S + s2[i]
    if l2 < l1:
        for i in range(l,l1):
            S = S + s1[i]
    return S
```



LUYỆN TẬP

1. Thiết lập hàm `power(a,b,c)` với a, b, c là số nguyên. Hàm trả lại giá trị $(a+b)^c$.

2. Viết chương trình thực hiện: Nhập hai số tự nhiên từ bàn phím, hai số cách nhau bởi dấu cách. Tính và in ra tổng của các số này.

Yêu cầu sử dụng hàm khi viết chương trình.



VẬN DỤNG

1. Viết chương trình thực hiện: Nhập hai số tự nhiên từ bàn phím, hai số cách nhau bởi dấu phẩy, in ra ước chung lớn nhất (ƯCLN) của hai số.

Yêu cầu sử dụng hàm khi viết chương trình.

2. Thiết lập hàm `change()` có hai tham số là xâu `ho_ten` và số c. Hàm sẽ trả lại xâu kí tự `ho_ten` là chữ in hoa nếu c = 0. Nếu tham số c khác 0 thì hàm trả lại xâu `ho_ten` là chữ in thường.

Gợi ý: Sử dụng các phương thức s.upper() và s.lower() để chuyển đổi các kí tự của xâu s sang chữ in hoa và in thường.

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

BÀI 28 PHẠM VI CỦA BIẾN

SAU BÀI NÀY EM SẼ:

- Biết và trình bày được ý nghĩa của phạm vi hoạt động của biến trong chương trình và hàm.

1. Một biến được định nghĩa trong chương trình chính (bên ngoài các hàm) thì sẽ được sử dụng như thế nào bên trong các hàm?
2. Một biến được khai báo bên trong một hàm thì có sử dụng được ở bên ngoài hàm đó hay không?

Bài này sẽ giúp em tìm câu trả lời cho các câu hỏi trên.

1. PHẠM VI CỦA BIẾN KHAI BÁO TRONG HÀM

Hoạt động 1 **Phạm vi của biến khi khai báo trong hàm**

Quan sát các lệnh sau để tìm hiểu phạm vi có hiệu lực của biến khi khai báo bên trong một hàm.

Các biến được khai báo bên trong một hàm chỉ được sử dụng bên trong hàm đó. Chương trình chính không sử dụng được.

```
>>> def func(a,b):
        n = 10
        a = a*2
        b = a+b
        return a+b+n
>>> a = 1
>>> b = 2
>>> func(a,b)
16
>>> a,b
(1, 2)
>>> n
Traceback (most recent call last):
  File "<pyshell#11>", line 1, in <module>
    n
NameError: name 'n' is not defined
```

Bên trong hàm này có các biến n, a, b đang hoạt động. n = 10, a và b được thay đổi.

Đây là các biến bên ngoài hàm: a, b. Các biến này được gán: a = 1, b = 2.

Sau khi chạy hàm, các biến a, b vẫn không thay đổi.

Biến n chỉ có tác dụng bên trong hàm func, gọi bên ngoài hàm này sẽ bị báo lỗi.

Trong Python tất cả các biến khai báo bên trong hàm đều có tính địa phương (cục bộ), không có hiệu lực ở bên ngoài hàm.

1. Giả sử có các lệnh sau:

```
>>> a,b = 1,2
>>> def f(a,b):
        a = a + b
        b = b*a
        return a + b
```

Giá trị của a, b bằng bao nhiêu sau khi thực hiện lệnh sau?

a) `f(1, 2)` b) `f(10, 20)`

2. Ta có thể khai báo một biến bên trong hàm trùng tên với biến đã khai báo trước đó bên ngoài hàm không?

2. PHẠM VI CỦA BIẾN KHAI BÁO NGOÀI HÀM

Hoạt động 2 **Phạm vi của biến khi khai báo bên ngoài hàm**

Quan sát các lệnh sau, tìm hiểu phạm vi có hiệu lực của biến khi khai báo bên ngoài một hàm.

Ví dụ 1. Biến khai báo bên ngoài hàm không có tác dụng bên trong hàm.

```
>>> def f(n):
        t = n + 1
        return t

>>> t = 10
>>> f(5)
6
>>> t
10
```

Trong chương trình chính biến t được khai báo bên ngoài hàm f() và gán giá trị 10. Khi gọi f(5), t sẽ được gán 6. Hàm trả lại giá trị 6. Nhưng khi thoát khỏi f(), t vẫn có giá trị 10. Do vậy biến t không có tác dụng bên trong hàm f().

Ví dụ 2. Bên trong hàm có thể truy cập để sử dụng giá trị của biến đã khai báo trước đó ở bên ngoài hàm.

```
>>> def f(a,b):
    return a + b + N

>>> N = 10
>>> f(1,2)
13
```

Trong chương trình chính, biến N được khai báo và gán giá trị 10. Khi gọi hàm f(1,2), giá trị trả lại là biểu thức có N tham gia. Vậy trong hàm f() được phép truy cập giá trị của biến N.

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

Lưu ý: Nếu muốn biến bên ngoài vẫn có tác dụng bên trong hàm thì cần khai báo lại biến này bên trong hàm với từ khoá **global**.

```
>>> def f(n):
        global t
        t = 2*n + 1
        return t
>>> t = 10
>>> f(1)
3
>>> t
3
```

Biến t được khai báo với từ khoá **global**.

Biến t khai báo trong chương trình chính và được gán giá trị 10.
Sau khi thực hiện f(1), giá trị t đã thay đổi.

Biến đã khai báo bên ngoài sẽ không có tác dụng bên trong hàm như một biến. Nếu muốn có tác dụng thì cần khai báo lại biến này trong hàm với từ khoá **global**.

Giả sử hàm f(x,y) được định nghĩa như sau:

```
>>> def f(x,y):
        a = 2*(x + y)
        print(a + n)
```

Kết quả nào được in ra khi thực hiện các lệnh sau?

```
n = 10
f(1,2)
```

THỰC HÀNH

Phạm vi của biến

Nhiệm vụ 1. Viết hàm với đầu vào là danh sách A chứa các số và số thực x. Hàm trả lại một danh sách kết quả B từ danh sách A bằng cách chỉ giữ lại các phần tử lớn hơn hoặc bằng x.

Hướng dẫn. Biến B kiểu danh sách cần được định nghĩa trong hàm và được bổ sung thêm các phần tử từ A nếu thoả mãn điều kiện lớn hơn hoặc bằng x.

```
def Select(A,x):
    B = []
    for k in range(len(A)):
        if A[k] >= x:
            B.append(A[k])
    return B
```

Nhiệm vụ 2. Viết hàm với đầu vào là xâu kí tự Str và số c, đầu ra là danh sách các từ được tách ra từ xâu Str nhưng đã được chuyển thành chữ in hoa hoặc chữ in thường, hoặc chỉ chuyển kí tự đầu các từ thành chữ in hoa tuỳ thuộc vào tham số đầu vào c như sau:

– Nếu c = 0, danh sách B là các từ được chuyển thành chữ in hoa.

– Nếu c = 1, danh sách B là các từ được chuyển thành chữ in thường.

– Nếu c = 2, danh sách B là các từ được chuyển viết chữ hoa kí tự đầu của mỗi từ.

Hướng dẫn. Chúng ta cần sử dụng các lệnh sau:

Str.upper() – chuyển kí tự của xâu thành chữ in hoa.

Str.lower() – chuyển kí tự của xâu thành chữ in thường.

Str.title() – chuyển kí tự đầu mỗi từ của xâu thành chữ in hoa, các kí tự khác chuyển về chữ thường.

Hàm được định nghĩa có dạng **Tach_tu(Str,c)**. Đầu tiên xâu Str cần được tách từ bằng lệnh split(). Sau đó danh sách kết quả sẽ được chuyển đổi chữ in hoa, in thường sử dụng một trong các lệnh trên tuỳ thuộc vào giá trị của đối số c.

```
def Tach_tu(Str, c):
    A = Str.split()
    for k in range(len(A)):
        if c == 0:
            A[k] = A[k].upper()
        if c == 1:
            A[k] = A[k].lower()
        if c == 2:
            A[k] = A[k].title()
    return A
```

Nhiệm vụ 3. Viết chương trình yêu cầu thực hiện lần lượt các việc sau, mỗi việc cần được thực hiện bởi một hàm:

1. Nhập từ bàn phím một dãy các số nguyên, mỗi số cách nhau bởi dấu cách. Chuyển các số này vào danh sách A và in danh sách A ra màn hình.
2. Trích từ danh sách A ra một danh sách B gồm các phần tử lớn hơn 0. In danh sách B ra màn hình.
3. Trích từ danh sách A ra một danh sách C gồm các phần tử nhỏ hơn 0. In danh sách C ra màn hình.

Hướng dẫn. Với mỗi việc trên được viết thành một hàm. Toàn bộ chương trình có thể như sau:

```
def Nhap_Dulieu():
    s = input("Nhập các số nguyên cách nhau bởi dấu cách: ")
    A = s.split()
    for k in range(len(A)):
        A[k] = int(A[k])
    return A
def getB(A):
    B = []
```

```
    for x in A:
        if x > 0:
            B.append(x)
    return B
def getC(A):
    C = []
    for x in A:
        if x < 0:
            C.append(x)
    return C
# Chương trình chính
A = Nhap_Dulieu()
print("Danh sách A:",A)
B = getB(A)
C = getC(A)
print("Danh sách B:",B)
print("Danh sách C:",C)
```



LUYỆN TẬP

1. Viết hàm với đầu vào, đầu ra như sau:

– Đầu vào là danh sách `sList`, các phần tử là xâu kí tự.

– Đầu ra là danh sách `cList`, các phần tử là kí tự đầu tiên của các xâu kí tự tương ứng trong danh sách `sList`.

2. Viết hàm `Tach_day()` với đầu vào là danh sách `A`, đầu ra là hai danh sách B, C được mô tả như sau:

– Danh sách B thu được từ A bằng cách lấy ra các phần tử có chỉ số chẵn.

– Danh sách C thu được từ A bằng cách lấy ra các phần tử có chỉ số lẻ.



VẬN DỤNG

1. Viết hàm có hai tham số đầu vào là m, n. Đầu ra trả lại hai giá trị là:

– ƯCLN của m, n.

– Bội chung nhỏ nhất (BCNN) của m, n.

Gợi ý: Sử dụng công thức ƯCLN(m, n) × BCNN(m, n) = m × n.

2. Viết chương trình nhập ba số tự nhiên từ bàn phím day, month, year, các số cách nhau bởi dấu cách. Các số này biểu diễn giá trị của ngày, tháng, năm nào đó. Chương trình cần kiểm tra và in ra thông báo số liệu đã nhập vào đó có hợp lệ hay không.

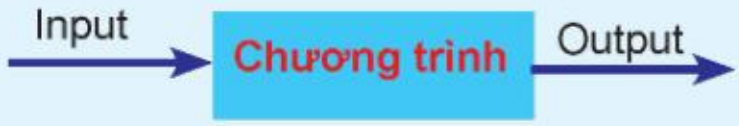
BÀI 29

NHẬN BIẾT LỖI CHƯƠNG TRÌNH

SAU BÀI NÀY EM SẼ:

- Biết và phân loại được một số loại lỗi chương trình.
- Biết được một vài lỗi ngoại lệ thường gặp.

Một chương trình hoàn chỉnh được mô tả như Hình 29.1: Tiếp nhận các dữ liệu đầu vào, xử lí theo yêu cầu bài toán và đưa ra kết quả đúng theo yêu cầu. Theo em nếu chương trình có lỗi, thì các lỗi này sẽ như thế nào và có thể ở đâu?



Hình 29.1

1. NHẬN BIẾT LỖI CHƯƠNG TRÌNH

Hoạt động 1 **Nhận biết và phân biệt một số loại lỗi chương trình**

Quan sát các trường hợp chương trình gặp lỗi như sau, từ đó nhận biết và phân biệt một số loại lỗi của chương trình.

Trường hợp 1. Người lập trình viết sai cú pháp lệnh, chương trình lập tức dừng lại và thông báo lỗi cú pháp.

```
>>> while True print("Hello")
SyntaxError: invalid syntax
```

Trường hợp 2. Người dùng nhập dữ liệu sai, chương trình thông báo lỗi nhập dữ liệu không đúng khuôn dạng.

```
>>> n = int(input("Nhập số nguyên n: "))
Nhập số nguyên n: 1.5
Traceback (most recent call last):
  File "<pyshell#0>", line 1, in <module>
    n = int(input("Nhập số nguyên n: "))
ValueError: invalid literal for int() with base 10: '1.5'
```

Trường hợp 3. Chương trình thông báo lỗi chỉ số vượt quá giới hạn cho phép.

Array_b1.py

```
A = [1,3,10,0]
for i in range(5):
    print(A[i],end = " ")
```

Khi chạy chương trình sẽ báo lỗi:

```
1 3 10 0 Traceback (most recent call last):
  File "C:\Python\Array_b1.py", line 3, in <module>
    print(A[i],end = " ")
IndexError: list index out of range
```

Trường hợp 4. Chương trình thực hiện bình thường nhưng kết quả không đúng với yêu cầu của bài toán. Đây là lỗi lôgic bên trong chương trình.

```
#Tính tổng của ba số nguyên dương đầu tiên
>>> s = 0
>>> for i in range(3)
      s = s + i
>>> print(s)
   3
```

Chương trình cho kết quả là 3 mà kết quả đúng là 1 + 2 + 3 = 6. Lí do là hàm `range(3)` trả lại vùng giá trị là 0, 1, 2 chứ không phải là 1, 2, 3.

Giải thích:

– Với trường hợp 1, chương trình dừng và thông báo lỗi **`Syntax Error`** (lỗi cú pháp).

– Với trường hợp 2, khi người dùng nhập dữ liệu sai, hàm `int()` không thể thực hiện được, chương trình dừng lại và báo lỗi. Mã lỗi là **`ValueError`**. Đây là lỗi **`Runtime`** (lỗi trong khi đang thực hiện) hay còn gọi là lỗi ngoại lệ (Exceptions error).

– Với trường hợp 3, chương trình phát hiện lỗi chỉ số vượt quá giới hạn tại dòng 3. Chương trình dừng và báo lỗi. Mã lỗi là **`IndexError`**. Đây là lỗi **`Runtime`**.

– Với trường hợp 4, chương trình không còn lỗi **`Runtime`**, nhưng kết quả đưa ra sai. Không có mã lỗi nào được trả lại. Đây là lỗi ngữ nghĩa hoặc lỗi lôgic bên trong chương trình.

- Tổng thể có thể phân biệt lỗi chương trình Python làm ba loại:
  1. Lỗi khi có lệnh viết sai cú pháp hoặc sai cấu trúc ngôn ngữ Python quy định. Chương trình sẽ lập tức dừng và thông báo lỗi `Syntax Error`.
  2. Lỗi khi không thể thực hiện một lệnh trong chương trình. Chương trình dừng lại và thông báo một mã lỗi. Lỗi này gọi lỗi ngoại lệ (`Exceptions Error`), mã lỗi trả lại gọi là mã lỗi ngoại lệ.
  3. Chương trình chạy không lỗi ngoại lệ, nhưng kết quả đưa ra sai, không chính xác. Đây là lỗi lôgic bên trong chương trình.
- Với mỗi loại lỗi trên, cách xử lí và kiểm soát lỗi sẽ khác nhau.



1. Khi gõ sai cú pháp một lệnh, chương trình sẽ dừng lại và báo lỗi, đó là lỗi loại gì?
2. Bài toán yêu cầu sắp xếp dãy số ban đầu thành dãy tăng dần. Giả sử dãy số ban đầu là [3, 1, 8, 10, 0]. Kết quả thu được dãy [1, 3, 8, 10, 0]. Chương trình có lỗi không? Nếu có thì lỗi đó thuộc loại gì?

2. MỘT SỐ LỖI NGOẠI LỆ THƯỜNG GẶP

Chúng ta đã biết, nếu gặp lỗi ngoại lệ, chương trình Python sẽ dừng lại và báo lỗi. Một trong những vấn đề được đưa ra khi kiểm soát lỗi là làm thế nào để vẫn phát hiện lỗi, xử lí lỗi nhưng chương trình không bị dừng lại trong khi thực hiện.

Hoạt động 2 **Nhận biết một số lỗi ngoại lệ thường gặp**

Đọc, thảo luận để nhận biết một số lỗi ngoại lệ thường gặp trong chương trình Python.

Mã lỗi ngoại lệ	Mô tả lỗi
`ZeroDivisionError`	Lỗi này xảy ra khi lệnh thực hiện phép chia cho giá trị 0.
`IndexError`	Lỗi xảy ra khi lệnh cố gắng truy cập phần tử của danh sách nhưng chỉ số vượt quá giới hạn.
`NameError`	Lỗi xảy ra khi chương trình muốn tìm một tên nhưng không thấy. Ví dụ khi lệnh gọi một hàm nhưng không có hàm đó.
`TypeError`	Lỗi kiểu dữ liệu. Một số ví dụ lỗi loại này: – Lệnh truy cập một phần tử của danh sách nhưng chỉ số không là số nguyên. – Lệnh tính biểu thức số nhưng lại có một toán hạng không phải là số.
`ValueError`	Lỗi liên quan đến giá trị của đối tượng. Lỗi khi thực hiện lệnh chuyển đổi kiểu dữ liệu, đối số của hàm có giá trị mà hàm không hỗ trợ. Ví dụ khi thực hiện lệnh `int("1.55")` sẽ sinh lỗi loại này.
`IndentationError`	Lỗi khi các dòng lệnh thụt vào không thẳng hàng hoặc không đúng vị trí.
`SyntaxError`	Lỗi cú pháp.

Hãy nêu mã lỗi ngoại lệ của mỗi lệnh sau nếu xảy ra lỗi.

a) `A[1.5]` b) `int("abc")`

c) `"10"*3.5` d) `12 + x(10)`

THỰC HÀNH

Lập trình và kiểm tra khả năng sinh lỗi khi chạy chương trình.

Nhiệm vụ 1. Viết chương trình nhập các số nguyên m, n từ bàn phím, cách nhau bởi dấu cách. Chương trình đưa ra tổng, hiệu, thương của hai số đã nhập.

Hướng dẫn. Chương trình chính là khối các lệnh nhập từ bàn phím hai số nguyên m, n. Các số này được nhập bằng lệnh `input()`, kết quả là một xâu kí tự. Xâu này sẽ được tách thành danh sách các xâu con bằng lệnh `split()`. Kết quả thu được sẽ

chuyển đổi thành hai số m, n bằng lệnh `int()`. Nhập chương trình sau và kiểm tra khả năng sinh lỗi khi chạy chương trình.

```
s = input("Nhập hai số m, n cách nhau bởi dấu cách: " )
sline = s.split()
m, n = int(sline[0]),int(sline[1])
print("Tổng, hiệu, thương 2 số đã nhập là:",m+n, m-n,m/n)
```

Gợi ý. Các khả năng sinh lỗi của chương trình:

- Các số m, n khi nhập vào không là số nguyên.
- Giữa hai số m, n không có dấu cách.
- Số n nhập vào là số 0.

Nhiệm vụ 2. Viết chương trình nhập số tự nhiên n và nhập lần lượt n số nguyên đưa vào danh sách số A. Sau khi nhập xong in danh sách A ra màn hình.

Hướng dẫn. Nhập chương trình sau và kiểm tra khả năng sinh lỗi khi chạy.

```
n = int(input("Nhập số tự nhiên n: "))
A = []
for k in range(n):
    num = int(input("Nhập số thứ "+str(k+1)+" : "))
    A.append(num)
print("Dãy đã nhập:",A)
```

Gợi ý. Các khả năng sinh lỗi của chương trình:

- Số n được nhập không là số nguyên.
- Mỗi số hạng của danh sách nhập vào không là số nguyên.

LUYỆN TẬP

1. Các lệnh sau có sinh lỗi chương trình không? Nếu có thì mã lỗi là gì?

```
>>> A = [1,3,5,10,0]
>>> for k in
range(1,len(A)+1):
   print(A[k])
```

```
>>> s1, s2 = "101010",1010101
>>> s = s1 + s2
```

2. Để tính giá trị trung bình của một danh sách số A, người lập trình đã dùng lệnh sau để tính:

```
gttb = sum(A)/len(A)
```

lệnh này có thể sinh lỗi ngoại lệ không? Nếu có thì là những lỗi gì?

VẬN DỤNG

1. Giả sử em được yêu cầu viết một chương trình nhập số tự nhiên n từ bàn phím, kết quả đưa ra là danh sách các ước số thực sự của n, tính cả 1 và không tính n. Hãy viết chương trình và kiểm tra các khả năng sinh lỗi khi thực hiện chương trình.

2. Em hãy viết một chương trình nhỏ để khi chạy sẽ sinh mã lỗi NameError.

BÀI 30 KIỂM THỬ VÀ GỠ LỖI CHƯƠNG TRÌNH

SAU BÀI NÀY EM SẼ:

- Biết được một vài phương pháp đơn giản kiểm thử chương trình.
- Biết được một vài cách gỡ lỗi đơn giản một chương trình.



Bài học trước em đã biết khái niệm lỗi ngoại lệ khi chạy chương trình Python. Tuy nhiên, một chương trình chạy không có lỗi ngoại lệ (chương trình không bị dừng) thì không có nghĩa là chương trình không có lỗi. Thậm chí các "lỗi" không tường minh này (các lỗi này được gọi bug) càng khó phát hiện và khó sửa.

Theo em, làm thế nào để kiểm tra (test) và gỡ lỗi (debug) một chương trình? Môi trường lập trình có công cụ nào hỗ trợ việc đó không?

1. MỘT VÀI PHƯƠNG PHÁP KIỂM THỬ CHƯƠNG TRÌNH

Hoạt động Tìm hiểu một số phương pháp kiểm thử chương trình

Đọc và thảo luận nhóm các phương pháp, công cụ sau để biết chức năng, tác dụng của từng công cụ trong công việc kiểm thử chương trình.



Có rất nhiều phương pháp và công cụ khác nhau để kiểm thử chương trình. Các công cụ này không những có mục đích *tìm ra lỗi* (hay bug) của chương trình mà còn có tác dụng *phòng ngừa* và *ngăn chặn* các lỗi phát sinh tiếp trong tương lai.

a) Quan sát mã lỗi Runtime và bắt lỗi ngoại lệ

Nếu chương trình có lỗi Runtime (tức là đang chạy bị dừng lại), cần quan sát các mã lỗi (mã lỗi ngoại lệ) để kiểm tra vị trí dòng lệnh sinh ra lỗi này. Từ đó phân tích, tìm và sửa lỗi.

b) Kiểm thử chương trình với các bộ dữ liệu test

Chương trình cần được thử với một số bộ dữ liệu test gồm đầu vào tiêu biểu phụ thuộc đặc thù của bài toán và kết quả đầu ra đã biết trước. Các bộ test có thể có đầu vào theo các tiêu chí khác nhau như độ lớn và tính đa dạng của dữ liệu. Cần chú ý một số điểm sau:

– *Cần có nhiều bộ test* (theo các tiêu chí khác nhau như độ lớn, tính đa dạng của dữ liệu,...).

– *Cần có bộ test ngẫu nhiên*. Việc sinh ngẫu nhiên dữ liệu đầu vào trong miền xác định của chương trình làm tăng khả năng tìm lỗi nếu có.

– *Cần có bộ test dữ liệu ở vùng biên*. Ví dụ dữ liệu đầu vào là cặp (x, y) xác định trên miền $0 \le x, y \le 1$. Khi đó cần kiểm tra chương trình với bộ dữ liệu biên là (0; 0), (0; 1), (1; 0) và (1; 1). Thực tế cho thấy thường phát sinh lỗi tại các vùng biên hoặc lân cận của biên. Một ví dụ khác của dữ liệu biên là cần tìm các bộ test với n và các giá trị $(x_1, x_2, ..., x_n)$ rất lớn (vùng cận biên lớn).

c) In các thông số trung gian

Bổ sung vào giữa các dòng lệnh các lệnh `print()` để in ra các biến trung gian, qua đó kiểm tra các quy trình hay thuật toán được viết có đúng không.

Giả sử chương trình có đầu vào là $(x_1, x_2, ..., x_n)$, đầu ra là $(a_1, a_2, ..., a_m)$ nhưng có sử dụng các biến trung gian $(y_1, y_2, ..., y_k)$. Khi đó với mỗi bộ test đầu vào, chúng ta sẽ bổ sung vào các dòng lệnh của chương trình để in ra các giá trị trung gian:

$$(x_1, x_2,..., x_n), (y_1, y_2, .., y_k), (a_1, a_2,..., a_m).$$

Thông qua các giá trị trung gian trong quá trình thực hiện chương trình, nếu kết quả cuối cùng có lỗi thì sẽ dễ tìm ra lỗi đó.

d) Sử dụng công cụ break point (điểm dừng)

Công cụ break point cho phép tạo ra các "điểm dừng" bên trong chương trình. Khi chạy, chương trình sẽ tạm dừng lại tại các "điểm dừng" cho phép người kiểm thử có thể quan sát các thông tin khác bên trong chương trình, qua đó kiểm tra tính đúng đắn của chương trình.

Trên thực tế sử dụng phương pháp điểm dừng thường kết hợp với phương pháp in các giá trị trung gian sẽ là hiệu quả hơn để kiểm thử chương trình.

Một số ghi nhớ:

- Sử dụng công cụ in các biến trung gian.
- Sử dụng công cụ sinh các bộ dữ liệu test.
- Sử dụng công cụ điểm dừng trong phần mềm soạn thảo lập trình.
- Quan sát các mã lỗi của chương trình nếu phát sinh.

2. VÍ DỤ MINH HOẠ



Xét ví dụ sau: Nhập từ bàn phím hai số tự nhiên m, n, tính ƯCLN của hai số này.

Gọi `gcd(m, n)` là ƯCLN của hai số tự nhiên m, n. Thuật toán của bài toán này dựa trên thuật toán sau:

(1) gcd(m, m) = m.

(2) Nếu n > m thì gcd(m, n) = gcd(m, n–m).

(3) Nếu n < m thì gcd(m, n) = gcd(m–n, n).

Phần cơ bản nhất của chương trình sẽ là một vòng lặp `while`, vòng lặp sẽ kết thúc khi m = n. Chương trình như sau:

```
1  # Tính ƯCLN của m, n
2  m = int(input("Nhập số tự nhiên m: "))
3  n = int(input("Nhập số tự nhiên n: "))
4  while m != n:
5      if m < n:
6          n = n - m
7      else:
8          m = m - n
9  print("Đáp số:",m)
```

Chúng ta sẽ tiến hành kiểm thử chương trình này. Cần tập trung kiểm tra kĩ khối lệnh của lệnh lặp **while**.

Cách 1: In ra các giá trị trung gian để kiểm soát chương trình.

Bổ sung biến k và hai lệnh `print()` vào chương trình như mô tả như sau:

```
# Tính ƯCLN của m, n
m = int(input("Nhập số tự nhiên m: "))
n = int(input("Nhập số tự nhiên n: "))
k = 0
while m!= n:
    k = k + 1
    print("Vòng lặp",k,":", m, n)
    if m < n:
        n = n - m
    else:
        m = m - n
print("Kết thúc vòng lặp:", m, n)
print("Đáp số:",m)
```

Bổ sung thêm biến k và hai lệnh `print()` để in các giá trị trung gian k, m, n.

Kết quả thực hiện chương trình trên như sau:

```
Nhập số tự nhiên m: 20
Nhập số tự nhiên n: 16
Vòng lặp 1 : 20 16
Vòng lặp 2 : 4 16
Vòng lặp 3 : 4 12
Vòng lặp 4 : 4 8
Kết thúc vòng lặp: 4 4
Đáp số: 4
```

Quan sát sự thay đổi của giá trị các biến k, m, n trong quá trình thực hiện chương trình để phát hiện lỗi (nếu có), đồng thời hiểu được lỗi này và tìm cách sửa lỗi.

Cách 2: Sử dụng công cụ tạo điểm dừng của phần mềm soạn thảo lập trình.

Thiết lập điểm dừng tại dòng 4 của chương trình như hình sau. Đây là vị trí bắt đầu chuẩn bị vào vòng lặp.

```
1  # Tính ƯCLN của m, n
2  m = int(input("Nhập số tự nhiên m: "))
3  n = int(input("Nhập số tự nhiên n: "))
4  while m != n:
5      if m < n:
6          n = n - m
7      else:
8          m = m - n
9  print("Đáp số:",m)
```

Thiết lập điểm dừng tại dòng 4 của chương trình, đây là vị trí bắt đầu một vòng lặp mới của lệnh **while**.

Khi chạy chương trình sẽ dừng lại trước mỗi vòng lặp, chúng ta sẽ ghi lại các giá trị m, n vào một bảng như bảng sau. Khi kết thúc hết vòng lặp thì kết quả chương trình chính là giá trị m.

Vòng lặp	m	n	Kết quả
1	20	16	
2	4	16	
3	4	12	
4	4	8	
Kết thúc lặp	4	4	4

Cả hai cách để kiểm soát lỗi là in các giá trị trung gian và thiết lập điểm dừng đều hiệu quả.



LUYỆN TẬP

1. Chương trình của em khi chạy phát sinh lỗi ngoại lệ ZeroDivisionError. Đó là lỗi gì và em sẽ xử lí lỗi này như thế nào?

2. Chương trình sau có lỗi không? Nếu có thì tìm và sửa lỗi.

```
m = input("Nhập số tự nhiên m: ")
n = input("Nhập số tự nhiên n: ")
print("Tổng hai số đã nhập là:",m+n)
```



VẬN DỤNG

1. Chương trình sau có chức năng sắp xếp một dãy số cho trước. Hãy kiểm tra xem chương trình có lỗi không. Nếu có thì tìm và sửa lỗi.

```
A = [10,1,5,2,8,0,4]
for i in range(len(A)-1):
    j = i
    while j > 1 and A[j] < A[j-1]:
        A[j],A[j-1] = A[j-1],A[j]
        j = j - 1
print(A)
```

2. Để kiểm thử một chương trình, nếu chỉ bằng việc kiểm tra thông qua các bộ dữ liệu test thì có bảo đảm tìm ra hết lỗi của chương trình hay không? Vì sao?

BÀI 31 THỰC HÀNH VIẾT CHƯƠNG TRÌNH ĐƠN GIẢN

SAU BÀI NÀY EM SẼ:

- Thực hành viết chương trình đơn giản bằng ngôn ngữ Python.
- Thực hành được các bước gỡ rối chương trình bằng công cụ debug – thiết lập điểm dừng và chạy theo từng lệnh.

Nhiệm vụ 1. Viết chương trình nhập từ bàn phím số tự nhiên n, kiểm tra n có phải là số nguyên tố hay không. Nếu n là hợp số thì in ra kết quả phân tích n thành tích các thừa số nguyên tố. Chú ý số 1 không là nguyên tố và cũng không là hợp số.

Hướng dẫn. Sử dụng biến danh sách `NT` để lưu các thừa số nguyên tố của n. Chương trình sẽ thiết lập danh sách NT chỉ khi n > 1. Kết quả của chương trình sẽ như sau:

– Nếu n = 1 thì danh sách NT sẽ rỗng.

– Nếu n > 1 thì danh sách NT không rỗng. Độ dài danh sách `len(NT)` sẽ bằng 1 khi và chỉ khi n là số nguyên tố.

Nếu len(NT) > 1 thì chương trình sẽ in ra khai triển n thành tích các thừa số nguyên tố, khai triển này sẽ có dạng: $n = p_1 \times p_1 \times ... \times p_k$.

phantichnt.py

```
# Nhập số tự nhiên từ bàn phím và kiểm tra n có phải là số nguyên tố hay không.
# Nếu n = 1 thì thông báo n không phải là số nguyên tố.
# Nếu n là hợp số thì in ra phân tích n thành tích các thừa số nguyên tố.
n = int(input("Nhập số tự nhiên n: "))
m = n
k = 2
NT = []
while m > 1:
    while m%k != 0:
        k = k + 1
    NT.append(k)
    m = m//k
count = len(NT)
if count == 0:
    print(n,"không là số nguyên tố")
elif count == 1:
    print(n,"là số nguyên tố")
else:
    print(n,"là hợp số")
    print(n,"=",end = " ")
    for i in range(count):
        if i < count - 1:
            print(NT[i],"x",end = " ")
        else:
            print(NT[i])
```

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

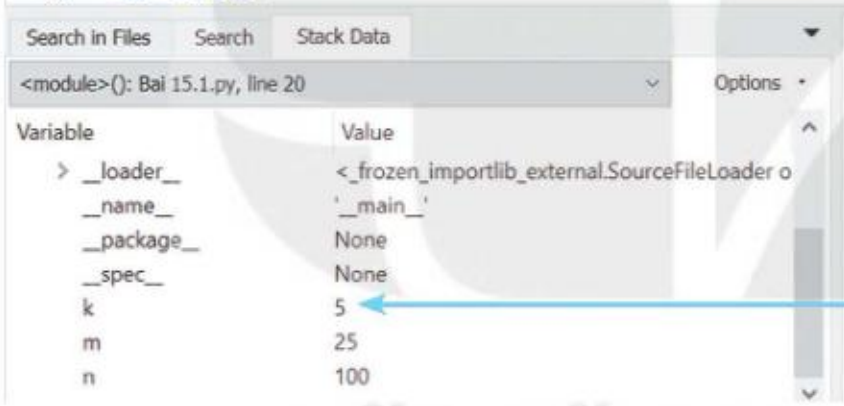
Chạy chương trình với công cụ gỡ lỗi của phần mềm lập trình. Thiết lập một điểm dừng tại dòng 20 của chương trình như sau:

```
13  m = n
14  k = 2
15  NT = []
16  while m > 1:
17      while m%k != 0:
18          k = k + 1
19      NT.append(k)
20      m = m//k
21  count = len(NT)
```

Điểm dừng của chương trình được đặt trước lệnh m = m//k, sau khi k là ước số nguyên tố tiếp theo được phát hiện và đưa vào danh sách NT. Quá trình gỡ lỗi được tiến hành để kiểm tra sự thay đổi các biến n, m, k có đúng theo thuật toán hay không.

```
14  k = 2
15  NT = []
16  while m > 1:
17      while m%k != 0:
18          k = k + 1
19      NT.append(k)
20      m = m//k
21  count = len(NT)
22  if count == 0:
23      print(n,"không là số nguyên tố")
24  elif count == 1:
25      print(n,"là số nguyên tố")
```

Khi chạy, chương trình sẽ chạy và dừng lại trước điểm dừng (trên màn hình dòng dừng lại được đánh dấu). Nháy nút ▶ để chạy tiếp chương trình.



Mỗi lần chương trình dừng lại có thể quan sát các biến n, m, k để kiểm tra tính đúng đắn của chương trình.

Thiết lập bảng theo dõi các giá trị trung gian k, m, n, NT sẽ như sau, giả sử giá trị nhập ban đầu của n = 100:

k	m	n	NT	Kết thúc
2	100	100	[2]	
2	50	100	[2, 2]	
5	25	100	[2, 2, 5]	
5	5	100	[2, 2, 5, 5]	
				Thông báo: 100 = 2 × 2 × 5 × 5

Nhiệm vụ 2. Viết chương trình nhập từ bàn phím ba số thực a, b, c và tìm nghiệm của phương trình bậc hai: $ax^2 + bx + c = 0$.

Chương trình cần xét đầy đủ các trường hợp xảy ra.

Hướng dẫn.

Với bộ dữ liệu a, b, c đã nhập (là các số thực), chúng ta cần xét đầy đủ các trường hợp sau:

– Nếu a = b = c = 0 phương trình có vô số nghiệm.

– Nếu a = b = 0; c ≠ 0, phương trình vô nghiệm.

– Nếu a = 0; b ≠ 0 phương trình là bậc nhất và có nghiệm duy nhất.

– Nếu a ≠ 0, giải phương trình bậc hai. Nghiệm sẽ phụ thuộc vào giá trị delta = $b^2 - 4ac$. Phương trình vô nghiệm, có một nghiệm kép hoặc hai nghiệm phân biệt phụ thuộc vào giá trị delta là nhỏ hơn 0, bằng 0 hay lớn hơn 0.

Chương trình được thiết kế thông qua các hàm sau:

– NhapDL(): hàm nhập ba số a, b, c từ bàn phím.

– GiaiPT1(b, c): hàm giải phương trình bậc nhất: bx + c = 0.

– GiaiPT2(a, b, c): hàm giải phương trình bậc hai: $ax^2 + bx + c = 0$.

Trong bài thực hành chúng ta sử dụng cấu trúc mở rộng của lệnh rẽ nhánh **if ... else** trong Python khi các lệnh này lồng nhau. Khi đó các lệnh rẽ nhánh lồng nhau trong mô hình bên trái sẽ được viết gọn hơn như mô hình bên phải.

```
if <điều kiện 1>:
    <nhóm lệnh 1>
else:
    if <điều kiện 2>:
        <nhóm lệnh 2>
    else:
        <nhóm lệnh 3>
```

```
if <điều kiện 1>:
    <nhóm lệnh 1>
elif <điều kiện 2>:
    <nhóm lệnh 2>
else:
    <nhóm lệnh 3>
```

Chú ý: Cấu trúc **if ... elif .... else** có thể lồng nhau nhiều lần.

Chương trình đầy đủ như sau:

giaipt.py

```
# Nhập từ bàn phím ba số thực a, b, c và tìm nghiệm của phương trình
# ax2 + bx + c = 0.
def sqrt(x):
    return x**0.5

def NhapDL():
    s = input("Nhập ba số a, b, c cách nhau bởi dấu cách: ")
    snum = s.split()
    return float(snum[0]), float(snum[1]), float(snum[2])

def GiaiPT1(b,c):
    if b != 0:
        print("Phương trình có một nghiệm duy nhất:",round(-c/b,1))
    elif c == 0:
        print("Phương trình có vô số nghiệm")
    else:
        print("Phương trình vô nghiệm")
```

```
def GiaiPT2(a,b,c):
    if a == 0:
        GiaiPT1(b,c)
    else:
        delta = b*b - 4*a*c
        if delta > 0:
            x1 = (-b + sqrt(delta))/(2*a)
            x2 = (-b - sqrt(delta))/(2*a)
            print("Phương trình có hai nghiệm khác biệt")
            print("x1 =",round(x1,1),"x2 =",round(x2,1))
        elif delta == 0:
            x = (-b / (2*a))
            print("Phương trình có nghiệm kép")
            print("x1,2 = ",round(x,1))
        else:
            print("Phương trình vô nghiệm")
# Chương trình chính
a,b,c = NhapDL()
GiaiPT2(a,b,c)
```



LUYỆN TẬP

1. Viết chương trình yêu cầu nhập số thực dương a. Chương trình cần kiểm tra dữ liệu nhập như sau: Nếu số đã nhập nhỏ hơn hoặc bằng 0 thì thông báo: "Nhập sai, số a phải lớn hơn 0. Hãy nhập lại". Chương trình chỉ dừng sau khi người dùng nhập đúng.

2. Viết chương trình in bảng cửu chương ra màn hình như sau:

- Hàng thứ nhất in ra bảng nhân 1, 2, 3, 4, 5.
- Hàng thứ hai in ra bảng nhân 6, 7, 8, 9, 10.



VẬN DỤNG

1. Viết chương trình nhập hai số tự nhiên Y1, Y2 là số năm, Y2 > Y1. Tính xem trong khoảng thời gian từ năm Y1 đến năm Y2 có bao nhiêu năm nhuận. Áp dụng tính xem trong thế kỉ XXI có bao nhiêu năm nhuận.

2. Gọi ƯCLN(a, b) là hàm ƯCLN của hai số tự nhiên a, b. Dễ thấy ta có ƯCLN(a, b) = ƯCLN(b, a%b) nếu b > 0 và ƯCLN(a, 0) = a. Từ đó hãy viết chương trình nhập hai số a, b và tính ƯCLN của a và b.

BÀI 32 ÔN TẬP LẬP TRÌNH PYTHON

SAU BÀI NÀY EM SẼ:

- Thực hành ôn tập lập trình Python.
- Thực hành lập trình giải bài toán có tính liên môn.

Nhiệm vụ 1. Viết chương trình nhập họ tên đầy đủ từ bàn phím, ví dụ "Nguyễn Thị Mai Hương", sau đó tách riêng phần tên, họ, đệm và in ra màn hình.

Hướng dẫn. Trong Bài 25 chúng ta sẽ biết cách tách phần họ đệm và tên từ một xâu họ và tên. Bài này yêu cầu thêm tách phần đệm, tức là phần nằm giữa tên và họ ra bằng cách sử dụng lệnh `join()`. Xâu kí tự ban đầu được tách thành một danh sách dùng hàm `split()`. Sau khi lấy phần họ và tên, phần đệm sẽ lấy ra theo lệnh sau:

```
dem = " ".join(slist[1:n-1])
```

Trong đó `slist` là danh sách được tách ra từ xâu ban đầu, n là độ dài của xâu `slist`.

Nhập, chạy thử và kiểm tra chương trình sau:

```
hoten = input("Nhập họ tên đầy đủ của em: ")
slist = hoten.split()
n = len(slist)
ho = slist[0]
ten = slist[n-1]
dem = " ".join(slist[1:n-1])
print("Tên của em là",ten)
print("Họ của em là",ho)
if n > 2:
    print("Đệm của em là",dem)
```

Nhiệm vụ 2. Trọng lượng của em trên các hành tinh khác.

Chương trình yêu cầu nhập trọng lượng của em (tính theo đơn vị N – Newton) trên Trái Đất và tính trọng lượng của em trên một hành tinh khác (ví dụ Mặt Trăng, Hoả tinh, Kim tinh, Thổ tinh, Mộc tinh, Mặt Trời).

Hướng dẫn. Trọng lượng đo lực hút của Trái Đất (hay hành tinh) lên vật thể. Trọng lượng có đơn vị đo N (Newton). Khối lượng vật thể tính bằng kg và giá trị này không thay đổi. Chúng ta có công thức:

$$P = m \times g \qquad (1)$$

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

Trong đó P là trọng lượng tính bằng N, m là khối lượng tính bằng kg, g là gia tốc trọng trường của Trái Đất (hay hành tinh), tính theo m/s^2. Trên Trái Đất, g = 9.8 m/s^2. Trên mỗi hành tinh các giá trị g sẽ khác nhau. Danh sách các hành tinh được lưu trong biến `planet`, các trọng lực tương ứng lưu trong danh sách `gravities`.

Biết trọng lượng của một người trên Trái Đất (ví dụ P_0) thì sẽ dễ dàng tính được trọng lượng của người này trên một hành tinh khác nếu biết giá trị g của hành tinh đó. Gọi P là trọng lượng cần tìm, khi đó ta có công thức sau, suy trực tiếp từ công thức (1).

$m = P_0/9.8 = P/g$, vậy suy ra $P = P_0 \times g/9.8$. (2)

Em hãy nhập chương trình sau và kiểm tra tính đúng đắn của chương trình.

weight.py

```
def string(w):
    s = ""
    for i in range(len(w)):
        s = s+str(i+1)+". "+w[i] + " "
    return s
planet = ["Mặt Trăng","Hoả tinh","Kim tinh","Mộc tinh","Thổ tinh", "Mặt Trời"]
gravities = [1.62, 3.711, 8.83, 24.79, 10.44, 274.0]
P_earth = float(input("Nhập trọng lượng của em, tính theo N: "))
k = int(input("Nhập số thứ tự hành tinh\n"+string(planet)+": "))
Grp = gravities[k-1]
P = round(P_earth*Grp/9.8,3)
print("Trọng lượng của em trên",planet[k-1],"là:",P,"N")
```

Nhiệm vụ 3. Kiểm tra tính hợp lệ của ba tham số ngày, tháng, năm.

Chương trình sẽ yêu cầu nhập ba số tự nhiên: ngày, tháng, năm từ bàn phím theo khuôn dạng, ví dụ nhập 08-02-2021. Chương trình sẽ thông báo bộ dữ liệu đã nhập là hợp lệ hay không hợp lệ.

Hướng dẫn. Bộ dữ liệu chính cần nhập sẽ đặt tên là **day**, **month**, **year**. Nhiệm vụ của bài toán là nhập bộ dữ liệu này và kiểm tra tính hợp lệ theo các yêu cầu về lịch của ngày, tháng, năm.

Điểm đặc biệt nhất cần chú ý là kiểm tra năm **year** có phải là nhuận không, nếu là nhuận thì tháng 2 phải có 29 ngày so với các năm không nhuận tháng 2 có 28 ngày. Chúng ta sử dụng biến danh sách số **thang** để lưu số ngày của các tháng trong năm. Sau mỗi lần nhập ba số **day**, **month**, **year** cần kiểm tra năm nhuận để cập nhật tháng 2. Khi đó, chương trình kiểm tra có thể viết đơn giản như sau:

date.py

```
thang = [31,28,31,30,31,30,31,31,30,31,30,31]
def nhuan(year):
    return year%400 == 0 or (year%4 == 0 and year%100 != 0)
date = input("Nhập thời gian theo dạng ngày - tháng - năm: ")
tg = date.split("-")
day,month,year = int(tg[0]),int(tg[1]),int(tg[2])
if nhuan(year):
    thang[1] = 29
else:
    thang[1] = 28
if year > 0 and 1 <= month <= 12 and 1<= day <= thang[month-1]:
    print(day,"-",month,"-",year, "là hợp lệ")
else:
    print("Bộ dữ liệu đã nhập không hợp lệ")
```

LUYỆN TẬP

Viết chương trình nhập số n, sau đó nhập danh sách tên học sinh với họ, đệm, tên. Sắp xếp tên học sinh trong lớp theo bảng chữ cái. Đưa kết quả ra màn hình.

VẬN DỤNG

1. Trong các phần mềm bảng tính điện tử, dữ liệu ngày tháng được coi là số ngày tính từ ngày 1-1-1990. Viết chương trình:

– Nhập số tự nhiên n từ bàn phím và tính xem số đó ứng với ngày, tháng, năm nào.

– Nhập thời gian theo khuôn dạng ngày - tháng - năm (ví dụ 8-10-2021), tính số ngày ứng với ngày này theo phần mềm bảng tính điện tử.

2. Mở rộng bài tập trong phần luyện tập như sau:

– Việc sắp xếp thứ tự phải ưu tiên tính theo tên trước, rồi đến họ, rồi đến đệm.

– Sắp xếp theo thứ tự của bảng chữ cái tiếng Việt.

Chú ý: Bảng chữ cái tiếng Việt (bao gồm cả dấu thanh) được sắp xếp theo thứ tự sau: AÀÁẢÃẠĂẰẮẲẴẶÂẦẤẨẪẬBCDĐEÈÉẺẼẸÊỀẾỂỄỆGHIÌÍỈĨỊKLMNOÒÓỎÕỌÔỒỐỔỖỘƠỜỚỞỠỢPQRSTUÙÚỦŨỤƯỪỨỬỮỰVXYỲÝỶỸỴ.

3. Nếu n là hợp số thì dễ thấy n phải có ước số nguyên tố nhỏ hơn hoặc bằng $\sqrt{n}$. Viết chương trình tối ưu hoá hơn nhiệm vụ 1, bài 31, theo cách sau: để tìm ước số nguyên tố nhỏ nhất chỉ cần tìm trong các số 2, 3, ..., $\sqrt{n}$. Nếu trong dãy trên không tìm thấy ước của n thì kết luận ngay n là nguyên tố.

Chủ đề 6

BÀI 33

Hướng nghiệp với Tin học

NGHỀ THIẾT KẾ ĐỒ HOẠ MÁY TÍNH

SAU BÀI NÀY EM SẼ:

- Biết được khái niệm, kiến thức và kĩ năng cần có của nghề thiết kế đồ hoạ.
- Biết các ngành học và nhu cầu nhân lực liên quan đến nghề thiết kế đồ hoạ.
- Tự tìm kiếm và khai thác được thông tin hướng nghiệp về lĩnh vực thiết kế đồ hoạ, giao lưu và chia sẻ với bạn bè qua các kênh truyền thông tin số về thông tin nghề nghiệp.

Trong các công việc sau, theo em công việc nào có liên quan trực tiếp đến nghề thiết kế đồ hoạ?



Thợ may



Phát thanh viên



Kiến trúc sư



Thư kí

Hình 33.1

1. KHÁI NIỆM VỀ THIẾT KẾ ĐỒ HOẠ

Hoạt động 1 Thiết kế đồ hoạ là gì?

Em hiểu thế nào là thiết kế đồ hoạ? Em đã bao giờ vẽ tranh hay làm phim chưa? Em tạo ra các sản phẩm đó bằng cách nào?

Thiết kế đồ hoạ là việc dàn dựng bố cục, sắp xếp, chỉnh sửa hình ảnh, lựa chọn màu sắc để sáng tạo các thông điệp truyền thông hấp dẫn và thu hút, đáp ứng yêu cầu truyền đạt thông tin một cách hiệu quả nhằm phục vụ mục đích tuyên truyền hoặc kinh doanh. Tuỳ theo phương thức thể hiện, thông điệp truyền thông có thể là các ấn phẩm (tấm thiếp, tờ rơi, logo, biển hiệu, áp phích, tài liệu quảng cáo/giới thiệu sản phẩm, bìa sách/tạp chí,...), các trang web,...

Các hình ảnh đồ hoạ thường bao gồm nhiều thành phần như văn bản, các đối tượng hình ảnh như các đường, các hình cơ bản hay hình vẽ, ảnh chụp, màu sắc,... Nhiệm vụ của người thiết kế đồ hoạ là lựa chọn, vẽ, cắt, ghép, sắp xếp các thành phần trên để tạo thành sản phẩm hoàn chỉnh.

Nghề thiết kế đồ hoạ đã có từ thời xa xưa, ban đầu nhằm mục đích truyền tin hoặc ghi nhớ. Cùng với sự phát triển của xã hội loài người, thiết kế đồ hoạ đã trở thành một lĩnh vực nghệ thuật ứng dụng nhằm mục đích trang trí, làm đẹp không thể thiếu trong đời sống tinh thần của con người. Ngày nay, nhờ máy tính và các phần mềm đồ hoạ, thiết kế đồ hoạ đã trở thành một nghề phổ biến, dễ tiếp cận hơn với nhiều người.

Thiết kế đồ hoạ đem lại nhiều lợi ích cho mọi ngành nghề, lĩnh vực khác nhau:

– Giúp tạo dựng hình ảnh chuyên nghiệp của tổ chức hoặc cá nhân đối với mọi người thông qua các sản phẩm như logo, áp phích, danh thiếp, thẻ nhân viên, hình ảnh trên mạng xã hội,...

– Mang lại trải nghiệm đặc biệt cho độc giả, người xem thông qua các hình ảnh truyền thông thu hút và hấp dẫn.

– Tăng hiệu quả tiếp thị và tăng doanh thu nhờ các tờ rơi, quảng cáo,... với các hình ảnh sản phẩm bắt mắt, ấn tượng.



Hình 33.2. *Áp phích quảng bá du lịch*

- Thiết kế đồ hoạ là sáng tạo các thông điệp truyền thông kết hợp giữa hình ảnh, kiểu chữ, màu sắc để truyền tải thông tin đến người xem.
- Thiết kế đồ hoạ đem lại nhiều lợi ích cho mọi ngành nghề, lĩnh vực.

Hãy chọn một công việc được nêu trong Hình 33.1 liên quan trực tiếp tới thiết kế đồ hoạ và cho biết thiết kế đồ hoạ có thể hỗ trợ những gì cho công việc đó.

2. KIẾN THỨC, KĨ NĂNG CẦN CÓ CỦA NGƯỜI THIẾT KẾ ĐỒ HOẠ

Hoạt động 2 **Các kĩ năng của người làm thiết kế đồ hoạ**

Theo em để làm nghề thiết kế đồ hoạ cần có những kĩ năng nào? Em có thấy bản thân mình phù hợp với nghề này không?

Bất cứ ngành nghề nào cũng cần phải có kiến thức, kĩ năng nhất định. Đối với ngành thiết kế đồ hoạ, ngoài kĩ năng vẽ, sắp xếp các đối tượng đồ hoạ thì còn đòi hỏi những yêu cầu sau:

– Có kiến thức về công nghệ nói chung và thành thạo kĩ năng máy tính và các thiết bị thông minh nói riêng, đặc biệt là kiến thức và kĩ năng làm việc trên các phần mềm đồ hoạ máy tính như Adobe Photoshop, CorelDraw, GIMP, InDesign, Scribus, AutoCad, Corel Designer, Solid Works,... Ngoài ra, có kiến thức về công nghệ in ấn cũng là điểm cộng đối với những người làm thiết kế đồ hoạ.

– Người làm đồ hoạ máy tính cần luôn học hỏi những điều mới, cần có kiến thức rộng về các lĩnh vực như toán học, vật lí, nghệ thuật, xã hội,... để có thể ứng dụng trong công việc của mình. Đồng thời, họ cần phải có kĩ năng nhận biết được những xu hướng, nắm bắt được nhu cầu của xã hội, tìm kiếm thông tin, tra cứu tài liệu để học hỏi và theo kịp với xu thế của đời sống, xã hội.

Bên cạnh đó, người làm thiết kế đồ hoạ không thể thiếu được khả năng sáng tạo, sự yêu thích cái đẹp, kĩ năng đánh giá, phản biện, phân tích, cũng như tư duy với những con số và khả năng ngoại ngữ.

Người làm nghề thiết kế đồ hoạ cần có:

- Khả năng sáng tạo, yêu thích và cảm nhận được cái đẹp.
- Kiến thức về công nghệ nói chung, công nghệ in ấn, công nghệ thông tin và truyền thông nói riêng.
- Kiến thức rộng về các lĩnh vực như toán học, vật lí, nghệ thuật, xã hội,...
- Kĩ năng vẽ, sắp xếp các đối tượng đồ hoạ.
- Kĩ năng sử dụng máy tính và thiết bị thông minh, sử dụng thành thạo phần mềm thiết kế đồ hoạ.
- Kĩ năng học hỏi những điều mới, công nghệ mới, kĩ năng tìm kiếm thông tin.
- Kĩ năng đánh giá, phản biện, phân tích cũng như tư duy với những con số.

Theo em, những kĩ năng, tố chất nào là cần thiết nhất cho người thiết kế đồ hoạ:

A. Có hiểu biết sâu về toán học.

B. Có khả năng sử dụng thành thạo phần mềm đồ hoạ máy tính và có kiến thức về công nghệ.

C. Biết chơi nhiều nhạc cụ khác nhau.

D. Có khả năng cảm nhận cái đẹp và khả năng sáng tạo.

3. HỌC TẬP VÀ VIỆC LÀM TRONG NGÀNH THIẾT KẾ ĐỒ HOẠ

Hoạt động 3 **Học tập và việc làm trong ngành thiết kế đồ hoạ**

1. Theo em, để theo học thiết kế đồ hoạ ở bậc đại học, cao đẳng, cần chuẩn bị tốt những môn học gì?
2. Em biết trường đại học nào có đào tạo chuyên ngành thiết kế đồ hoạ?
3. Sau khi tốt nghiệp chuyên ngành thiết kế đồ hoạ, người học có thể làm những công việc gì?

Để bắt đầu với lĩnh vực thiết kế đồ hoạ, các em có thể theo học tại các trung tâm, trường dạy nghề. Các em cũng có thể theo học bậc đại học, cao đẳng tại các trường về mĩ thuật, kiến trúc, thiết kế hoặc nhiều trường đào tạo ngành Công nghệ thông tin cũng đào tạo chuyên ngành Thiết kế đồ hoạ trên máy tính.

Các em có thể tìm kiếm thông tin về hướng nghiệp, việc làm trên Internet thông qua các công cụ tìm kiếm phổ biến như Google Search, Bing,... với các từ khoá về nghề như thiết kế đồ hoạ, thiết kế mĩ thuật, thiết kế 3D, thiết kế giao diện, nhận diện thương hiệu, thiết kế quảng cáo,... Các em cũng có thể truy cập vào các diễn đàn, dịch vụ tìm kiếm việc làm như LinkedIn, Vietnamworks,... để tìm kiếm cũng như trao đổi thông tin.

Hiện nay, nhu cầu lao động làm việc trong ngành Thiết kế đồ hoạ rất cao. Vì thế, theo học ngành này, em có nhiều cơ hội phát triển nghề nghiệp trong tương lai. Những cơ hội nghề nghiệp dành cho sinh viên tốt nghiệp ngành này có thể kể đến: chuyên viên thiết kế, tư vấn thiết kế tại các công ty quảng cáo, công ty thiết kế, công ty truyền thông và tổ chức sự kiện, studio nghệ thuật, xưởng phim hoạt hình và truyện tranh, các toà soạn, các nhà xuất bản, cơ quan truyền hình, báo chí,... Ngoài ra, sau khi tốt nghiệp, sinh viên có thể tự thành lập doanh nghiệp, các công ty thiết kế, dịch vụ studio hoặc tư vấn, giảng dạy tại các trường học, trung tâm, câu lạc bộ,... Hơn nữa, như một đặc thù ưu ái, ngành Thiết kế đồ hoạ luôn mang lại những cơ hội làm thêm hấp dẫn tại nhà như thiết kế website, thiết kế logo để nhận diện thương hiệu,... Khi đã có những kinh nghiệm cần thiết, em hoàn toàn có thể tự mở công ty riêng cho mình, nhận dự án của các công ty, tổ chức,...

- Theo học lĩnh vực thiết kế đồ hoạ tại các trung tâm, trường dạy nghề, các trường đại học, cao đẳng có chuyên ngành thiết kế đồ hoạ, thiết kế đồ hoạ trên máy tính.
- Có thể tìm kiếm thông tin về hướng nghiệp trên Internet hay qua các diễn đàn nghề nghiệp.
- Nhu cầu nhân sự cao với nhiều công việc và cách thức làm việc đa dạng.

LUYỆN TẬP

1. Thiết kế đồ hoạ là thao tác:

A. Tạo ra các thành phần đồ hoạ.
B. Lựa chọn các thành phần đồ hoạ.
C. Sắp xếp các thành phần đồ hoạ.
D. Tất cả các thao tác trên.

2. Sau khi tốt nghiệp các trường đào tạo về thiết kế đồ hoạ, em có thể làm việc ở những đơn vị nào?

VẬN DỤNG

1. Hãy tìm các kênh thông tin giới thiệu việc làm liên quan đến thiết kế đồ hoạ và chia sẻ với bạn bè về kênh thông tin đó.

2. Sử dụng công cụ tìm kiếm trên Internet để biết các phần mềm công cụ đồ hoạ như Illustrator, Photoshop, InDesign, AutoCad,... được dùng để làm gì?

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

BÀI 34 NGHỀ PHÁT TRIỂN PHẦN MỀM

SAU BÀI NÀY EM SẼ:

- Hiểu được khái niệm nghề phát triển phần mềm và một số kiến thức, kĩ năng cần có của người làm nghề phát triển phần mềm.
- Biết các ngành học ở bậc đại học, cao đẳng liên quan đến phát triển phần mềm và cơ hội nghề nghiệp liên quan đến phát triển phần mềm.

Theo em, phát triển phần mềm có phải chỉ là việc viết các đoạn mã lệnh bằng một ngôn ngữ lập trình nào đó để máy tính có thể hiểu và giải quyết một bài toán trong thực tế?

1. PHÁT TRIỂN PHẦN MỀM LÀ GÌ?

Hoạt động 1 **Hiểu về phát triển phần mềm**

Nhiều em mong muốn biết lập trình để làm ra các phần mềm ứng dụng. Vậy em có biết việc sản xuất phần mềm gồm các công đoạn nào không?

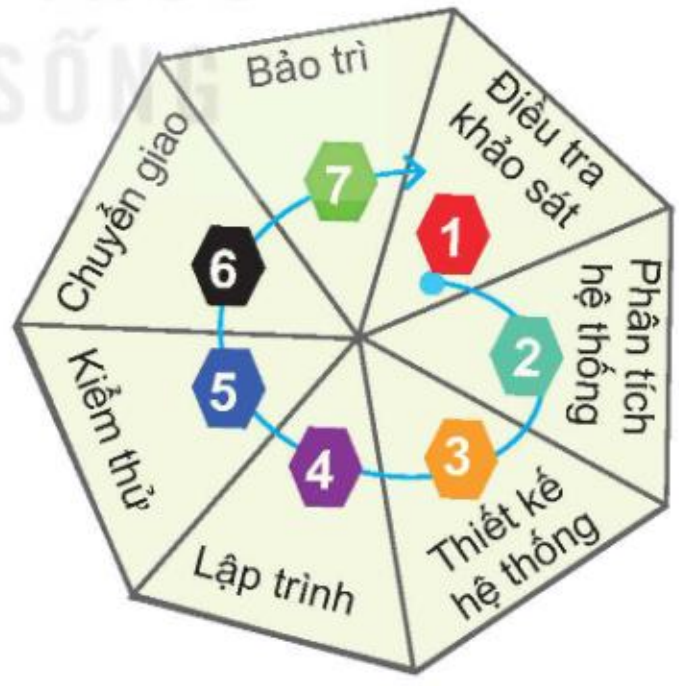
Nếu như kết quả của một bài tập lập trình thông thường chỉ dài vài chục câu lệnh và thường do một người làm ra, thì phần mềm ứng dụng có thể lên tới hàng vạn, hàng triệu dòng lệnh và nói chung do nhiều người thực hiện. Các công việc cơ bản, cũng chính là các công đoạn cần thực hiện để sản xuất một phần mềm gồm có:

Điều tra khảo sát: Tiếp xúc với khách hàng, tìm hiểu về yêu cầu nghiệp vụ, xây dựng hồ sơ yêu cầu của hệ thống.

Phân tích hệ thống: Dựa trên các tài liệu điều tra khảo sát, chuyên viên phân tích sẽ tạo ra tài liệu mô tả đầy đủ yêu cầu của phần mềm.

Thiết kế hệ thống: Dựa vào tài liệu phân tích, chuyên viên thiết kế sẽ đưa ra thiết kế tổng thể, thiết kế dữ liệu và thiết kế chức năng và có thể cả giao diện chi tiết.

Lập trình: Dựa vào tài liệu thiết kế, các lập trình viên sẽ tiến hành tạo cơ sở dữ liệu nếu cần và viết các đoạn mã thực hiện các chức năng.



Kiểm thử: Phát hiện để loại bỏ các lỗi cũng như các bất hợp lí trong sử dụng chương trình nếu có; kiểm tra kết quả thực hiện theo chức năng đã thiết kế,...

Chuyển giao: Cài đặt, khởi tạo dữ liệu, hướng dẫn sử dụng và chuyển giao.

Bảo trì: Nói chung, các phần mềm khi mới đưa vào vận hành có thể còn những sai sót hoặc không đáp ứng đầy đủ yêu cầu của người dùng. Bảo trì là công việc rất quan trọng nhằm khắc phục triệt để các lỗi, nâng cấp cả về tính năng và giao diện của phần mềm. Công việc này có thể là một vòng phát triển mới, liên quan tới tất cả các công việc sản xuất phần mềm nêu trên.

Hoạt động có tính bao trùm lên toàn bộ các công việc cơ bản của sản xuất phần mềm là quản trị dự án phần mềm, bao gồm lập kế hoạch, điều phối nhân sự, tài chính, phương tiện, kiểm soát chất lượng, để đảm bảo thành công của dự án.

Tất cả các công việc và hoạt động nêu trên được gọi chung là phát triển phần mềm mà lập trình chỉ là một hoạt động trong đó. Những người tham gia vào các công việc và hoạt động đó đều được gọi là người phát triển phần mềm (Software Developer).

> Phát triển phần mềm gồm các công việc: điều tra, khảo sát, phân tích và thiết kế hệ thống; lập trình, kiểm thử, chuyển giao; bảo trì và quản trị dự án.

Theo em điều nào là đúng nhất trong các điều sau khi nói về phát triển phần mềm?

A. Phát triển phần mềm là lập trình.

B. Phát triển phần mềm là quá trình gồm nhiều công việc và hoạt động.

C. Phát triển phần mềm là quá trình gồm nhiều công việc và hoạt động, có thể lặp đi lặp lại.

D. Phát triển phần mềm là quản trị dự án phần mềm.

2. KIẾN THỨC, KĨ NĂNG CỦA NGƯỜI PHÁT TRIỂN PHẦN MỀM

Hoạt động 2 Tìm hiểu về các vị trí trong phát triển phần mềm

Theo em, phát biểu "tất cả những người phát triển phần mềm đều có vai trò như nhau" là đúng hay sai?

Có ba hoạt động chính trong phát triển phần mềm là:

- Lập trình.
- Các hoạt động khác đảm bảo cho lập trình như phân tích, thiết kế, kiểm thử, bảo trì,...
- Quản trị dự án phát triển phần mềm.

Khởi đầu, lập trình viên chỉ cần có các hiểu biết cơ bản về một ngôn ngữ lập trình phù hợp để có thể bắt đầu phụ trách những đoạn mã ngắn, đơn giản theo thiết kế sau khi trải qua một khoá đào tạo cơ bản tại các trường nghề, các trung tâm đào tạo hay các công ty phần mềm. Với kiến thức và kĩ năng có được, ngoài việc lập trình, họ có thể tham gia một số công đoạn khác như kiểm thử, chuyển giao hay bảo trì phần mềm.

Ở cấp độ cao hơn, lập trình viên được trang bị thêm các kiến thức về thuật toán, cấu trúc dữ liệu, trí tuệ nhân tạo, mật mã,... để có thể viết các chương trình phức tạp đòi hỏi hiểu biết chuyên sâu về toán học và khoa học máy tính.

Hiểu biết thuật toán, cấu trúc dữ liệu, có những kiến thức về khoa học máy tính, có đức tính cẩn thận, tỉ mỉ, làm chủ được môi trường lập trình là các tố chất cần thiết đối với lập trình viên.

Khái niệm kĩ sư phần mềm thường để chỉ những người tổ chức làm phần mềm. Họ có thể phụ trách các khâu quan trọng như phân tích, thiết kế hay trực tiếp tham gia hoặc chủ trì quản trị dự án phần mềm.

Thông thường, những người thiết kế đồng thời là những người phân tích hệ thống. Những hiểu biết về toán học, lôgic, phân tích, thiết kế là các yếu tố giúp họ thiết lập kiến trúc của phần mềm, xác định các thành phần chức năng (phần mềm làm gì), tổ chức các dữ liệu và có thể định ra các thuật toán phù hợp,... đồng thời giúp họ xây dựng tài liệu phân tích, thiết kế hệ thống. Người phân tích, thiết kế phần mềm không chỉ biết xây dựng các giải pháp tin học mà còn phải có khả năng học hỏi, nắm bắt công việc trong lĩnh vực ứng dụng. Chẳng hạn để viết một phần mềm kế toán thì các chuyên viên phân tích phải hiểu nghề như một kế toán giỏi,...

Sự khác biệt giữa các kĩ sư phần mềm và lập trình viên tương tự như giữa kiến trúc sư và thợ xây trong xây dựng công trình. Kĩ sư phần mềm không nhất thiết phải lập trình nhưng hiểu biết về lập trình rất quan trọng giúp họ có giải pháp thiết kế tốt. Trong thực tế, chuyên viên phân tích và thiết kế nói chung đều trải qua hoạt động lập trình.

Quản trị dự án xuyên suốt quá trình hình thành, triển khai và kết thúc dự án phần mềm.

Người quản trị dự án cần có tầm nhìn, hiểu biết về quy trình làm phần mềm, hiểu biết xu hướng công nghệ, có khả năng tổ chức, lập kế hoạch, điều phối các nguồn lực, tổ chức giám sát. Nếu đã từng trải qua công việc lập trình hay phân tích và thiết kế hệ thống thì đây sẽ là một lợi thế cho người quản trị dự án.

Đối với các dự án phần mềm lớn, hoạt động quản trị dự án đóng vai trò cốt yếu cho sự thành công của dự án phần mềm.

- Lập trình viên; kĩ sư phần mềm; người quản trị dự án là những người đảm nhận những công việc quan trọng nhất trong phát triển phần mềm.
- Có những kiến thức nhất định về toán học, cấu trúc dữ liệu và giải thuật nói riêng và về khoa học máy tính nói chung ở các mức độ khác nhau cùng khả năng vận dụng thuần thục các kiến thức ấy vào thực tế là những yêu cầu cần có đối với lập trình viên và kĩ sư phần mềm - người đảm nhận những vị trí quan trọng trong tổ chức phát triển phần mềm.
- Quản trị dự án là công việc xuyên suốt quá trình sản xuất phần mềm, có vai trò cốt yếu cho sự thành công của dự án phần mềm. Có tầm nhìn, hiểu biết về quy trình làm phần mềm, hiểu biết xu hướng công nghệ, có khả năng tổ chức, lập kế hoạch, điều phối các nguồn lực, tổ chức giám sát,... là những yêu cầu không thể thiếu đối với người quản trị dự án phát triển phần mềm.

Hãy ghép mỗi hoạt động phát triển phần mềm ở cột trái với tố chất ở cột phải cho thích hợp.

Hoạt động	Tố chất
1) Lập trình (viết code). 2) Phân tích và thiết kế hệ thống. 3) Quản trị dự án.	a) Cẩn thận, tỉ mỉ. b) Có khả năng học hỏi, tìm hiểu nghiệp vụ của lĩnh vực ứng dụng. c) Có kiến thức toán, thuật toán tốt. d) Làm chủ được ngôn ngữ lập trình. e) Hiểu biết công nghệ. g) Có tầm nhìn, kĩ năng tổ chức và điều phối nguồn lực.

3. CÔNG VIỆC PHÁT TRIỂN PHẦN MỀM

Hoạt động 3 **Công việc của người phát triển phần mềm**

Em có biết làm thế nào để trở thành người tham gia phát triển phần mềm? Theo em có những cơ hội nghề nghiệp nào cho người phát triển phần mềm?

Để trở thành người phát triển phần mềm, các em có thể bắt đầu với các khoá đào tạo về lập trình, phát triển phần mềm tại các trung tâm, các trường dạy nghề, hoặc các công ty, tập đoàn, dần dần tích luỹ kinh nghiệm thông qua các công việc thực tế. Nếu muốn tham gia phát triển phần mềm ở vị trí kĩ sư phần mềm, em cần theo học ở bậc đại học về tin học hay công nghệ thông tin.

Sau khi tốt nghiệp các khoá, ngành đào tạo, em có thể tham gia các công việc phát triển phần mềm ở nhiều lĩnh vực như:

- Lập trình ứng dụng: Viết chương trình với tác vụ cụ thể.
- Phát triển giao diện người dùng: Xây dựng các giao diện thân thiện với người dùng.
- Phát triển ứng dụng trên web, các phần mềm hệ thống hoặc quản trị các hệ thống thông tin, các kho dữ liệu,...
- Lập trình trí tuệ nhân tạo/máy học: Các chương trình có thể bắt chước các hành động của con người, có khả năng học và cải thiện kết quả hành động.
- Phát triển games: Xây dựng các phần mềm trò chơi trên máy tính.
- Phát triển ứng dụng di động: Viết các ứng dụng cho điện thoại di động, máy tính bảng và các thiết bị di động khác.

Ở Việt Nam, phát triển phần mềm luôn là công việc được đánh giá cao trong lĩnh vực kĩ thuật, công nghệ cũng như các lĩnh vực liên quan tới STEM. Nhu cầu nhân lực phát triển phần mềm chiếm tỉ lệ cao nhất trong nhu cầu tuyển dụng ngành Công nghệ thông tin. Nhu cầu đó thường xuyên thay đổi và phát triển cùng với sự phát triển ứng dụng khoa học và công nghệ trong mọi lĩnh vực khác nhau của đời sống xã hội. Đặc biệt, trong bối cảnh cách mạng công nghiệp lần thứ tư, với sự phát triển mạnh mẽ các ứng dụng của AI, IoT, điện toán đám mây, chuỗi khối (BlockChain),... những cơ hội nghề nghiệp lớn lao đang không ngừng mở ra cho những người phát triển phần mềm.

Người muốn tham gia phát triển phần mềm có thể tìm kiếm cơ hội tại các hội chợ việc làm do các tỉnh, thành phố, các công ty, tập đoàn công nghệ như FPT, Viettel, VNPT,... hay các trường đại học tổ chức. Các em cũng có thể tìm kiếm cơ hội việc

làm thông qua các trang thông tin tuyển dụng trực tuyến của các doanh nghiệp, hay các trang chuyên về tuyển dụng như TopDev, Vietnamworks, LinkedIn.

- Có thể theo học phát triển phần mềm tại nhiều nơi khác nhau: các trung tâm, trường nghề, các công ty, các nhà trường,...
- Các cơ hội nghề nghiệp cho người phát triển phần mềm rất đa dạng. Nhu cầu nhân lực phát triển phần mềm không ngừng tăng cao cùng với sự phát triển ứng dụng khoa học và công nghệ.

1. Em đánh giá thế nào về cơ hội việc làm trong tương lai đối với nghề phát triển phần mềm?
2. Theo em, người tốt nghiệp các trường đại học về công nghệ thông tin có thể làm những công việc gì? Cho những đơn vị như thế nào?

LUYỆN TẬP

1. Mô tả quy trình phát triển phần mềm.
2. Theo em, để theo học ngành phát triển phần mềm, em cần chuẩn bị tốt những môn học nào?
3. Hãy liệt kê một vài phần mềm ứng dụng mà em biết.

VẬN DỤNG

1. Ở khu vực nơi em sinh sống hay các tỉnh/thành phố lân cận, trường đại học nào đào tạo nghề phát triển phần mềm? Khối thi ngành liên quan đến phát triển phần mềm của trường đó là gì?
2. Ở tỉnh/thành phố nơi em cư trú có Trung tâm dạy nghề phát triển phần mềm nào không? Liệt kê một vài khoá học tiêu biểu mà họ cung cấp. Chia sẻ thông tin em tìm hiểu được với các bạn.
3. Ở tỉnh/thành phố nơi em cư trú có doanh nghiệp nào chuyên về phát triển phần mềm không? Họ có cung cấp các chương trình đào tạo cho người muốn trở thành người phát triển phần mềm của công ty hay không?

PHỤ LỤC

Bảng mã ASCII mở rộng

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	NULL	SOH	STX	ETX	EOT	ENQ	ACK	BELL	BS	HT	LF	VT	FF	CR	SO	SI
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0001	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0010		!	"	#	$	%	&	'	(	)	*	+	,	–	.	/
	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
0011	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
0100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
0101	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
0110	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
0111	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL
	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
1000	Ç	ü	é	â	ä	à	ă	ç	ê	ë	è	ï	î	ì	Ä	Å
	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
1001	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	ø	£	Ø	×	ƒ
	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
1010	á	ù	ó	ú	ñ	Ñ	ª	º	¿	®	¬	½	¼	¡	«	»
	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
1011	▒	▒	▓	|	┤	Á	Â	À	©	╣	‖	╗	╝	¢	¥	┐
	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
1100	└	┴	┬	├	─	┼	ã	Ã	╚	╔	╩	╦	╠	=	╬	¤
	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
1101	ð	Đ	Ê	Ë	È	ı	Í	Î	Ï	┘	┌	▉	▄	¦	Ì	▀
	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
1110	Ó	ß	Ô	Ò	õ	Õ	µ	þ	Þ	Ú	Û	Ù	ý	Ý	¯	´
	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
1111	–	±	‗	¾	¶	§	÷	¸	°	¨	·	¹	³	²	■	
	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

BẢNG GIẢI THÍCH THUẬT NGỮ

	Thuật ngữ	Giải thích	Trang
B	Biến (Variable)	Một tên (định danh) được tạo ra khi lập trình để tham chiếu dữ liệu trong bộ nhớ.	91
D	Dịch vụ đám mây (Cloud Service)	Dịch vụ phân phối/chia sẻ các tài nguyên tin học (phần cứng, phần mềm,...) theo nhu cầu qua Internet với chính sách thanh toán theo hạn mức sử dụng.	41
Đ	Điểm chỉ hướng (Node handles)	Điểm có thể tác động để điều khiển hình dạng của đoạn cong kề với điểm neo ở phía tương ứng.	76
	Điểm neo (Nodes)	Điểm đầu mút hoặc điểm nối giữa các đoạn của đối tượng dạng đường thường được dùng để biến đổi hình. Điểm neo trơn có hai đường chỉ hướng của hai đoạn có phương trùng nhau. Điểm neo góc không bắt buộc phương của hai đường chỉ hướng (nếu có) phải cùng phương.	76
	Điều chế mã xung (Pulse Code Modulation - PCM)	Mã hoá tín hiệu âm thanh bằng cách lấy mẫu biên độ tín hiệu theo những khoảng thời gian đều đặn. Biên độ tức thời của tín hiệu tại mỗi lần lấy mẫu được làm tròn gần nhất với các mức đã định trước. PCM là dạng tiêu chuẩn của âm thanh kĩ thuật số.	28
	Độ sâu màu (bit depth)	Số bit cần thiết để mang thông tin màu của một điểm ảnh trong biểu diễn ảnh.	30
	Đối số (Argument) của hàm	Giá trị thực tế truyền vào cho tham số khi gọi hàm.	131
	Đối tượng dạng đường (Paths)	Một chuỗi các đoạn thẳng hay đoạn cong nối với nhau.	75
	Đường chỉ hướng (Direction lines)	Đường xác định hướng của đường tiếp tuyến và cung trên đoạn cong kề với điểm neo ở phía tương ứng.	76
H	Hàm (Function)	Một nhóm các lệnh có liên quan được đóng gói thành một đơn vị độc lập để thực hiện một tác vụ/nhiệm vụ cụ thể nào đó.	127
K	Kết nối vạn vật (Internet of things)	Liên kết các thiết bị thông minh để tự động thu thập, truyền và xử lí dữ liệu phục vụ cho mục đích ứng dụng.	12
	Kiểm thử (Test)	Công việc kiểm tra và chạy thử chương trình với mục đích tìm ra lỗi của chương trình.	145
	Kiểu dữ liệu (Data type)	Phân loại dữ liệu sử dụng trong chương trình có chung cách biểu diễn và các phép xử lí, ví dụ: số nguyên (int), số thực (float), xâu kí tự (str), lôgic (bool), danh sách (list).	88
M	Màu tô (Fill color)	Màu tô phía trong đối tượng được chọn	65
	Màu vẽ (Stroke color)	Màu vẽ đường viền của đối tượng.	65

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

	Thuật ngữ	Giải thích	Trang
L	Lớp (Layer)	Thuộc tính cho phép người dùng tổ chức các đối tượng theo thứ tự hiển thị được mong muốn trên vùng làm việc.	66
	Lỗi ngoại lệ (Exception error)	Lỗi gây ra do điều kiện xử lí không thích hợp như chia cho 0, mất tín hiệu mạng, mở một tệp không tồn tại,... Việc bắt các lỗi ngoại lệ giúp có thể kiểm soát được hoạt động xử lí, ví dụ thông báo lỗi và dừng xử lí.	141
	Lỗi trong khi đang chạy (Runtime error)	Lỗi xảy ra khi chương trình đang chạy.	142
	Lỗi chương trình (Bug) và gỡ lỗi (Debug)	Lỗi trong chương trình dẫn đến chương trình không chạy được hoặc chạy không đúng. Debug là tìm để khắc phục các bug.	141
P	Phần mềm độc hại (Malware)	Phần mềm được viết ra với ý đồ xấu, gây hại cho người dùng.	45
	Phần mềm nội gián (Trojan)	Phần mềm độc hại, thực hiện các hành vi nội gián như ăn cắp thông tin hay chiếm đoạt quyền sử dụng máy tính.	46
	Phương thức (Method)	Hàm đặc biệt được thiết lập riêng cho từng kiểu dữ liệu.	113
S	Sâu máy tính (Worm)	Phần mềm độc hại, hoàn chỉnh, tồn tại độc lập, có khả năng nhân bản và lây lan nhờ khai thác lỗ hổng bảo mật của hệ điều hành hay sự bất cẩn của người dùng.	46
T	Tham số (Parameters)	Biến được mô tả trong khai báo của hàm.	131
	Thiết bị số	Thiết bị dùng với thông tin số - thông tin được mã hoá trong hệ nhị phân.	6
	Thiết bị thông minh	Thiết bị có khả năng kết nối với thiết bị khác qua mạng, có khả năng hoạt động tự động, thích ứng với bối cảnh không cần sự can thiệp của con người; thiết bị tính toán thông dụng; thiết bị có các yếu tố trí tuệ nhân tạo.	11
	Tốc độ bit (Bit rate)	Số bit cần thiết lưu âm thanh để phát âm thanh trong một giây.	29
	Từ khoá (Keyword)	Từ dành riêng quy định ý nghĩa câu lệnh của chương trình, không được dùng để đặt tên cho các đối tượng của chương trình.	91
V	Virus	Một đoạn mã độc hại, lây lan bằng cách gắn vào một chương trình, khi thực hiện chương trình, đoạn mã sẽ thực hiện chức năng gây hại và giải phóng mã độc để gắn vào các chương trình khác.	46

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

Nhà xuất bản Giáo dục Việt Nam xin trân trọng cảm ơn các tác giả có tác phẩm, tư liệu được sử dụng, trích dẫn trong cuốn sách này.

Chịu trách nhiệm xuất bản:
Chủ tịch Hội đồng thành viên – Tổng Giám đốc NGUYỄN TIẾN THANH

Chịu trách nhiệm nội dung:
Tổng biên tập PHẠM VĨNH THÁI

Biên tập lần đầu: NGUYỄN THỊ NGUYÊN THÚY– PHẠM THỊ THANH NAM
Biên tập tái bản: PHẠM THỊ THANH NAM
Biên tập mĩ thuật: NGUYỄN BÍCH LA
Thiết kế sách: PHAN THỊ THU HƯƠNG
Trình bày bìa: NGUYỄN BÍCH LA
Minh hoạ: NGUYỄN HỒNG QUÂN
Sửa bản in: PHẠM THỊ TÌNH - NGUYỄN NGỌC TÚ
Chế bản: CÔNG TY CỔ PHẦN MĨ THUẬT VÀ TRUYỀN THÔNG

Bản quyền © (2022) thuộc Nhà xuất bản Giáo dục Việt Nam.

Xuất bản phẩm đã đăng kí quyền tác giả. Tất cả các phần của nội dung cuốn sách này đều không được sao chép, lưu trữ, chuyển thể dưới bất kì hình thức nào khi chưa có sự cho phép bằng văn bản của Nhà xuất bản Giáo dục Việt Nam.

TIN HỌC 10
Mã số: G1HHXI001h26
In ..... bản, (QĐ .....) khổ 19 x 26,5 cm.
Đơn vị in:.....
Cơ sở in:.....
Số ĐKXB: 7-2026/CXBIPH/103-09/GD
Số QĐXB: .../..... ngày ... tháng ... năm .....
In xong và nộp lưu chiểu quý ..... năm .....
Mã số ISBN: 978-604-0-31089-7



HUÂN CHƯƠNG HỒ CHÍ MINH

BỘ SÁCH GIÁO KHOA LỚP 10 – KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

1. Ngữ văn 10, tập một
2. Ngữ văn 10, tập hai
3. Chuyên đề học tập Ngữ văn 10
4. Toán 10, tập một
5. Toán 10, tập hai
6. Chuyên đề học tập Toán 10
7. Lịch sử 10
8. Chuyên đề học tập Lịch sử 10
9. Địa lí 10
10. Chuyên đề học tập Địa lí 10
11. Giáo dục Kinh tế và Pháp luật 10
12. Chuyên đề học tập Giáo dục Kinh tế và Pháp luật 10
13. Vật lí 10
14. Chuyên đề học tập Vật lí 10
15. Hoá học 10
16. Chuyên đề học tập Hoá học 10
17. Sinh học 10
18. Chuyên đề học tập Sinh học 10
19. Công nghệ 10 – Thiết kế và Công nghệ
20. Chuyên đề học tập Công nghệ 10 – Thiết kế và Công nghệ
21. Công nghệ 10 – Công nghệ trồng trọt
22. Chuyên đề học tập Công nghệ 10 – Công nghệ trồng trọt
23. Tin học 10
24. Chuyên đề học tập Tin học 10 – Định hướng Tin học ứng dụng
25. Chuyên đề học tập Tin học 10 – Định hướng Khoa học máy tính
26. Mĩ thuật 10 – Thiết kế mĩ thuật đa phương tiện
27. Mĩ thuật 10 – Thiết kế đồ hoạ
28. Mĩ thuật 10 – Thiết kế thời trang
29. Mĩ thuật 10 – Thiết kế mĩ thuật sân khấu, điện ảnh
30. Mĩ thuật 10 – Lí luận và lịch sử mĩ thuật
31. Mĩ thuật 10 – Điêu khắc
32. Mĩ thuật 10 – Kiến trúc
33. Mĩ thuật 10 – Hội hoạ
34. Mĩ thuật 10 – Đồ hoạ (tranh in)
35. Mĩ thuật 10 – Thiết kế công nghiệp
36. Chuyên đề học tập Mĩ thuật 10
37. Âm nhạc 10
38. Chuyên đề học tập Âm nhạc 10
39. Hoạt động trải nghiệm, hướng nghiệp 10
40. Giáo dục thể chất 10 – Bóng chuyền
41. Giáo dục thể chất 10 – Bóng đá
42. Giáo dục thể chất 10 – Cầu lông
43. Giáo dục thể chất 10 – Bóng rổ
44. Giáo dục quốc phòng và an ninh 10
45. Tiếng Anh 10 – Global Success – Sách học sinh

Website: https://www.nxbgd.vn

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

ISBN 978-604-0-31089-7

9 786040 310897

Giá: 20.500 đ