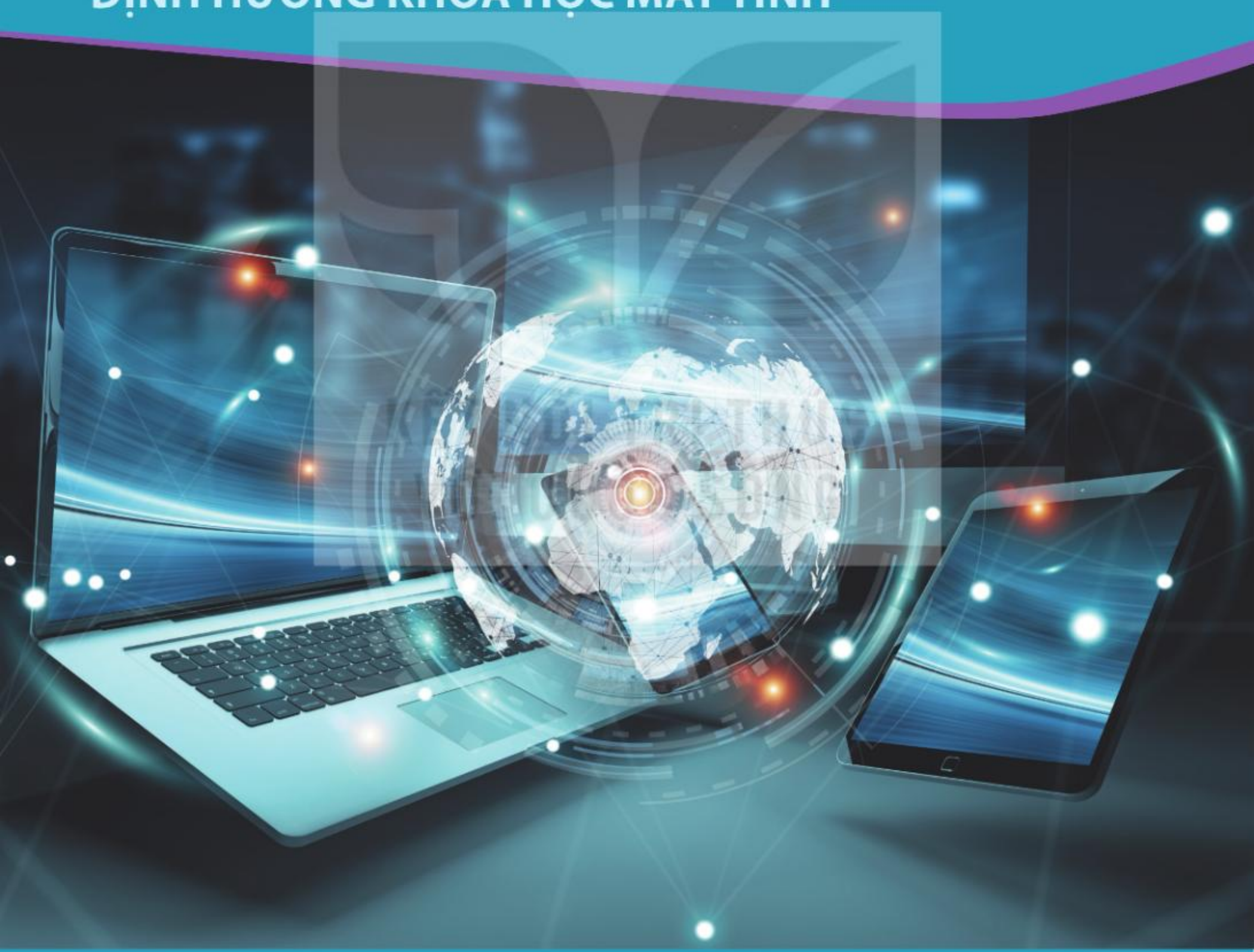
BỘ GIÁO DỤC VÀ ĐÀO TẠO

# TIN HỌC 11

## ĐỊNH HƯỚNG KHOA HỌC MÁY TÍNH



NHÀ XUẤT BẢN GIÁO DỤC VIỆT NAM

# HỘI ĐỒNG QUỐC GIA
# THẨM ĐỊNH SÁCH GIÁO KHOA

## Môn: Tin học – Lớp 11

*(Theo Quyết định số 2026/QĐ-BGDĐT ngày 21 tháng 7 năm 2022 của Bộ trưởng Bộ Giáo dục và Đào tạo)*

| Họ và tên | Chức vụ Hội đồng |
|---|---|
| Ông LÊ HOÀI BẮC | Chủ tịch |
| Ông TRẦN ĐĂNG HƯNG | Phó Chủ tịch |
| Ông HỒ VĨNH THẮNG | Uỷ viên, Thư kí |
| Bà LƯU THỊ BÍCH HƯƠNG | Uỷ viên |
| Bà NGUYỄN THỊ VÂN KHÁNH | Uỷ viên |
| Bà PHAN THỊ MAY | Uỷ viên |
| Ông NGUYỄN ĐỨC NHUẬN | Uỷ viên |
| Ông HOÀNG VĂN QUYẾN | Uỷ viên |
| Bà NGUYỄN THỊ HỒNG THÁI | Uỷ viên |
| Ông ĐỖ NĂNG TOÀN | Uỷ viên |
| Ông NGUYỄN TRUNG TRỰC | Uỷ viên |

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

PHẠM THẾ LONG (Tổng Chủ biên)
BÙI VIỆT HÀ – ĐÀO KIẾN QUỐC (đồng Chủ biên)
NGUYỄN HOÀNG HÀ – NGUYỄN NAM HẢI – NGUYỄN THỊ HIỀN
DƯƠNG QUỲNH NGA – TRƯƠNG VÕ HỮU THIÊN – LÊ HỮU TÔN
PHẠM THỊ BÍCH VÂN – ĐẶNG BÍCH VIỆT

# TIN HỌC

11

## ĐỊNH HƯỚNG KHOA HỌC MÁY TÍNH

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

*(Tái bản lần thứ hai)*

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.



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



**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.

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

# Lời nói đầu

*Các em học sinh thân mến!*

Như các em đã biết, bắt đầu từ lớp 10, môn Tin học có sự phân hoá sâu theo hai định hướng nghề nghiệp: *Tin học ứng dụng* (ICT) và *Khoa học máy tính* (CS). Ngoài kiến thức cốt lõi chung, nội dung môn học được tổ chức thành các nhóm chủ đề kiến thức ICT và CS khác nhau. Sách Tin học 11 được kết cấu tách riêng theo hai định hướng chuyên sâu ICT và CS để tạo thuận lợi cho việc lựa chọn và sử dụng.

Trên tay các em là cuốn sách **Tin học 11 – Định hướng Khoa học máy tính** thuộc bộ sách *Kết nối tri thức với cuộc sống* của Nhà xuất bản Giáo dục Việt Nam. Sách được biên soạn dựa trên các yêu cầu cần đạt về kiến thức, kĩ năng môn Tin học nêu trong Chương trình Giáo dục phổ thông năm 2018.

Nội dung sách gồm 31 bài học được kết cấu thành 6 chủ đề kiến thức. Các bài học từ Bài 1 tới Bài 16 là nội dung kiến thức cốt lõi chung cho cả hai định hướng ICT và CS. Nội dung các bài học này tập trung trình bày khái niệm cơ bản về hệ điều hành và phần mềm ứng dụng; phần mềm nguồn mở, phần mềm chạy trên Internet; những hiểu biết mang tính tổng quan về hệ cơ sở dữ liệu (CSDL) với trọng tâm sâu hơn về một trong các hệ CSDL được ứng dụng phổ biến và rộng rãi nhất - hệ CSDL quan hệ. Các kiến thức mang tính hệ thống và sâu hơn về bên trong máy tính, chức năng và hoạt động của một số thiết bị ngoại vi và thiết bị số thông dụng; các vấn đề nâng cao hiệu quả tìm kiếm và trao đổi thông tin; yêu cầu ứng xử văn hoá và an toàn trên mạng; hỗ trợ hướng nghiệp với tin học cũng được trình bày trong các bài học này.

Các bài học từ Bài 17 tới Bài 31 là nội dung kiến thức dành riêng cho định hướng CS. Nội dung các bài học xoay quanh những hiểu biết cơ bản ban đầu nhưng hết sức quan trọng về một số kĩ thuật thiết kế thuật toán và lập trình; tổ chức dữ liệu, tổ chức chương trình và kiểm thử; những quy tắc thực hành để xác định độ phức tạp thời gian của một số thuật toán, chương trình đã biết,... Đây là những kiến thức nền tảng nhằm giúp các em củng cố và phát triển hơn nữa tư duy giải quyết vấn đề, khả năng đưa ra ý tưởng và chuyển giao nhiệm vụ cho máy tính thực hiện.

Các bài học của sách đều được cấu trúc một cách thống nhất với các hoạt động khám phá, luyện tập, vận dụng kiến thức một cách đa dạng, giúp các em có điều kiện trao đổi, tự nhận biết và tiếp thu kiến thức mới một cách hiệu quả và chủ động.

Hi vọng sách sẽ mang đến cho các em những kiến thức mới hấp dẫn và bổ ích.

*Chúc các em học tốt!*

CÁC TÁC GIẢ

# Mục lục


Trang

**CHỦ ĐỀ 1. MÁY TÍNH VÀ XÃ HỘI TRI THỨC ..... 5**

Bài 1. Hệ điều hành ..... 5
Bài 2. Thực hành sử dụng hệ điều hành ..... 10
Bài 3. Phần mềm nguồn mở và phần mềm chạy trên Internet ..... 15
Bài 4. Bên trong máy tính ..... 21
Bài 5. Kết nối máy tính với các thiết bị số ..... 27

**CHỦ ĐỀ 2. TỔ CHỨC LƯU TRỮ, TÌM KIẾM VÀ TRAO ĐỔI THÔNG TIN ..... 32**

Bài 6. Lưu trữ và chia sẻ tệp tin trên Internet ..... 32
Bài 7. Thực hành tìm kiếm thông tin trên Internet ..... 36
Bài 8. Thực hành nâng cao sử dụng thư điện tử và mạng xã hội ..... 39

**CHỦ ĐỀ 3. ĐẠO ĐỨC, PHÁP LUẬT VÀ VĂN HOÁ TRONG MÔI TRƯỜNG SỐ ..... 43**

Bài 9. Giao tiếp an toàn trên Internet ..... 43

**CHỦ ĐỀ 4. GIỚI THIỆU CÁC HỆ CƠ SỞ DỮ LIỆU ..... 49**

Bài 10. Lưu trữ dữ liệu và khai thác thông tin phục vụ quản lí ..... 49
Bài 11. Cơ sở dữ liệu ..... 53
Bài 12. Hệ quản trị cơ sở dữ liệu và hệ cơ sở dữ liệu ..... 58
Bài 13. Cơ sở dữ liệu quan hệ ..... 64
Bài 14. SQL - ngôn ngữ truy vấn có cấu trúc ..... 69
Bài 15. Bảo mật và an toàn hệ cơ sở dữ liệu ..... 73

**CHỦ ĐỀ 5. HƯỚNG NGHIỆP VỚI TIN HỌC ..... 77**

Bài 16. Công việc quản trị cơ sở dữ liệu ..... 77

**CHỦ ĐỀ 6. KĨ THUẬT LẬP TRÌNH ..... 81**

Bài 17. Dữ liệu mảng một chiều và hai chiều ..... 81
Bài 18. Thực hành dữ liệu mảng một chiều và hai chiều ..... 86
Bài 19. Bài toán tìm kiếm ..... 89
Bài 20. Thực hành bài toán tìm kiếm ..... 94
Bài 21. Các thuật toán sắp xếp đơn giản ..... 99
Bài 22. Thực hành bài toán sắp xếp ..... 104
Bài 23. Kiểm thử và đánh giá chương trình ..... 106
Bài 24. Đánh giá độ phức tạp thời gian thuật toán ..... 111
Bài 25. Thực hành xác định độ phức tạp thời gian thuật toán ..... 115
Bài 26. Phương pháp làm mịn dần trong thiết kế chương trình ..... 118
Bài 27. Thực hành thiết kế chương trình theo phương pháp làm mịn dần ..... 123
Bài 28. Thiết kế chương trình theo mô đun ..... 127
Bài 29. Thực hành thiết kế chương trình theo mô đun ..... 132
Bài 30. Thiết lập thư viện cho chương trình ..... 137
Bài 31. Thực hành thiết lập thư viện chương trình ..... 143
Bảng giải thích thuật ngữ ..... 146

Chủ đề 1

# Máy tính và xã hội tri thức

BÀI 1

## HỆ ĐIỀU HÀNH

SAU BÀI HỌC NÀY EM SẼ:

- Trình bày được sơ lược lịch sử phát triển của các hệ điều hành thông dụng cho máy tính cá nhân (PC).
- Chỉ ra được một số đặc điểm của hệ điều hành cho thiết bị di động.
- Trình bày được một cách khái quát mối quan hệ giữa phần cứng, hệ điều hành và phần mềm ứng dụng cũng như vai trò của mỗi thành phần trong hoạt động chung của cả hệ thống.

Khi chưa có hệ điều hành, con người phải can thiệp vào hầu hết quá trình hoạt động của máy tính nên hiệu quả khai thác sử dụng máy tính rất thấp. Sự ra đời của hệ điều hành đã giúp khắc phục được tình trạng đó. Việc sử dụng máy tính về cơ bản được thực hiện thông qua hệ điều hành. Em hãy chỉ ra một số công việc mà hệ điều hành thực hiện.

### 1. LỊCH SỬ PHÁT TRIỂN CỦA HỆ ĐIỀU HÀNH MÁY TÍNH CÁ NHÂN

**Hoạt động 1** Tìm hiểu các chức năng của hệ điều hành

Hệ điều hành của các loại máy tính nói chung có năm nhóm chức năng sau:

– Quản lí thiết bị (CPU, bộ nhớ hay thiết bị ngoại vi).

– Quản lí việc lưu trữ dữ liệu (quản lí tệp và thư mục).

– Tổ chức thực hiện các chương trình, điều phối tài nguyên cho các tiến trình xử lí trên máy tính. Nói cách khác, hệ điều hành là môi trường để chạy các ứng dụng.

– Cung cấp môi trường giao tiếp với người sử dụng.

– Cung cấp một số tiện ích giúp nâng cao hiệu quả sử dụng máy tính như định dạng đĩa, nén tệp, kiểm tra lỗi đĩa cứng, cấu hình kết nối mạng,...

Theo em, nhóm chức năng nào thể hiện rõ nhất đặc thù của hệ điều hành máy tính cá nhân?

Khác với các máy chủ, siêu máy tính do kĩ sư vận hành, máy tính cá nhân dành cho người dùng phổ thông nên sự thân thiện, dễ sử dụng là tiêu chí quan trọng nhất. Quá trình hình thành và phát triển của hệ điều hành máy tính cá nhân có liên quan chặt chẽ tới tiêu chí này và được thể hiện ở hai điểm chính sau:

- Giao diện đồ hoạ.
- Cơ chế plug & play để tự động nhận biết thiết bị ngoại vi khi khởi động máy tính.

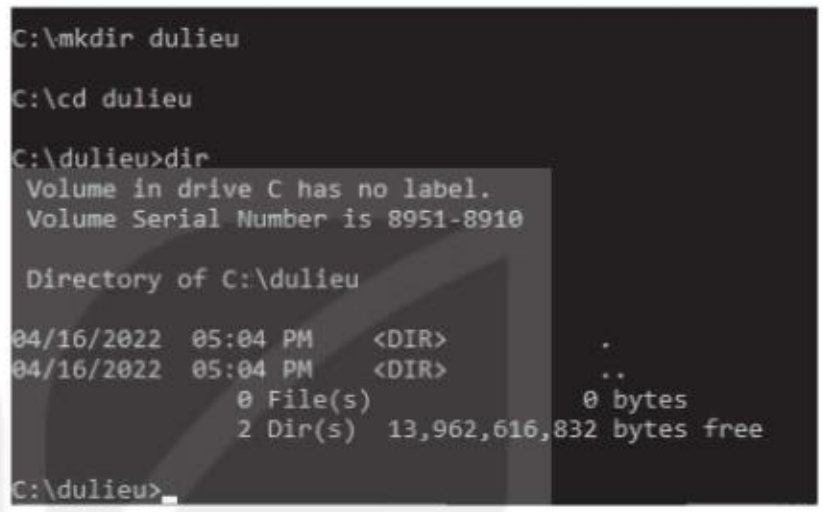
Có thể nói cơ chế plug & play ("cắm và chạy", còn được hiểu là "cắm vào là chạy") là bước phát triển hết sức quan trọng của hệ điều hành máy tính cá nhân. Thời kì đầu, thiết bị ngoại vi nhiều khi gây phiền phức cho người sử dụng, bởi lẽ mỗi thiết bị ngoại vi của một hãng đòi hỏi phải có một phần mềm điều khiển riêng, việc cài đặt không phải luôn dễ dàng với những người ít hiểu biết về tính năng của thiết bị ngoại vi và hoạt động của nó. Cơ chế plug & play giúp hệ điều hành nhận biết các thiết bị ngoại vi ngay khi khởi động máy và có thể hỗ trợ cài đặt các chương trình điều khiển một cách tự động.

Về giao diện, ban đầu hệ điều hành máy tính cá nhân sử dụng *giao diện dòng lệnh*, người dùng phải gõ các lệnh, chẳng hạn trong hệ điều hành DOS, lệnh:

"erase C:\dulieu\danhsach.txt"

sẽ xoá tệp danhsach.txt nằm trong thư mục dulieu của ổ đĩa C. Mặc dù đơn giản, nhưng DOS thiếu tính trực quan, đòi hỏi người dùng phải nhớ cú pháp của từng câu lệnh.

Bước phát triển tiếp theo của hệ điều hành là sử dụng *giao diện đồ hoạ* với các đối tượng thể hiện bằng hình ảnh. Một số thành phần cơ bản của giao diện đồ hoạ bao gồm:




*Hình 1.1. Giao diện lệnh của DOS*

- *Cửa sổ* là một vùng hình chữ nhật trên màn hình dành cho một ứng dụng. Cửa sổ có thể phóng to, thu nhỏ, ẩn đi hoặc đóng lại.
- *Biểu tượng* dễ gợi nhớ, cho phép quan sát đối tượng dưới dạng đồ hoạ.
- *Chuột* là phương tiện chỉ định điểm làm việc trên màn hình thể hiện bởi một con trỏ màn hình.

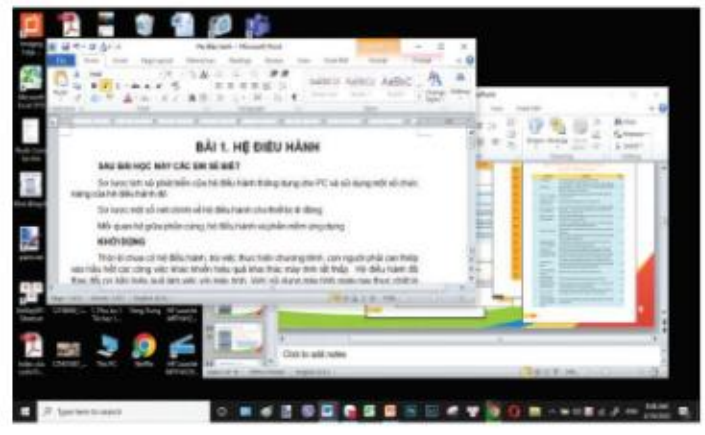
Giao diện đồ hoạ có tính trực quan, giúp người sử dụng giao tiếp với máy dễ dàng hơn. Các hệ điều hành cho máy tính cá nhân càng ngày càng thân thiện, một số hệ điều hành đã hỗ trợ giao tiếp bằng giọng nói.

Hai dòng máy tính cá nhân chủ đạo là dòng Mac (MacBook, iMac) sử dụng hệ điều hành đồ hoạ macOS của Apple và dòng PC sử dụng hệ điều hành đồ hoạ Windows của Microsoft. Sau đây chúng ta sẽ tìm hiểu kĩ hơn lịch sử phát triển của một hệ điều hành thương mại và một hệ điều hành nguồn mở (mã nguồn mở) phổ biến nhất.

## a) Hệ điều hành Windows

Windows đã trải qua nhiều phiên bản. Sau đây là một số phiên bản quan trọng, đánh dấu các mốc phát triển của Windows.

- Phiên bản 1 của Windows phát hành vào năm 1985 với giao diện đồ hoạ.
- Phiên bản 3, phát hành năm 1990 bắt đầu có khả năng đa nhiệm, cho phép chạy nhiều chương trình đồng thời, giúp nâng cao hiệu quả máy tính. Chức năng kéo thả tiện lợi bắt đầu có từ phiên bản 3.1. Các tính năng làm việc với mạng bắt đầu có từ phiên bản 3.11.




*Hình 1.2. Giao diện đồ hoạ của Windows*

- Trong vòng 10 năm đầu, về cơ bản Windows chỉ là một vỏ bọc đồ hoạ và từ đó gọi các dịch vụ của DOS. Chỉ từ phiên bản Windows 95 (1995) nhiều tính năng cơ bản của hệ điều hành mới được tích hợp trực tiếp vào Windows. Không những thế Windows 95 còn có giao diện đẹp, giao tiếp tiện lợi. Nhiều công cụ bắt đầu có từ phiên bản này hiện nay vẫn được dùng như bảng chọn Start, thanh trạng thái (Status bar), thanh công việc (Taskbar) và biểu tượng shortcut. Cơ chế plug & play lần đầu tiên được sử dụng.
- Năm 2001, Microsoft phát hành Windows XP với nhiều cải tiến đáng kể về giao diện và hiệu suất làm việc với một nâng cấp quan trọng để chạy trên các bộ xử lí tiên tiến thế hệ 64 bit. Đây là một trong các phiên bản hệ điều hành thành công nhất của Microsoft với số người sử dụng rất lớn.
- Các phiên bản Windows 7 (2009), Windows 8 (2012), Windows 10 (2015) và Windows 11 (2021) là một thế hệ mới với những thay đổi lớn so với Windows XP về độ an toàn, ổn định và hiệu quả sử dụng tài nguyên. Chúng dễ dùng hơn và hầu như không còn lỗi bất thường như các phiên bản trước.

Windows gần như thống trị thị trường hệ điều hành máy tính cá nhân. Một thống kê vào năm 2018 cho thấy hơn 86% người dùng máy tính sử dụng Windows.

## b) Hệ điều hành LINUX và các phiên bản

LINUX có nguồn gốc từ hệ điều hành UNIX – một hệ điều hành đa nhiệm (có thể chạy đồng thời nhiều chương trình) và đa người dùng (nhiều người có thể làm việc đồng thời) được phát triển từ năm 1969. UNIX đã chứng tỏ được tính hiệu quả, ổn định và an toàn cao. Phần lớn các phiên bản UNIX thương mại đều có giá thành khá cao.

Ý tưởng xây dựng một hệ điều hành kiểu UNIX chạy trên các máy tính cá nhân được quan tâm từ giữa những năm 1980 nhưng chỉ thực sự thành công với hệ điều hành LINUX, do Linus Torvalds viết vào năm 1991. Phiên bản LINUX 1.0 được công bố chính thức năm 1994 dưới dạng mã nguồn mở cho phép bất cứ ai cũng có thể sử dụng và phát triển thêm. LINUX đã khởi đầu trào lưu phần mềm nguồn mở, có ảnh hưởng rất lớn đến sự phát triển của công nghệ thông tin sau này.



*Hình 1.3. Màn hình nền của Ubuntu*

LINUX được cộng đồng người dùng đánh giá cao và sử dụng rộng rãi. LINUX không chỉ dùng cho máy tính cá nhân mà còn dùng cho cả máy chủ và các thiết bị nhúng – các thiết bị có phần mềm được tích hợp vào phần cứng và được thiết kế riêng như ti vi, xe ô tô tự lái,...

Đối với máy tính cá nhân, đã có nhiều biến thể khác nhau ra đời từ LINUX như Red Hat (viết năm 1994 và phát hành rộng rãi năm 1999), Suse (1996), Ubuntu (2004), thậm chí hệ điều hành Android của phần lớn điện thoại di động cũng được xây dựng trên lõi của LINUX.

Các hệ điều hành máy tính cá nhân phát triển theo hướng ngày càng dễ sử dụng, thể hiện ở các điểm sau:

- Giao diện thân thiện, từ giao diện dòng lệnh chuyển sang giao diện đồ hoạ và tích hợp với nhận dạng tiếng nói.
- Khả năng nhận biết các thiết bị ngoại vi với cơ chế plug & play giúp người sử dụng không cần quan tâm tới trình điều khiển của thiết bị ngoại vi.

Các hệ điều hành thông dụng nhất trên máy tính cá nhân là MacOS trên dòng máy MAC và Windows trên dòng máy PC. Đặc biệt LINUX và các biến thể của nó như Red Hat, Suse hay Ubuntu là hệ điều hành nguồn mở, mang đến cho người dùng các hệ điều hành mạnh mẽ, tin cậy và chi phí thấp.



1. Nêu các nhóm chức năng chính của hệ điều hành.
2. Nêu các đặc điểm cơ bản của hệ điều hành máy tính cá nhân.

## 2. HỆ ĐIỀU HÀNH CHO THIẾT BỊ DI ĐỘNG

**Hoạt động 2** **Một số đặc điểm của hệ điều hành cho thiết bị di động**

Điện thoại thông minh, máy tính bảng (gọi chung là thiết bị di động) thực chất là các máy tính cá nhân. Sự khác nhau giữa hệ điều hành cho thiết bị di động và hệ điều hành của máy tính có nguồn gốc từ sự khác biệt về tính năng, tác dụng của hai loại thiết bị này. Hãy cùng thảo luận để chỉ ra những điểm khác nhau đó.



Ra đời sau máy tính nhưng điện thoại thông minh và máy tính bảng được phổ cập nhanh, thúc đẩy mạnh mẽ tiến trình tin học hoá xã hội.

Điện thoại thông minh không chỉ dùng để nghe, gọi mà còn được trang bị rất nhiều tiện ích như chụp ảnh, quay phim, định vị, ghi nhận tình trạng sức khoẻ.

Do tính di động mà hệ điều hành cho thiết bị di động chú trọng đến khả năng kết nối mạng không dây như wifi, Internet di động (dịch vụ 3G, 4G, 5G,...), bluetooth hay giao tiếp gần NFC.

Các thiết bị di động phần lớn nhỏ, gọn và giao tiếp phổ biến nhờ các thao tác vuốt, chạm, lắc,... Chúng thường được trang bị màn hình cảm ứng, bàn phím ảo và nhiều cảm biến để tạo các giao tiếp hay ứng dụng độc đáo.

Có nhiều hệ điều hành cho các thiết bị di động nhưng phổ biến hơn cả là iOS của Apple dùng cho iPhone, iPad và Android của Google dùng cho hầu hết các dòng điện thoại khác.

Một số khác biệt của hệ điều hành cho thiết bị di động so với hệ điều hành cho máy tính cá nhân:

- Giao diện đặc biệt thân thiện nhờ nhận dạng hành vi của người dùng thông qua các cảm biến.
- Dễ dàng kết nối mạng di động.
- Nhiều tiện ích hỗ trợ cá nhân.

Hai hệ điều hành phổ biến cho thiết bị di động là iOS của Apple và Android của Google.



1. Vì sao hệ điều hành di động ưu tiên cao cho giao tiếp thân thiện và kết nối mạng di động?
2. Kể tên ba tiện ích thường có trên thiết bị di động và chức năng của nó.

## 3. QUAN HỆ GIỮA HỆ ĐIỀU HÀNH, PHẦN CỨNG VÀ PHẦN MỀM ỨNG DỤNG

### Hoạt động 3 Vai trò của hệ điều hành

Có hay không trường hợp phần mềm chạy trên một thiết bị không có hệ điều hành? Khi nào cần phải có hệ điều hành?

Thời kì mới có máy tính, chưa có hệ điều hành, người sử dụng phải nạp thủ công chương trình vào bộ nhớ như gõ lệnh hay nhập chương trình được mã hoá trên băng giấy hoặc bìa đục lỗ, sau đó nhấn một phím trên bàn điều khiển và đợi kết quả.

Ngày nay, có nhiều thiết bị được điều khiển bởi các bộ vi xử lí, cài sẵn chương trình ghi trong bộ nhớ ROM, bật lên là chạy không cần hệ điều hành. Ví dụ hệ thống điều khiển lò vi sóng cho phép người dùng chọn lựa các chế độ nấu ăn.

Tuy nhiên cách này chỉ phù hợp với những thiết bị chuyên dụng, chỉ làm một việc, không thích hợp với các thiết bị đa năng như máy tính có khả năng thực hiện nhiều công việc. Với thiết bị đa năng, người dùng có nhu cầu nạp nhiều phần mềm ứng dụng và dữ liệu vào bộ nhớ ngoài (cần tổ chức dữ liệu), chọn phần mềm để chạy (cần giao diện làm việc). Khi chạy, cần điều phối tài nguyên cho các ứng dụng như bộ nhớ, công suất CPU, các thiết bị ngoại vi. Cần có hệ điều hành để đáp ứng các nhu cầu trên. Hệ điều hành cung cấp các dịch vụ điều khiển máy tính để thực hiện các công việc cơ bản mà nhiều chương trình ứng dụng cần đến. Ví dụ, hệ điều hành có các dịch vụ tìm kiếm tệp trong bộ nhớ ngoài, mở tệp, ghi dữ liệu vào tệp và đóng tệp. Các ứng dụng muốn ghi dữ liệu vào tệp không tự điều khiển máy tính làm những công việc trên mà chỉ gọi các dịch vụ do hệ điều hành cung cấp để thực hiện.




*Hình 1.4. Mối quan hệ giữa phần cứng, phần mềm ứng dụng và hệ điều hành*

Như vậy, phần cứng là thiết bị xử lí thông tin, hệ điều hành là môi trường trung gian giúp phần mềm ứng dụng khai thác phần cứng. Quan hệ giữa hệ điều hành, phần cứng và phần mềm ứng dụng được minh hoạ trong Hình 1.4.

Hệ điều hành là môi trường để phần mềm ứng dụng khai thác hiệu quả phần cứng.

1. Nêu lí do thiết bị xử lí đa năng cần có hệ điều hành.
2. Nêu mối quan hệ giữa phần cứng, phần mềm ứng dụng và hệ điều hành.

## LUYỆN TẬP

1. Em hiểu thế nào về tính thân thiện của hệ điều hành?
2. Hệ điều hành cung cấp môi trường giao tiếp với người sử dụng như thế nào? Môi trường giao tiếp đó thể hiện như thế nào trên hệ điều hành Windows?

## VẬN DỤNG

1. Em hãy tìm hiểu xem ngoài máy tính còn có thiết bị điện gia dụng nào sử dụng hệ điều hành không.
2. Thực ra, LINUX là hệ điều hành có nguồn gốc từ hệ điều hành UNIX. Hãy tìm hiểu lịch sử của hệ điều hành LINUX để biết thêm về hệ điều hành UNIX.

# BÀI 2 THỰC HÀNH SỬ DỤNG HỆ ĐIỀU HÀNH

SAU BÀI HỌC NÀY EM SẼ:

- Sử dụng được một số chức năng cơ bản của hệ điều hành cho máy tính cá nhân.
- Sử dụng được một vài tiện ích của hệ điều hành nâng cao hiệu quả của máy tính cá nhân.
- Sử dụng được một vài tiện ích cơ bản của hệ điều hành trên thiết bị di động.

Các thiết bị di động thực tế cũng là máy tính cá nhân. Hệ điều hành của các loại máy tính cá nhân có nhiều tiện ích khác nhau nhưng giao diện người dùng có nhiều điểm tương đồng. Em hãy chỉ ra một vài điểm tương đồng đó.

## Nhiệm vụ 1. Sử dụng một số chức năng cơ bản của hệ điều hành cho máy tính cá nhân

Ở Bài 1, em đã biết hệ điều hành cho máy tính cá nhân như Windows và LINUX. Em hãy thực hành sử dụng các chức năng sau (với hệ điều hành Windows hoặc hệ điều hành mã nguồn mở Ubuntu):

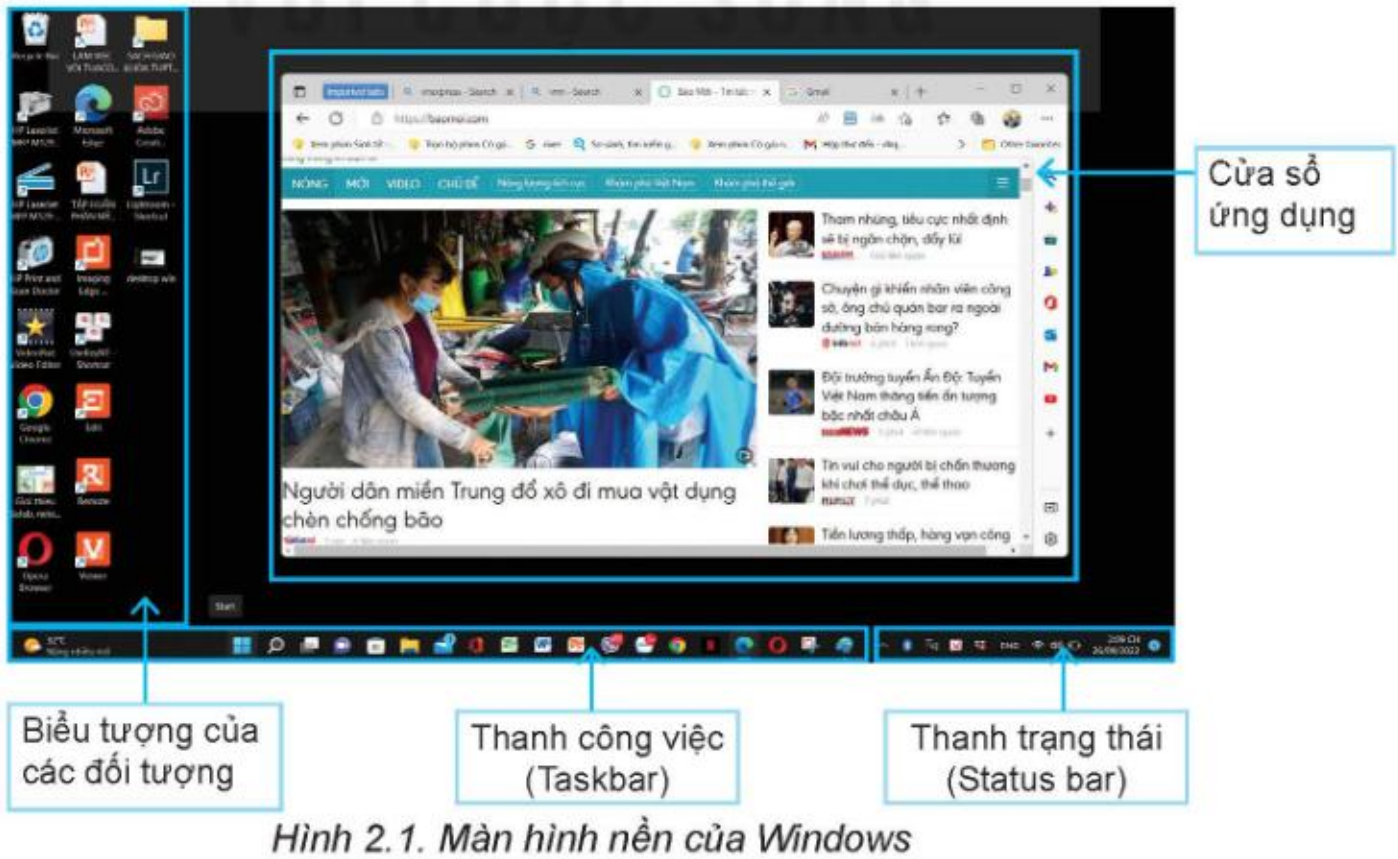
- Cung cấp môi trường giao tiếp với người sử dụng.
- Quản lí tệp và thư mục.

**Hướng dẫn:**

- Cung cấp môi trường giao tiếp với người sử dụng.

– Quan sát giao diện đồ hoạ với các cửa sổ, các biểu tượng và con trỏ. Mỗi cửa sổ hay biểu tượng đều có tên. Con trỏ dùng để chọn đối tượng làm việc.

– Nhận biết các biểu tượng trên màn hình như: tệp, thư mục, nút lệnh,...




*Hình 2.1. Màn hình nền của Windows*




*Hình 2.2. Màn hình nền của Ubuntu*

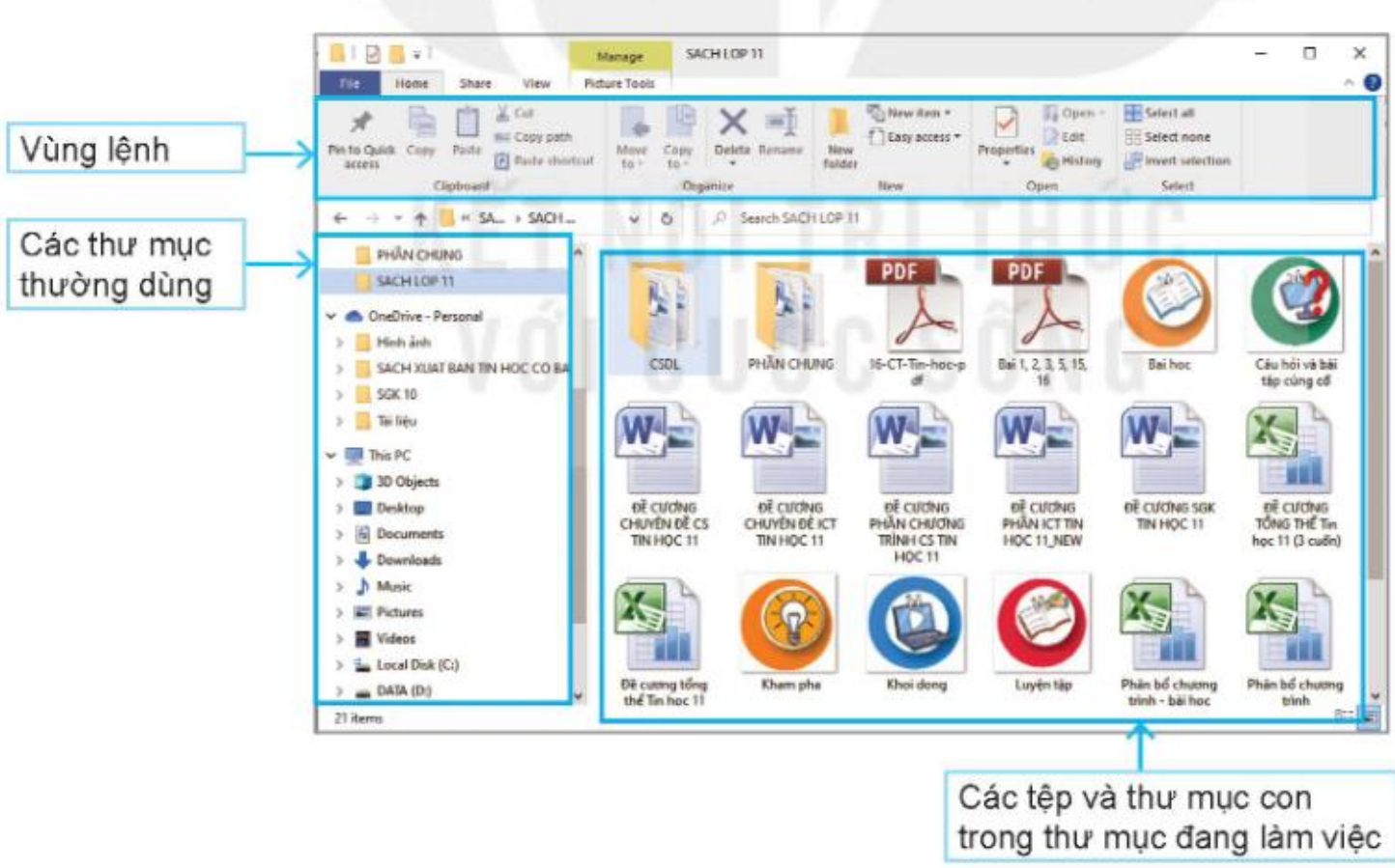
– Truy cập nhanh các phần mềm ứng dụng nhờ thanh công việc hay nút Start.

– Quan sát thanh trạng thái hiển thị các biểu tượng và cho biết trạng thái làm việc của máy tính như kết nối mạng, dung lượng pin, mức loa, chế độ bàn phím,…

– Thực hành các thao tác làm việc với biểu tượng như nháy chuột, nháy đúp chuột, nháy nút phải chuột, kéo thả chuột.

- Quản lí tệp và thư mục.

– Sử dụng tiện ích File Explorer của Windows để quản lí tệp và thư mục.




*Hình 2.3. Quản lí thư mục và tệp bằng tiện ích File Explorer của Windows*

– Trong cửa sổ File Explorer, nháy chuột (nháy đúp chuột) vào biểu tượng thư mục để xem nội dung bên trong (các tệp và thư mục con).

– Thực hành quản lí thư mục gồm: tạo mới, đổi tên, xoá, di chuyển thư mục.

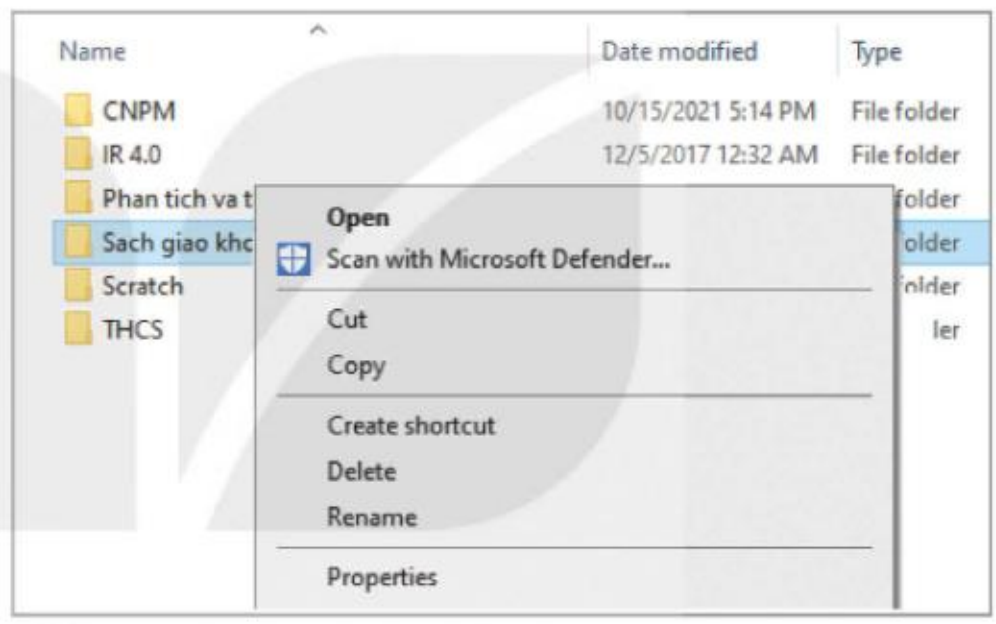
– Thực hành quản lí tệp trên Ubuntu gồm: đổi tên, xoá, di chuyển tệp và chạy ứng dụng với tệp chương trình.




*Hình 2.4. Giao diện chương trình quản lí thư mục và tệp trên Ubuntu*

– Nháy đúp chuột vào biểu tượng của tệp ứng dụng hoặc tệp dữ liệu để kích hoạt ứng dụng tương ứng.

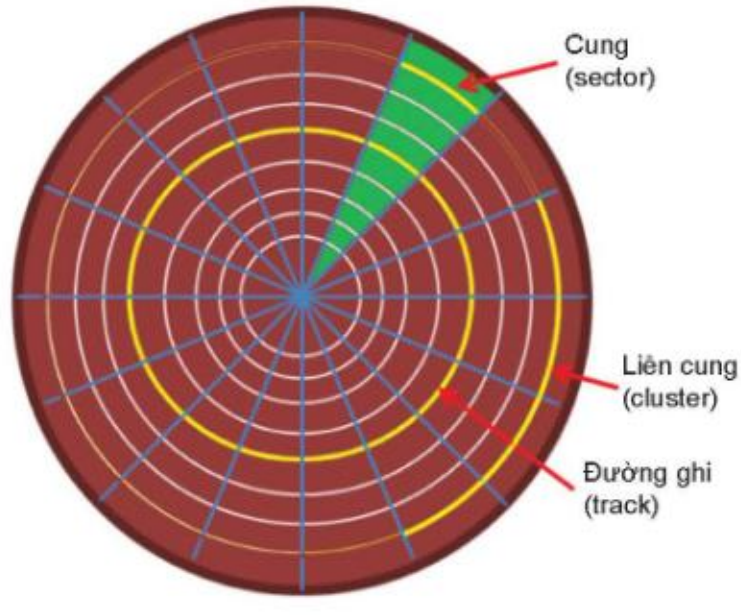
– Nháy nút phải chuột vào một đối tượng để làm xuất hiện bảng chọn gồm các lệnh có thể thực hiện được với đối tượng đó như minh hoạ trong Hình 2.5.




*Hình 2.5. Bảng chọn xuất hiện khi nháy nút phải chuột*

## Nhiệm vụ 2. Sử dụng một số tiện ích trên hệ điều hành máy tính cá nhân nhằm nâng cao hiệu quả sử dụng máy

Tiện ích là những phần mềm công cụ hỗ trợ nhiều công việc khác nhau như ứng dụng tính toán, chụp ảnh màn hình, gõ tiếng Việt hoặc các phần mềm nâng cao hiệu quả làm việc với máy tính như nén tệp, quét virus. Có một số tiện ích được cài đặt cùng với hệ điều hành như tiện ích kiểm tra đĩa cứng hay kiểm tra kết nối mạng nhưng cũng có những tiện ích không có sẵn, được cài đặt sau. Sau khi được cài đặt, chúng được tích hợp như một dịch vụ trên hệ điều hành, ví dụ các tiện ích nén dữ liệu zip hay bộ gõ bàn phím Unikey.




*Hình 2.6. Tổ chức ghi dữ liệu trên đĩa cứng*

Em hãy thực hành sử dụng tiện ích kiểm tra đĩa và hợp mảnh trên đĩa cứng.

**Hướng dẫn:**

Đĩa cứng (Hard Disk Driver – HDD) là loại đĩa kim loại phủ vật liệu từ tính. Dữ liệu được đọc, ghi bởi các đầu từ theo các đường tròn đồng tâm gọi là đường ghi (track), mỗi đường gồm nhiều cung (sector), mỗi cung ghi 512 byte dữ liệu. Việc đọc, ghi được thực hiện theo đơn vị liên cung (cluster), thường gồm 8 cung.

Khi ghi tệp, hệ điều hành sẽ tìm các vùng trống trên đĩa để ghi, những vùng này có thể nằm trên các đường ghi khác nhau làm tệp bị phân mảnh. Khi đó, thời gian đọc, ghi tệp tăng lên nhiều lần vì đầu từ phải dịch chuyển (một cách cơ học) từ đường ghi này sang đường ghi khác. Việc tổ chức lại tệp sao cho các liên cung của một tệp được ghi liên tục, giảm hoạt động di chuyển đầu từ sẽ giúp tăng tốc độ truy cập đĩa cứng. Tiện ích tối ưu hoá (Optimize), còn có tên là hợp mảnh (Defragment) cho phép thực hiện công việc này.

Do nhiều nguyên nhân, có thể xảy ra các lỗi tệp như một liên cung mất liên kết với tệp, tạo thành các đoạn dữ liệu "mồ côi", có trên đĩa nhưng không khai thác được hoặc tình trạng chồng chéo, khi có vài tệp liên kết đến cùng một liên cung. Ngoài ra, còn có tình trạng một số cung bị hỏng về vật lí (bad sector), đọc ghi không được, cần phải loại khỏi danh sách sử dụng. Tiện ích kiểm tra đĩa (Check disk) dùng để khử các lỗi trên để việc đọc, ghi đĩa trở lại bình thường.

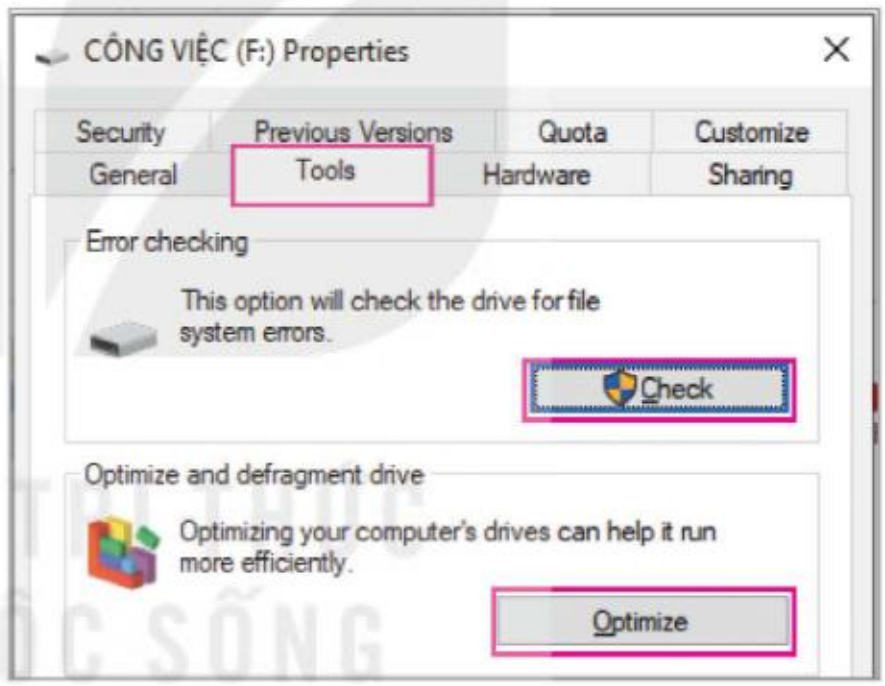
Các bước để sửa lỗi đĩa và hợp mảnh:

*Bước 1.* Sử dụng **File Explorer** và tìm danh sách các ổ đĩa. Nháy nút phải chuột vào ổ đĩa muốn xử lí rồi chọn **Properties**.

*Bước 2.* Trong cửa sổ **Properties** của đĩa cứng, chọn **Tools**.

*Bước 3.* Chọn **Check** để kiểm tra và khắc phục lỗi đĩa; Chọn **Optimize** để tối ưu hoá, hợp mảnh.

*Lưu ý:* Chức năng hợp mảnh chỉ có tác dụng đối với đĩa từ. Việc hợp mảnh có thể mất nhiều thời gian nếu đĩa có dung lượng lớn.




*Hình 2.7. Cửa sổ **Properties** của đĩa cứng*

## Nhiệm vụ 3. Sử dụng một số tiện ích của hệ điều hành cho thiết bị di động

**1.** Em hãy kể tên một số các tiện ích của hệ điều hành Android hoặc iOS cho thiết bị di động mà em biết.

Thiết bị di động như điện thoại thông minh và máy tính bảng cung cấp cho người dùng nhiều tiện ích cá nhân như quản lí danh bạ, nhắn tin, hẹn giờ, lịch, quản lí ảnh, quản lí tệp,...

**2.** Giao diện quản lí danh bạ

Hãy tìm hiểu trên điện thoại em đang sử dụng để thực hiện các chức năng sau:

– Hiển thị danh bạ.

– Thêm một người vào danh bạ với các thông tin về số điện thoại, địa chỉ, nhóm; sửa thông tin của một người trong danh bạ; chia sẻ thông tin danh bạ.

– Xoá một người khỏi danh bạ.

– Truy cập danh bạ để gọi điện thoại nhanh.

**3.** Đặt lịch, hẹn giờ, nhắc việc

– Mở giao diện đồng hồ, sau đó đặt một công việc được nhắc hẹn vào một giờ định trước, một ngày định trước.

– Đặt hẹn một công việc hằng ngày để được nhắc hằng ngày.

**4.** Quản lí ứng dụng

– Xem các ứng dụng được tải và cài trên máy.

– Xoá ứng dụng không cần thiết.



*Hình 2.8. Một số biểu tượng ứng dụng trên màn hình chính của điện thoại thông minh*




*Hình 2.9. Truy cập tiện ích quản lí ứng dụng*

## LUYỆN TẬP

**1.** Tiện ích danh bạ còn có chức năng quản lí nhóm. Mỗi nhóm có thể gồm nhiều số điện thoại, mỗi số điện thoại có thể thuộc nhiều nhóm. Hãy thực hiện việc tạo nhóm, xoá nhóm, đăng kí vào danh bạ.

**2.** Đồng hồ là một tiện ích cơ bản của hệ điều hành di động. Ứng dụng này, không chỉ cho phép đặt nhắc hẹn (một lần hay định kì) mà còn có khả năng đếm thời gian chính xác đến 1% giây, rất cần cho các hoạt động cần độ chính xác cao như đo các kỉ lục thể thao. Hãy sử dụng các chức năng đếm thời gian tiến hay lùi của tiện ích này.

## VẬN DỤNG

**1.** Hình 2.7 là cửa sổ Properties mở theo trang Tools. Nếu mở theo trang General em sẽ thấy có nút lệnh của tiện ích dọn đĩa (Disk Cleanup). Hãy tìm hiểu chức năng và cách sử dụng tiện ích này.

**2.** Ngoài cách đăng nhập dùng mật khẩu, các thiết bị di động còn cung cấp nhiều phương pháp đăng nhập khác như nhận dạng vân tay, nhận dạng khuôn mặt hay dùng khẩu hình. Hãy tìm hiểu các phương pháp đó và so sánh các điểm mạnh, điểm yếu của chúng.

BÀI 3

# PHẦN MỀM NGUỒN MỞ VÀ PHẦN MỀM CHẠY TRÊN INTERNET

SAU BÀI HỌC NÀY EM SẼ:

- Trình bày được một số khái niệm và so sánh phần mềm nguồn mở với phần mềm thương mại; nêu được vai trò của phần mềm nguồn mở và phần mềm thương mại đối với sự phát triển của công nghệ thông tin.
- Làm quen với phần mềm chạy trên Internet.

Với ngôn ngữ lập trình bậc cao, chương trình được viết dưới dạng văn bản gần với ngôn ngữ tự nhiên. Văn bản này gọi là mã nguồn. Để máy tính có thể chạy được trực tiếp, chương trình được dịch thành dãy lệnh máy gọi là mã máy. Mã máy rất khó đọc, hiểu nên việc dịch sang mã máy còn giúp bảo vệ chống đánh cắp ý tưởng hay sửa đổi phần mềm. Phần mềm chuyển giao dưới dạng mã máy thường được gọi là phần mềm nguồn đóng.

Vào những năm 1970, trong một số trường đại học ở Mỹ đã xuất hiện việc chia sẻ mã nguồn để cùng phát triển phần mềm, dẫn tới sự ra đời của phần mềm nguồn mở – một xu hướng có ảnh hưởng lớn tới sự phát triển của công nghệ phần mềm sau này.

Theo em, lợi ích đối với cộng đồng trong việc chia sẻ mã nguồn là gì?

## 1. PHẦN MỀM NGUỒN MỞ

**Hoạt động 1** Tìm hiểu các cách chuyển giao phần mềm

Cách thức chuyển giao phần mềm cho người sử dụng theo chiều hướng "mở dần" như sau:

1. Bán phần mềm dưới dạng mã máy.
2. Cho sử dụng phần mềm miễn phí có điều kiện hoặc không điều kiện, không cung cấp mã nguồn.
3. Cho sử dụng phần mềm tự do, cung cấp cả mã nguồn để có thể sửa, nâng cấp, phát triển và chuyển giao (phân phối) lại phần mềm.

Hãy thảo luận xem lợi ích của người dùng được tăng dần như thế nào theo hướng mở nói trên.

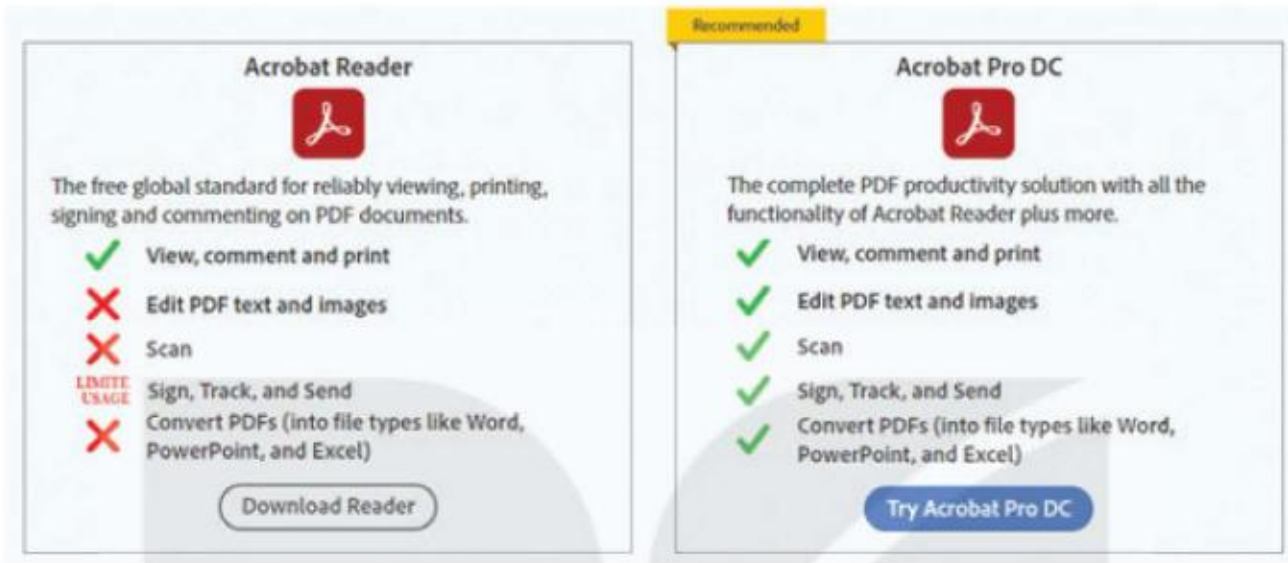
### a) Phân loại phần mềm theo cách chuyển giao sử dụng

Các loại phần mềm tương ứng với ba cách thức chuyển giao trong Hoạt động 1 được gọi lần lượt là:

- **Phần mềm thương mại** (commercial software) là phần mềm để bán. Hầu hết các phần mềm thương mại là loại nguồn đóng để bảo vệ ý tưởng và chống sửa đổi. Phần mềm soạn thảo văn bản Microsoft Word và phần mềm chỉnh sửa ảnh Photoshop,... là các ví dụ về phần mềm thương mại.

• **Phần mềm tự do** (free software) là phần mềm không chỉ miễn phí mà còn được tự do sử dụng mà không phải xin phép. Phần mềm tự do có thể ở dạng mã máy hoặc mã nguồn.

Hình 3.1 cho thấy một phần của trang web cho phép tải phần mềm làm việc với các tệp pdf. Bản Acrobat Reader là phần mềm tự do ở dạng mã máy, chỉ có thể đọc, ghi chú và in văn bản. Nếu người dùng muốn có các tính năng cao hơn như soạn thảo, quét một ảnh thành một bản pdf, kí, theo dõi vết chỉnh sửa, chuyển định dạng,... thì phải mua bản thương mại Acrobat Pro DC.




*Hình 3.1. Acrobat có bản tự do Acrobat Reader và bản thương mại Acrobat Pro DC*

• **Phần mềm nguồn mở** (open-source software) là phần mềm được cung cấp cả mã nguồn để người dùng có thể tự sửa đổi, cải tiến, phát triển, phân phối lại theo một quy định gọi là giấy phép được nêu dưới đây. Thông thường, phần mềm nguồn mở là tự do và không được bảo hành.

Một số phần mềm mã nguồn mở được giới thiệu trong bộ sách giáo khoa này có thể kể tới đó là Inkscape, GIMP, môi trường lập trình cho ngôn ngữ Python,...

### b) Giấy phép đối với phần mềm nguồn mở

Khi sử dụng mỗi loại phần mềm nói trên, người dùng cần tuân thủ các điều kiện được phép sử dụng, thường được gọi là "giấy phép" với nghĩa của từ "license" trong tiếng Anh. Ví dụ, khi trả tiền cho mỗi giấy phép, phần mềm thương mại chỉ được phép cài trên một số lượng máy tính nhất định.

**Hoạt động 2** **Giấy phép đối với phần mềm nguồn mở**

Theo quy định về bản quyền, các tác giả của phần mềm có quyền bảo vệ chống phần mềm bị sửa đổi gây phương hại đến uy tín và danh dự của tác giả. Nếu là người đầu tư, các tác giả còn giữ cả quyền tạo bản sao, sửa đổi, nâng cấp phần mềm, quyền chuyển giao sử dụng,...

Em hãy so sánh quyền sử dụng phần mềm nguồn mở với quy định về bản quyền và cho biết một số điểm mâu thuẫn.

Chính vì sự mâu thuẫn giữa quy định về bản quyền và quyền sử dụng phần mềm nguồn mở mà cần có giấy phép cho phần mềm nguồn mở. Giấy phép sẽ cung cấp cho người dùng các quyền vốn bị cấm bởi các quy định bản quyền.

Giấy phép không chỉ đề cập đến quyền sử dụng mà còn liên quan đến nhiều vấn đề khác như:

- Các tác giả có được miễn trừ bảo hành hay không, có bị kiện vì những sai sót của phần mềm hay không.
- Người sửa đổi phần mềm có bắt buộc phải công bố rõ các tác giả trước đó hay không, bản sửa đổi có phải công khai dưới dạng nguồn mở hay không.

Có nhiều loại giấy phép phần mềm nguồn mở, trong đó giấy phép công cộng GNU GPL (GNU General Public License) được áp dụng rộng rãi nhất. Nó có những quy định không chỉ đảm bảo quyền tiếp cận của mọi người đối với các phần mềm nguồn mở mà còn đảm bảo sự phát triển bền vững của phần mềm nguồn mở. Giấy phép GNU GPL 3.0 phát hành năm 2007 (xem www.gnu.org/licenses/gpl.html) có một số nội dung đáng chú ý sau:

- Được sao chép và phân phối phần mềm; có quyền yêu cầu trả phí cho việc chuyển giao đó nhưng phải thông báo rõ ràng về bản quyền gốc và thông báo miễn trừ trách nhiệm bảo hành.
- Được sửa đổi và phân phối bản sửa đổi với điều kiện phải công bố mã nguồn phần sửa đổi, nêu rõ đó là bản đã được thay đổi, chỉ rõ các thành phần được thay đổi; đồng thời phải áp dụng giấy phép GNU GPL cho chính phần thay đổi đó. Nói cách khác, phần mềm có nguồn gốc từ việc sửa đổi một phần mềm nguồn mở theo GPL cũng phải là phần mềm nguồn mở theo GPL.

- Phần mềm thương mại dùng để bán, người dùng phải mua mới được quyền sử dụng. Hầu hết phần mềm thương mại được bán ở dạng mã máy, gọi là phần mềm nguồn đóng.
- Phần mềm nguồn mở là phần mềm được cung cấp cả mã nguồn mà người dùng có quyền sử dụng, thay đổi và phân phối lại theo các "giấy phép" thích hợp.
- Giấy phép công cộng GNU GPL là giấy phép điển hình đối với phần mềm nguồn mở. Nó đảm bảo quyền tiếp cận của người sử dụng đối với mã nguồn để dùng, thay đổi hoặc phân phối lại; bảo đảm quyền miễn trừ của các tác giả về hậu quả sử dụng phần mềm; bảo đảm quyền đứng tên của các tác giả tham gia phát triển, đảm bảo sự phát triển bền vững của phần mềm nguồn mở bằng cách công bố rõ ràng các thay đổi của các phiên bản và buộc phần phát triển dựa trên phần mềm nguồn mở theo giấy phép GPL cũng phải mở theo GPL.

1. Em hãy cho biết ý nghĩa của yêu cầu "người sửa đổi, nâng cấp phần mềm nguồn mở phải công bố rõ ràng phần nào đã sửa, sửa thế nào so với bản gốc".
2. Ý nghĩa của yêu cầu "phần mềm sửa đổi một phần mềm nguồn mở theo GPL cũng phải mở theo giấy phép của GPL" là gì?

## 2. VAI TRÒ CỦA PHẦN MỀM THƯƠNG MẠI VÀ PHẦN MỀM NGUỒN MỞ

Phần mềm nguồn mở đã trở thành cơ hội cho những ai muốn có những giải pháp phần mềm tốt với đầu tư thấp. Đến nay, hầu như lĩnh vực nào của tin học cũng có các phần mềm nguồn mở có thể thay thế được các phần mềm nguồn đóng. Bảng 3.1 là một số ví dụ về phần mềm nguồn mở và phần mềm thương mại cùng chức năng.

**Bảng 3.1.** *Phần mềm thương mại và phần mềm nguồn mở có thể thay thế*

| Lĩnh vực | Phần mềm thương mại | Phần mềm nguồn mở cùng chức năng |
|---|---|---|
| Phần mềm văn phòng | Word, Excel và PowerPoint (của Microsoft) | Writer, Calc và Impress (trong bộ OpenOffice) |
| Hệ điều hành | Windows (của Microsoft) | LINUX và các phiên bản (Ubuntu, Red Hat,...) |
| | iOS (cho iPhone, iPad) | Android (cho hầu hết các dòng điện thoại thông minh khác) |
| Hệ quản trị cơ sở dữ liệu | Oracle, SQL server (của Microsoft) | MySQL, PostGreSQL |
| Phần mềm xử lí ảnh | Photoshop (của Adobe System) | GIMP |

**Hoạt động 3** **Vai trò của phần mềm thương mại và phần mềm nguồn mở**

Hãy thảo luận xem phần mềm nguồn mở có thay thế hoàn toàn được phần mềm thương mại hay không. Tại sao?

Phần mềm thương mại có hai loại:

- **Phần mềm "đặt hàng"** (phần mềm "may-đo") được thiết kế theo yêu cầu của từng khách hàng. Ví dụ phần mềm điều khiển một dây chuyền lắp ráp hay phần mềm đặt xe trên thiết bị di động của các hãng taxi là các ví dụ về phần mềm đặt hàng. Điều quan trọng là phần mềm đặt hàng không những được thiết kế chính xác theo yêu cầu, mà còn được bảo hành theo hợp đồng.

- **Phần mềm "đóng gói"** được thiết kế dựa trên những yêu cầu chung của nhiều người. Chúng được viết rất hoàn chỉnh và kèm theo công cụ cài đặt tự động giúp người dùng dễ sử dụng. Người bán không có trách nhiệm sửa chữa nâng cấp theo yêu cầu của từng người dùng nhưng có thể nâng cấp định kì. Phần mềm xử lí ảnh Photoshop, phần mềm soạn thảo văn bản Microsoft Word là các ví dụ về phần mềm thương mại đóng gói.

Rõ ràng là phần mềm nguồn mở không thể thay thế được cho phần mềm thương mại trong thực tế. Mỗi phần mềm nguồn mở đáp ứng nhu cầu chung của nhiều người, trong khi đó những nhu cầu riêng, vốn phong phú hơn rất nhiều so với những nhu cầu chung thì phần mềm "đặt hàng" mới có thể đáp ứng được.

Một điều quan trọng là chính các phần mềm thương mại mới đem lại nguồn tài chính chủ yếu để duy trì các tổ chức làm phần mềm.

Cần lưu ý, phần mềm thương mại thường liên quan đến những giải pháp riêng của người cung cấp, nên người dùng dễ bị lệ thuộc vào nhà cung cấp cả về giải pháp cũng như hỗ trợ kĩ thuật.

**Bảng 3.2.** *So sánh phần mềm thương mại và phần mềm nguồn mở*

| Yếu tố | Phần mềm thương mại nguồn đóng | Phần mềm nguồn mở |
|---|---|---|
| Chi phí | Mất chi phí mua phần mềm và phí chuyển giao | Chỉ mất phí chuyển giao nếu có |
| Hỗ trợ kĩ thuật | Có | Không, nhưng có thể được hỗ trợ từ cộng đồng |
| Tính minh bạch | Khó kiểm soát những gì được cài cắm bên trong | Có thể kiểm soát được mã nguồn |
| Sự phụ thuộc của người dùng | Bị phụ thuộc vào nhà cung cấp về giải pháp và hỗ trợ kĩ thuật | Được cộng đồng phát triển theo chuẩn chung, không phụ thuộc vào riêng ai |

1. Ưu điểm của phần mềm thương mại:
- Phần mềm dạng "đặt hàng" đáp ứng nhu cầu riêng và người dùng được hỗ trợ kĩ thuật.
- Phần mềm "đóng gói" có tính hoàn chỉnh cao, đáp ứng nhu cầu rộng rãi.

2. Ưu điểm của phần mềm nguồn mở: chi phí thấp, minh bạch, không bị phụ thuộc nhiều vào nhà cung cấp.

3. Vai trò của hai loại phần mềm:
- Phần mềm thương mại là nguồn thu nhập chính của các tổ chức, cá nhân làm phần mềm chuyên nghiệp, góp phần tạo ra thị trường phần mềm phong phú, đáp ứng các nhu cầu riêng của cá nhân, tổ chức và các nhu cầu chung của xã hội.
- Phần mềm nguồn mở giúp những người có nhu cầu được sử dụng phần mềm dùng chung chất lượng tốt, ổn định với chi phí thấp.



1. Cho ví dụ về phần mềm đóng gói và phần mềm đặt hàng. Ưu điểm của phần mềm thương mại là gì?
2. Cho ví dụ về một phần mềm thương mại và một phần mềm nguồn mở có thể thay thế. Ưu điểm của phần mềm nguồn mở là gì?

## 3. PHẦN MỀM CHẠY TRÊN INTERNET

**Hoạt động 4** **Phần mềm chạy trên Internet**

Phần mềm chạy trên Internet là gì? Em hãy cho một ví dụ về phần mềm như vậy. Hãy nêu ưu điểm của phần mềm chạy trên Internet.

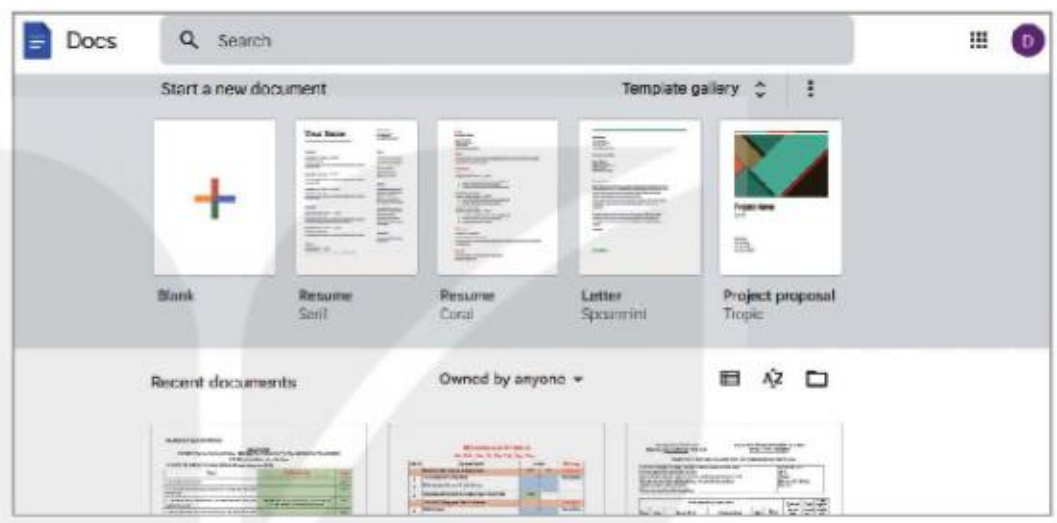
Phần mềm chạy trên Internet được hiểu là phần mềm cho phép sử dụng qua Internet mà không cần phải cài đặt vào máy.

Phần mềm chạy trên Internet (phần mềm trực tuyến) rất phổ biến, chẳng hạn phần mềm mạng xã hội, thư điện tử và các ứng dụng mua sắm trên mạng,... Lợi ích của các phần mềm này là có thể sử dụng ở bất cứ đâu, bất cứ khi nào, bất cứ máy tính nào miễn là có kết nối Internet, chi phí rẻ hoặc không mất phí.

Ví dụ: Google cung cấp nhiều phần mềm trực tuyến, trong đó phần mềm Google Docs giúp soạn thảo văn bản, Google Sheets giúp tạo lập các bảng tính, Google Slides giúp trình chiếu trực tuyến có thể thay thế cho Word, Excel hay PowerPoint của Microsoft.

*Lưu ý:* Để sử dụng được các phần mềm trực tuyến của Google, cần có tài khoản Google và truy cập trang docs.google.com, sheets.google.com, slides.google.com.

Ví dụ khi truy cập trang docs.google.com để soạn thảo văn bản, em sẽ nhận được trang màn hình, phía trên là các mẫu văn bản, phía dưới là các văn bản của em đã soạn từ trước, nếu có. Nếu muốn sửa văn bản đã có thì chỉ cần nháy chuột chọn văn bản đó, còn nếu muốn tạo mới thì nháy chuột chọn biểu tượng + (Hình 3.2). Em có thể soạn thảo tương tự như Word. Văn bản được lưu tự động trên không gian lưu trữ của em trên đám mây của Google.




*Hình 3.2. Truy cập Google Docs*

1. Em hãy nêu những ưu điểm của phần mềm chạy trên Internet.
2. Em hãy nêu tên một phần mềm trực tuyến khác với các phần mềm đã nêu trong bài.

## LUYỆN TẬP

1. Có thể nói "Phần mềm nguồn mở ngày càng phát triển thì thị trường phần mềm thương mại càng suy giảm" hay không? Tại sao?
2. Phần mềm ở các trạm ATM (rút tiền tự động) có phải là phần mềm trực tuyến không?

## VẬN DỤNG

1. Em hãy tìm trên Internet và cho biết tên một số phần mềm đồ hoạ nguồn mở và một số phần mềm đồ hoạ thương mại.
2. Nói chung, các môi trường lập trình trên ngôn ngữ Python đều không có chức năng biên dịch để chuyển mã nguồn thành mã máy. Các chương trình Python đều ở dạng mã nguồn. Liệu có thể coi mọi phần mềm viết bằng Python đều là phần mềm nguồn mở hay không?

BÀI 4

# BÊN TRONG MÁY TÍNH

SAU BÀI HỌC NÀY EM SẼ:

- Nhận diện được một số thiết bị bên trong máy tính với chức năng và các thông số đo hiệu năng của chúng.
- Nhận biết được sơ đồ của các mạch lôgic AND, OR, NOT và giải thích được vai trò của các mạch lôgic đó trong thực hiện các tính toán nhị phân.

Trong chương trình tin học ở các lớp dưới, em đã biết cấu trúc chung của máy tính bao gồm: bộ xử lí trung tâm, bộ nhớ trong, bộ nhớ ngoài, các thiết bị vào – ra. Tuy nhiên, hầu hết các em mới chỉ nhìn thấy các thiết bị bên ngoài như màn hình, bàn phím, chuột, máy chiếu, bộ nhớ ngoài (đĩa cứng rời hay thẻ nhớ USB).

Em có biết cụ thể trong máy tính có những bộ phận nào không?




*Hình 4.1. Sơ đồ cấu tạo chức năng của máy tính điện tử*

## 1. CÁC THIẾT BỊ BÊN TRONG MÁY TÍNH

**Hoạt động 1** Các thiết bị bên trong máy tính

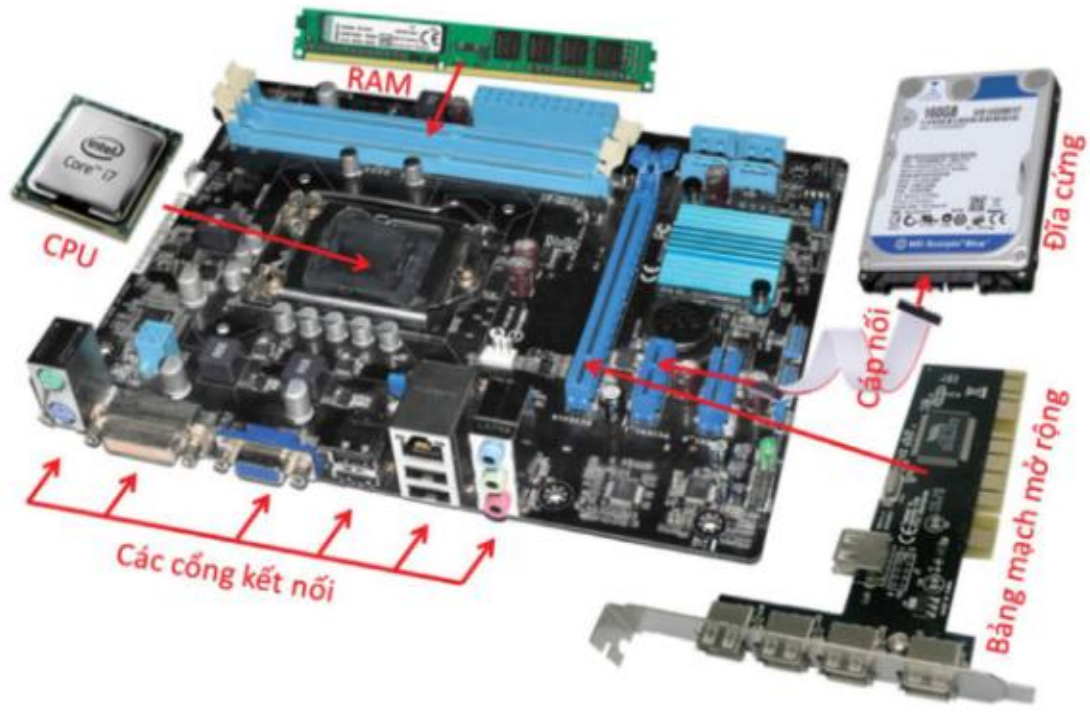
Dưới đây là một số thiết bị bên trong máy tính, em có biết chúng là các thiết bị gì không?




a) b) c) d)

*Hình 4.2. Một số thiết bị bên trong máy tính*

Tất cả các thiết bị bên trong máy tính được gắn với một bảng mạch, gọi là bảng mạch chính (mainboard) như trong Hình 4.3.




*Hình 4.3. Bên trong máy tính*

## a) Bộ xử lí trung tâm

Bộ xử lí trung tâm (Central Proccessing Unit – CPU) còn được gọi là bộ xử lí là thành phần quan trọng nhất của máy tính, đảm nhận việc thực hiện các chương trình máy tính. CPU được cấu tạo từ hai bộ phận chính:

- Bộ số học và lôgic (Arithmetic & Logic Unit, viết tắt là ALU): thực hiện tất cả các phép tính số học và lôgic trong máy tính.
- Bộ điều khiển (Control Unit): phối hợp đồng bộ các thiết bị của máy tính, đảm bảo máy tính thực hiện đúng chương trình.

CPU có một đồng hồ xung, tạo ra các xung điện áp gửi đến mọi thành phần của máy để đồng bộ các hoạt động. Mỗi phép tính sẽ được thực hiện trong một số xung đồng hồ. Vì thế người ta thường dùng tần số đồng hồ xung, thường là GHz (1 tỉ xung một giây) để đánh giá tốc độ của CPU.

Ngoài ra, CPU còn có thêm một số thành phần khác như thanh ghi (register) và bộ nhớ đệm (bộ nhớ truy cập nhanh – cache).

Thanh ghi là vùng nhớ đặc biệt được dùng để lưu trữ tạm thời các lệnh và dữ liệu đang được xử lí cho phép CPU truy cập tới với tốc độ rất nhanh. Bộ nhớ đệm là một bộ nhớ nhỏ chứa dữ liệu được nạp trước từ bộ nhớ trong nhằm giảm thời gian đọc dữ liệu.

Thời kì đầu, mỗi máy tính chỉ có một đơn vị xử lí, sau này người ta chế tạo các máy tính có nhiều đơn vị xử lí được đóng gói trong cùng một chíp. Mỗi đơn vị xử lí như thế được gọi là một lõi hoặc một nhân (core). CPU đa lõi cho phép máy tính xử lí nhanh hơn vì có thể thực hiện song song nhiều công việc.

## b) Bộ nhớ trong ROM và RAM

Tuỳ theo cách sử dụng, bộ nhớ trong (memory) chia thành hai loại RAM (Random Access Memory) và ROM (Read Only Memory).

RAM là bộ nhớ có thể ghi được, dùng để ghi dữ liệu tạm thời trong khi chạy các chương trình nhưng không giữ được lâu dài (khi tắt máy, dữ liệu trong RAM sẽ bị xoá).

ROM là bộ nhớ được ghi bằng phương tiện chuyên dùng, các chương trình ứng dụng chỉ có thể đọc mà không thể ghi hay xoá. ROM không cần nguồn nuôi nên có thể lưu dữ liệu và chương trình lâu dài. Nó thường được dùng để lưu các dữ liệu hệ thống cố định và các chương trình kiểm tra hay khởi động máy tính.

Các tham số của bộ nhớ trong thường là:

- Dung lượng của bộ nhớ (dung lượng nhớ) tính theo MB, GB, ví dụ 8 GB, 16 GB hay 32 GB.

- Thời gian truy cập trung bình của bộ nhớ là thời gian cần thiết để ghi hay đọc thông tin. Việc giảm thời gian truy cập bộ nhớ trong có ý nghĩa quan trọng để nâng cao hiệu suất tổng thể của máy tính.

So với RAM, ROM thường có dung lượng nhỏ hơn và thời gian truy cập trung bình lớn hơn.

### c) Bộ nhớ ngoài

Bộ nhớ ngoài có thể đặt bên trong hay bên ngoài thân máy. Bộ nhớ ngoài thường là đĩa từ (đĩa cứng, đĩa mềm), đĩa thể rắn (Solid State Disk – SSD) hay đĩa quang,... Bộ nhớ ngoài dùng để lưu dữ liệu lâu dài, không cần nguồn nuôi, giá thành rẻ hơn RAM và có dung lượng lớn. Các tham số đo hiệu năng của bộ nhớ ngoài cũng giống như bộ nhớ trong bao gồm:

- Dung lượng của bộ nhớ thường tính theo GB hay TB. Các bộ nhớ ngoài cỡ TB ngày nay đã rất phổ biến.

- Thời gian truy cập trung bình là thời gian cần thiết để ghi hay đọc dữ liệu. Đĩa cứng là thiết bị điện cơ nên tốc độ truy cập chậm hơn nhiều so với đĩa SSD nhưng vẫn nhanh hơn nhiều so với đĩa quang.

- Các thiết bị bên trong máy tính được gắn trên bảng mạch chính, gồm có bộ xử lí, bộ nhớ trong, bộ nhớ ngoài và có thể gắn thêm các bảng mạch mở rộng.
- Bộ xử lí là nơi thực hiện các phép toán và điều khiển toàn bộ máy tính hoạt động theo chương trình. Tốc độ của bộ xử lí đo bằng tần số xung nhịp, thường được tính theo đơn vị GHz. Bộ xử lí có thể có nhiều lõi, mỗi lõi là một đơn vị xử lí, cho phép thực hiện đồng thời nhiều nhiệm vụ.
- Bộ nhớ trong là nơi chứa dữ liệu trong khi máy hoạt động còn bộ nhớ ngoài chứa dữ liệu lưu trữ. Các thông số quan trọng nhất của bộ nhớ là dung lượng nhớ, thường được tính theo KB, MB hoặc GB và thời gian truy cập trung bình.



1. Có thể đo tốc độ của CPU bằng số phép tính thực hiện trong một giây không?
2. Giá tiền của mỗi thiết bị nhớ có phải là một thông số đo chất lượng không?

## 2. MẠCH LÔGIC VÀ VAI TRÒ CỦA MẠCH LÔGIC



CPU là thiết bị quan trọng nhất bên trong thân máy. Về bản chất, nó là một mạch điện tử được dùng để biến đổi (xử lí) các dữ liệu nhị phân. Cách thức xử lí dữ liệu của CPU được dựa trên cơ sở hoạt động của các mạch lôgic.

## a) Một số phép toán lôgic và thể hiện vật lí của chúng

Các phép toán lôgic và hệ đếm nhị phân đã được nêu trong chương trình lớp 10 đối với định hướng khoa học máy tính. Ở đây ta chỉ nêu một cách ngắn gọn các kiến thức về lôgic và số học nhị phân mà các em theo định hướng tin học ứng dụng chưa được giới thiệu.

Các đại lượng lôgic là các đại lượng chỉ nhận một trong hai giá trị lôgic là "Đúng" và "Sai", được thể hiện tương ứng bởi các bit 1 và 0.

Trên các đại lượng lôgic người ta xây dựng một số các phép toán lôgic, trong đó có phép cộng (kí hiệu là OR hoặc $\vee$), phép nhân (kí hiệu là AND hoặc $\wedge$), phép phủ định lôgic (kí hiệu là NOT hoặc một dấu gạch ngang trên đối tượng phủ định), phép hoặc loại trừ (kí hiệu là XOR hoặc $\oplus$) như trong Bảng 4.1.

**Bảng 4.1.** *Một số phép toán lôgic*

| x | y | x AND y<br>$x \wedge y$ | x OR y<br>$x \vee y$ | NOT x<br>$\overline{x}$ | x XOR y<br>$x \oplus y$ |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 |

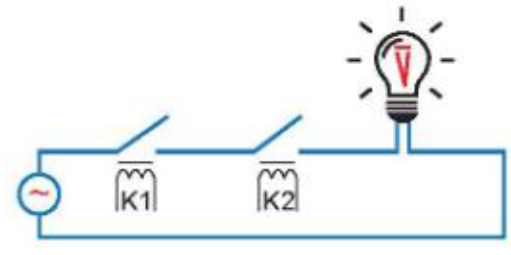
Như vậy,

- Phép nhân hai đại lượng lôgic chỉ nhận giá trị 1 khi và chỉ khi cả hai đại lượng x VÀ y đều bằng 1.
- Phép cộng hai đại lượng lôgic chỉ bằng 1 khi và chỉ khi ít nhất một trong hai đại lượng x HOẶC y bằng 1.
- Phép phủ định một đại lượng lôgic sẽ cho giá trị ngược lại. Phủ định của 0 là 1 và phủ định của 1 là 0.
- Phép hoặc loại trừ XOR (OR EXCLUSIVE) của hai đại lượng lôgic cho kết quả bằng 1 khi và chỉ khi hai đại lượng đó có giá trị khác nhau.

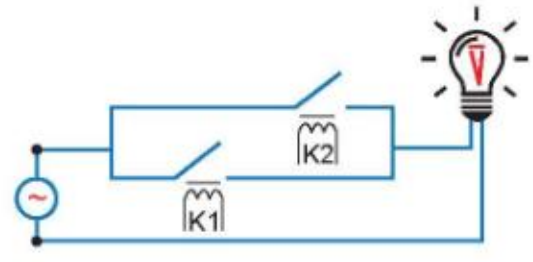
Có thể xây dựng các mạch điện hoặc điện tử để thực hiện các phép toán lôgic. Để dễ hình dung, ta minh hoạ qua các rơ le (relay) điện từ nhưng thực tế các mạch xử lí trong máy tính là mạch điện tử hoặc vi mạch có cùng tính năng.

Rơ le điện từ có một cuộn cảm ứng, khi được cấp điện chúng sẽ hút các tiếp điểm để đóng một mạch điện khác. Rơ le như thế gọi là loại thường mở, chỉ đóng mạch khi được cấp điện. Còn loại rơ le thường đóng sẽ luôn đóng mạch, khi được cấp điện rơ le sẽ tách các tiếp điểm để ngắt mạch.

**Sơ đồ mạch lôgic AND.** Hình 4.4 là mạch điện mắc nối tiếp hai rơ le K1 và K2 với một bóng điện. Quy ước trạng thái có dòng điện có giá trị lôgic 1, còn trạng thái không có dòng điện có giá

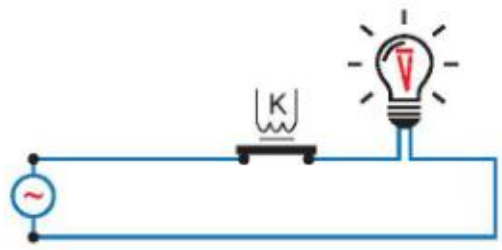



*Hình 4.4. Mạch điện thực hiện phép nhân lôgic*




*Hình 4.5. Mạch điện thực hiện phép cộng lôgic*

trị lôgic 0. Dễ thấy, chỉ khi cả hai rơ le có điện (tương ứng với giá trị lôgic 1), thì chúng mới đóng mạch và đèn mới có điện (nhận giá trị lôgic 1). Như vậy trạng thái lôgic của đèn chính là kết quả phép nhân trạng thái lôgic của K1 và K2.

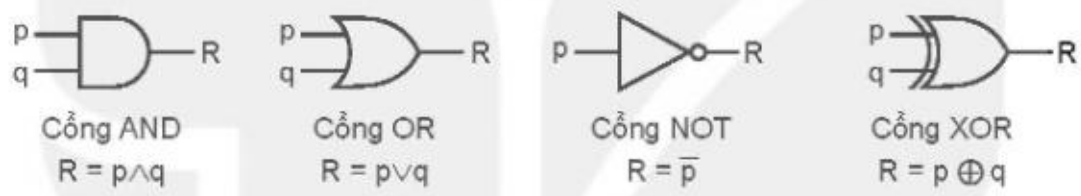



*Hình 4.6. Mạch điện thực hiện phép phủ định lôgic*

**Sơ đồ mạch lôgic OR.** Tương tự, sơ đồ mạch điện như Hình 4.5 cho phép thực hiện phép cộng lôgic. Chỉ cần ít nhất một trong hai rơ le có điện là mạch đóng, đèn sáng.

**Sơ đồ mạch lôgic NOT.** Dùng một rơ le thường đóng như Hình 4.6. Khi cấp điện cho rơ le (ứng với giá trị lôgic 1) thì nó sẽ ngắt mạch điện (cho giá trị lôgic 0) và ngược lại, khi không cấp điện cho rơ le (ứng với giá trị lôgic 0) thì mạch điện cho đèn lại đóng (cho giá trị lôgic 1).

Các mạch điện có đầu vào và đầu ra được dùng để thể hiện các giá trị lôgic được gọi chung là các mạch lôgic hay mạch số. Nói cách khác, mạch lôgic là một mạch thực hiện được các biến đổi lôgic. Những mạch lôgic thực hiện các phép toán lôgic cơ bản như AND, OR, NOT, XOR,... được gọi là các cổng lôgic (logic gate). Mỗi cổng như thế đều có một kí hiệu riêng như sau:




*Hình 4.7. Kí hiệu một số cổng lôgic cơ bản*

Tất cả các thiết bị số đều phải dùng mạch lôgic. Không có mạch lôgic sẽ không có thiết bị số trong đó có máy tính điện tử. Vì vậy mạch lôgic rất quan trọng.

## b) Phép cộng trên hệ nhị phân

Hệ nhị phân chỉ dùng hai chữ số 0, 1. Mỗi số trong hệ nhị phân được biểu diễn bằng một dãy chữ số nhị phân, ví dụ số 19 ở hệ thập phân được viết trong hệ nhị phân là 10011. Trong hệ thập phân, một chữ số ở hàng thứ k tính từ phải sang trái có giá trị được nhân với $10^{k-1}$. Trong hệ nhị phân cũng tương tự, giá trị của chữ số 1 ở hàng thứ k tính từ phải sang trái sẽ là $2^{k-1}$.

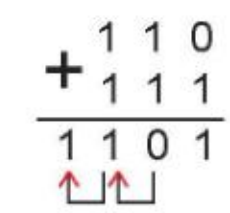
| x | y | x + y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 10 |

*Hình 4.8. Bảng cộng*

Ví dụ giá trị của số 10011 sẽ là:

$$1 \times 2^4 + 1 \times 2^1 + 1 \times 2^0 = 16 + 2 + 1 = 19$$

Trên hệ nhị phân, có thể thực hiện các phép tính số học thông thường. Để cộng các số nhị phân, phải cộng từng chữ số, có thể có nhớ sang hàng bên trái. Bảng cộng trên hệ nhị phân như trong Hình 4.8.




*Hình 4.9. Cộng hai số có nhiều chữ số*

Trong bảng cộng, ta thấy, chỉ trong trường hợp x và y đều bằng 1, phép cộng sẽ phát sinh số nhớ bằng 1. Các trường hợp khác số nhớ bằng 0.

Ví dụ phép cộng 6 với 7 trong hệ nhị phân được minh hoạ trong Hình 4.9 với hai lần có nhớ sang hàng bên trái.

## c) Minh hoạ cách dùng mạch lôgic xây dựng mạch điện thực hiện phép cộng 2 bit

**Hoạt động 2** Cộng hai bit

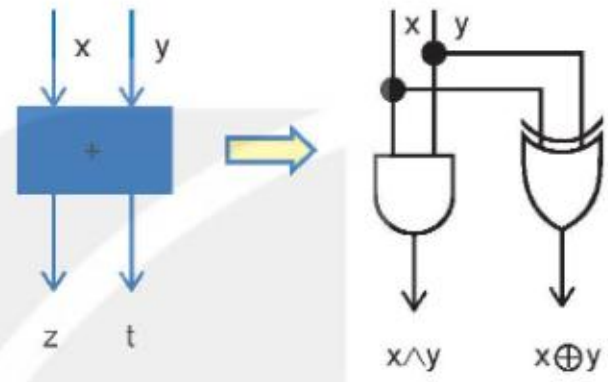
Bảng cộng trong Hình 4.8 cho thấy việc cộng hai số 1 bit có thể cho kết quả là một số 2 bit nếu phép cộng có nhớ. Khi cộng hai số nhiều bit, thì số nhớ được cộng tiếp vào hàng bên trái.

| x | y | x + y = zt | Số nhớ z | Kết quả t |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 10 | 1 | 0 |

Em hãy cho biết z và t là kết quả của phép toán lôgic nào của x và y.

Ta hình dung mạch lôgic cộng hai số 1 bit là mạch có hai đầu vào (x, y) và hai đầu ra (z, t). Có thể thấy z chính là $x \wedge y$, còn t chính là $x \oplus y$.

Như vậy mạch lôgic thực hiện phép cộng sẽ như Hình 4.10.




*Hình 4.10. Mạch lôgic thực hiện phép cộng*

Ở trên đã có sơ đồ các cổng lôgic AND, OR, NOT. Có thể chứng minh được cổng XOR cũng như mọi cổng lôgic đều có thể tổng hợp được từ các cổng AND, OR, NOT. Nói cách khác, bất cứ mạch lôgic nào cũng có thể xây dựng được từ các cổng AND, OR, NOT.

- Mạch lôgic hay mạch số là các mạch điện hay điện tử có đầu vào và đầu ra thể hiện các giá trị lôgic. Mọi mạch lôgic đều có thể xây dựng từ các cổng AND, OR và NOT.
- Tất cả các thiết bị số, gồm cả máy tính đều được chế tạo từ các mạch lôgic.

1. Thế nào là một mạch lôgic?
2. Nêu tầm quan trọng của mạch lôgic.

## LUYỆN TẬP

1. Trong các thiết bị của máy tính, thiết bị nào có ảnh hưởng đến tốc độ xử lí của máy tính? Tại sao?
2. Thực hiện những phép cộng các số nhị phân nhiều chữ số sau đây rồi chuyển các số sang hệ thập phân. Ví dụ 111 + 110 = 1101, chuyển thành 7 + 6 = 13.

   a) 1010 + 101.

   b) 1001 + 1011.

## VẬN DỤNG

Có một chỉ số đo hiệu quả của máy tính là flops (floating operation per second). Hãy tìm hiểu flops là gì và tại sao lại ít dùng với máy tính cá nhân.

# BÀI 5 KẾT NỐI MÁY TÍNH VỚI CÁC THIẾT BỊ SỐ

SAU BÀI HỌC NÀY EM SẼ:

- Biết một số thông số cơ bản của thiết bị vào – ra thông dụng, cách kết nối với máy tính cũng như tuỳ chỉnh được một vài chức năng cơ bản để nâng cao hiệu quả và đáp ứng nhu cầu sử dụng.
- Đọc hiểu và giải thích được một số thông số cơ bản của các thiết bị số thông dụng trong các tài liệu để kết nối chúng với máy tính.

Các thiết bị số có khả năng trao đổi dữ liệu với máy tính rất đa dạng. Trong số đó, một số thiết bị không có khả năng xử lí thông tin độc lập, chúng chỉ làm việc khi được kết nối với máy tính. Việc kết nối máy tính với các thiết bị số được thực hiện như thế nào, cần phải tuỳ chỉnh gì khi kết nối chúng với nhau?

## 1. MỘT SỐ THIẾT BỊ VÀO – RA THÔNG DỤNG

**Hoạt động 1** **Các thiết bị số có thể kết nối với máy tính**

Hãy kể ra một số các thiết bị có thể kết nối với máy tính và nêu chức năng của nó.

Các thiết bị vào – ra là nhóm các thiết bị ngoại vi đa dạng và phong phú nhất của máy tính. Các *thiết bị vào* cho phép nhập dữ liệu vào máy tính như bàn phím, chuột, máy đọc mã vạch, máy quét ảnh, camera số; các *thiết bị ra* chuyển thông tin từ máy tính ra ngoài như màn hình, máy in, máy chiếu. Một số thiết bị có khả năng trao đổi thông tin hai chiều với máy tính có thể được coi vừa là thiết bị vào, vừa là thiết bị ra, ví dụ các bộ nhớ ngoài kết nối theo cổng USB.

### a) Một số thiết bị vào thông dụng

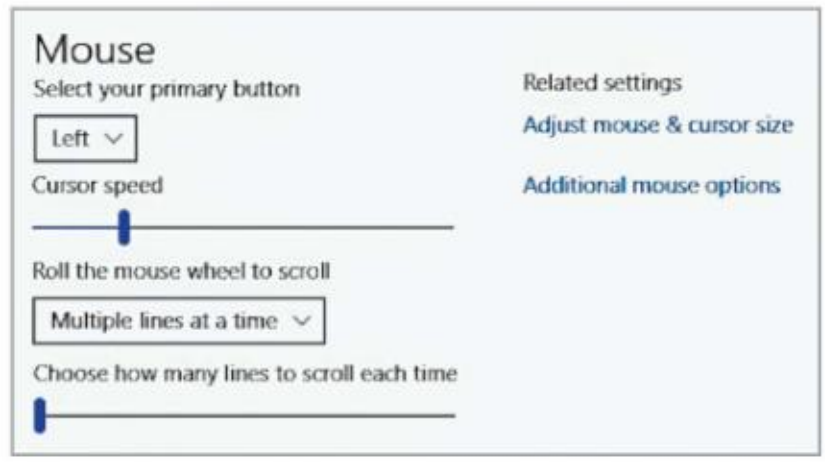
• **Bàn phím** là thiết bị thông dụng nhất để nhập dữ liệu. Bàn phím rời có thể kết nối không dây hoặc có dây với máy tính bằng cáp. Nếu nối bằng cáp thì chỉ cần cắm đúng cổng, không phải thiết lập gì thêm.

• **Chuột:** Trong các thiết bị chỉ định đối tượng làm việc trên màn hình (pointing device) có loại luôn gắn với máy tính như bàn cảm ứng (trackpad), bi lăn (trackball), có loại được kết nối với máy tính từ bên ngoài như chuột.

Chuột là thiết bị vào rất phổ biến vì dễ điều khiển chính xác. Hai thông số quan trọng nhất của chuột là phương thức kết nối (có dây hoặc không dây) và độ phân giải, đo bằng dpi (dots per inch), là số điểm riêng rẽ mà chuột xác định được khi dịch chuyển được một inch (2,54 cm).

Tuy nhiên khi sử dụng thì tốc độ của chuột lại rất quan trọng. Tốc độ thể hiện tỉ lệ giữa khoảng cách con trỏ màn hình di chuyển được so với khoảng cách di chuyển của chuột trên mặt bàn. Khi chỉnh sửa ảnh cần thực hiện những di chuyển rất nhỏ, do vậy sẽ khó làm việc nếu tốc độ chuột quá cao. Vì thế, cần biết tuỳ chỉnh chế độ làm việc của chuột.

Để tuỳ chỉnh chế độ làm việc của chuột, em cần mở tiện ích Setting, chọn nhóm chức năng Devices sau đó chọn Mouse để làm xuất hiện cửa sổ tuỳ chỉnh chuột như Hình 5.1. Ví dụ để tăng/giảm tốc độ của chuột chỉ cần kéo con trượt trên thanh "Cursor speed" sang phải/trái; chọn lại nút ưu tiên là phải (right) thay vì trái (left) dưới vị trí "Select your primary button" cho người thuận tay trái. Ngoài ra còn nhiều tuỳ chỉnh khác như chọn hình dáng của chuột, chọn để lại hay không để lại vết di chuyển trên màn hình của chuột,...




*Hình 5.1. Tuỳ chỉnh chuột máy tính*

## b) Thiết bị ra

• **Màn hình** là thiết bị ra phổ biến nhất.

Có nhiều loại màn hình sử dụng các công nghệ khác nhau như đèn chân không (CRT), tinh thể lỏng (LCD), LED hay plasma. Nhưng dù dùng công nghệ gì thì chúng đều có một số thông số chung như sau:



*Hình 5.2. Màn hình máy tính*

– *Kích thước:* được đo bằng độ dài đường chéo màn hình, tính theo inch.

– *Độ phân giải:* thể hiện bởi số điểm ảnh theo chiều ngang và chiều dọc của màn hình. Ví dụ, độ phân giải VGA: 640 × 480 pixel, độ phân giải Full HD: 1920 × 1080 pixel. Số điểm ảnh càng lớn thì màn hình càng nét.

– *Khả năng thể hiện màu:* loại đơn sắc (monochrome) chỉ có hai màu, còn loại màu 24 bit có thể thể hiện được khoảng 16,7 triệu sắc màu khác nhau.

– *Tần số quét:* hình ảnh trên màn hình được tạo lại liên tục. Tần số quét là số lần hiển thị lại hình ảnh trong một giây. Khi tần số quét cao, thời gian tái hiện hình ảnh ngắn hơn thời gian lưu ảnh trên võng mạc, ảnh sẽ không bị giật, đỡ mỏi mắt. Tần số quét thường là 50 Hz, 60 Hz, 75 Hz hay 100 Hz.

– *Thời gian phản hồi:* là khoảng thời gian cần thiết để có thể đổi màu một điểm ảnh. Những màn hình có chất lượng tốt có thể có thời gian phản hồi là 1 ms.

Có thể tuỳ chỉnh màn hình, chủ yếu là độ sáng. Đối với màn hình rời, việc tuỳ chỉnh thực hiện qua các nút trên màn hình. Đối với máy tính xách tay, có thể chỉnh trên bàn phím, ví dụ phím F11 để giảm sáng và F12 để tăng sáng.

• **Máy in:** Có nhiều loại máy in.

– *Máy in kim* sử dụng một hàng kim gõ vào băng mực để lại vết trên giấy. Các chữ hay hình vẽ được tạo thành từ một lưới các điểm nên máy in kim còn gọi là máy in ma trận (matrix printer) hay máy in điểm (dot printer). Tuy chất lượng ảnh rất thấp nhưng chưa có phương pháp nào tốt hơn khi phải in hoá đơn có nhiều liên.

– *Máy in laser* dùng tia laser để tạo ra sự thay đổi điện áp trên bề mặt của một trống tĩnh điện tương tự với hình cần in. Sau đó trống tĩnh điện sẽ hút các hạt mực siêu nhỏ rồi áp vào mặt giấy, đốt nóng để tạo thành bản in. Máy in laser được dùng phổ biến để in tài liệu.

– *Máy in phun* sẽ phun các hạt mực màu nước siêu nhỏ để tạo ảnh. Máy in loại này thích hợp để in ảnh màu, phông bạt quảng cáo với kích thước đa dạng và chi phí thấp.

– *Máy in nhiệt* in trên giấy cảm ứng nhiệt, vùng giấy bị làm nóng đổi sang màu đen. Máy in nhiệt thường dùng để in hoá đơn bán hàng.

Dù đa dạng nhưng các máy in đều có những thông số chung như sau:

– *Độ phân giải* tính bằng dpi là số điểm ảnh trên một inch theo cả hai chiều ngang/dọc tương tự như cách tạo ảnh trên màn hình. Máy in laser hay in phun có thể đạt được từ 300 dpi đến 600 dpi hoặc hơn, trong khi đó máy in kim hay in nhiệt có độ phân giải thấp hơn nhiều.

– *Kích thước giấy* có thể in được. Máy in khổ giấy A4 phổ biến nhất. Các máy in phông, bạt có thể in được khổ lớn đến vài mét.



*a) Máy in kim* *b) Máy in laser*

*c) Máy in phun* *d) Máy in nhiệt*

*Hình 5.3. Một số loại máy in*

– *Tốc độ in* thường tính theo số trang in được trong một phút. Các máy in laser có thể in vài chục trang một phút, máy in phun có thể mất vài phút cho một trang.

– *Khả năng in màu.*

– *Cách kết nối với máy tính,...*

**Kết nối máy in với máy tính:** Một số loại máy in khi kết nối với máy tính, trước lần sử dụng đầu tiên đòi hỏi phải cài đặt trình điều khiển (driver) do hãng sản xuất máy in cung cấp. Nếu máy in kết nối với máy tính bằng cáp thì cần chọn cáp đúng loại. Nếu kết nối máy in với máy tính qua mạng (có dây hay không dây) thì có thể còn phải tuỳ chỉnh các tham số kết nối mạng.

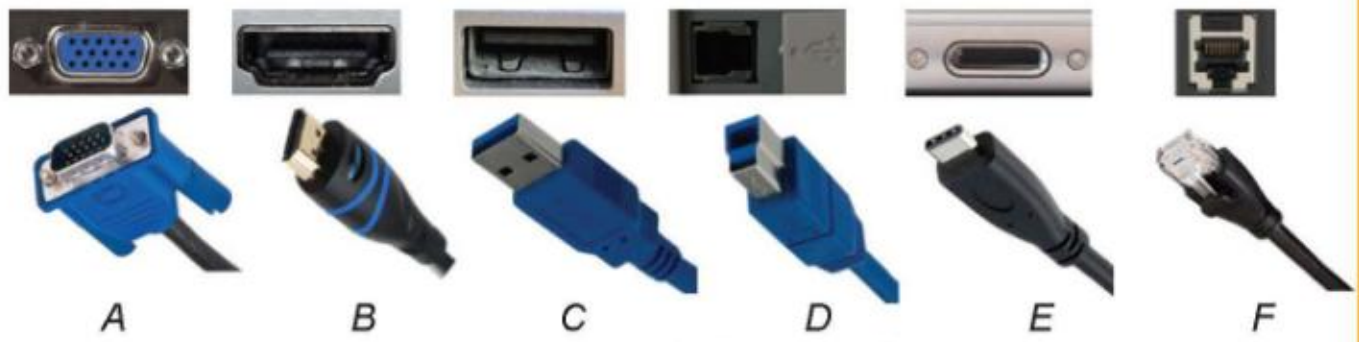
- Máy tính có thể kết nối được với nhiều thiết bị số, trong đó có các thiết bị vào – ra, thường chỉ làm việc được khi kết nối với máy tính.
- Mỗi thiết bị vào – ra đều có những thông số đặc trưng riêng. Việc hiểu các thông số này giúp ta lựa chọn thiết bị phù hợp. Trong trường hợp cần thiết có thể phải tuỳ chỉnh chức năng của thiết bị theo nhu cầu sử dụng để đạt hiệu quả tốt hơn.

1. Nêu và giải thích ý nghĩa các thông số của màn hình.
2. Nêu và giải thích ý nghĩa các thông số của máy in.

## 2. KẾT NỐI MÁY TÍNH VỚI THIẾT BỊ SỐ

**Hoạt động 2** **Cổng kết nối của máy tính**

Xem Hình 5.4 rồi cho biết tên các cổng kết nối của máy tính.



*A* *B* *C* *D* *E* *F*

*Hình 5.4. Một số cổng kết nối của máy tính*

## a) Các cổng kết nối

• Các cổng A, B dùng để kết nối với thiết bị ra như màn hình hoặc máy chiếu có tên lần lượt là cổng VGA và cổng HDMI.

– Cổng VGA không dùng tín hiệu số mà dùng tín hiệu tương tự để truyền hình ảnh tới màn hình hoặc máy chiếu, không truyền âm thanh.

– Cổng HDMI truyền tín hiệu số, truyền đồng thời cả âm thanh và hình ảnh với băng thông lớn nên có thể truyền video với độ phân giải cao.

• Các cổng C, D, E đều thuộc họ cổng USB (Universal Serial Bus). Đặc điểm chung của phương thức truyền dữ liệu USB là truyền tuần tự, đa năng. Với tốc độ được cải thiện nhanh, USB đã thay thế và giúp loại bỏ nhiều cổng khác được dùng trước đây như cổng COM, cổng Parallel dùng cho máy in, cổng PS/2 cho bàn phím và chuột.

• Cổng F là cổng mạng. Nhiều thiết bị có thể kết nối với máy tính qua mạng.

## b) Kết nối máy tính với thiết bị số

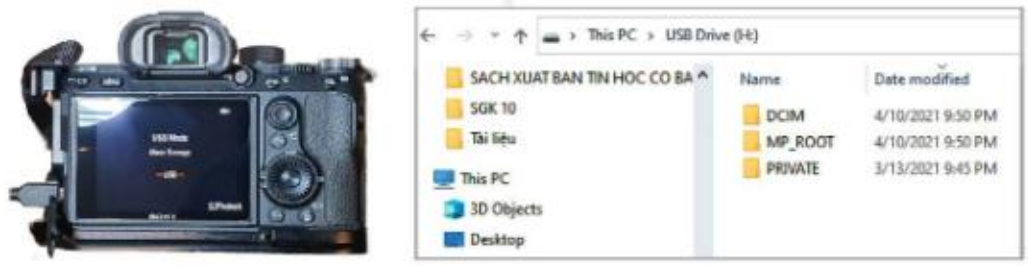
Không có một phương thức kết nối chung nào cho các thiết bị số với máy tính. Nói chung, các thiết bị số được cung cấp kèm theo tài liệu kĩ thuật, nêu rõ các đặc tính của thiết bị, cách sử dụng và cách liên lạc với nhà sản xuất để được hỗ trợ. Vì vậy, cần đọc kĩ các tài liệu kĩ thuật hoặc nhờ các chuyên gia tư vấn để biết cách sử dụng tốt nhất.

Nhờ cơ chế plug & play nên việc kết nối máy tính với các thiết bị số thông qua các cổng hay mạng ngày nay dễ dàng hơn nhiều. Trong nhiều trường hợp, chỉ cần cắm cáp kết nối là có thể sử dụng thiết bị ngay, ví dụ chuột hay bộ nhớ ngoài (USB, ổ đĩa rời,...). Trong một số trường hợp khác, cần thiết lập các tham số kết nối.

**Ví dụ 1.** Kết nối máy tính với máy ảnh

Máy ảnh số có thể hỗ trợ một số cách kết nối khác nhau với máy tính: qua cáp USB, wifi hoặc bluetooth.

Hình 5.5 minh hoạ việc kết nối máy tính với máy ảnh qua cáp nối USB. Khi cắm cáp nối với máy tính, máy ảnh sẽ hiện thông báo "USB Mode, Mass Storage" với ý nghĩa bộ nhớ được chia sẻ qua cổng USB.




*Hình 5.5. Kết nối máy ảnh với máy tính để lấy ảnh*

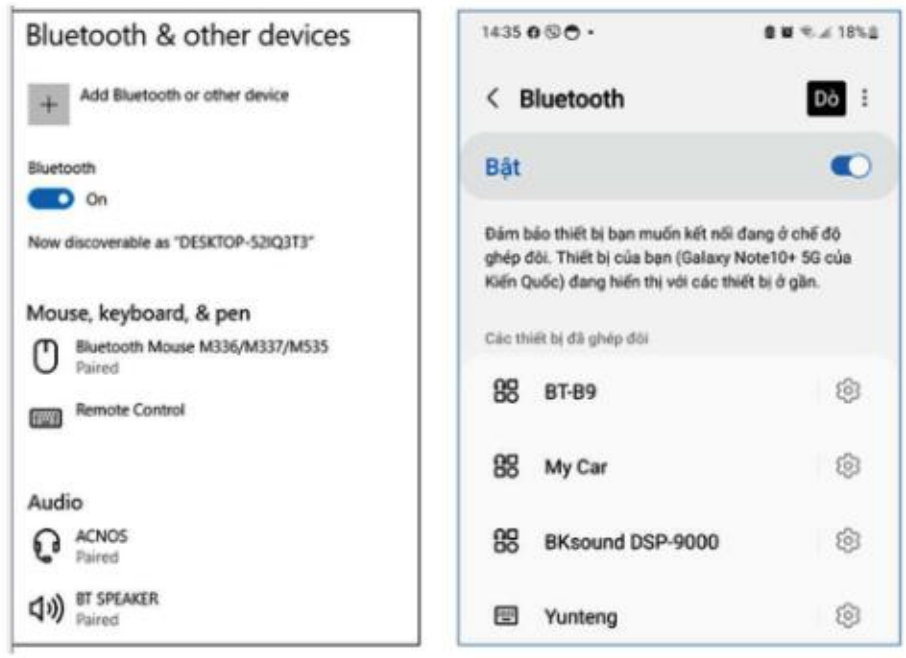
Khi đó có thể truy cập thẻ nhớ của máy ảnh như một bộ nhớ ngoài. Dùng tiện ích File Explorer, ta có thể mở thư mục DCIM, nơi chứa các tệp ảnh để xem, xoá, sao chép,... các tệp ảnh trong máy ảnh số (Hình 5.5).

**Ví dụ 2.** Kết nối máy tính với thiết bị số qua bluetooth

Bluetooth là phương thức kết nối không dây trong khoảng cách gần (dưới 10 m) để truyền tệp, thu phát âm thanh hay video hoặc điều khiển thiết bị.

Tại mỗi thời điểm, mỗi thiết bị chỉ có thể kết nối qua bluetooth với một thiết bị khác.

Trước khi kết nối với một thiết bị nào, máy tính hay điện thoại thông minh cần phải ghép đôi (pairing) với thiết bị đó để trao đổi tham số kết nối. Đối với máy tính cá nhân cần chạy tiện ích Setting sau đó chọn "Bluetooth & others devices"; đối với điện thoại thông minh cần chạy tiện ích Setting và chọn kết nối với bluetooth. Giao diện làm việc với bluetooth trên máy tính cá nhân và điện thoại thông minh được minh hoạ trong Hình 5.6. Những thiết bị bluetooth nào đã ghép đôi từ trước được hiển thị sẵn.




*Hình 5.6. Giao diện ghép đôi và kết nối với bluetooth trên PC và trên điện thoại thông minh*

Nếu muốn kết nối với thiết bị đã được ghép đôi sẵn, chỉ cần bật bluetooth của thiết bị đó rồi chọn tên thiết bị và đợi thông báo kết nối thành công.

Để làm việc với một thiết bị mới, ta cần ghép đôi theo các bước sau:

– Bật chức năng bluetooth của thiết bị rồi tìm thiết bị bằng cách chọn "+ Add Bluetooth or other device" hoặc "Dò".

– Khi thấy hiện tên thiết bị bluetooth muốn ghép đôi, hãy chọn. Một số thiết bị bluetooth có thể gửi một mã xác thực, yêu cầu gõ lại mã đó rồi mới thực hiện ghép đôi.

Để kết nối một thiết bị số với máy tính, cần tìm hiểu tài liệu kĩ thuật để nắm được các thông số và cách kết nối.

1. Cách kết nối thiết bị số với máy tính có phụ thuộc vào loại thiết bị không?
2. Em hiểu thế nào về tham số kết nối?

## LUYỆN TẬP

1. Thực hiện kết nối máy tính với một điện thoại thông minh qua cổng USB để lấy ảnh từ điện thoại về máy tính tương tự như Ví dụ 1.
2. Thực hiện kết nối máy tính hay điện thoại di động với một tai nghe hay một loa bluetooth theo Ví dụ 2. Sau đó hãy bật nhạc từ máy tính hay điện thoại để trải nghiệm âm nhạc phát tới thiết bị bluetooth.

## VẬN DỤNG

1. Tìm hiểu máy quét ảnh theo các gợi ý sau: Máy quét ảnh là thiết bị ra hay vào? Mô tả chức năng. Những công nghệ khác nhau để chế tạo máy quét ảnh. Các thông số của máy quét ảnh là gì?
2. Tìm hiểu máy chiếu theo các gợi ý sau: Máy chiếu là thiết bị ra hay vào? Mô tả chức năng. Những công nghệ khác nhau để chế tạo máy chiếu. Các thông số của máy chiếu là gì?

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

Chủ đề 2

# Tổ chức lưu trữ, tìm kiếm và trao đổi thông tin

BÀI 6

## LƯU TRỮ VÀ CHIA SẺ TỆP TIN TRÊN INTERNET

SAU BÀI HỌC NÀY EM SẼ:

- Sử dụng được một số công cụ trực tuyến như Google Drive hay Dropbox,... để lưu trữ và chia sẻ tệp tin.

Nhóm em đang cùng nhau làm một bài tập nhóm. Em được giao lập kế hoạch và phân công công việc cho cả nhóm. Thông tin về kế hoạch và phân công được lưu trong một tệp văn bản. Có những cách nào để chia sẻ văn bản này cho các thành viên trong nhóm?

### 1. LƯU TRỮ VÀ CHIA SẺ TỆP TIN TRÊN INTERNET – Ổ ĐĨA TRỰC TUYẾN

Cùng với sự phát triển của công nghệ thông tin nói chung và Internet nói riêng, nhu cầu ứng dụng công nghệ thông tin trong tất cả các lĩnh vực học tập, khoa học, đời sống, kinh tế, xã hội,... ngày càng lớn. Do đó lượng thông tin, dữ liệu được tạo ra cũng tăng lên gấp bội.

Nếu như trước đây, việc lưu trữ thông tin, dữ liệu bị hạn chế bởi các tài nguyên máy tính (dung lượng đĩa cứng, dung lượng thiết bị nhớ ngoài,...) thì ngày nay, Internet đã giúp khắc phục hoàn toàn những rào cản đó bằng sự ra đời của các dịch vụ lưu trữ và chia sẻ tệp tin trên Internet.

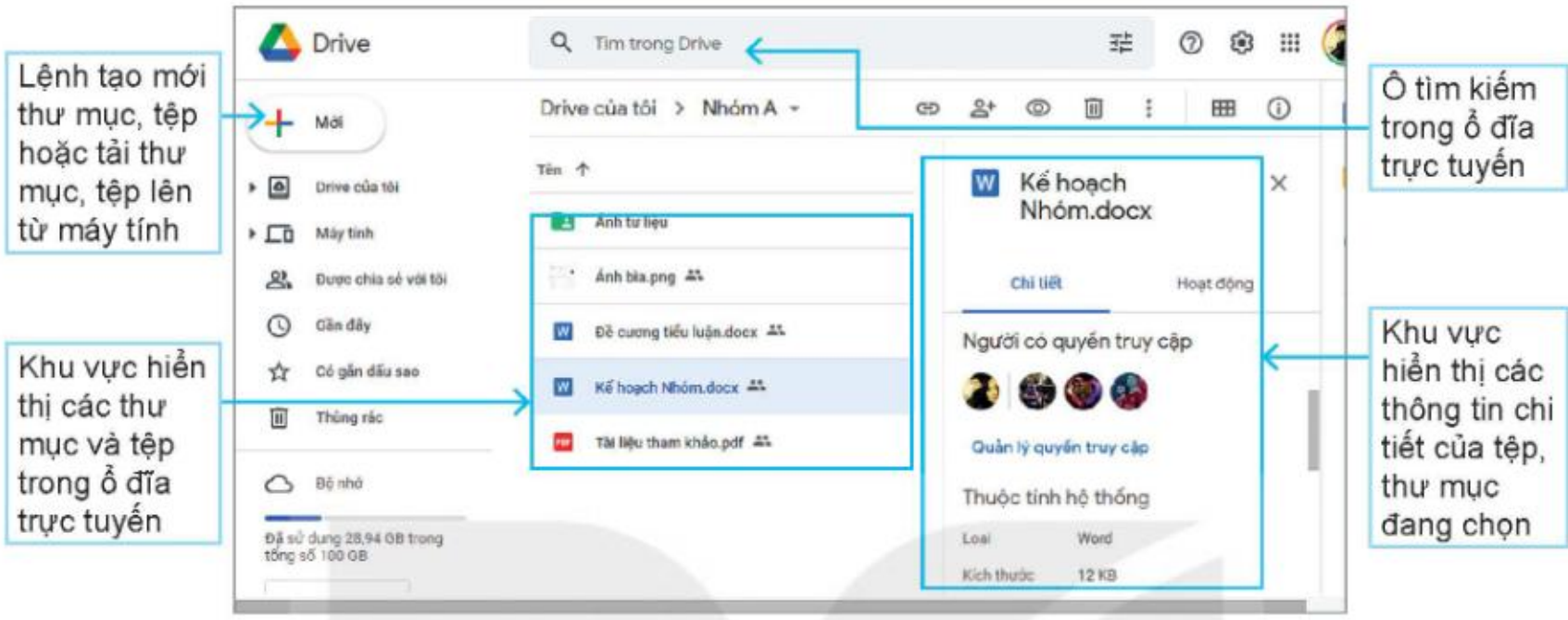
**Hoạt động 1** **Khám phá dịch vụ lưu trữ và chia sẻ tệp tin trên Internet**

Hình 6.1 minh hoạ tính năng cơ bản của một dịch vụ lưu trữ và chia sẻ tệp tin trên Internet.
Các em hãy quan sát, thảo luận nhóm và đưa ra mô tả các tính năng đó. Từ đó cho biết tại sao dịch vụ lưu trữ và chia sẻ tệp tin trên Internet còn được gọi là dịch vụ "Lưu trữ đám mây"?




*Hình 6.1. Minh hoạ dịch vụ lưu trữ và chia sẻ tệp tin trên Internet*

Có nhiều nhà cung cấp các dịch vụ này, chẳng hạn Google với Google Drive, Microsoft với One Drive, Apple với iCloud, Dropbox với ứng dụng cùng tên. Khi đăng kí sử dụng dịch vụ, người dùng sẽ được cấp một không gian nhớ trực tuyến gọi là "ổ đĩa trực tuyến" để lưu trữ các tệp cũng như thư mục của mình (Hình 6.2) với những tính năng cơ bản sau:




*Hình 6.2. Giao diện làm việc của một ổ đĩa trực tuyến thông dụng*

## a) Tải tệp lên ổ đĩa trực tuyến

Cho phép tải các tệp hay thư mục từ máy tính của mình lên ổ đĩa trực tuyến để lưu trữ và sử dụng.

## b) Tạo mới và quản lí thư mục, tệp trên ổ đĩa trực tuyến

Cho phép tạo mới thư mục, tệp và quản lí, sắp xếp chúng trên ổ đĩa trực tuyến. Một số nhà cung cấp dịch vụ còn cung cấp tính năng cho phép mở và chỉnh sửa trực tuyến các tệp được tạo ra bởi các ứng dụng văn phòng (phần mềm soạn thảo văn bản, phần mềm bảng tính, phần mềm trình chiếu,...).

## c) Chia sẻ thư mục và tệp

Một trong những ưu điểm nổi bật của việc lưu trữ thông tin, dữ liệu trực tuyến là cho phép người dùng có thể chia sẻ thư mục, tệp cho những người dùng khác. Khi chia sẻ, chủ sở hữu của thư mục, tệp quyết định một trong ba chế độ chia sẻ (quyền của người được chia sẻ đối với thư mục, tệp đó) như sau: (1) quyền chỉ xem; (2) quyền được nhận xét, tức là được xem và nhận xét; và (3) quyền chỉnh sửa. Quyền chỉnh sửa là quyền cao nhất cho phép xem, nhận xét và thực hiện mọi thao tác với thư mục và tệp (chỉnh sửa, xoá, tạo mới, đổi tên,...).

Sau khi chia sẻ thư mục, tệp, người dùng có thể huỷ bỏ việc chia sẻ này hoặc thay đổi chế độ chia sẻ.

- Sử dụng dịch vụ lưu trữ thư mục và tệp trực tuyến, người dùng sẽ được cung cấp một ổ đĩa trực tuyến.
- Trên ổ đĩa trực tuyến, người dùng có thể thực hiện tải tệp, thư mục lên để lưu trữ, tạo mới, chia sẻ tệp, thư mục,... và các tính năng hữu ích khác.

Thảo luận nhóm để chỉ ra một vài ưu điểm và nhược điểm của việc lưu trữ và chia sẻ tệp trên Internet.

## 2. THỰC HÀNH: LƯU TRỮ VÀ CHIA SẺ TỆP TIN TRÊN Ổ ĐĨA TRỰC TUYẾN

*(Hướng dẫn sau đây sử dụng Google Drive để minh hoạ)*

### Nhiệm vụ 1. Lưu trữ tệp tin trên ổ đĩa trực tuyến

**Hướng dẫn:**

Trước tiên, cần đảm bảo rằng mọi thành viên trong nhóm đã đăng kí tài khoản với nhà cung cấp dịch vụ. Chẳng hạn, tài khoản Google khi muốn sử dụng Google Drive.

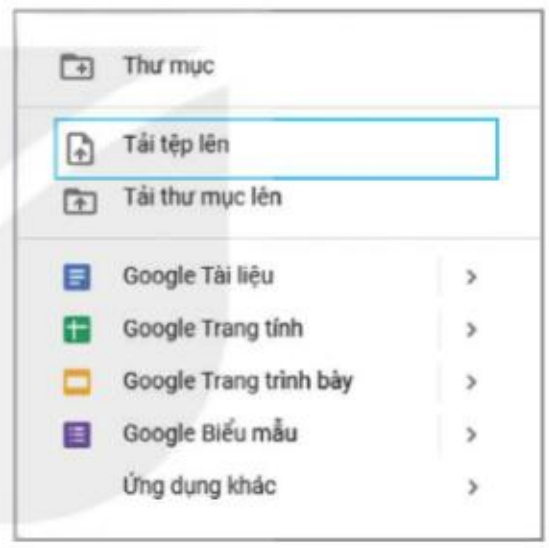
*Bước 1.* Mở dịch vụ Google Drive tại địa chỉ http://drive.google.com

*Bước 2.* Đăng nhập bằng tài khoản Google của em để truy cập ổ đĩa trực tuyến được cung cấp bởi Google.

*Bước 3.* Tải tệp từ máy tính lên lưu trữ tại ổ đĩa trực tuyến bằng cách:

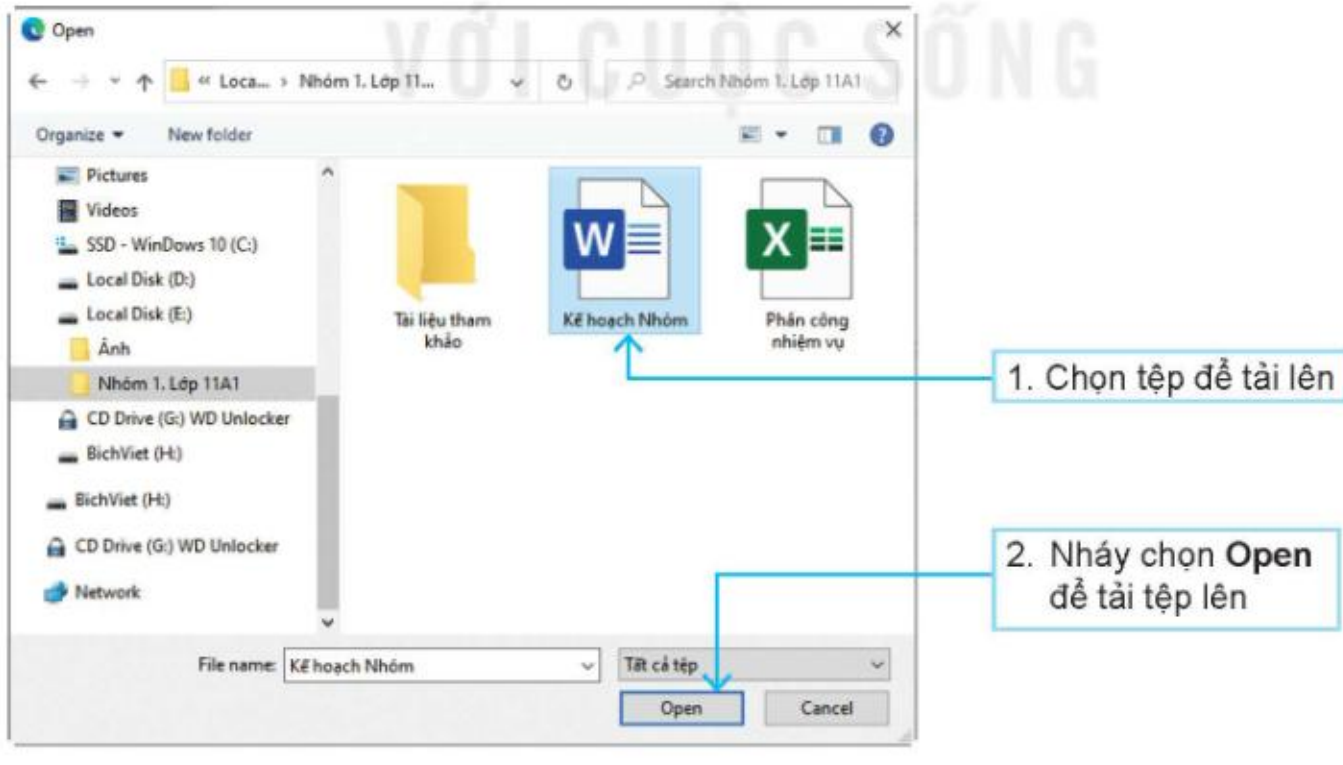
– Nháy chuột vào lệnh + Mới ở góc trên bên trái cửa sổ Google Drive (Hình 6.2).

– Trong bảng chọn hiện ra, chọn lệnh **Tải tệp lên** (Hình 6.3).




*Hình 6.3. Bảng lệnh hiện ra khi nháy chuột vào lệnh + Mới*

– Trong hộp thoại **Open**, thực hiện các bước như minh hoạ trong Hình 6.4.




*Hình 6.4. Hộp thoại **Open***

– Tệp tải lên sẽ xuất hiện trên ổ đĩa trực tuyến (Hình 6.2).

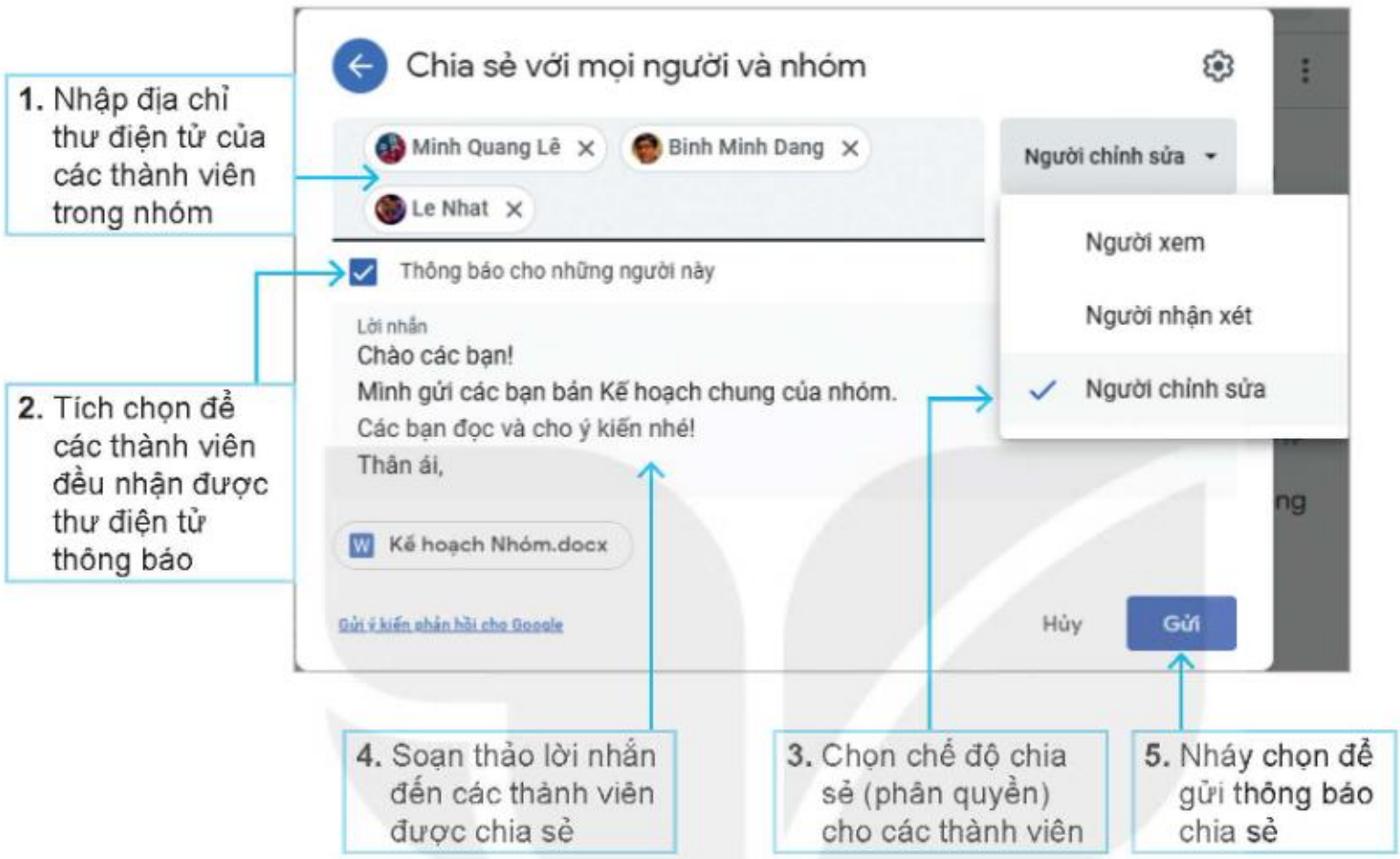
KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

## Nhiệm vụ 2. Chia sẻ tệp tin cho các thành viên trong nhóm

### Hướng dẫn:

*Bước 1.* Nháy nút phải chuột vào tệp tin cần chia sẻ trên ổ đĩa trực tuyến, bảng lệnh hiện ra, chọn Chia sẻ.

*Bước 2.* Thực hiện các bước như minh hoạ trong Hình 6.5 để chia sẻ tệp tin.




*Hình 6.5. Hộp thoại chia sẻ tệp*

*Bước 3.* Kiểm tra thông tin chia sẻ của tệp tại khu vực hiển thị các thông tin chi tiết của tệp, thư mục đang được chọn (Hình 6.2).

## LUYỆN TẬP

1. Thực hành tải thêm các tệp từ máy tính lên ổ đĩa trực tuyến.
2. Chia sẻ một vài tệp cho các bạn trong nhóm với chế độ chia sẻ khác nhau và kiểm tra sự khác biệt giữa các chế độ chia sẻ đó.

## VẬN DỤNG

1. Em hãy sử dụng thêm dịch vụ lưu trữ trực tuyến của ít nhất hai nhà cung cấp khác và đưa ra các ý kiến đánh giá, so sánh về: Dung lượng miễn phí mà người dùng được sử dụng; Có cho phép tải lên, tải xuống hay chia sẻ với người dùng khác hay không? Có thân thiện, dễ sử dụng hay không? Đánh giá của em cũng như người khác về dịch vụ đó như thế nào?
2. Khám phá, sử dụng các chức năng khác của ổ đĩa trực tuyến. Chia sẻ, thảo luận với bạn bè các khám phá của em.

BÀI 7

# THỰC HÀNH TÌM KIẾM THÔNG TIN TRÊN INTERNET

SAU BÀI HỌC NÀY EM SẼ:

- Sử dụng được máy tìm kiếm trên PC và thiết bị số thông minh bằng cách gõ từ khoá hoặc bằng tiếng nói.
- Xác lập được các lựa chọn theo tiêu chí tìm kiếm để nâng cao hiệu quả tìm kiếm thông tin.

Có ý kiến cho rằng "Ngày nay, tất cả mọi thông tin đều có thể tìm thấy trên Internet". Em có đồng ý với ý kiến đó không? Theo em, tại sao khi cùng tìm thông tin về một vấn đề, nhưng có người sẽ tìm được rất nhanh và chính xác, có người thì không?

## Nhiệm vụ 1. Tìm kiếm thông tin trên Internet bằng máy tìm kiếm

**Hướng dẫn:**

*Bước 1.* Khởi động công cụ tìm kiếm.

– Mở trình duyệt Internet trên máy tính của em.

– Gõ địa chỉ URL máy tìm kiếm, chẳng hạn Google.com.

*Bước 2.* Tìm kiếm bằng từ khoá nhập từ bàn phím.

Nhập từ khoá bằng bàn phím (chẳng hạn "tuyển sinh Đại học Dược Hà Nội 2022") rồi nhấn phím **Enter**.

*Bước 3.* Đọc kết quả tìm kiếm. Nếu kết quả tìm kiếm chưa được như ý muốn của em, quay lại bước 2 với từ khoá khác để việc tìm kiếm hiệu quả hơn.

## Nhiệm vụ 2. Khám phá cách thực hiện tìm kiếm bằng tiếng nói

**Hướng dẫn:**

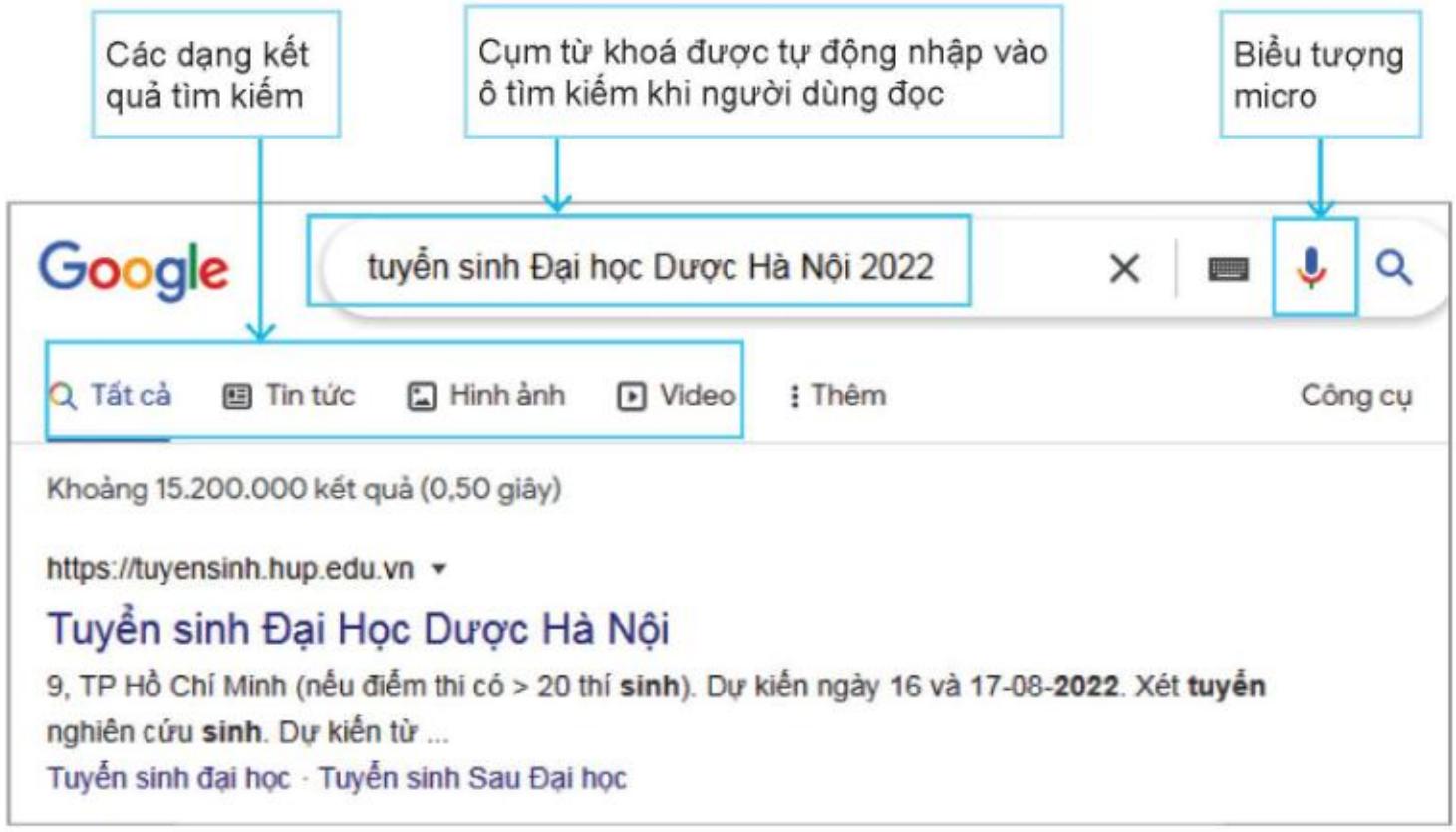
*Bước 1.* Khởi động công cụ tìm kiếm.

*Bước 2.* Tìm kiếm bằng từ khoá nhập bằng tiếng nói. Nháy chuột vào biểu tượng micro cạnh ô nhập từ khoá tìm kiếm (Hình 7.1), sau đó, đọc từ khoá tìm kiếm, sau khi dừng đọc, máy tìm kiếm sẽ hiển thị kết quả tìm kiếm như Hình 7.1.

*Bước 3.* Kiểm tra từ khoá được tự động điền sau khi đọc tại Bước 2 (Hình 7.1). Nếu không khớp, thực hiện lại Bước 2 để đọc lại từ khoá.

*Bước 4.* Đọc kết quả tìm kiếm. Nếu kết quả tìm kiếm chưa được như ý muốn của em, quay lại Bước 2 với từ khoá khác để việc tìm kiếm hiệu quả hơn.

*Lưu ý:* Cần có micro để thực hiện tìm kiếm bằng tiếng nói.




*Hình 7.1. Minh hoạ giao diện và kết quả tìm kiếm*

## Nhiệm vụ 3. Xác lập được các lựa chọn theo tiêu chí tìm kiếm để nâng cao hiệu quả tìm kiếm thông tin

Có nhiều tiêu chí giúp nâng cao hiệu quả tìm kiếm thông tin. Có thể tìm kiếm thông tin dạng hình ảnh, tin tức hay video bằng cách chọn dạng phân loại kết quả tìm kiếm tương ứng như Hình 7.1, hoặc quy định dạng tệp chứa thông tin cần tìm. Chẳng hạn, khi tìm kiếm thông tin về thông báo của một tổ chức, đơn vị nào đó (ví dụ thông báo tuyển sinh của một trường Đại học), để nâng cao tính chính xác và hiệu quả tìm kiếm, người dùng có thể yêu cầu tìm dưới dạng tệp tin .pdf để xem được nội dung thông báo đó bằng văn bản.

**Hướng dẫn:**

*Bước 1*. Khởi động công cụ tìm kiếm.

*Bước 2*. Nhập từ khoá cần tìm và bổ sung cụm từ "filetype:pdf" (ví dụ như Hình 7.2).




*Hình 7.2. Ví dụ tìm kiếm thông tin dưới dạng tệp .pdf*

*Bước 3*. Đọc kết quả tìm kiếm. Nếu kết quả tìm kiếm chưa được như ý muốn của em, quay lại Bước 2 với từ khoá khác để việc tìm kiếm hiệu quả hơn.

Gợi ý: Trong ví dụ tại Hình 7.2, kết quả tìm kiếm có thể bao gồm thông báo tuyển sinh của các trường Đại học khác nữa. Để tìm chính xác hơn, tên trường Đại học mà em cần tìm nên đặt trong dấu nháy kép, cụ thể: tuyển sinh 2022 "đại học Dược Hà Nội" + filetype:pdf.

Việc đặt từ khoá cần tìm trong dấu nháy kép là một trong những cách thông dụng để thu hẹp phạm vi tìm kiếm, cho kết quả tìm kiếm chính xác hơn. Bên cạnh đó, xây dựng từ khoá tìm kiếm cũng là một trong những kĩ năng quan trọng làm tăng độ chính xác, hiệu quả và tiết kiệm thời gian tìm kiếm.

Hãy tham khảo các "thủ thuật tìm kiếm" trên Internet để nâng cao kĩ năng tìm kiếm của em.



### Nhiệm vụ 4. Trải nghiệm và so sánh giữa các máy tìm kiếm phổ biến

**Hướng dẫn:**

*Bước 1*. Chọn máy tìm kiếm để trải nghiệm.

– Khởi động công cụ tìm kiếm mà em quen sử dụng.

– Nhập từ khoá cần tìm. Gợi ý: sử dụng từ khoá "máy tìm kiếm" hay "công cụ tìm kiếm", kết hợp với các từ khoá "tốt nhất" hay "phổ biến nhất".

– Đọc thông tin từ các kết quả tìm được, chọn và ghi lại địa chỉ URL của ba công cụ/máy tìm kiếm được đánh giá là phổ biến nhất hoặc tốt nhất.

*Bước 2*. Trải nghiệm các máy tìm kiếm đã chọn.

– Khởi động trình duyệt trên máy tính của em.

– Gõ địa chỉ URL của máy tìm kiếm mà em muốn trải nghiệm.

– Gõ từ khoá để tìm một thông tin mà em quan tâm (đội bóng, ban nhạc, thời tiết hay kì quan thiên nhiên,...).

– Quan sát và nhận xét về kết quả tìm kiếm.

– Lập bảng so sánh tính năng cơ bản của các máy tìm kiếm rồi điền thông tin vào bảng. Ví dụ: Có cho phép tìm kiếm bằng giọng nói hay không? Có phân loại kết quả tìm kiếm như tin tức, hình ảnh, video,... hay không? Có cho phép tìm dưới dạng tệp tin, chẳng hạn .pdf hay không?

– Lặp lại các thao tác trên với hai máy tìm kiếm còn lại.

*Bước 3*. Đọc lại bảng so sánh đã được điền đầy đủ các thông tin sau Bước 2 và rút ra kết luận về các máy tìm kiếm đã trải nghiệm. Chia sẻ với bạn bè các thông tin mà em thu được.



## LUYỆN TẬP

Thực hiện lại các nhiệm vụ ở phần thực hành bằng thiết bị số thông minh.



## VẬN DỤNG

Sử dụng máy tìm kiếm để tìm hiểu thông tin về nghề nghiệp mà em mơ ước được làm trong tương lai dưới các dạng văn bản, hình ảnh và video.

# BÀI 8 THỰC HÀNH NÂNG CAO SỬ DỤNG THƯ ĐIỆN TỬ VÀ MẠNG XÃ HỘI

SAU BÀI HỌC NÀY EM SẼ:

- Biết đánh dấu và phân loại thư điện tử.
- Khai thác được một số chức năng nâng cao của mạng xã hội.

Thư điện tử trong hộp thư (Inbox) thường được hiển thị theo trình tự thời gian thư được gửi tới. Em hãy trao đổi với các bạn trong nhóm về những cách giúp em tìm đọc lại được những thư em đã nhận trước đây.

## 1. ĐÁNH DẤU VÀ PHÂN LOẠI THƯ ĐIỆN TỬ

Việc đánh dấu, phân loại thư điện tử để sắp xếp hộp thư một cách hợp lí sẽ giúp việc sử dụng và tìm kiếm thư điện tử thuận tiện hơn. Dưới đây là một số cách đơn giản để thực hiện điều này của dịch vụ thư điện tử phổ biến nhất hiện nay là Gmail.

### Nhiệm vụ 1. Tìm hiểu dấu hiệu thư quan trọng trong Gmail

Gmail hỗ trợ tự động xác định và đánh dấu thư thuộc loại quan trọng bằng dấu ➤ (dấu quan trọng màu vàng – Hình 8.1) bằng cách dựa vào các dấu hiệu như: người gửi và tần suất gửi cho một người; thư điện tử được mở và trả lời; từ khoá có trong thư điện tử thường xuyên đọc; thư điện tử được gắn dấu sao, lưu trữ hoặc xoá,...




*Hình 8.1. Một số thư được đánh dấu quan trọng*

**Hướng dẫn:**

*Bước 1*. Vào hộp thư đến, di chuyển con trỏ chuột vào dấu quan trọng màu vàng để biết lí do thư đó được đánh dấu là quan trọng.

*Bước 2*. Nháy chuột vào dấu quan trọng ➤ để thay đổi trạng thái quan trọng/ không quan trọng của thư điện tử đó.

*Bước 3*. Thực hiện tìm kiếm **is: important** trong Gmail để hiển thị danh sách tất cả thư điện tử quan trọng.

### Nhiệm vụ 2. Sắp xếp, phân loại thư trong Gmail bằng Nhãn

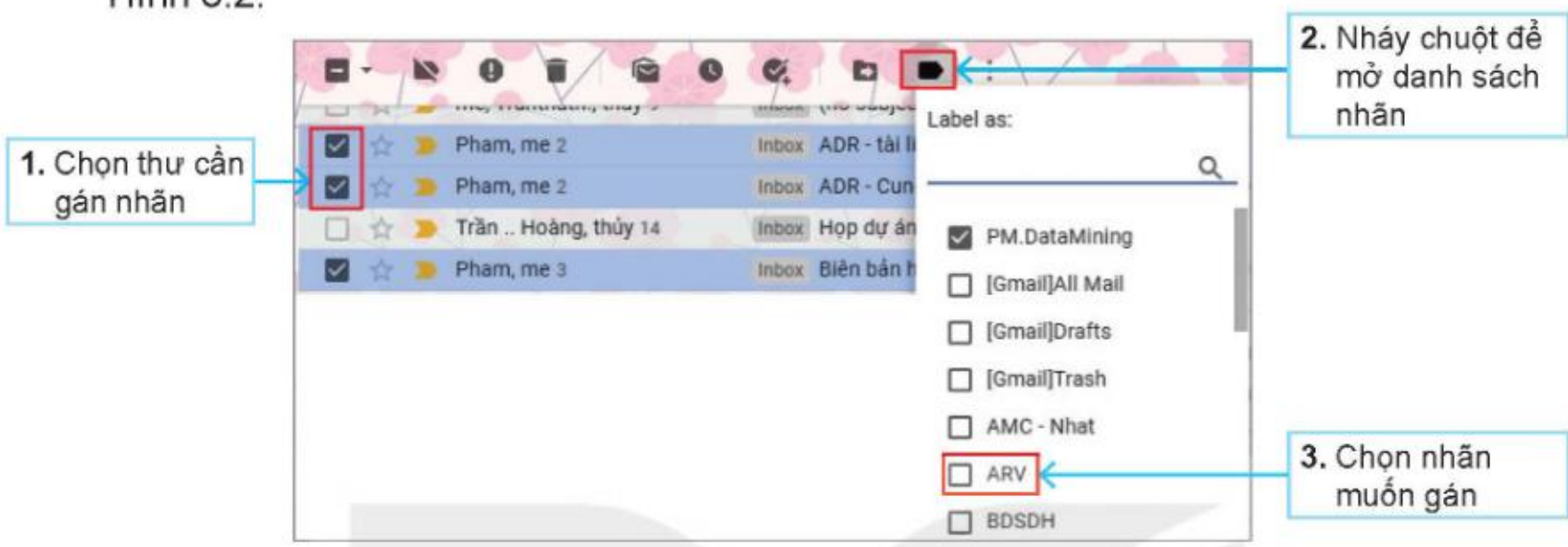
Sử dụng Nhãn (Label) để sắp xếp, phân loại thư trong hộp thư đến giúp em tiết kiệm thời gian khi tìm kiếm lại các thư, tránh thất lạc thông tin ở các thư cũ và quản lí việc nhận thư từ các địa chỉ thư điện tử dễ dàng hơn.

*Lưu ý*: Nhãn khác với thư mục. Khi em xoá một thư, thư đó sẽ bị xoá khỏi mọi nhãn đính kèm cũng như trong hộp thư đến.

**Hướng dẫn:**

*Bước 1.* Tạo nhãn: Truy cập vào Gmail. Nháy chuột vào **Danh sách mở rộng** ở bên trái cửa sổ; nháy chuột vào **Tạo nhãn mới**. Sau khi đặt tên nhãn nháy chọn **Save** để lưu lại.

*Bước 2.* Gán nhãn cho các thư trong hộp thư đến. Thực hiện theo các bước ở Hình 8.2.




*Hình 8.2. Gán hoặc thay đổi nhãn cho thư điện tử*

# 2. KHAI THÁC MỘT SỐ CHỨC NĂNG NÂNG CAO CỦA MẠNG XÃ HỘI

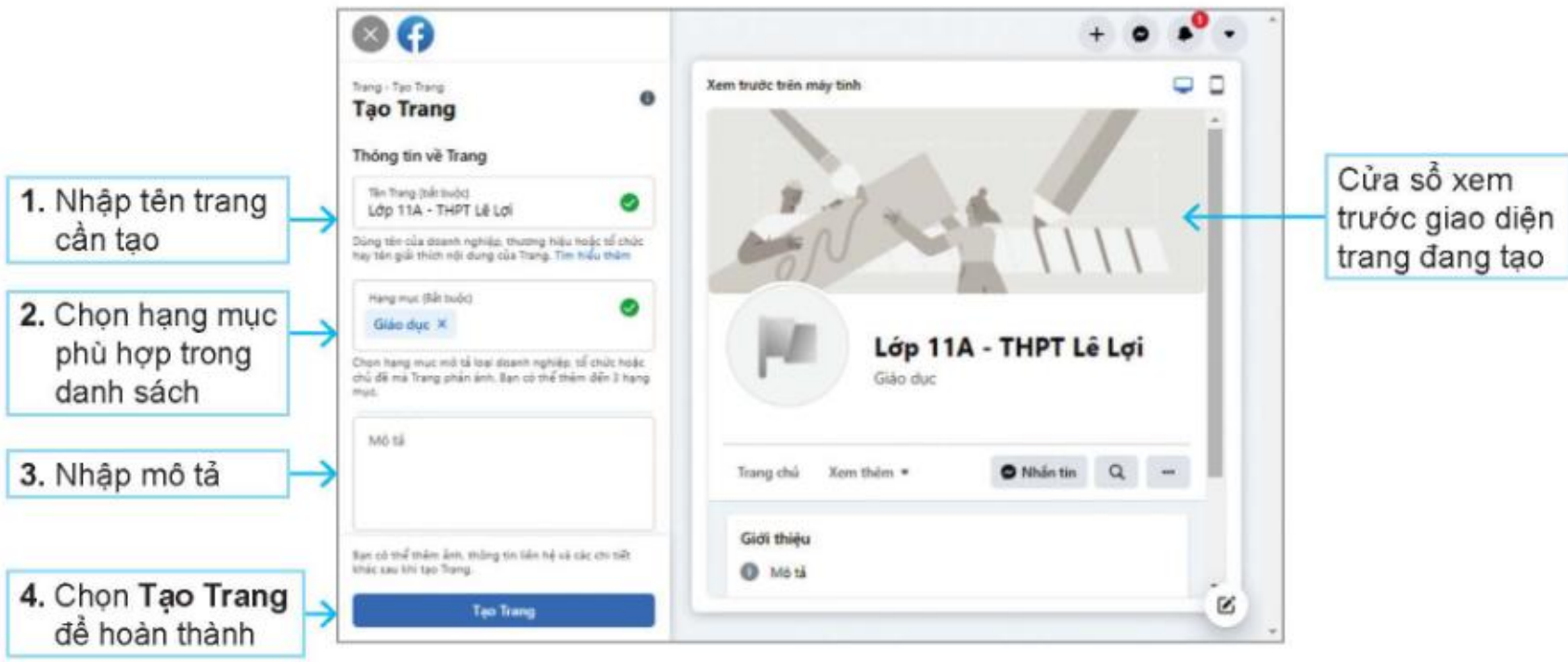
## Nhiệm vụ 3. Tạo Fanpage trên Facebook

Fanpage là trang web trên Facebook giúp các tổ chức, doanh nghiệp quảng bá hình ảnh, thương hiệu và sản phẩm. Em hãy tạo Fanpage của lớp mình để đăng tải các bài viết, ảnh, video và sự kiện của trường, lớp.

**Hướng dẫn:**

*Bước 1.* Đăng nhập vào Facebook. Nháy chuột vào biểu tượng **Menu** để mở danh sách các lệnh. Trong mục **Tạo**, chọn **Trang** Trang.

*Bước 2.* Làm theo các hướng dẫn tại Hình 8.3.




*Hình 8.3. Minh hoạ giao diện tạo mới một trang Fanpage*

*Bước 3.* Nhập nội dung, hình ảnh, các bài viết cho trang Fanpage vừa tạo và chia sẻ với bạn bè về trang này.

*Bước 4.* Khám phá các tính năng về quản lí trang để thực hiện quản lí trang Fanpage của lớp em.

*Lưu ý:* Việc tạo và quản lí các Fanpage để quảng bá thương hiệu của một tổ chức hay một trang thương mại điện tử được thực hiện theo cách tương tự.

## Nhiệm vụ 4. Tìm hiểu và cài đặt quyền riêng tư trên Facebook

– Thiết lập những người có thể xem các bài viết của mình trong tương lai:

*Bước 1.* Đăng nhập vào Facebook. Chọn **Cài đặt** → **Quyền riêng tư** để mở trang thông tin, hướng dẫn.

*Bước 2.* Đọc kĩ các thông tin giải thích về **Cài đặt quyền riêng tư và công cụ** để thực hiện.

*Bước 3.* Tìm hiểu và thiết lập những người có thể xem các bài viết của em trong tương lai (Hình 8.4).

*Bước 4.* Kiểm thử các cài đặt vừa thiết lập.




*Hình 8.4. Các lựa chọn người được xem bài viết*

– Thiết lập các quyền liên quan đến trang cá nhân và gắn thẻ

*Bước 1.* Chọn **Cài đặt** →**Trang cá nhân và gắn thẻ** trong Facebook.

*Bước 2.* Quan sát và giải thích ý nghĩa các lựa chọn tại Hình 8.5.

*Bước 3.* Thực hiện các thiết lập phù hợp tại **Trang cá nhân và gắn thẻ**.

*Bước 4.* Kiểm thử lựa chọn vừa thiết lập.

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG




*Hình 8.5. Các lựa chọn thiết lập Trang cá nhân và gắn thẻ*

## LUYỆN TẬP

1. Tạo mới một vài nhãn trong hộp thư của em để phân loại các thư liên quan đến học tập và giải trí. Gán nhãn phù hợp cho các thư và tìm kiếm chúng theo nhãn. Thực hiện việc chỉnh sửa và xoá nhãn.
2. Kiểm tra việc cài đặt quyền riêng tư hiện tại trong tài khoản Facebook của em. Thực hiện các cài đặt phù hợp để tăng tính bảo mật cho tài khoản.

## VẬN DỤNG

1. Khám phá phân loại thư bằng dấu sao. Nhận xét, so sánh ưu, nhược điểm với cách phân loại thư bằng dấu quan trọng.
2. Khám phá và sử dụng các tính năng liên quan tới cài đặt riêng tư cho tài khoản người dùng mạng xã hội Facebook:

   a) Xoá lịch sử hoạt động Facebook để ngăn Facebook phát tán quảng cáo đến người dùng.

   b) Bật, tắt dịch vụ chia sẻ vị trí của người dùng.

Chủ đề 3

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

BÀI 9

# GIAO TIẾP AN TOÀN TRÊN INTERNET

SAU BÀI HỌC NÀY EM SẼ:

- Nêu được một số dạng lừa đảo phổ biến trên không gian số và những biện pháp phòng tránh.
- Biết giao tiếp một cách văn minh, phù hợp với các quy tắc và văn hoá ứng xử trong môi trường số.

Mạng Internet mang lại nhiều lợi ích, tuy nhiên, việc sử dụng không đúng cách sẽ ảnh hưởng xấu tới sức khoẻ và tinh thần cùng nhiều rủi ro: mất thông tin cá nhân, bị lừa đảo, quấy rối, đối mặt với các thông tin sai lệch, bị lôi kéo tham gia những thử thách nguy hại,... Do vậy, khi tham gia mạng xã hội nói riêng, không gian số nói chung, mỗi người cần trang bị cho mình một số kĩ năng cơ bản như: kĩ năng giao tiếp; kĩ năng đối phó với dư luận và vượt qua khủng hoảng; kĩ năng nhận biết và phòng tránh lừa đảo,...

Hãy nêu vài ví dụ cho thấy sự cần thiết phải có các kĩ năng đó.

## 1. NHẬN BIẾT VÀ PHÒNG TRÁNH MỘT SỐ DẠNG LỪA ĐẢO TRÊN KHÔNG GIAN SỐ

**Hoạt động 1** Trao đổi về cách xử lí những tình huống bất thường

Em hoặc người quen đã có khi nào gặp những tình huống tương tự như sau chưa?

- Nhận được tin nhắn (qua thư điện tử, điện thoại, mạng xã hội Facebook, Zalo, Tiktok,...) yêu cầu chuyển tiền ngay vào tài khoản số 123 456 789 nếu không muốn gặp những rắc rối liên quan tới bản thân hay với cơ quan luật pháp.
- Nhận được thư điện tử từ địa chỉ lạ yêu cầu mở một tài liệu hoặc một đường link đính kèm và thực hiện theo hướng dẫn.

Trong những tình huống ấy, em hoặc người quen đã xử lí như thế nào?

### a) Một số nguyên tắc nhận biết và phòng tránh lừa đảo trên không gian số

Những kẻ lừa đảo thường hướng tới mục đích hưởng lợi tài chính hoặc gây ảnh hưởng tinh thần, thể xác người bị hại. Trong điều kiện bình thường, nhìn nhận sự việc một cách bình tĩnh, những câu chuyện mà chúng đưa ra thật sự khó tin. Do vậy, ta có thể ngăn chặn hành động của chúng bằng cách ứng xử linh hoạt theo ba nguyên tắc mang tính định hướng chung sau đây:

### Nguyên tắc thứ nhất: Hãy chậm lại!

Những kẻ lừa đảo thường tạo ra cảm giác cấp bách để chúng có thể vượt qua khả năng nhận định một cách sáng suốt của nạn nhân.



*... Phải thực hiện chuyển tiền ngay, nếu không chúng tôi buộc phải tiến hành các thủ tục truy tố bạn hoặc người thân...*

### Nguyên tắc thứ hai: Kiểm tra ngay!

Thực hiện ngay việc tra cứu số điện thoại, địa chỉ cơ quan hoặc tổ chức mà người gửi thông tin mang danh để liên hệ xác minh trực tiếp.



*... Đây là thông tin từ Cục Cảnh sát giao thông. Bạn chưa thanh toán tiền phạt vi phạm an toàn giao thông. Bạn đang vi phạm pháp luật nghiêm trọng...*

### Nguyên tắc thứ ba: Dừng lại, không gửi!

Mọi yêu cầu thanh toán hoặc gửi tiền ngay lập tức đều cần phải đặt dấu hỏi. Vì vậy, nếu cảm thấy giao dịch này không đáng tin, hãy dừng lại vì đó có thể là lừa đảo.



*... Mình là X đây. Đang cần rất gấp. Hãy gửi vào tài khoản 123... hoặc gửi cho mình mã thẻ cào điện thoại mệnh giá 200 nghìn hoặc 500 nghìn mỗi loại 20 thẻ....*

## b) Vận dụng vào một số tình huống cụ thể

**Lừa đảo hỗ trợ kĩ thuật**

Kẻ lừa đảo hỗ trợ kĩ thuật cố gắng thuyết phục rằng thiết bị của nạn nhân đang gặp sự cố và yêu cầu thanh toán ngay lập tức cho các dịch vụ để khắc phục sự cố đó, mà trên thực tế, nó không hề tồn tại.

**HÃY CHẬM LẠI**

Hãy tự đặt câu hỏi khi thông báo hiện lên có vẻ rất khẩn cấp. Tuy nhiên, các cảnh báo và tin nhắn bảo mật thực sự sẽ không bao giờ yêu cầu phải thực hiện ngay một cuộc điện thoại hay một khoản thanh toán. Nếu thấy cần, việc đầu tiên nên làm đó là cập nhật phần mềm bảo mật và quét virus.

**KIỂM TRA NGAY**

Thử tìm kiếm tên công ty hoặc số điện thoại kèm những từ khoá như "lừa đảo" hoặc "khiếu nại". Nếu cần sự hỗ trợ về mặt kĩ thuật, hãy tìm đến một đơn vị có uy tín và tin cậy.

**DỪNG LẠI, KHÔNG GỬI**

Các đơn vị hỗ trợ công nghệ hợp pháp sẽ không yêu cầu thanh toán ngay dưới dạng thẻ điện thoại, chuyển khoản, ứng dụng chuyển tiền hoặc tiền mã hoá, nhất là khi dịch vụ còn chưa được thực hiện.

| Lừa đảo dưới dạng thông báo tin tốt | |
|---|---|
| Kẻ lừa đảo có thể bất ngờ thông báo nạn nhân có cơ hội trúng thưởng hay nhận phiếu mua hàng trị giá cao, nhưng phải thanh toán một khoản phí để được nhận thưởng. | **HÃY CHẬM LẠI**<br>Hãy tìm kiếm lời khuyên từ người hiểu biết (thành viên trong gia đình, hoặc bạn bè) nếu không chắc đây có phải là thông tin đáng tin cậy.<br><br>**KIỂM TRA NGAY**<br>Tìm hiểu thêm thông tin về giải thưởng hoặc chương trình khuyến mại liên quan trên Internet. Nếu không có thông tin gì thì đây có khả năng là một trò lừa đảo.<br><br>**DỪNG LẠI, KHÔNG GỬI**<br>Không bao giờ trả phí trước để nhận thưởng sau cho dù mức phí đó rất nhỏ so với phần thưởng sắp nhận được. Cần nhớ, khi kẻ lừa đảo nhận được tiền, món tiền thưởng đó sẽ không bao giờ đến tay người trả phí. |

| Lừa đảo dưới dạng thông báo tin xấu | |
|---|---|
| Trong nhiều vụ lừa đảo, bạn sẽ nhận được cuộc gọi hoặc thư điện tử tự xưng là nhân viên làm việc trong cơ quan chức năng, cơ quan nhà nước và yêu cầu thanh toán ngay một khoản tiền nào đó. | **HÃY CHẬM LẠI**<br>Hãy đặt những câu hỏi để làm rõ vấn đề. Nếu người liên hệ tỏ ra cáu kỉnh khi thấy quá trình thanh toán không được thực hiện ngay, có thể đó là một vụ lừa đảo.<br><br>**KIỂM TRA NGAY**<br>Hãy liên lạc trực tiếp với tổ chức hay đơn vị mà người liên hệ mang danh. Không sử dụng thông tin liên hệ được cung cấp từ người gọi mà hãy tra cứu để tìm số điện thoại liên lạc hay trang web chính thức, hoặc gọi cho một đường dây nóng về lừa đảo.<br><br>**DỪNG LẠI, KHÔNG GỬI**<br>Không đồng ý với các khoản giao dịch bất thường mà không có các văn bản cụ thể kèm theo để có thời gian tra cứu, xác minh, nhất là với những yêu cầu thanh toán ngay bằng các phương thức chuyển tiền nhanh. |

**Lừa đảo qua website giả mạo các trang thương mại điện tử phổ biến**

Lợi dụng nhu cầu mua sắm online tăng cao, một số đối tượng đã tạo các trang web giả mạo và gửi đường liên kết truy cập các trang web lừa đảo này qua tin nhắn SMS, thư điện tử,... hoặc gọi điện thoại hướng dẫn khách hàng thực hiện giao dịch mua những món hàng "giá rẻ bất ngờ", thậm chí "miễn phí", so với các hàng hoá cùng chủng loại từ các siêu thị uy tín nhằm chiếm đoạt tiền của khách hàng trong thẻ cũng như tài khoản.

**HÃY CHẬM LẠI**

Đừng vội trả lời những yêu cầu hoặc nghe theo các hướng dẫn cung cấp thông tin thẻ, thông tin tài khoản, mật khẩu đăng nhập, mã số OTP từ những đối tượng không quen biết. Đó có thể là một vụ lừa đảo.

**KIỂM TRA NGAY**

Không ít trang web giả mạo tìm cách lừa đảo rằng chúng thuộc về những thương hiệu tên tuổi. Vì thế hãy kiểm tra các liên kết chỉ ra trên trang đó có tồn tại, nội dung trang web có phong phú, được trình bày cẩn trọng, các đánh giá của người dùng hoặc người mua hàng có đáng tin cậy hay là giả mạo,... Kẻ lừa đảo thường không có thời gian chăm chút cho các trang web của mình nên nội dung các trang web đó thường chỉ là những cóp nhặt sơ sài với nhiều lỗi chính tả,...

**DỪNG LẠI, KHÔNG GỬI**

Không vội thanh toán mà chưa xác minh, nhất là những yêu cầu thanh toán ngay bằng các phương thức chuyển tiền nhanh, không phải qua các kênh thanh toán uy tín như thẻ tín dụng, thẻ ghi nợ. Luôn nhớ, khi mua bán hàng qua mạng, quyền lựa chọn phương thức "nhận hàng – trả tiền" là một giao dịch an toàn nên chọn, nếu có thể.

Cần tỉnh táo và bình tĩnh ứng xử linh hoạt theo ba nguyên tắc mang tính định hướng chung: *Hãy chậm lại*; *Kiểm tra ngay*; *Dừng lại, không gửi* để nhận biết và phòng tránh lừa đảo trên không gian số.



Với các tình huống nêu trong Hoạt động 1, những cách nào sau đây là ứng xử cần thiết để phòng tránh những rủi ro?

A. Thực hiện các yêu cầu để đề phòng câu chuyện trở nên phức tạp.

B. Hãy dành thời gian và đặt câu hỏi để tránh bị dồn vào tình huống xấu.

C. Trao đổi với thầy cô giáo, người thân, bạn bè,... để được nghe ý kiến tư vấn.

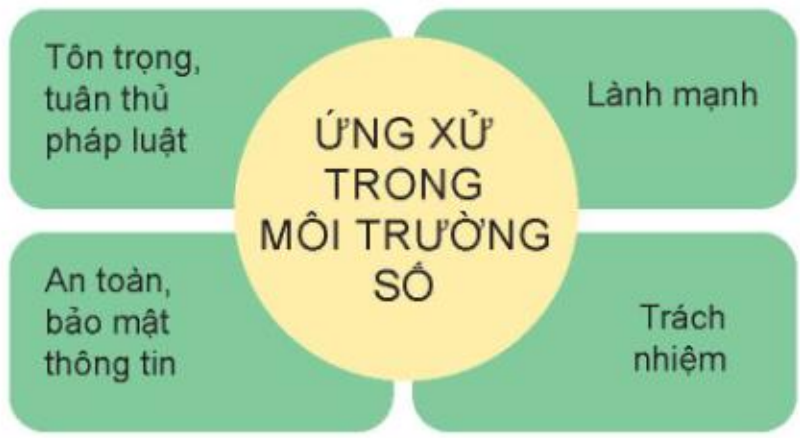
D. Tìm cách liên hệ trực tiếp với người gửi để làm rõ.

## 2. GIAO TIẾP VÀ ỨNG XỬ TRONG MÔI TRƯỜNG SỐ

**Hoạt động 2** **Quy tắc ứng xử chung trong môi trường số**

Khi bắt đầu sử dụng mạng Internet là em bắt đầu trở thành một công dân số, được tiếp cận với cả những lợi ích và rủi ro trên mạng. Hãy cùng trao đổi để chỉ ra một vài quy tắc ứng xử chung trong môi trường số.

Môi trường văn hoá của mạng xã hội và không gian số phụ thuộc nhiều vào hành vi của mỗi người sử dụng. Chính vì vậy, việc xây dựng chuẩn mực đạo đức về hành vi, ứng xử trên mạng xã hội, tạo thói quen tích cực trong các hành vi ứng xử của người dùng, góp phần xây dựng môi trường mạng an toàn, lành mạnh là việc làm rất cần thiết. Việc xây dựng và hình thành các chuẩn mực đó đòi hỏi cả một quá trình không ngừng vun đắp trên cơ sở bốn quy tắc chính được nêu trong Hình 9.1 (theo Quyết định số 874/QĐ-BTTTT về Bộ Quy tắc ứng xử trên mạng xã hội ngày 17/6/2021).

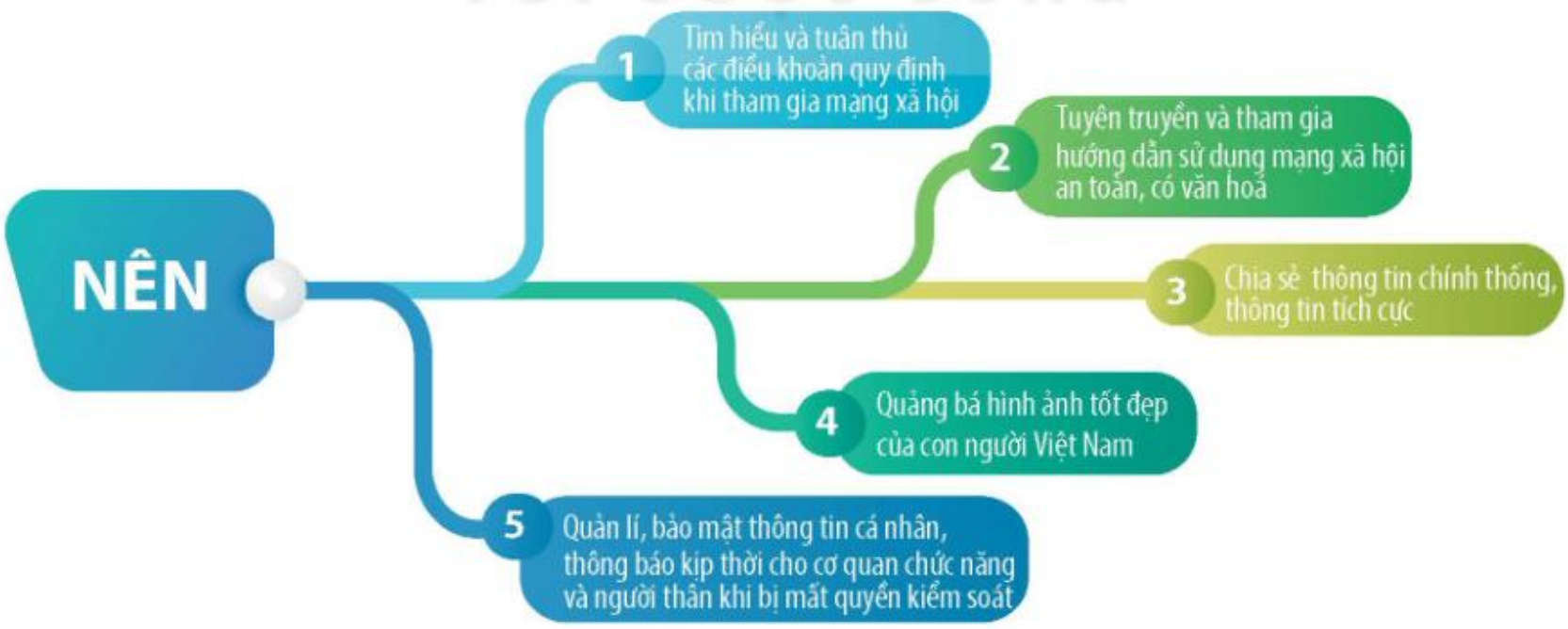



*Hình 9.1. Quy tắc ứng xử trong môi trường số*

Nội hàm của các quy tắc đó có thể được trình bày ngắn gọn như sau:

- **Quy tắc tôn trọng, tuân thủ pháp luật** là yêu cầu phải tuân thủ luật pháp Việt Nam, tôn trọng quyền và lợi ích hợp pháp của tổ chức, cá nhân.
- **Quy tắc lành mạnh** đòi hỏi mọi hành vi, ứng xử trên mạng xã hội phải phù hợp với các giá trị đạo đức, văn hoá truyền thống tốt đẹp của dân tộc Việt Nam.
- **Quy tắc an toàn, bảo mật thông tin** yêu cầu phải tuân thủ các quy định và hướng dẫn về bảo vệ an toàn và bảo mật thông tin cá nhân và tổ chức.
- **Quy tắc trách nhiệm** hướng tới việc cần chịu trách nhiệm về các hành vi, ứng xử trên mạng xã hội; khi có yêu cầu, phải chủ động phối hợp, hợp tác với cơ quan chức năng để xử lí hành vi, nội dung thông tin vi phạm pháp luật.

Không gian số nói chung và mạng xã hội nói riêng là không gian chung của tất cả mọi người, không phân biệt tôn giáo, tuổi tác, màu da,... Chính vì thế, với tư cách là những công dân có trách nhiệm, có tri thức, có văn hoá trong không gian ấy, em cần luôn nhớ mỗi hành vi, mỗi biểu cảm, mỗi ngôn từ khi đưa lên mạng xã hội chỉ trong chốc lát đã lan truyền tới mọi nơi trên thế giới. Hãy cân nhắc một cách cẩn trọng về những gì nên làm (Hình 9.2) và không nên làm (Hình 9.3) khi tham gia mạng xã hội.




*Hình 9.2. Một số điều nên làm khi tham gia mạng xã hội*




*Hình 9.3. Một số điều không nên làm khi tham gia mạng xã hội*

Cần tuân thủ các quy tắc ứng xử trong môi trường số như: 1) tôn trọng, tuân thủ pháp luật; 2) lành mạnh; 3) an toàn, bảo mật thông tin; 4) trách nhiệm.

**1.** Những việc nào sau đây cần được khuyến khích khi tham gia môi trường số?

A. Tìm hiểu và tuân thủ các điều khoản hướng dẫn sử dụng của nhà cung cấp dịch vụ mạng xã hội trước khi đăng kí, tham gia mạng xã hội.

B. Chia sẻ thông tin từ mọi nguồn khác nhau.

C. Mạng xã hội là môi trường ảo, do vậy không cần quá câu nệ về câu chữ.

D. Cần được sự đồng ý khi chia sẻ hình ảnh và chuyện riêng tư của bạn bè.

**2.** Những quan niệm nào sau đây là không đúng?

A. Mọi tin nhắn, hình ảnh và video đăng tải lên mạng đều có thể thu hồi.

B. Cần nhanh chóng thông báo tới các cơ quan chức năng, nhà cung cấp dịch vụ khi tài khoản của tổ chức, cá nhân bị mất quyền kiểm soát, bị giả mạo.

C. Cần phê phán các từ ngữ không mang tính phổ thông, nặng bản sắc vùng miền.

D. Trong ứng xử trên mạng xã hội được phép làm mọi điều pháp luật không cấm.

E. Không cho mượn, cho thuê giấy tờ cá nhân hoặc thẻ ngân hàng; không bán, cho mượn tài khoản; không nhận chuyển khoản hay nhận tiền cho người không quen,...

## LUYỆN TẬP

**1.** Em nhận được tin nhắn trên Facebook từ tài khoản mang tên bạn em với nội dung bạn cần tiền gấp và yêu cầu em chuyển tiền ngay cho số điện thoại lạ hoặc một số tài khoản ngân hàng mang tên bạn em. Có thể vận dụng ba nguyên tắc phòng chống lừa đảo trong trường hợp này như thế nào?

**2.** Ngoài những điều nên làm và không nên làm khi tham gia mạng xã hội (Hình 9.2, Hình 9.3), em có thể bổ sung thêm một vài điều khác nữa hay không?

## VẬN DỤNG

Các hình thức lừa đảo trên không gian số rất đa dạng. Hãy sử dụng các từ khoá thích hợp để tìm hiểu thêm các tình huống lừa đảo trong thực tế và áp dụng ba nguyên tắc phòng tránh đã được nêu trong bài học.

Chủ đề 4

# Giới thiệu các hệ cơ sở dữ liệu

BÀI 10

## LƯU TRỮ DỮ LIỆU VÀ KHAI THÁC THÔNG TIN PHỤC VỤ QUẢN LÍ

SAU BÀI HỌC NÀY EM SẼ:

- Biết được sự cần thiết phải lưu trữ dữ liệu và khai thác thông tin từ dữ liệu lưu trữ cho các bài toán quản lí.

Các công việc quản lí trong thực tế rất đa dạng: quản lí nhân viên, tài chính, thiết bị,... tại các cơ quan, tổ chức; quản lí chỗ ngồi trên các chuyến bay, tàu xe tại các phòng bán vé; quản lí hồ sơ bệnh án tại bệnh viện; quản lí học sinh và kết quả học tập trong các trường.

Để quản lí kết quả học tập, như em biết, phải quản lí điểm của từng môn học bao gồm điểm đánh giá (ĐĐG) thường xuyên, ĐĐG giữa kì, ĐĐG cuối kì,... Theo em, hoạt động này có cần lưu trữ dữ liệu không? Nếu có, đó là những dữ liệu gì?

### 1. CẬP NHẬT DỮ LIỆU

**Hoạt động** Thảo luận về ghi chép điểm môn học

Giáo viên dạy môn Toán dùng cuốn sổ điểm môn học để ghi lại điểm của từng học sinh lớp 11A (Bảng 10.1). Hãy cùng thảo luận để xác định xem có thể khai thác được những thông tin gì từ sổ điểm môn học này. Ngoài việc ghi điểm vào sổ điểm, có thể có những công việc nào khác?

Việc ghi điểm vào sổ điểm được thực hiện thường xuyên, mỗi khi có ĐĐG thường xuyên, giữa kì hay cuối kì. Việc ghi chép này là gọi là lưu trữ dữ liệu điểm.

**Bảng 10.1.** *Bảng điểm môn Toán lớp 11A (học kì I)*

| STT | Họ và tên | ĐĐG thường xuyên | | | | ĐĐG giữa kì | ĐĐG cuối kì |
|---|---|---|---|---|---|---|---|
| 1 | Dương Hồng Anh | 8 | 7 | | 7 | 9 | 8 |
| 2 | Lương Việt Anh | | 5 | 6 | 5 | 5 | 5 |
| 3 | Nguyễn Kì Duyên | 7 | | 8 | 0 | 10 | 9 |
| 4 | Bùi Quý Dương | 5 | 7 | | 7 | 6 | 6 |
| 5 | Đỗ Hồng Dương | | 7 | 8 | 6 | 6 | 7 |
| | ... | | | | | | |

Việc ghi chép điểm có thể có sai sót, nhầm lẫn. Vì vậy:

- Có thể phải xoá một dữ liệu điểm. Chẳng hạn xoá điểm 0 trong điểm đánh giá thường xuyên của Nguyễn Kì Duyên vì nghỉ học có lí do.
- Có thể phải sửa một dữ liệu nào đó. Chẳng hạn trong danh sách có tên Dương Hồng Anh nhưng thực tế tên chính xác phải là Dương Hoàng Anh.

**Bảng 10.2.** *Bảng điểm môn Toán lớp 11A (học kì I) sau khi chỉnh sửa*

| STT | Họ và tên | ĐĐG thường xuyên | | | | ĐĐG giữa kì | ĐĐG cuối kì |
|---|---|---|---|---|---|---|---|
| 1 | Dương Hoàng Anh | 8 | 7 | | 7 | 9 | 8 |
| 2 | Lương Việt Anh | | 5 | 6 | 5 | 5 | 5 |
| 3 | Nguyễn Kì Duyên | 7 | | 8 | | 10 | 9 |
| 4 | Bùi Quý Dương | 5 | 7 | | 7 | 6 | 6 |
| 5 | Đỗ Hồng Dương | | 7 | 8 | 6 | 6 | 7 |
| | ... | | | | | | |

Việc thêm, xoá và chỉnh sửa dữ liệu tương tự như trên là những công việc thường được thực hiện với dữ liệu của tất cả các bài toán quản lí và chúng được gọi chung là *cập nhật dữ liệu*.

## 2. TRUY XUẤT DỮ LIỆU VÀ KHAI THÁC THÔNG TIN

Rõ ràng việc ghi chép điểm các môn học như trên không chỉ nhằm mục đích lưu trữ. Từ dữ liệu lưu trữ này, có thể:

- Lập được danh sách học sinh với điểm học kì từ cao xuống thấp.
- Tìm kiếm và lập danh sách học sinh có điểm học kì >= 7, >= 8.

Việc tìm kiếm, sắp xếp hay lọc ra các dữ liệu theo những tiêu chí nào đó từ dữ liệu đã có thường được gọi là *truy xuất dữ liệu*.

Một ví dụ phức tạp hơn về truy xuất dữ liệu là lập bảng điểm cuối học kì I (Bảng điểm lớp 11A) đòi hỏi ghép dữ liệu từ các bảng điểm môn học theo danh sách lớp (Hình 10.1).

DANH SÁCH LỚP 11A

| STT | Họ và tên |
|---|---|
| 1 | Dương Hoàng Anh |
| 2 | Lương Việt Anh |
| 3 | Nguyễn Kì Duyên |
| | ... |

ĐĐG CUỐI KÌ I MÔN TOÁN LỚP 11A

| STT | Họ và tên | ĐĐG cuối kì |
|---|---|---|
| 1 | Dương Hoàng Anh | 8 |
| 2 | Lương Việt Anh | 5 |
| 3 | Nguyễn Kì Duyên | 9 |
| | ... | |

...

ĐĐG CUỐI KÌ I MÔN ĐỊA LÍ LỚP 11A

| STT | Họ và tên | ĐĐG cuối kì |
|---|---|---|
| 1 | Dương Hoàng Anh | 6 |
| 2 | Lương Việt Anh | 8 |
| 3 | Nguyễn Kì Duyên | 7 |
| | ... | |

BẢNG ĐĐG CÁC MÔN CUỐI HỌC KÌ I LỚP 11A

| STT | Họ và tên | Toán | Tin học | Vật lí | Hoá học | Sinh học | Ngữ văn | Lịch sử | Địa lí |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Dương Hoàng Anh | 8 | 10 | 10 | 9 | 10 | 6 | 8 | 6 |
| 2 | Lương Việt Anh | 5 | 6 | 6 | 7 | 6 | 7 | 8 | 8 |
| 3 | Nguyễn Kì Duyên | 9 | 10 | 10 | 10 | 8 | 6 | 7 | 7 |
| | ... | | | | | | | | |

*Hình 10.1. Ví dụ truy xuất dữ liệu*

Cũng có những công việc khác, chẳng hạn lập bảng phân loại kết quả học tập như Bảng 10.3, đòi hỏi phải phân tích, thống kê, tính toán từ dữ liệu đã có để được thông tin cần thiết. Những công việc kiểu như vậy được gọi là khai thác thông tin từ dữ liệu đã có.

**Bảng 10.3.** *Thống kê kết quả trung bình năm học môn Toán*

| STT | Phân loại | Số lượng | Tỉ lệ |
|---|---|---|---|
| 1 | Tốt | 8 | 20% |
| 2 | Khá | 16 | 40% |
| 3 | Đạt | 16 | 40% |
| 4 | Chưa đạt | 0 | 0% |

Từ bảng điểm của lớp 11A, tương tự như với các môn học, có thể tìm kiếm hoặc thống kê theo các tiêu chí khác nhau, để khai thác thông tin phục vụ đánh giá phân loại, so sánh kết quả các môn học của lớp,...

Quản lí kết quả học tập của học sinh từ bảng điểm môn học đến bảng điểm chung của lớp chỉ là một ví dụ cho thấy nhu cầu lưu trữ dữ liệu và khai thác thông tin rất đa dạng. Trong thực tế, nhiều lĩnh vực hoạt động khác, khối lượng dữ liệu được lưu trữ và khai thác thường xuyên lớn hơn rất nhiều, chẳng hạn ở bệnh viện là dữ liệu về các bệnh nhân đến khám chữa bệnh, các loại thuốc, vật tư y tế được mua, sử dụng; ở ngân hàng là dữ liệu về khách hàng, lượng tiền gửi vào rút ra hằng ngày; ở các trung tâm dự báo thời tiết là các dữ liệu về những thay đổi nhiệt độ, độ ẩm, hướng và cường độ gió;... Chính vì thế, ngày nay dữ liệu đã trở thành đối tượng nghiên cứu của một lĩnh vực khoa học liên ngành mới nổi đó là Khoa học dữ liệu.

- Bài toán quản lí là bài toán phổ biến trong thực tế. Cần phải tổ chức lưu trữ dữ liệu để phục vụ các yêu cầu quản lí đa dạng.
- Dữ liệu lưu trữ có thể được cập nhật thường xuyên, được truy xuất theo nhiều tiêu chí khác nhau để thu được các thông tin hữu ích.



Cập nhật dữ liệu là gì? Tại sao dữ liệu cần được cập nhật thường xuyên?

## 3. THU THẬP DỮ LIỆU TỰ ĐỘNG



Hầu hết các hoạt động quản lí truyền thống đều phải nhập dữ liệu thủ công. Trong bối cảnh tự động hoá trên cơ sở máy tính, đặc biệt là sự xuất hiện của các hệ thống kết nối vạn vật (IoT) nói riêng và Cách mạng Công nghiệp 4.0 nói chung, rất nhiều hoạt động quản lí đã thực hiện việc thu thập dữ liệu tự động.




*Hình 10.2. Mã vạch*

Ở các siêu thị lớn, mỗi ngày có hàng nghìn khách mua hàng, nhiều người mua hàng chục mặt hàng trong số hàng trăm mặt hàng có trong siêu thị. Nếu nhập dữ liệu thủ công, phải mất nhiều thời gian cho mỗi đơn hàng. Việc này có thể gây tắc nghẽn tại các quầy thanh toán, làm giảm năng suất bán hàng. Để khắc phục tình trạng này, người ta đã tạo các mã vạch mang thông tin về mặt hàng dán trên bao bì (Hình 10.2). Tại các quầy thanh toán, người thu tiền không cần nhập dữ liệu, chỉ cần đưa hàng qua đầu đọc mã vạch. Đầu đọc sẽ đọc mã và gửi cho máy tính để lập đơn hàng. Toàn bộ dữ liệu về hàng hoá cùng doanh thu được lưu trữ tự động. Dựa trên các dữ liệu đó, máy tính sẽ giúp lập các báo cáo doanh thu, thống kê, tổng hợp, phân tích để cải thiện hoạt động kinh doanh, báo tồn kho (để đặt thêm hàng khi cần), lượng hàng tồn trên quầy (để bổ sung từ kho) và nhiều thông tin có ích khác để quản lí siêu thị.

Một ví dụ khác đó là việc lưu trữ chỉ số tiêu thụ điện. Người ta thay các đồng hồ đo điện với hộp số cơ khí trước đây bằng công tơ điện tử. Các công tơ điện tử có thể đều đặn gửi về công ti điện lực không chỉ lượng điện tiêu thụ mà còn cả các giá trị tức thời của điện áp, dòng, tần số và độ lệch pha,... Với công tơ điện tử, nhân viên điện lực không cần phải ghi số thủ công hằng tháng rồi nhập vào máy tính, giảm bớt nhiều công sức làm hoá đơn tiền điện. Tuy nhiên lợi ích lớn nhất mà ngành điện có được từ giải pháp này là có thể quản lí kĩ thuật qua phân tích dữ liệu từ các công tơ điện tử liên tục gửi về. Chẳng hạn có thể biết khu vực nào đang mất điện, khu vực nào đang quá tải,... để đưa ra những quyết định phù hợp.

- Quản lí là hoạt động rất phổ biến. Mục đích chính của quản lí là xử lí thông tin để đưa ra các quyết định. Vì vậy việc thu thập, lưu trữ dữ liệu có ý nghĩa quan trọng hàng đầu.
- Việc thu thập dữ liệu tự động mang lại nhiều lợi ích, không chỉ giảm bớt công sức thu thập mà còn cung cấp một khối lượng dữ liệu lớn giúp nâng cao hiệu quả của việc ra các quyết định cần thiết.

Hãy nêu tầm quan trọng của việc thu thập và lưu trữ dữ liệu đối với các bài toán quản lí.

## LUYỆN TẬP

1. Quản lí điểm chỉ là một ứng dụng quản lí trong trường học. Hãy tìm thêm các nhu cầu quản lí khác trong nhà trường và chỉ ra hoạt động quản lí đó cần những dữ liệu nào.
2. Người ta thường nói, ở bất cứ nơi nào có một tổ chức là nơi ấy có nhu cầu quản lí. Hãy kể tên một vài bài toán quản lí mà em biết.

## VẬN DỤNG

1. Hãy cho một ví dụ về một bài toán quản lí và nêu những dữ liệu mà hoạt động quản lí đó cần thu thập.
2. Tại các trạm bán xăng, việc thu thập dữ liệu về lượng xăng bán và doanh thu được thực hiện như thế nào?

# BÀI 11 CƠ SỞ DỮ LIỆU

SAU BÀI HỌC NÀY EM SẼ:

- Hiểu được khái niệm và các thuộc tính cơ bản của cơ sở dữ liệu.

Theo em, việc lưu trữ dữ liệu phục vụ các bài toán quản lí có phải chỉ là việc chuyển các ghi chép trên giấy thành văn bản trên máy tính không?

## 1. YÊU CẦU TỔ CHỨC LƯU TRỮ DỮ LIỆU MỘT CÁCH KHOA HỌC

**Hoạt động 1** **Có cần lưu trữ bảng điểm lớp học không?**

Giáo viên dạy mỗi môn học bắt buộc phải có một sổ điểm - bảng điểm môn học. Một bản sao của bảng điểm môn học được gửi cho giáo viên chủ nhiệm lớp. Hãy cùng thảo luận xem có cần lưu trữ bảng điểm lớp học không.

### a) Hạn chế dư thừa trong lưu trữ dữ liệu

Ví dụ về bảng điểm môn học và bảng điểm của lớp ở Mục 1, Bài 10 được xem xét từ thực tế thầy cô giáo ghi chép dữ liệu trên giấy.

Khi ghi chép, lưu trữ trên giấy, giáo viên chủ nhiệm lớp thường lưu trữ cả các bảng điểm môn học và bảng điểm lớp học. Tuy nhiên, có thể thấy nhiều dữ liệu lặp lại trong các bảng điểm này và có thể dẫn đến sự không nhất quán dữ liệu, ví dụ:

Ở bảng điểm môn Toán, ĐĐG cuối kì của Nguyễn Kì Duyên là 9.

| STT | Họ và tên | ĐĐG thường xuyên | | | | ĐĐG giữa kì | ĐĐG cuối kì |
|---|---|---|---|---|---|---|---|
| ... | ... | | | | | | |
| 3 | Nguyễn Kì Duyên | 7 | | 8 | | 10 | 9 |
| ... | ... | | | | | | |

*Hình 11.1. Trích xuất bảng điểm môn Toán*

Nhưng ở bảng điểm lớp học, ĐĐG cuối kì môn Toán của Nguyễn Kì Duyên có thể bị ghi nhầm là 8 (Hình 11.2).

| STT | Họ và tên | Toán | Tin học | Vật lí | Hoá học | Sinh học | Ngữ văn | Lịch sử | Địa lí |
|---|---|---|---|---|---|---|---|---|---|
| ... | ... | | | | | | | | |
| 3 | Nguyễn Kì Duyên | 8 | 10 | 10 | 10 | 8 | 6 | 7 | 7 |
| ... | ... | | | | | | | | |

*Hình 11.2. Trích xuất bảng điểm lớp học có điểm trung bình*

Thói quen cá nhân của người lưu trữ cũng có thể dẫn đến sự không nhất quán của dữ liệu khi lưu trữ thủ công. Ví dụ, điểm có thể ghi bằng chữ hoặc các kí hiệu riêng thay vì các cột điểm bằng số dễ nhận biết và tính toán.

Khi dữ liệu được lưu trữ trên máy tính, việc lập bảng điểm lớp học từ dữ liệu cơ sở (các bảng điểm môn học) có thể được thực hiện một cách dễ dàng. Vì thế, không cần lưu trữ bảng điểm lớp học do bảng này chỉ là một khung nhìn tổng hợp từ dữ liệu cơ sở bằng cách ghép các bảng điểm môn học.

Ví dụ trên cho thấy việc lưu trữ dữ liệu trên máy tính không đơn thuần là việc chuyển các văn bản trên giấy thành các tệp văn bản trên máy tính. Cần phải tổ chức việc lưu trữ sao cho có thể hạn chế trùng lặp làm dư thừa dữ liệu, khắc phục những lỗi không nhất quán về dữ liệu. Đây cũng là yêu cầu chung đối với việc lưu trữ dữ liệu của mọi bài toán quản lí.

## b) Sự phụ thuộc phần mềm và dữ liệu

Việc lưu trữ dữ liệu không tách rời với việc khai thác thông tin nhờ các phần mềm ứng dụng vì khai thác thông tin chính là mục đích của việc lưu trữ dữ liệu. Để thấy rõ hơn sự cần thiết phải tổ chức lưu trữ dữ liệu một cách khoa học, ta sẽ xem xét một cách tiếp cận gắn kết chương trình - dữ liệu để khai thác thông tin.

Việc lưu trữ dữ liệu điểm các môn học trên máy tính đòi hỏi cần có những phần mềm hỗ trợ cập nhật dữ liệu điểm và khai thác thông tin từ những dữ liệu ấy. Có thể chỉ ra một số thành phần cần có của phần mềm (thường được gọi là các mô đun phần mềm) đó là: cập nhật điểm môn học (Điểm môn học), quản lí danh sách lớp học (Danh sách lớp) và lập bảng điểm lớp học từ dữ liệu điểm môn học (Lập bảng điểm). Nếu viết các mô đun phần mềm này bằng một ngôn ngữ lập trình, ví dụ Python, giải pháp lưu trữ đơn giản nhất thường được nghĩ đến là sử dụng trực tiếp hệ thống tệp, ví dụ tệp văn bản (text).

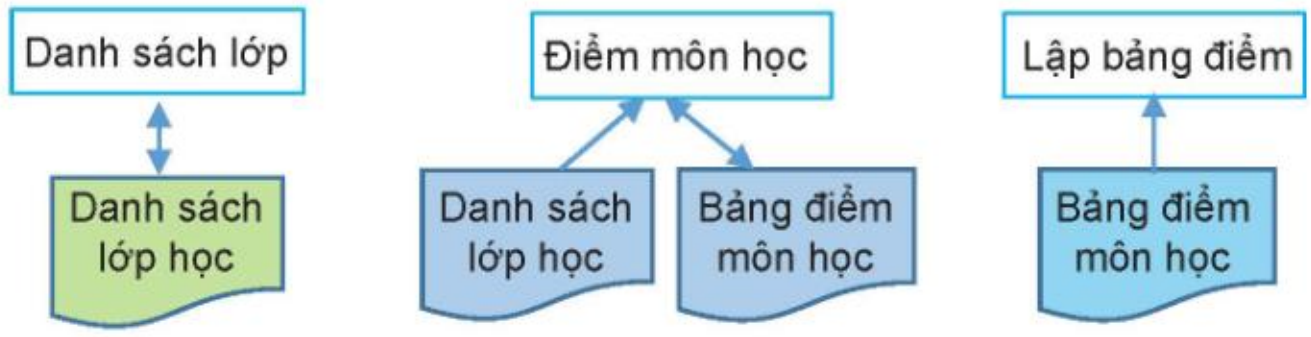
Danh sách lớp học sẽ gồm các dòng, mỗi dòng có nội dung lần lượt là: tên lớp, số thứ tự, họ và tên của một học sinh, ngăn cách nhau bởi dấu phẩy, ví dụ:

```
11A, 1, Dương Hoàng Anh
11A, 2, Lương Việt Anh
11A, 3, Nguyễn Kì Duyên
...
```

Bảng điểm môn học sẽ gồm các dòng, mỗi dòng có nội dung lần lượt là: tên lớp, số thứ tự, họ và tên của một học sinh, 4 ĐĐG thường xuyên, 1 ĐĐG giữa kì và 1 ĐĐG cuối kì, ngăn cách nhau bởi dấu phẩy (,). Ví dụ:

```
11A, 1, Dương Hoàng Anh, 8, 7,  , 7, 9, 8
11A, 2, Lương Việt Anh,  , 5, 6, 5, 5, 5
11A, 3, Nguyễn Kì Duyên, 7,  , 8,  , 10, 9
...
```

Hình 11.3 là sơ đồ cho thấy quan hệ giữa các mô đun phần mềm và các tệp dữ liệu.




*Hình 11.3. Các mô đun phần mềm và quan hệ với các tệp dữ liệu*

Khi viết mỗi mô đun phần mềm, người lập trình phải biết cấu trúc của các tệp dữ liệu để lần lượt đọc từng dòng, rồi tách các thành phần dữ liệu tương ứng theo dấu phẩy,...

Mô đun *Danh sách lớp* chịu trách nhiệm đọc và ghi dữ liệu cập nhật vào Danh sách lớp học. Mô đun *Điểm môn học* phải đọc dữ liệu từ Danh sách lớp học, tạo lập tệp Bảng điểm môn học nếu chưa có, cập nhật tệp Bảng điểm môn học nếu xuất hiện những dòng mới trong Danh sách lớp học; đọc và ghi dữ liệu cập nhật vào Bảng điểm môn học. Mô đun *Lập bảng điểm* phải đọc dữ liệu từ Bảng điểm môn học, xử lí dữ liệu để tạo và kết xuất ra bảng điểm lớp học.

Khi thay đổi cấu trúc các dòng ghi dữ liệu thì các mô đun phần mềm liên quan bắt buộc phải chỉnh sửa theo. Ví dụ quy định mới về điểm có phần lẻ thập phân, dùng dấu phẩy làm dấu ngăn cách phần nguyên và phần thập phân của điểm dẫn tới việc phải dùng dấu khác dấu phẩy để ngăn cách các thành phần của dòng dữ liệu. Khi đó, người viết chương trình buộc phải sửa mô đun *Điểm môn học* và *Lập bảng điểm* ở những câu lệnh phân tích, tách dòng dữ liệu.

Như vậy, tình trạng phụ thuộc giữa chương trình và dữ liệu dẫn tới việc nếu thay đổi cách lưu trữ dữ liệu phải sửa đổi phần mềm làm cho việc thiết kế, bảo trì, phát triển phần mềm mất nhiều thời gian và công sức. Bên cạnh đó, bài toán quản lí, nhất là khi liên quan tới những lĩnh vực lớn hoặc có nhiều biến động, lại thường xuất hiện các yêu cầu khai thác thông tin đa dạng khác nhau từ dữ liệu lưu trữ. Do vậy việc tổ chức dữ liệu độc lập để phần mềm không cần "nhìn thấy" chi tiết về cách lưu trữ mà vẫn sử dụng được dữ liệu là một trong các ý tưởng quan trọng để hình thành nên khoa học về cơ sở dữ liệu (CSDL) được nêu trong mục tiếp theo.

Dữ liệu cần được tổ chức lưu trữ một cách độc lập với việc xây dựng phát triển phần mềm, đảm bảo dễ dàng chia sẻ, dễ dàng bảo trì phát triển, đồng thời đảm bảo hạn chế tối đa việc dữ liệu lặp lại, gây dư thừa dữ liệu và hỗ trợ đảm bảo tính nhất quán dữ liệu.



**1.** Hãy giải thích yêu cầu về tính nhất quán dữ liệu trong lưu trữ dữ liệu.

**2.** Tại sao cần tổ chức lưu trữ dữ liệu độc lập với phần mềm?

# 2. CƠ SỞ DỮ LIỆU VÀ MỘT SỐ THUỘC TÍNH CƠ BẢN

**Hoạt động 2** **Sự cần thiết phải lưu trữ dữ liệu theo một cấu trúc xác định**

Hãy so sánh cách thức ghi chép và lưu trữ kết quả điểm môn học nêu trong Mục 1 với cách ghi chép và lưu trữ dưới dạng bảng. Theo em, cách nào là phù hợp hơn? Thông qua ví dụ bảng điểm môn học hãy chỉ ra một vài lí do cần lưu trữ dữ liệu theo một cấu trúc xác định.

## a) Khái niệm CSDL

Không thể tiếp tục với cách tiếp cận có sự phụ thuộc lẫn nhau giữa dữ liệu và phần mềm như mô tả ở trên. Cần phải chọn cách tiếp cận khác, theo đó việc lưu trữ dữ liệu phải là một vấn đề độc lập cần được xem xét một cách khái quát. Nhu cầu tổ chức dữ liệu sao cho việc khai thác dữ liệu thuận tiện, giảm được công sức và thời gian làm phần mềm là lí do chính cần tổ chức lưu trữ dữ liệu dưới dạng các CSDL.

*CSDL là một tập hợp các dữ liệu có liên quan với nhau, được lưu trữ một cách có tổ chức trên hệ thống máy tính.*

Bảng điểm các môn học khi được lưu trữ trong máy tính có thể được xem là ví dụ về CSDL. Thông tin về tài khoản ngân hàng bao gồm tên chủ tài khoản, số căn cước công dân, số dư có trong tài khoản,... có thể được tổ chức thành CSDL để quản lí và phục vụ khách hàng một cách nhanh chóng và tiện lợi,...

## b) Một số thuộc tính cơ bản của CSDL

- **Tính cấu trúc:** Hoạt động 2 cho ta hình dung về *tính cấu trúc* của CSDL. Dễ hình dung nhất về tính cấu trúc của CSDL khi dữ liệu được lưu trữ dưới dạng bảng gồm các hàng và các cột.

- **Tính không dư thừa:** Trong phần trên ta đã thấy cần hạn chế việc lưu trữ các dữ liệu trùng lặp cũng như những dữ liệu dễ dàng có được thông qua việc khai thác thông tin từ dữ liệu đã có. Thuộc tính này được gọi là *tính không dư thừa* của CSDL. Dưới đây ta sẽ xem xét thêm một số thuộc tính quan trọng khác của CSDL.

- **Tính độc lập dữ liệu:** Ở Mục 1, các em có thể thấy sự cần thiết phải đảm bảo tính độc lập dữ liệu với phần mềm. Trong phạm vi kiến thức phổ thông, tính độc lập dữ liệu có thể được hiểu là khả năng các mô đun phần mềm ứng dụng không cần phải cập nhật khi thay đổi cách thức tổ chức hoặc lưu trữ dữ liệu. Trong thực tế, các CSDL được thiết kế nhằm phục vụ nhiều người dùng với nhiều mục đích quản lí khác nhau, không phụ thuộc vào bài toán quản lí cụ thể hay phương tiện lưu trữ, xử lí dữ liệu, không đòi hỏi họ phải biết được các chi tiết kĩ thuật liên quan đến CSDL. Do vậy, việc đảm bảo độc lập dữ liệu là một trong các thuộc tính quan trọng nhất của CSDL.

- **Tính toàn vẹn:** Các giá trị dữ liệu lưu trữ phải thoả mãn những ràng buộc cụ thể tuỳ thuộc vào thực tế mà nó phản ánh, ví dụ điểm đánh giá học tập phải là số nguyên (hay số thập phân) không âm và nhỏ hơn hoặc bằng 10, tuỳ theo quy định về đánh giá điểm học tập của tổ chức.

- **Tính nhất quán:** Trong Mục 1 ta đã biết sơ suất khi cập nhật dữ liệu trong trường hợp CSDL tổ chức không tốt có thể làm ảnh hưởng tới *tính nhất quán* của dữ liệu. Khái niệm về tính nhất quán còn đòi hỏi dữ liệu trong các CSDL được đảm bảo đúng đắn sau các thao tác cập nhật dữ liệu, kể cả khi xảy ra sự cố ngay trong quá trình cập nhật. Ví dụ phải có cơ chế để đảm bảo không xảy ra hiện tượng số tiền đã bị trừ bớt trong tài khoản chuyển đi nhưng lại chưa xuất hiện trong tài khoản nhận về; hoặc hai đại lí vé máy bay bán cùng một vé ngồi cho hai khách hàng khác nhau,... Để xảy ra những trường hợp như vậy đều là sự vi phạm tới tính nhất quán của dữ liệu.

- **Tính bảo mật và an toàn:** Điều này có nghĩa là dữ liệu phải được bảo vệ an toàn, ngăn chặn được những truy xuất trái phép, chống được việc sao chép dữ liệu không hợp lệ. CSDL cần được tổ chức sao cho không phải ai cũng có quyền truy cập hay cập nhật dữ liệu. Ví dụ với CSDL Bảng điểm môn học không phải ai cũng có thể vào sửa chữa điểm; với CSDL tài khoản ngân hàng người không có thẩm quyền không được truy xuất để lấy thông tin cá nhân hay sửa đổi số dư tài khoản,... Bên cạnh đó, dữ liệu phải được bảo vệ an toàn, không dễ bị sai lạc, mất mát, có thể khôi phục dù có xảy ra các sự cố liên quan tới phần cứng hay phần mềm của máy tính.

- CSDL là một tập hợp dữ liệu có liên quan với nhau, được lưu trữ một cách có tổ chức trong hệ thống máy tính.
- Một số thuộc tính cơ bản của CSDL bao gồm: tính cấu trúc; tính không dư thừa; tính độc lập; tính toàn vẹn; tính nhất quán; tính bảo mật và an toàn.

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG



Hãy nêu ví dụ minh hoạ cho một vài thuộc tính cơ bản của CSDL.



## LUYỆN TẬP

1. Khi lưu trữ trên máy tính, theo em, có cần lưu trữ cột điểm trung bình trong bảng điểm môn học không?
2. Hãy lấy một ví dụ minh hoạ cho sự cần thiết của việc lưu trữ dữ liệu độc lập với phần mềm khai thác dữ liệu.



## VẬN DỤNG

Thư viện là nơi em có thể đến để đọc hay mượn sách. Hãy đề xuất các dữ liệu cần quản lí của một thư viện.

# BÀI 12 HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU VÀ HỆ CƠ SỞ DỮ LIỆU

SAU BÀI HỌC NÀY EM SẼ:

- Hiểu được khái niệm hệ quản trị CSDL.
- Hiểu được khái niệm hệ CSDL.
- Phân biệt được CSDL tập trung và CSDL phân tán.

Một CSDL lưu dữ liệu trên hệ thống máy tính dưới dạng các tệp có cấu trúc được thiết kế để nhiều người dùng có thể cùng khai thác dữ liệu trong CSDL đó. Tuy nhiên, không phải tất cả người dùng đều biết về cấu trúc các tệp lưu dữ liệu và tự viết chương trình khai thác dữ liệu. Theo em, có thể giải quyết vấn đề này như thế nào?

## 1. KHÁI NIỆM HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU

**Hoạt động 1** **Thảo luận về một phần mềm hỗ trợ thao tác dữ liệu**

Để tạo ra, lưu trữ và sửa đổi một văn bản trên máy tính chúng ta cần một phần mềm soạn thảo văn bản. Để tạo ra và cập nhật một bảng tính điện tử chúng ta cần một phần mềm bảng tính.

Theo em, một phần mềm hỗ trợ làm việc với các CSDL cần thực hiện được những yêu cầu nào dưới đây?

A. Cung cấp công cụ tạo lập CSDL.

B. Cập nhật dữ liệu và tự động kiểm tra tính đúng đắn của dữ liệu.

C. Hỗ trợ truy xuất dữ liệu.

D. Cung cấp giao diện để ai cũng có thể xem nội dung của các bảng dữ liệu một cách dễ dàng.

Để hỗ trợ làm việc với các CSDL người ta đã xây dựng những bộ phần mềm chuyên dụng gọi là hệ quản trị cơ sở dữ liệu (Database management systems, sau đây sẽ gọi tắt là hệ QTCSDL) với các nhóm chức năng sau:

### a) Nhóm chức năng định nghĩa dữ liệu

Với các chức năng này người dùng có thể:

- Khai báo CSDL với tên gọi xác định. Một hệ QTCSDL có thể quản trị nhiều CSDL.
- Tạo lập, sửa đổi kiến trúc bên trong mỗi CSDL.
- Nhiều hệ QTCSDL cho phép cài đặt các ràng buộc toàn vẹn dữ liệu để có thể kiểm soát tính đúng đắn của dữ liệu.

**b) Nhóm chức năng cập nhật và truy xuất dữ liệu** hay là nhóm *chức năng thao tác dữ liệu* với các chức năng sau:

- *Chức năng cập nhật dữ liệu.* CSDL sau khi được khởi tạo chưa có dữ liệu, cần phải nhập dữ liệu vào. Theo thời gian, do biến động của thế giới thực hoặc do sai sót khi nhập dữ liệu, dữ liệu trong CSDL không còn đúng nữa. Hệ QTCSDL cần cung cấp các chức năng thêm, xoá, sửa dữ liệu.

- *Chức năng truy xuất dữ liệu* theo những tiêu chí khác nhau.

**c) Nhóm chức năng bảo mật, an toàn CSDL**

- Không phải mọi hệ QTCSDL đều cung cấp công cụ để mọi người có thể dễ dàng đọc nội dung các bảng dữ liệu. Dữ liệu cần được bảo mật, chỉ cung cấp cho người có thẩm quyền. Do vậy, nhiều hệ QTCSDL cung cấp phương tiện kiểm soát quyền truy cập dữ liệu.

- Khi nhiều người được truy cập đồng thời vào CSDL sẽ nảy sinh ra vấn đề tranh chấp dữ liệu, chẳng hạn một người đang sửa trường dữ liệu của một bản ghi thì người kia ra lệnh xoá cả bản ghi. Trong những trường hợp như vậy, hệ QTCSDL cần cung cấp chức năng kiểm soát các giao dịch để đảm bảo tính nhất quán của dữ liệu.

- Hệ QTCSDL cung cấp các phương tiện thực hiện sao lưu dự phòng (backup) để đề phòng các sự cố gây mất dữ liệu và khôi phục dữ liệu khi cần thiết.

Điều quan trọng là các chức năng nói trên của hệ QTCSDL được xây dựng một cách tổng quát, theo mô hình CSDL không phụ thuộc các CSDL và ứng dụng cụ thể.

Ngoài ba nhóm chức năng chính nêu trên, các hệ QTCSDL còn cung cấp Giao diện lập trình ứng dụng (Application Programing Interface – API) dưới dạng thư viện hàm riêng cho từng ngôn ngữ lập trình. Nhờ có API, những phát triển ứng dụng CSDL có thể gửi truy vấn đến CSDL từ phần mềm mà họ phát triển để giải quyết các yêu cầu công việc cụ thể.

Những hệ QTCSDL được dùng phổ biến hiện nay là Oracle, MySQL, SQL Server, DB2, PostGreSQL, SQLlite,... Cũng có thể nhắc đến Microsoft Access, tuy nhiên do có nhiều hạn chế về hiệu suất, hệ QTCSDL này không được sử dụng cho các ứng dụng phức tạp, đặc biệt với các ứng dụng trực tuyến đang trở nên phổ biến.

- Hệ QTCSDL là phần mềm cung cấp phương thức để lưu trữ, cập nhật và truy xuất dữ liệu của CSDL, bảo mật và an toàn dữ liệu.
- Hệ QTCSDL cũng cung cấp giao diện lập trình ứng dụng cho các nhà phát triển ứng dụng và người dùng.



**1.** Nêu những khó khăn trong việc khai thác CSDL nếu không có hệ QTCSDL.

**2.** Tóm tắt các nhóm chức năng của hệ QTCSDL.

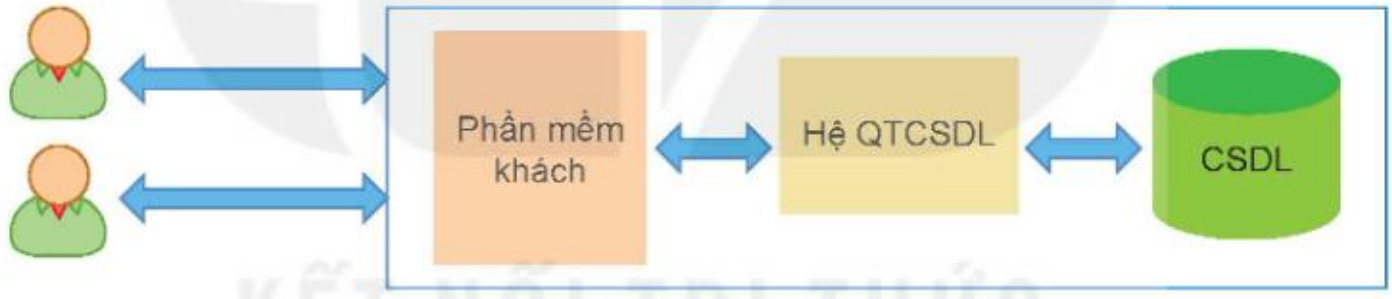
## 2. HỆ CƠ SỞ DỮ LIỆU

**Hoạt động 2** **Thảo luận về tra cứu trực tuyến điểm thi**

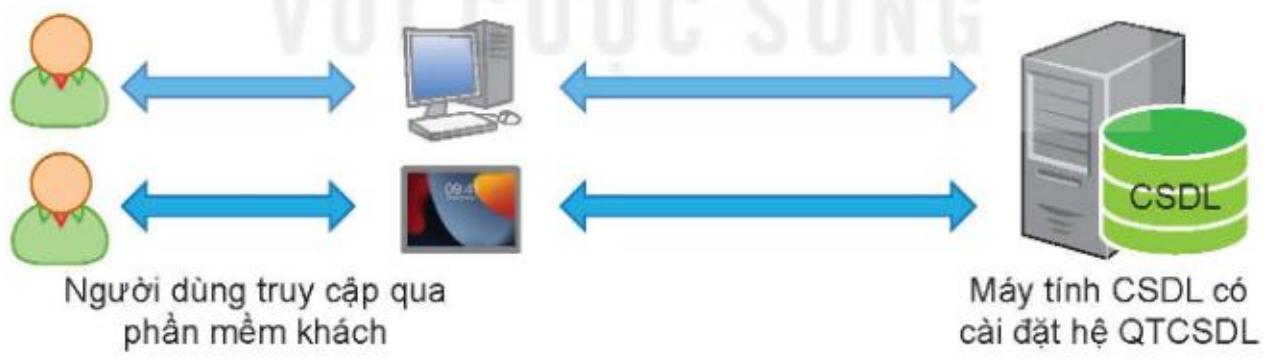
Khi lên mạng để tra cứu điểm thi vào lớp 10, thông thường trên màn hình chỉ yêu cầu nhập vài dữ liệu tối thiểu. Ví dụ sau khi nhập số báo danh gần như ngay lập tức em nhận được đầy đủ thông tin họ tên, trường lớp, điểm thi cùng kết quả xét tuyển. Vậy, ngoài CSDL điểm thi, hệ thống tra cứu trực tuyến cần có những gì để có thể cung cấp cho em thông tin như vậy?

Câu trả lời dễ suy đoán là đã có một phần mềm tổ chức giao tiếp với người dùng (tra cứu điểm thi) làm cầu nối giữa người dùng với CSDL điểm thi.

Vì mục đích hỗ trợ nhiều người dùng, các hệ QTCSDL nhiều người dùng thường được xây dựng theo mô hình hai thành phần, có thể được cài đặt trên cùng một máy tính hay được cài đặt trên các máy tính khác nhau. Một thành phần có nhiệm vụ chính là thực hiện các nhiệm vụ tính toán, xử lí dữ liệu (được gọi là phần "chủ"), thành phần còn lại (được gọi là phần mềm "khách") có nhiệm vụ tổ chức giao diện tương tác với người dùng, kết nối với phần "chủ", gửi các yêu cầu tính toán xử lí dữ liệu đến phần "chủ" và nhận lại, hiển thị kết quả tính toán xử lí dữ liệu. Hai thành phần này luôn được cung cấp trong một gói cài đặt hệ QTCSDL và thành phần chủ thường được gọi luôn là hệ QTCSDL.




*Hình 12.1. Phần mềm khách và hệ QTCSDL cài đặt trên cùng một máy tính*




*Hình 12.2. Hệ QTCSDL và phần mềm khách cài đặt trên các máy tính khác nhau*

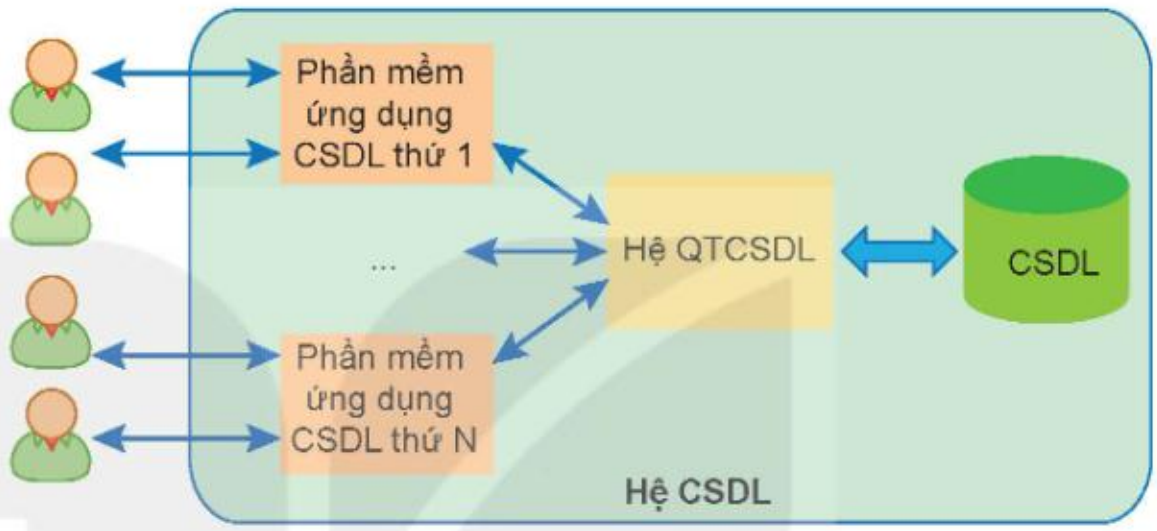
Các hệ QTCSDL cũng cung cấp công cụ (giao diện lập trình ứng dụng) để người dùng có thể tự viết phần mềm khách chuyên biệt theo nhu cầu. Phần mềm khách chuyên biệt như thế được gọi là phần mềm ứng dụng CSDL.

Nói một cách khái quát thì phần mềm ứng dụng CSDL là phần mềm được xây dựng tương tác với hệ QTCSDL nhằm mục đích hỗ trợ người dùng khai thác thông tin từ CSDL một cách thuận tiện theo các yêu cầu xác định. Có thể có nhiều phần mềm ứng dụng CSDL được xây dựng với các mục tiêu yêu cầu khác nhau cùng truy xuất, khai thác thông tin từ một CSDL.

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

Khi lên mạng tra cứu điểm thi vào lớp 10, em đã tương tác với một phần mềm ứng dụng CSDL tra cứu điểm thi được thiết kế với giao diện đơn giản, dễ dàng nhất cho người dùng. Sau khi tiếp nhận thông tin (ví dụ số báo danh), nó sẽ kết nối với hệ QTCSDL quản lí điểm thi, yêu cầu lấy ra những thông tin cần thiết (họ tên, trường lớp, điểm tất cả các môn thi cùng kết quả xét tuyển) và hiển thị kết quả nhận được cho em. Ngoài phần mềm ứng dụng CSDL tra cứu điểm thi, có thể còn có nhiều phần mềm ứng dụng CSDL nữa được xây dựng để khai thác thông tin từ CSDL điểm thi, như phần mềm xét tuyển, phần mềm thống kê đánh giá kết quả toàn bộ kì thi,...

Các ứng dụng mua bán trực tuyến, đặt xe công nghệ, thanh toán điện tử,... mà các em thường gặp trong cuộc sống đều là các phần mềm ứng dụng CSDL của một hệ thống CSDL cụ thể.




*Hình 12.3. Hệ cơ sở dữ liệu*

- Phần mềm ứng dụng CSDL là phần mềm được xây dựng tương tác với hệ QTCSDL nhằm mục đích hỗ trợ người dùng khai thác thông tin từ CSDL một cách thuận tiện theo các yêu cầu xác định.
- Một hệ thống gồm ba thành phần: CSDL, hệ QTCSDL và các phần mềm ứng dụng CSDL được gọi là một hệ CSDL.

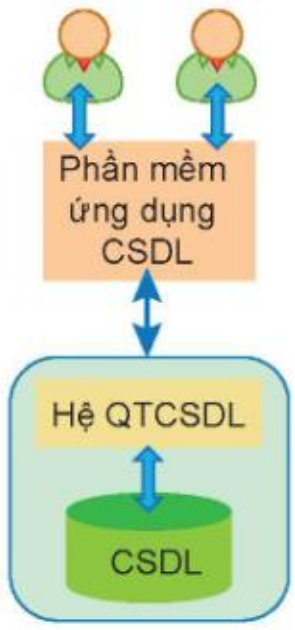
Hệ QTCSDL và hệ CSDL khác nhau như thế nào?

## 3. HỆ CƠ SỞ DỮ LIỆU TẬP TRUNG VÀ PHÂN TÁN

### a) Hệ CSDL tập trung

Hệ CSDL mà CSDL được lưu trữ tập trung trên một máy tính được gọi là *hệ cơ sở dữ liệu tập trung*.

Hệ CSDL tập trung bao gồm cả những CSDL một người dùng trên một máy (như các CSDL của Microsoft Access); người dùng vừa là người thiết kế, tạo lập và bảo trì CSDL, vừa là người viết phần mềm ứng dụng CSDL, vừa là người dùng đầu cuối hệ thống, khai thác thông tin theo những mục tiêu đã được đặt ra.




*Hình 12.4. Hệ CSDL tập trung*

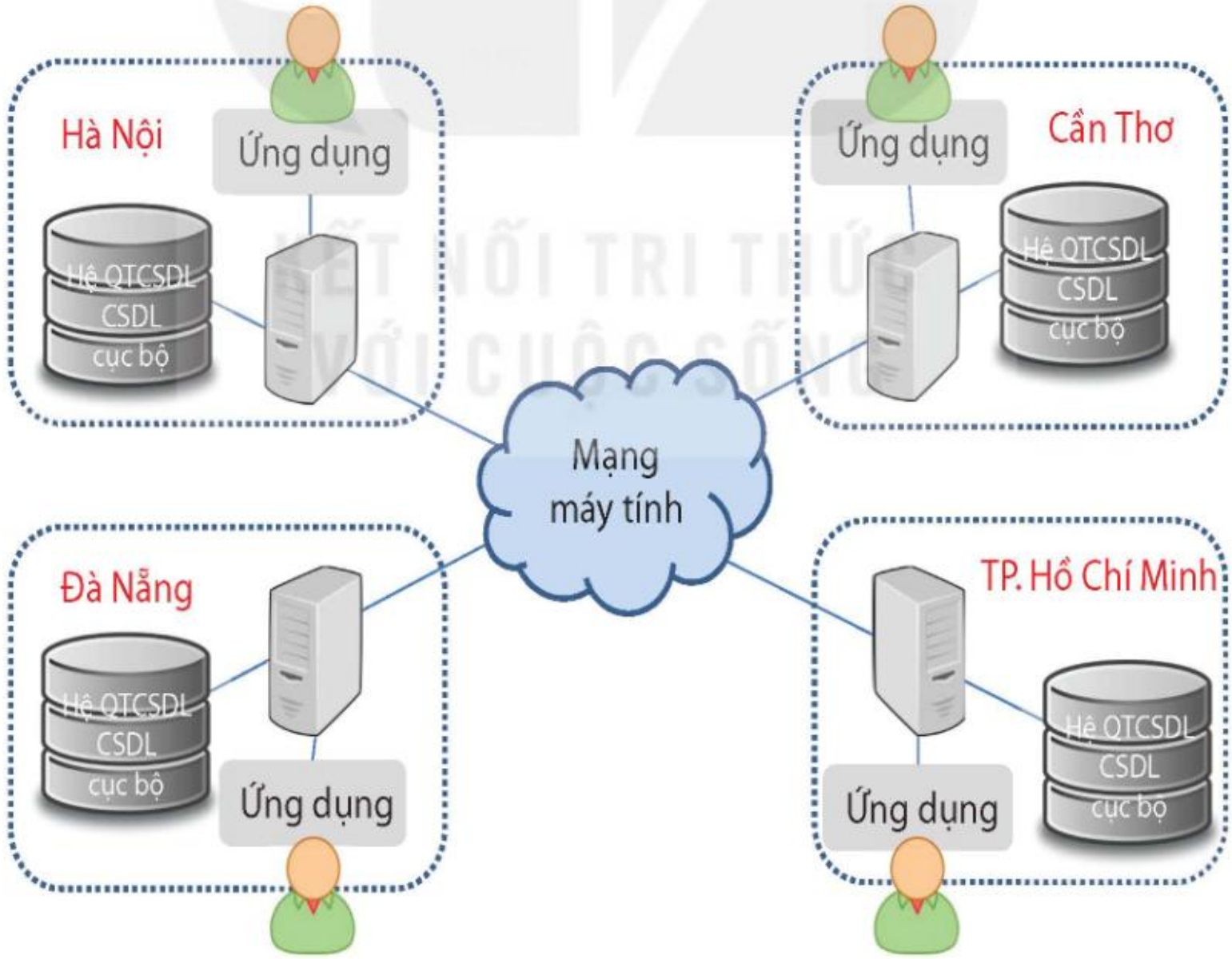
## b) Hệ CSDL phân tán

Khi một tổ chức có nhiều đơn vị phân tán về mặt địa lí, họ có thể chọn giải pháp tổ chức hệ CSDL phân tán để giải quyết các bài toán quản lí liên quan. Khác với hệ CSDL tập trung, quản lí dữ liệu tập trung trong một CSDL đặt trên một máy tính, hệ CSDL phân tán cho phép người dùng truy cập dữ liệu được lưu trữ ở nhiều máy tính khác nhau của mạng máy tính. Như vậy, để thiết lập một hệ CSDL phân tán, trước tiên cần xây dựng một CSDL phân tán tương ứng.

Ví dụ: Một doanh nghiệp có trụ sở chính tại Hà Nội và các chi nhánh tại Đà Nẵng, TP. Hồ Chí Minh, Cần Thơ. Mỗi máy tính tại trụ sở Hà Nội, Đà Nẵng, TP. Hồ Chí Minh, Cần Thơ hình thành một trạm; các trạm này được kết nối với nhau bởi một mạng máy tính.

*Cơ sở dữ liệu phân tán* là tập hợp dữ liệu được phân tán trên các trạm khác nhau của một mạng máy tính. Dữ liệu được lưu trữ tại mỗi trạm hình thành một CSDL cục bộ của trạm này. Mỗi trạm phải thực hiện các ứng dụng cục bộ. Ứng dụng cục bộ là ứng dụng chạy tại một trạm và chỉ sử dụng dữ liệu cục bộ của trạm này để cho ra kết quả cuối cùng. Mỗi trạm phải tham gia thực hiện ứng dụng toàn cục, còn được gọi là ứng dụng phân tán. Ứng dụng toàn cục là ứng dụng chạy tại một trạm và sử dụng dữ liệu của ít nhất hai trạm để cho ra kết quả cuối cùng.

Các chương trình ứng dụng chạy tại mỗi trạm, các hệ quản trị CSDL tại mỗi trạm và các CSDL của mỗi trạm hình thành hệ CSDL phân tán.




*Hình 12.5. Hệ cơ sở dữ liệu phân tán*

- Hệ CSDL mà CSDL được lưu trữ tập trung trên một máy tính được gọi là *hệ cơ sở dữ liệu tập trung*.
- Hệ CSDL phân tán cho phép người dùng truy cập dữ liệu được lưu trữ ở nhiều máy tính khác nhau trên mạng máy tính.

So với hệ CSDL tập trung, thiết kế và triển khai hệ CSDL phân tán phức tạp, khó khăn hơn trong đảm bảo tính nhất quán và bảo mật dữ liệu, chi phí duy trì cao hơn. Tuy nhiên, hệ CSDL phân tán có ưu điểm:

- Dễ dàng mở rộng, luôn có thể bổ sung thêm trạm dữ liệu vào hệ thống khi cần mà không làm ảnh hưởng đến hoạt động của các trạm dữ liệu đang hoạt động.

- Tính sẵn sàng và độ tin cậy được nâng cao. Hệ thống hoạt động ổn định, hạn chế tối đa việc mất mát dữ liệu dù có thể có trạm dữ liệu gặp sự cố vì dữ liệu có thể được sao lưu nhiều bản đặt ở các trạm dữ liệu khác.

Những hệ CSDL lớn dùng trong những tổ chức lớn, có nhiều người truy xuất trên phạm vi địa lí rộng lớn thường được tổ chức ở dạng hệ cơ sở dữ liệu phân tán để tối ưu hoá được tốc độ, giảm tải đường truyền khi truy xuất, cập nhật dữ liệu. Ví dụ điển hình là các CSDL của Facebook, Google, Amazon,... được cài đặt trên nhiều máy chủ ở nhiều quốc gia và hỗ trợ khai thác thông tin theo mô hình khách chủ trên nền web.

*Lưu ý:* Cần phân biệt hệ CSDL phân tán với hệ CSDL tập trung nhưng xử lí dữ liệu phân tán. Một ví dụ cụ thể của mô hình CSDL tập trung nhưng xử lí dữ liệu phân tán là mô hình máy chủ tệp. Toàn bộ CSDL và phần mềm ứng dụng CSDL được đặt trên máy chủ tệp. Các máy tính tham gia vào hệ thống được gọi là máy trạm. Mỗi khi có yêu cầu thao tác từ máy trạm, toàn bộ CSDL và phần mềm được chuyển qua đường truyền của mạng về bộ nhớ RAM của máy trạm. Công việc xử lí dữ liệu được thực hiện trên máy trạm, khi kết thúc toàn bộ dữ liệu lại được lưu về máy chủ tệp.

Hệ CSDL tập trung và hệ CSDL phân tán khác nhau như thế nào?

## LUYỆN TẬP

1. Hãy lập danh sách các chức năng của hệ QTCSDL trong từng nhóm chức năng của hệ QTCSDL.
2. Hãy phân tích điểm mạnh và điểm yếu của hệ CSDL phân tán so với hệ CSDL tập trung.

## VẬN DỤNG

1. Cho ví dụ về một hệ CSDL trên thực tế, chỉ rõ những thành phần của nó.
2. Hãy tìm hiểu qua Internet tên một số hệ quản trị CSDL quan hệ được sử dụng phổ biến.

# BÀI 13 CƠ SỞ DỮ LIỆU QUAN HỆ

SAU BÀI HỌC NÀY EM SẼ:

- Hiểu được mô hình CSDL quan hệ.
- Hiểu được các thuật ngữ và khái niệm liên quan: bản ghi, trường (thuộc tính), khoá, khoá chính, khoá ngoài, liên kết dữ liệu.

Trong bài trước các em đã biết khái niệm CSDL. Đã có khá nhiều mô hình CSDL khác nhau. Từ những năm 1970, Edgar Frank Codd (1923 – 2003) đã đề xuất mô hình CSDL quan hệ. Mô hình này nhanh chóng trở thành mô hình được dùng phổ biến nhất, nó xuất hiện trong hầu khắp các ứng dụng quản lí, kể cả trong các ứng dụng thư tín điện tử, mạng xã hội,... Vậy mô hình CSDL quan hệ là gì?

## 1. KHÁI NIỆM CƠ SỞ DỮ LIỆU QUAN HỆ

**Hoạt động 1** **Tìm hiểu về một CSDL thông tin âm nhạc**

Một CSDL các bản nhạc, trên một website âm nhạc, được tổ chức như mô tả trong Hình 13.1, gồm có danh sách các tên nhạc sĩ với mã (định danh) là Aid (Hình 13.1a), danh sách các tên ca sĩ với mã (định danh) là Sid (Hình 13.1b), danh sách các bản nhạc với tên bản nhạc, mã nhạc sĩ (tác giả bản nhạc) và mã Mid – định danh bản nhạc (Hình 13.1c), danh sách các bản thu âm gồm có mã bản nhạc và mã ca sĩ (Hình 13.1d). Hãy quan sát và trả lời các câu hỏi sau:

1. Nhạc sĩ sáng tác bản nhạc "Trường ca sông Lô" là nhạc sĩ nào? Nhạc sĩ sáng tác bản nhạc "Xa khơi" là nhạc sĩ nào?
2. Bản thu âm trong Hình 13.1d tương ứng với dòng 0004 TN là bản thu âm của bản nhạc nào, do ca sĩ nào thể hiện?

**Nhạc sĩ**

| Aid | TenNS |
|---|---|
| 1 | Đỗ Nhuận |
| 2 | Văn Cao |
| 3 | Hoàng Việt |
| 4 | Nguyễn Tài Tuệ |

*a)*

**Ca sĩ**

| Sid | TenCS |
|---|---|
| TK | Trần Khánh |
| LD | Lê Dung |
| TN | Tân Nhân |
| QH | Quốc Hương |

*b)*

**Bản nhạc**

| Mid | Aid | TenBN |
|---|---|---|
| 0001 | 1 | Du kích Sông Thao |
| 0002 | 2 | Trường ca Sông Lô |
| 0003 | 3 | Tình ca |
| 0004 | 4 | Xa khơi |
| 0005 | 1 | Việt Nam quê hương tôi |
| 0006 | 2 | Tiến về Hà Nội |

*c)*

**Bản thu âm**

| Mid | Sid |
|---|---|
| 0001 | TK |
| 0002 | LD |
| 0003 | TK |
| 0004 | TN |
| 0005 | QH |
| 0006 | QH |

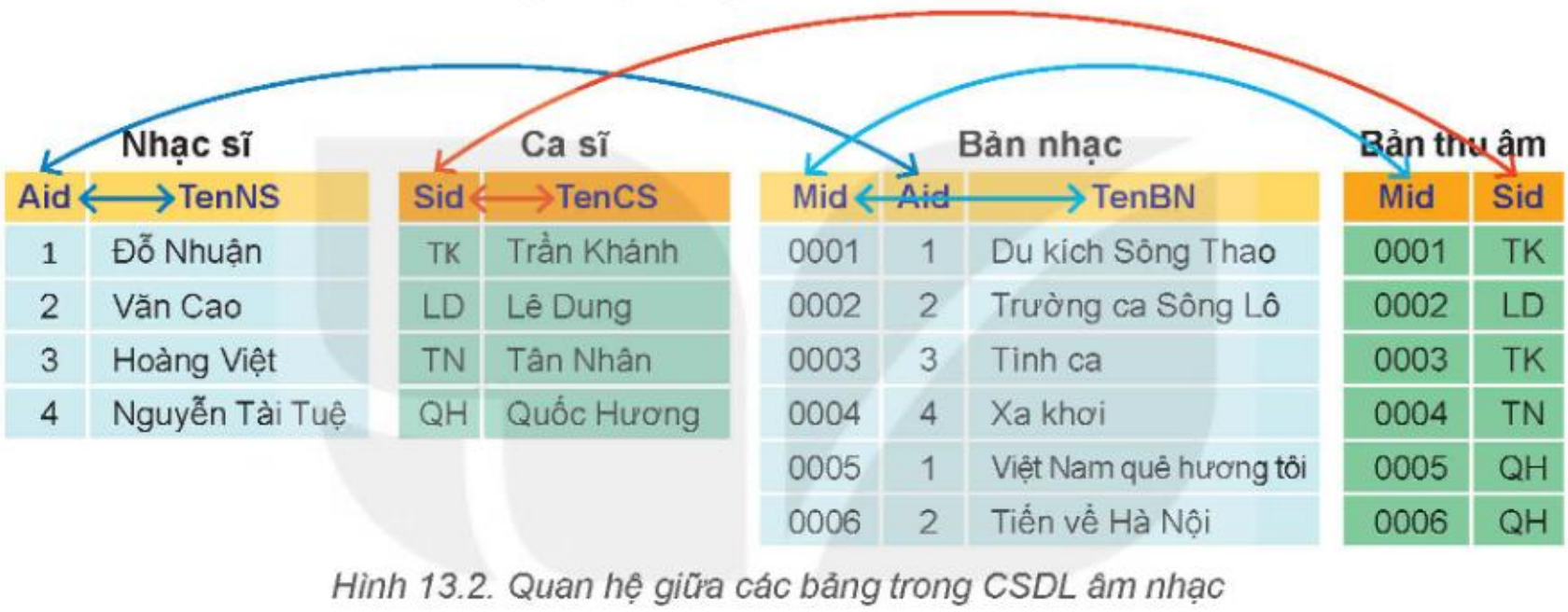
*d)*

*Hình 13.1. CSDL âm nhạc*

CSDL này được xây dựng nhằm đáp ứng nhu cầu tìm kiếm các bản nhạc, bản thu âm lưu trữ trong hệ thống theo tên bản nhạc, tên nhạc sĩ hay tên ca sĩ. Có thể thấy tất cả các dữ liệu của CSDL này được tổ chức ở dạng các bảng gồm các hàng và các cột. Mỗi bảng lưu trữ dữ liệu của các đối tượng có các thuộc tính (được hiểu là các đặc tính xác định đối tượng) giống nhau, mỗi đối tượng một hàng. Dữ liệu của mỗi thuộc tính được ghi trong một cột. Ví dụ bảng Nhạc sĩ lưu trữ những thông tin liên quan đến nhạc sĩ với các cột thuộc tính là mã định danh (Aid) và tên nhạc sĩ (TenNS). Còn bảng Ca sĩ mang thông tin của các ca sĩ với các cột thuộc tính là mã định danh (Sid) và tên ca sĩ (TenCS).

Mỗi bảng có quan hệ với một số bảng còn lại thông qua một thuộc tính nào đó: bảng Bản nhạc và bảng Nhạc sĩ có chung thuộc tính Aid, bảng Bản thu âm và bảng Bản nhạc có chung thuộc tính Mid.

Từ các phân tích trên, có thể thấy Hình 13.2 chính là tóm tắt cách tìm câu trả lời cho các câu hỏi nêu trong Hoạt động 1.



**Nhạc sĩ**

| Aid | TenNS |
|---|---|
| 1 | Đỗ Nhuận |
| 2 | Văn Cao |
| 3 | Hoàng Việt |
| 4 | Nguyễn Tài Tuệ |

**Ca sĩ**

| Sid | TenCS |
|---|---|
| TK | Trần Khánh |
| LD | Lê Dung |
| TN | Tân Nhân |
| QH | Quốc Hương |

**Bản nhạc**

| Mid | Aid | TenBN |
|---|---|---|
| 0001 | 1 | Du kích Sông Thao |
| 0002 | 2 | Trường ca Sông Lô |
| 0003 | 3 | Tình ca |
| 0004 | 4 | Xa khơi |
| 0005 | 1 | Việt Nam quê hương tôi |
| 0006 | 2 | Tiến về Hà Nội |

**Bản thu âm**

| Mid | Sid |
|---|---|
| 0001 | TK |
| 0002 | LD |
| 0003 | TK |
| 0004 | TN |
| 0005 | QH |
| 0006 | QH |

*Hình 13.2. Quan hệ giữa các bảng trong CSDL âm nhạc*

Mô hình tổ chức dữ liệu thành các bảng dữ liệu của các đối tượng có các thuộc tính giống nhau, có thể có quan hệ với nhau theo cách tương tự như trên được gọi *mô hình dữ liệu quan hệ* và các CSDL tương ứng được gọi là *CSDL quan hệ*.

Sau đây, chúng ta sẽ chỉ xem xét các CSDL quan hệ. Vì vậy để cho ngắn gọn, nếu không gây nhầm lẫn, ta sẽ dùng thuật ngữ CSDL thay cho CSDL quan hệ.

> CSDL quan hệ là CSDL lưu trữ dữ liệu dưới dạng các bảng có quan hệ với nhau.

**1.** Hãy chỉ ra các cột của bảng Bản nhạc.

**2.** Bảng Bản thu âm và bảng Ca sĩ có chung thuộc tính nào?

## 2. MỘT SỐ THUẬT NGỮ, KHÁI NIỆM LIÊN QUAN

### a) Bản ghi, trường

Về mặt cấu trúc, CSDL quan hệ tổ chức lưu trữ dữ liệu dưới dạng các *bảng* gồm các hàng và cột. Mỗi hàng của bảng được gọi là một *bản ghi* (record), là tập hợp các thông tin về một đối tượng cụ thể được quản lí trong bảng. Mỗi cột trong bảng được gọi là *trường* (field) thể hiện *thuộc tính* của đối tượng được quản lí trong bảng.

Nói cách khác mỗi hàng là một bản ghi chép dữ liệu cho một đối tượng, gồm một bộ giá trị ghi trong các trường tương ứng với các thuộc tính của đối tượng.

Ví dụ, bảng Nhạc sĩ có bốn bản ghi là (1, Đỗ Nhuận), (2, Văn Cao), (3, Hoàng Việt), (4, Nguyễn Tài Tuệ). Bảng này có hai trường dữ liệu là Aid và TenNS. Cũng nói mỗi bản ghi của bảng này có hai thuộc tính là Aid và TenNS.

Nhờ sự nhất quán về cấu trúc như thế, việc thực hiện các *thao tác dữ liệu* (cập nhật, truy xuất dữ liệu) sẽ có nhiều thuận lợi.

### b) Khoá chính

Mỗi bảng có thể có một hay một nhóm trường mà giá trị của chúng tại các bản ghi không trùng nhau, xác định duy nhất một bản ghi, nói cách khác là bộ giá trị của chúng cho phép xác định phân biệt các bản ghi của bảng. Trường hay nhóm trường ấy được gọi là *khoá* của bảng.

Ví dụ, ở bảng Nhạc sĩ, trường Aid có giá trị phân biệt tại mỗi bản ghi của bảng, nó là khoá của bảng Nhạc sĩ.

Một bảng có thể có nhiều khoá. Người ta có thể chọn (chỉ định) một khoá trong các khoá đó làm *khoá chính* của bảng và thường chọn khoá có số trường ít nhất.

Ví dụ, ở bảng Bản nhạc, nhóm ba trường Mid, Aid, TenBN có giá trị không trùng nhau tại các bản ghi, chúng làm thành một khoá của bảng Bản nhạc. Mỗi bản nhạc cũng được phân biệt bởi Aid và TenBN, vì vậy nhóm hai trường Aid và TenBN cũng làm thành một khoá của bảng Bản nhạc. Mỗi giá trị của trường Mid cũng xác định phân biệt một bản nhạc vì vậy Mid cũng là một khoá của bảng Bản nhạc. Để chọn khoá chính của bảng này, rõ ràng chọn Mid là tốt hơn cả. Khoá chính xác định duy nhất bản ghi nên có vai trò quan trọng trong sắp xếp, truy xuất dữ liệu.

### c) Khoá ngoài

Mỗi bảng (A) có thể có trường hay nhóm các trường (***k***) làm thành khoá chính ở một bảng khác (B). Khi đó ***k*** được gọi là *khoá ngoài* của bảng A. Hai bảng A và B được gọi là có *quan hệ* với nhau qua khoá ngoài ***k*** của bảng A.

Ví dụ: Trường Aid trong bảng Bản nhạc là một khoá ngoài của bảng này vì Aid là khoá chính ở bảng Nhạc sĩ.

### d) Liên kết dữ liệu

Có thể dùng khoá ngoài của bảng để thực hiện ghép nối dữ liệu hai bảng với nhau. Người ta gọi việc ghép nối như thế là liên kết (join) dữ liệu theo khoá. Ví dụ: Có thể liên kết bảng Bản nhạc với Nhạc sĩ theo trường Aid để biết được tên nhạc sĩ sáng tác bản nhạc.

**Nhạc sĩ**

| Aid | TenNS |
|---|---|
| 1 | Đỗ Nhuận |
| 2 | Văn Cao |
| 3 | Hoàng Việt |
| 4 | Nguyễn Tài Tuệ |

**Bản nhạc**

| Mid | Aid | TenBN |
|---|---|---|
| 0001 | 1 | Du kích Sông Thao |
| 0002 | 2 | Trường ca Sông Lô |
| 0003 | 3 | Tình ca |
| 0004 | 4 | Xa khơi |
| 0005 | 1 | Việt Nam quê hương tôi |
| 0006 | 2 | Tiến về Hà Nội |

**Bảng NB**

| Mid | Aid | TenNS | TenBN |
|---|---|---|---|
| 0001 | 1 | Đỗ Nhuận | Du kích Sông Thao |
| 0002 | 2 | Văn Cao | Trường ca Sông Lô |
| 0003 | 3 | Hoàng Việt | Tình ca |
| 0004 | 4 | Nguyễn Tài Tuệ | Xa khơi |
| 0005 | 1 | Đỗ Nhuận | Việt Nam quê hương tôi |
| 0006 | 2 | Văn Cao | Tiến về Hà Nội |

*Hình 13.3. Mô tả liên kết dữ liệu hai bảng Nhạc sĩ và Bản nhạc*

Nếu liên kết bảng Bản thu âm với bảng Ca sĩ theo khoá Sid và liên kết với Bảng NB theo khoá Mid, sẽ có được kết quả là một bảng với dữ liệu đầy đủ nhất.

**Bản thu âm**

| Mid | Sid |
|---|---|
| 0001 | TK |
| 0002 | LD |
| 0003 | TK |
| 0004 | TN |
| 0005 | QH |
| 0006 | QH |

**Ca sĩ**

| Sid | TenCS |
|---|---|
| TK | Trần Khánh |
| LD | Lê Dung |
| TN | Tân Nhân |
| QH | Quốc Hương |

**Bảng NB**

| Mid | Aid | TenNS | TenBN |
|---|---|---|---|
| 0001 | 1 | Đỗ Nhuận | Du kích Sông Thao |
| 0002 | 2 | Văn Cao | Trường ca Sông Lô |
| 0003 | 3 | Hoàng Việt | Tình ca |
| 0004 | 4 | Nguyễn Tài Tuệ | Xa khơi |
| 0005 | 1 | Đỗ Nhuận | Việt Nam quê hương tôi |
| 0006 | 2 | Văn Cao | Tiến về Hà Nội |

**Bảng NBC**

| Mid | Sid | TenCS | Aid | TenNS | TenBN |
|---|---|---|---|---|---|
| 0001 | TK | Trần Khánh | 1 | Đỗ Nhuận | Du kích Sông Thao |
| 0002 | LD | Lê Dung | 2 | Văn Cao | Trường ca Sông Lô |
| 0003 | TK | Trần Khánh | 3 | Hoàng Việt | Tình ca |
| 0004 | TN | Tân Nhân | 4 | Nguyễn Tài Tuệ | Xa khơi |
| 0005 | QH | Quốc Hương | 1 | Đỗ Nhuận | Việt Nam quê hương tôi |
| 0006 | QH | Quốc Hương | 2 | Văn Cao | Tiến về Hà Nội |

*Hình 13.4. Mô tả liên kết dữ liệu ba bảng*

Quan sát Bảng NBC (Bản thu âm – Bản nhạc – Ca sĩ), em lại thấy những điều quen thuộc: Bản thu âm thứ hai là bản thu âm do ca sĩ Lê Dung thể hiện bản nhạc Trường ca Sông Lô của nhạc sĩ Văn Cao, bản thu âm thứ năm là bản thu âm ca sĩ Quốc Hương thể hiện bản nhạc Việt Nam quê hương tôi của nhạc sĩ Đỗ Nhuận.

Lí do tạo CSDL âm nhạc với bốn bảng Nhạc sĩ, Ca sĩ, Bản nhạc, Bản thu âm như ở Hình 13.1 thay vì chỉ lập một bảng tương tự Bảng NBC có thể được nêu ngắn gọn như sau:

- Nếu chỉ lập một bảng như Bảng NBC, dữ liệu tên các ca sĩ, tên các nhạc sĩ và tên các bản nhạc có thể lặp lại nhiều lần với độ dài xâu kí tự khá lớn, gây ra dư thừa dữ liệu, tiêu tốn không gian lưu trữ và dễ mắc lỗi làm mất tính nhất quán của dữ liệu: chẳng hạn ở dòng trên ghi là Văn Cao, dòng dưới là Van Cao,... Mỗi lần sửa đổi tên một ca sĩ chẳng hạn cần phải tầm soát sửa đổi ở rất nhiều dòng khác.
- Bằng cách lập bốn bảng Nhạc sĩ, Ca sĩ, Bản nhạc, Bản thu âm như ở Hình 13.1, mỗi lần thay đổi một tên nhạc sĩ hay một tên ca sĩ ta chỉ cần sửa đổi một lần ở bảng Nhạc sĩ hay bảng Ca sĩ và khi thực hiện liên kết sẽ có được Bảng NBC với đầy đủ dữ liệu đã sửa đổi.

## e) Các trường và dữ liệu

Hãy cùng xem xét một bảng phức tạp hơn - bảng điểm thi tiếng Anh ở một trung tâm ngoại ngữ.

**Bảng 13.1.** *Bảng điểm thi tiếng Anh*

| SBD | Họ và tên | Giới tính | Ngày sinh | Điểm | Mã kì thi |
|---|---|---|---|---|---|
| 1 | Phan Việt An | Nam | 17/10/2005 | 6,25 | A1001 |
| 2 | Trần Duy Anh | Nam | 11/09/2004 | 8,00 | A1001 |

| | | | | | |
|---|---|---|---|---|---|
| 3 | Bùi Thị Hương | Nữ | 20/03/2006 | 9,50 | A1001 |
| 4 | Đỗ Hồng Hoa | Nữ | 15/09/2005 | 7,75 | A1001 |
| 5 | Nguyễn Văn Khoa | Nam | 22/11/2003 | 7,25 | A1001 |
| | ... | | | | |

Giả sử người ta chỉ thiết lập một bảng trong CSDL để lưu trữ dữ liệu từ bảng điểm trên, với tên bảng là bangdiem và các trường tương ứng là sbd, hoten, gt, ngaysinh, diem, makithi. Chúng ta cùng phân tích các đặc điểm dữ liệu trong các trường này.

- Trường sbd có các giá trị là các số nguyên.
- Trường hoten có các giá trị là xâu kí tự, độ dài giới hạn, ví dụ không quá 64 kí tự.
- Trường gt chỉ có hai giá trị là Nữ hoặc Nam, có thể xem nó là trường có giá trị lôgic là đúng (1) hoặc sai (0) tương ứng thay thế cho các giá trị là Nữ hoặc Nam.
- Trường ngaysinh có các giá trị là ngày/tháng/năm.
- Trường diem có các giá trị kiểu số thập phân có tối đa 5 chữ số với 2 chữ số sau dấu phẩy thập phân.
- Trường makithi có các giá trị kiểu xâu với độ dài 5 kí tự.

Những phân tích trên cho thấy mỗi trường có các dữ liệu cùng một kiểu. Chúng được gọi là kiểu dữ liệu của trường. Trường sbd có kiểu số nguyên, trường hoten có kiểu xâu kí tự độ dài không quá 64 kí tự, trường gt có kiểu lôgic, trường ngaysinh có kiểu ngày tháng, trường diem có kiểu số thập phân, trường makithi có kiểu xâu kí tự độ dài cố định 5 kí tự.

Việc xác định kiểu dữ liệu của các trường có mục đích:

- Hạn chế việc lãng phí dung lượng lưu trữ dữ liệu.
- Kiểm soát tính đúng đắn về lôgic của dữ liệu được nhập vào bảng. Chẳng hạn sẽ không thể nhập các kí tự dạng chữ vào trường sbd, không thể nhập giá trị số vào trường ngaysinh,...



1. Hãy chỉ ra khoá chính của bảng Ca sĩ và bảng Bản nhạc.
2. Hãy chỉ ra các khoá ngoài của bảng Bản nhạc và bảng Bản thu âm.



## LUYỆN TẬP

Cho CSDL học tập có các bảng sau: Hocsinh (họ tên, số CCCD, số thẻ học sinh, ngày sinh, địa chỉ), monhoc (tên, mã môn), Diem (số thẻ học sinh, mã môn, năm, học kì, loại điểm, điểm), trong đó loại điểm chỉ các loại ĐĐG thường xuyên, ĐĐG giữa kì, ĐĐG cuối kì.

Hãy xác định các khoá chính và các khoá ngoài của từng bảng, có thể lấy số CCCD làm khoá chính được không.



## VẬN DỤNG

Trong kì thi tốt nghiệp trung học phổ thông, học sinh được đánh số báo danh, có thể thi một số môn, được chia vào các phòng thi được đánh số, sau khi chấm sẽ có điểm thi với các môn đăng kí dự thi. Em hãy đề xuất một số bảng dữ liệu và các trường làm khoá chính và khoá ngoài cho các bảng đó.

# BÀI 14 SQL - NGÔN NGỮ TRUY VẤN CÓ CẤU TRÚC

SAU BÀI HỌC NÀY EM SẼ:

- Hiểu được ở mức nguyên lí: CSDL và các bảng được tạo lập, được thêm mới, cập nhật và truy xuất dữ liệu qua SQL.

Ở bài trước các em đã biết hệ QTCSDL với vai trò là một bộ phần mềm hỗ trợ khởi tạo, cập nhật, truy xuất CSDL để người dùng có thể cập nhật, truy xuất CSDL. Ngày nay người ta thực hiện công việc đó chủ yếu thông qua ngôn ngữ truy vấn có cấu trúc SQL. Sự khác biệt của việc sử dụng SQL so với việc truy xuất dữ liệu bằng ngôn ngữ lập trình là gì?

## 1. LỢI ÍCH CỦA NGÔN NGỮ TRUY VẤN

**Hoạt động 1** **Thảo luận về hai cách truy xuất dữ liệu**

Để lấy danh sách các bản nhạc do nhạc sĩ Văn Cao (mã định danh Aid = 1), sáng tác trong bảng dữ liệu Bản nhạc, ta có thể thực hiện theo một trong hai cách sau:

– Dùng một ngôn ngữ lập trình, viết chương trình mở tệp chứa bảng dữ liệu Bản nhạc, rồi lần lượt lấy ra từng nhóm dữ liệu liên quan đến từng bản nhạc, sau đó tách phần Aid để kiểm tra, nếu Aid = 1 thì đưa ra tên bản nhạc (TenBN).

– Dùng ngôn ngữ truy vấn, viết "CHỌN TenBN TỪ Bản nhạc VỚI Aid = 1" rồi gửi cho hệ QTCSDL thực hiện.

Sự khác biệt cơ bản trong cách truy vấn nhờ ngôn ngữ truy vấn so với lập trình trực tiếp theo em là gì?

Với cách thực hiện thứ nhất trong Hoạt động 1, người dùng phải biết rõ cấu trúc tệp dữ liệu, từ đó lập trình lấy ra đoạn dữ liệu liên quan tới từng bản nhạc để xử lí. Việc làm này rất mất công sức, lại dễ nhầm lẫn. Đây là kiểu lập trình "theo thủ tục" vì phải biết rõ thủ tục truy cập dữ liệu để xây dựng thuật toán. Hơn thế nữa, ở một bài toán khác có nội dung tương tự, ví dụ lập danh sách các học sinh có điểm trung bình môn Toán trên 8 lại phải viết lại chương trình với một thủ tục tương tự.

Với cách thực hiện thứ hai trong Hoạt động 1, người dùng chỉ cần viết ra yêu cầu dưới dạng một câu truy vấn – muốn làm gì, chứ không phải nghĩ cách để thực hiện yêu cầu ấy. Mọi việc còn lại sẽ do hệ QTCSDL giải quyết: tiếp nhận yêu cầu ở dạng xâu truy vấn rồi lấy ra kết quả theo đúng yêu cầu.

Ngôn ngữ truy vấn định chuẩn cho việc định nghĩa, cập nhật, truy xuất và điều khiển dữ liệu từ các CSDL quan hệ là SQL (Structured Query Language) được xây

dựng từ những năm 1970. SQL đã trở thành ngôn ngữ truy vấn tiêu chuẩn mà hầu hết các hệ QTCSDL đều sử dụng. Điều đó có nghĩa là chúng ta có thể dùng SQL để thao tác trên hầu hết các hệ QTCSDL phổ biến như Oracle, SQL server, MySQL, PostGreSQL,... SQL có ba thành phần là DDL (Data Definition Language – ngôn ngữ định nghĩa dữ liệu), DML (Data Manipulation Language – ngôn ngữ thao tác dữ liệu) và DCL (Data Control Language – ngôn ngữ kiểm soát dữ liệu).

Chúng ta sẽ sử dụng SQL để minh hoạ cách thức quản trị CSDL.

## 2. KHỞI TẠO CSDL

Thành phần DDL của SQL cung cấp các câu truy vấn khởi tạo CSDL, khởi tạo bảng, thiết lập các khoá, tóm tắt trong các bảng sau.

**Bảng 14.1.** *Các câu truy vấn định nghĩa CSDL*

| Câu truy vấn DDL | Ý nghĩa |
|---|---|
| CREATE DATABASE | Khởi tạo CSDL |
| CREATE TABLE | Khởi tạo bảng |
| ALTER TABLE | Thay đổi định nghĩa bảng |
| PRIMARY KEY | Khai báo khoá chính |
| FOREIGN KEY... REFERENCES... | Khai báo khoá ngoài |

Các kiểu dữ liệu được sử dụng cho các thuộc tính của các bảng trong SQL.

**Bảng 14.2.** *Kiểu dữ liệu*

| Kiểu dữ liệu | Ý nghĩa |
|---|---|
| CHAR (n) hay CHARACTER (n) | Xâu kí tự có độ dài cố định n kí tự, nếu xâu có ít hơn n kí tự, các kí tự trống được thêm vào phía bên phải |
| VARCHAR (n) | Xâu kí tự có độ dài thay đổi, không vượt quá n kí tự |
| BOOLEAN | Kiểu lôgic có giá trị Đúng (1) hay Sai (0) |
| INT hay INTEGER | Số nguyên |
| REAL | Số thực dấu phẩy động |
| DATE | Ngày tháng, dạng 'YYYY-MM-DD' |
| TIME | Thời gian, dạng 'HH:MM:SS' |

Ví dụ: Khởi tạo CSDL âm nhạc, đặt tên là music và khởi tạo các bảng Nhạc sĩ, Bản nhạc có tên tương ứng là nhacsi và bannhac.

```
CREATE DATABASE music;

CREATE TABLE bannhac (
      Mid CHAR (4),
      Aid INT,
      TenBN VARCHAR (128)
);
ALTER TABLE bannhac ADD PRIMARY KEY (Mid);

CREATE TABLE nhacsi (
      Aid INT,
      TenNS VARCHAR (64)
);
```

Khởi tạo CSDL music

Khởi tạo bảng bannhac

Thêm khoá chính Mid cho bảng bannhac

Khởi tạo bảng nhacsi

Các dấu chấm phẩy ";" được dùng để kết thúc câu truy vấn

1. Hãy viết câu truy vấn tạo bảng Ca sĩ như đã mô tả trong Bài 11 với tên bảng là casi.
2. Hãy viết câu truy vấn thêm khoá chính Sid cho bảng casi.

## 3. CẬP NHẬT VÀ TRUY XUẤT DỮ LIỆU

Thành phần DML của SQL cung cấp các câu truy vấn cập nhật và truy xuất dữ liệu. Sau đây là một vài câu truy xuất dữ liệu để minh hoạ.

**Bảng 14.3.** *Câu truy xuất dữ liệu*

| Câu truy xuất dữ liệu | Ý nghĩa |
|---|---|
| `SELECT <dữ liệu cần lấy>`<br>`FROM <tên bảng>` | <dữ liệu cần lấy> có thể là danh sách các trường hay hàm nào đó với các biến là trường trong bảng |
| `WHERE <điều kiện chọn>` | Chỉ định chọn chỉ các dòng thoả mãn điều kiện xác định |
| `ORDER BY <tên trường>` | Sắp xếp các dòng kết quả theo thứ tự chỉ định |
| `INNER JOIN` | Liên kết các bảng theo điều kiện |

Một số câu truy vấn cập nhật dữ liệu cho trong Bảng 14.4.

**Bảng 14.4.** *Các câu truy vấn cập nhật dữ liệu*

| Câu truy vấn cập nhật dữ liệu | Ý nghĩa |
|---|---|
| `INSERT INTO <tên bảng>`<br>`VALUES <danh sách giá trị>` | Thêm dữ liệu vào bảng <tên bảng> với giá trị lấy từ <danh sách giá trị> |
| `DELETE FROM <tên bảng>`<br>`WHERE <điều kiện>` | Xoá các dòng trong bảng <tên bảng> thoả mãn <điều kiện> |
| `UPDATE <tên bảng>`<br>`SET <tên trường> = <giá trị>` | Cập nhật <giá trị> cho trường có tên là <tên trường> trong bảng <tên bảng> |

Ví dụ:

```
SELECT Mid, TenBN
  FROM bannhac
  WHERE Aid=1
  ORDER by TenBN;
```

Chọn ra từ bảng bannhac các dòng có Aid = 1, ở mỗi dòng chỉ lấy giá trị các cột Mid và TenBN. Sắp xếp các dòng kết quả theo thứ tự TenBN.

```
SELECT bannhac.TenBN, nhacsi.TenNS
  FROM bannhac INNER JOIN nhacsi
  ON bannhac.Aid = nhacsi.Aid;
```

Chọn lấy tất cả các dòng từ bảng bannhac, liên kết với bảng nhacsi theo khoá Aid, ở mỗi dòng lấy cột TenBN ở bảng bannhac và cột TenNS ở bảng nhacsi.

```
INSERT INTO nhacsi VALUES
  (5, 'Phú Quang'),
  (6, 'Phan Huỳnh Điểu');
```

Thêm vào bảng nhacsi hai dòng mới.

```
DELETE FROM bannhac
  WHERE Mid = '0005';
```

Xoá dòng có Mid = '0005' trong bảng bannhac.

```
UPDATE nhacsi
  SET TenNS = 'Hoàng Hiệp'
  WHERE Aid = 6;
```

Thay đổi giá trị cột TenNS thành 'Hoàng Hiệp' ở dòng có cột Aid = 6 trong bảng nhacsi.

Giả sử đã có bảng diemtoan_11A, trong đó có cột tb ghi điểm trung bình môn Toán của tất cả các học sinh lớp 11A. Khi đó câu truy vấn sau lấy ra điểm trung bình cộng môn Toán của tất cả các học sinh lớp 11A.

```
SELECT AVG(tb) FROM diemtoan_11A;
```



1. Hãy viết câu truy vấn lấy tất cả các dòng của bảng nhacsi.
2. Hãy viết câu truy vấn thêm các dòng cho bảng casi với các giá trị là ('TK', 'Nguyễn Trung Kiên'), ('QD', 'Quý Dương'), ('YM', 'Y Moan').

## 4. KIỂM SOÁT QUYỀN TRUY CẬP

Thành phần DCL của SQL cung cấp các câu truy vấn kiểm soát quyền người dùng đối với CSDL, tóm tắt trong Bảng 14.5.

**Bảng 14.5.** *Các câu truy vấn kiểm soát quyền người dùng*

| Mẫu câu truy vấn | Ý nghĩa |
|---|---|
| GRANT | Cấp quyền cho người dùng |
| REVOKE | Thu hồi quyền đối với người dùng |

Ví dụ:

```
GRANT select
    ON music.*
    TO guest ;
```

Cấp quyền dùng truy vấn SELECT đối với tất cả các bảng trong CSDL music cho người dùng guest.

```
REVOKE create, alter
    ON music.bannhac
    FROM mod ;
```

Thu hồi quyền CREATE và ALTER cho bảng bannhac trong CSDL music đối với người dùng mod.



1. Hãy viết câu truy vấn cấp quyền UPDATE đối với tất cả các bảng trong CSDL music cho người dùng mod.
2. Hãy viết câu truy vấn thu hồi quyền DELETE đối bảng nhacsi trong CSDL music cho người dùng mod.



## LUYỆN TẬP

1. Hãy viết câu truy vấn tạo bảng Bản thu âm (banthuam) như đã mô tả trong Bài 13.
2. Viết câu truy vấn tạo khoá ngoài Mid và Sid cho bảng banthuam.
3. Viết câu truy vấn lấy ra tất cả các dòng trong liên kết bảng banthuam với bảng bannhac, mỗi dòng lấy các cột: Mid, Sid của bảng banthuam và cột TenBN của bảng bannhac.



## VẬN DỤNG

Viết câu truy vấn lấy ra tất cả các dòng trong liên kết bảng banthuam với bảng bannhac và bảng casi, mỗi dòng lấy các cột: Mid, Sid của bảng banthuam, cột TenBN của bảng bannhac và TenCS của bảng casi.

# BÀI 15 BẢO MẬT VÀ AN TOÀN HỆ CƠ SỞ DỮ LIỆU

SAU BÀI HỌC NÀY EM SẼ:

- Hiểu được tầm quan trọng và một số biện pháp bảo vệ hệ CSDL.

Mỗi hệ CSDL đều được xây dựng với mục đích xác định nhằm phục vụ một hệ thống quản lí như hệ thống bán vé máy bay, đặt chỗ khách sạn, quản lí bệnh án ở bệnh viện, quản lí kết quả học tập, quản lí website mạng xã hội,... Từng có nhiều thông tin về việc những khối lượng lớn dữ liệu bị đánh cắp, những tài khoản người dùng mạng xã hội bị gán những phát biểu sai trái,... Tình trạng này xảy ra một phần do các hệ CSDL liên quan chưa được bảo vệ đủ tốt. Cần phải làm gì để đảm bảo an ninh, an toàn cho các hệ CSDL?

## 1. BẢO MẬT HỆ CSDL

**Hoạt động 1** **Tổ chức phân quyền với website âm nhạc**

Tất cả người dùng Internet đều có thể được tìm kiếm, được xem danh sách các bản nhạc theo tên bản nhạc, tên ca sĩ, tên nhạc sĩ mà không cần đăng nhập hệ thống. Ngoài ra, một số người dùng xác định có quyền nhập thêm dữ liệu về bản nhạc mới, nhạc sĩ mới và ca sĩ mới.

Theo em, cần phải tổ chức phân quyền truy cập CSDL như thế nào để đáp ứng các yêu cầu trên?

Việc lập danh sách và xác định quyền hạn các nhóm người dùng đối với hệ CSDL chính là công việc đầu tiên cần phải thực hiện để *Xây dựng chính sách bảo mật CSDL*.

Trong trường hợp website âm nhạc, có thể thấy có bốn nhóm người dùng với quyền hạn khác nhau:

- Nhóm 1: Nhóm người dùng không cần khai báo, đăng nhập, được quyền chỉ tìm kiếm, xem, không có quyền cập nhật.
- Nhóm 2: Nhóm người dùng có quyền thêm vào CSDL các bản nhạc mới, tên nhạc sĩ, ca sĩ mới, nhưng không có quyền xoá, sửa dữ liệu.
- Nhóm 3: Nhóm người dùng có quyền xoá, sửa dữ liệu trong các bảng của CSDL, nhưng không có quyền thay đổi cấu trúc bảng, không có quyền xoá bảng.
- Nhóm 4: Nhóm người dùng có toàn quyền đối với các bảng trong CSDL, chính là người dùng có quyền tạo lập các bảng của CSDL.

Người ta sẽ thiết lập một bảng user để quản lí những người có quyền đăng nhập website âm nhạc, bao gồm các cột: tên đăng nhập, mật khẩu và nhóm người dùng.

Mỗi khi người dùng đăng nhập website âm nhạc sẽ biết được họ thuộc nhóm người dùng nào, 2 hay 3 hay 4. Người dùng nhóm 1 không cần đăng nhập website âm nhạc.

Để tất cả người dùng Internet có thể tìm kiếm, xem danh sách các bản nhạc mà không cần đăng nhập website âm nhạc, có thể tạo một tài khoản khách (guest) và cấp cho tài khoản này quyền SELECT đối với tất cả các bảng nhacsi, casi, bannhac, banthuam. Tất cả người dùng khi vào xem website âm nhạc sẽ mặc nhiên được xem như truy xuất CSDL music với tài khoản guest.

Tiếp theo có thể tạo một tài khoản, chẳng hạn là moderator và cấp quyền SELECT, INSERT đối với tất cả các bảng cho moderator. Tất cả những người dùng nhóm này có quyền nhập thêm dữ liệu về bản nhạc mới và ca sĩ mới, sau khi đăng nhập vào website âm nhạc sẽ truy xuất CSDL music với tài khoản này.

Đối với nhóm 3, có thể tạo tài khoản, chẳng hạn là master_mod, có tất cả quyền SELECT, INSERT, UPDATE, DELETE đối với tất cả các bảng nhacsi, casi, bannhac, banthuam trong CSDL music. Tất cả những người dùng nhóm này khi đăng nhập website âm nhạc sẽ dùng tài khoản master_mod để truy xuất, cập nhật CSDL.

Cuối cùng là tài khoản admin có toàn quyền đối với tất cả các bảng trong CSDL music cho người dùng thuộc nhóm 4.

| Tài khoản đăng nhập website | Nhóm 1 (không cần tài khoản) | Nhóm 2 | Nhóm 3 | Nhóm 4 |
|---|---|---|---|---|
| Tài khoản truy xuất CSDL | guest | moderator | master_mod | admin |

Như vậy các nhóm người dùng khác nhau đều có quyền truy xuất CSDL phù hợp, nhưng ngay cả người dùng có quyền cao nhất ở nhóm 4 cũng không thể can thiệp vào các CSDL khác được quản trị trong cùng hệ QTCSDL.

Tuy nhiên, việc đảm bảo an ninh CSDL còn phụ thuộc vào chính ý thức của người dùng. Nếu những người dùng thuộc các nhóm 2, 3, 4 không bảo vệ quyền của mình, để lộ hay để người khác chiếm được tài khoản của mình thì giải pháp bảo mật theo hình thức phân quyền nói trên sẽ hạn chế tác dụng. Nói rộng ra đối với những CSDL cần bảo mật như CSDL ngân hàng, chứng khoán,... nếu người dùng tiết lộ thông tin truy cập thì dữ liệu cũng không còn được bảo mật.

Vì vậy, khi xây dựng chính sách bảo mật CSDL, cần phải bổ sung cả những nội dung liên quan đến ý thức và trách nhiệm của người dùng đối với tài khoản của mình cũng như đối với dữ liệu trong CSDL.

Khi một ứng dụng CSDL hoạt động, nó có thể trở thành một hệ thống với hàng nghìn, hàng trăm nghìn, thậm chí hàng triệu người truy cập đồng thời và trở nên phần nào giống như một hộp đen, người quản trị không thể biết được hết người dùng đang làm gì, các hoạt động đang diễn ra bên trong hệ thống ra sao, có điều gì bất ổn hay có nguy cơ nào rình rập hay không,... Vì vậy, cũng cần có chính sách, với những kế hoạch cụ thể, tổ chức giám sát hoạt động của hệ thống: số người truy cập, tình trạng thiết bị, máy móc,... Những truy xuất của người dùng có thể cần phải được lưu lại dưới dạng biên bản (thường gọi là *log file*) để khi cần có thể kiểm tra, phân tích. Phải xây dựng kế hoạch xử lí các tình huống dự tính có thể xảy ra.

Cuối cùng cần chú ý khi hệ thống máy tính chứa CSDL được kết nối vào mạng cần phải có kế hoạch cụ thể về các giải pháp an ninh mạng (cả về phần cứng và phần mềm) để chống lại các cuộc tấn công qua mạng.

Công tác bảo mật CSDL cần được thực hiện với một chính sách bảo mật toàn diện bao gồm:

- Quy định liên quan đến ý thức và trách nhiệm của người dùng đối với tài khoản của mình và dữ liệu trong CSDL.
- Quy định về tổ chức đảm bảo an ninh mạng cùng với hệ thống phần cứng và phần mềm cụ thể.
- Danh sách các nhóm người dùng và danh sách tài khoản truy xuất CSDL với quyền hạn tương ứng.
- Biện pháp giám sát trạng thái hoạt động của hệ thống, người dùng. Có những quy định về làm biên bản lưu trữ hoạt động của hệ thống và kế hoạch xử lí những tình huống có thể xảy ra.



Nêu tóm tắt các quyền của các tài khoản moderator và admin.

## 2. BẢO ĐẢM AN TOÀN DỮ LIỆU

**Hoạt động 2**

Bảo đảm an toàn dữ liệu là việc đảm bảo để dữ liệu trong CSDL không bị sai lệch, mất mát khi hệ thống phần cứng, phần mềm gặp sự cố rủi ro. Hãy nêu một vài sự cố có thể xảy ra và cách hạn chế, khắc phục các sự cố này.



### a) Sự cố về nguồn điện

- Hệ thống cấp điện không đủ công suất. Giải pháp: Xây dựng hệ thống cấp điện đủ công suất.

- Hệ thống cấp điện bị quá tải do nhu cầu sử dụng điện tăng đột biến. Giải pháp: Thường xuyên kiểm tra hệ thống cấp điện, đặc biệt trong những thời gian nhu cầu sử dụng điện tăng vọt.

- Hệ thống cấp điện ngừng đột ngột vì những lí do khác. Giải pháp: Dùng bộ lưu điện để cấp điện ngay cho hệ thống máy tính quản trị CSDL khi mất điện đột ngột.

### b) Sự cố hư hỏng thiết bị lưu trữ

Các thiết bị lưu trữ (ví dụ ổ đĩa cứng) có khả năng gặp sự cố. Khi đó, nói chung rất khó lấy lại được toàn bộ và chính xác dữ liệu trong thiết bị lưu trữ bị hỏng.

- Thiết bị lưu trữ bị hư hỏng vì quá tuổi thọ. Giải pháp: Quản lí thời gian sử dụng của thiết bị lưu trữ, thay thế trước khi thiết bị đến giai đoạn thường bị hư hỏng.

• Thiết bị lưu trữ bị hư hỏng vì các lí do khác. Giải pháp: Sao lưu dữ liệu định kì. Tất cả các hệ QTCSDL đều hỗ trợ khả năng sao lưu toàn bộ dữ liệu ra thiết bị dự phòng. Dùng giải pháp thiết bị lưu trữ hỗ trợ bảo vệ dữ liệu (các hãng sản xuất thiết bị lưu trữ có những giải pháp để lưu trữ hai phiên bản dữ liệu, kiểm tra chéo để đảm bảo tính chính xác của dữ liệu).

Tóm lại, tuỳ theo những yêu cầu cụ thể của mỗi tổ chức, đặc điểm về CSDL để xây dựng những chính sách đảm bảo an toàn dữ liệu. Trong đó cần quan tâm tới các sự cố có thể xảy ra và giải pháp hạn chế, khắc phục. Chính sách này cũng phải bao gồm những quy định về ý thức, trách nhiệm đối với những người vận hành hệ thống.

- Để đảm bảo an toàn dữ liệu cần xây dựng chính sách an toàn dữ liệu cùng kế hoạch xử lí các sự cố có thể xảy ra và giải pháp hạn chế, khắc phục. Chính sách an toàn dữ liệu cũng phải bao gồm những quy định về ý thức, trách nhiệm đối với người dùng và người vận hành hệ thống.
- Các hệ QTCSDL đều hỗ trợ chức năng sao lưu định kì và phục hồi dữ liệu từ bản sao lưu gần nhất.



Vì sao cần phải sao lưu dữ liệu định kì?



## LUYỆN TẬP

1. Tại sao cần phải có những quy định về ý thức và trách nhiệm của người dùng đối với tài khoản của mình và dữ liệu trong CSDL?
2. Tại sao cần có những quy định về ý thức trách nhiệm của những người vận hành hệ thống?



## VẬN DỤNG

Ở một trung tâm dạy tiếng Anh, có bốn giáo viên dạy bốn kĩ năng là luyện nghe, luyện nói, luyện đọc, luyện viết. CSDL quản lí điểm học tập của học viên có các bảng là diemnghe, diemnoi, diemdoc, diemviet. Các học viên được quyền chỉ xem các bảng điểm, các giáo viên được quyền thêm mới, cập nhật, xoá các bản ghi trong bảng điểm môn học mình dạy, chỉ một người dùng có toàn quyền đối với tất cả các bảng trong CSDL. Hãy xây dựng mô hình phân nhóm người dùng truy cập CSDL nói trên.

Chủ đề
5

# Hướng nghiệp với tin học

BÀI 16

## CÔNG VIỆC QUẢN TRỊ CƠ SỞ DỮ LIỆU

SAU BÀI HỌC NÀY EM SẼ:

- Hiểu được các công việc cần thực hiện cùng các kiến thức và kĩ năng cần thiết để quản trị CSDL, các ngành học có liên quan và nhu cầu xã hội đối với công việc quản trị CSDL.
- Có thể tìm kiếm, khai thác và trao đổi thông tin hướng nghiệp liên quan đến công việc quản trị CSDL.

Trong các tổ chức, những CSDL phục vụ công tác quản lí có thể rất lớn, mỗi CSDL có thể gồm rất nhiều bảng với những quan hệ phức tạp. Không phải ai cũng biết rõ CSDL của đơn vị mình, ngay cả với những nhân viên tin học nếu không tìm hiểu chi tiết. Vậy làm thế nào để có thể vận hành, duy trì cho các CSDL hoạt động thông suốt, luôn sẵn sàng đáp ứng được nhu cầu khai thác? Có cần những cán bộ chuyên trách quản trị CSDL không?

## 1. NHÀ QUẢN TRỊ CƠ SỞ DỮ LIỆU

**Hoạt động 1** Quản trị cơ sở dữ liệu là gì?

Quản trị là hoạt động hỗ trợ song song với một hoạt động cơ bản để thực hiện tốt nhất hoạt động cơ bản. Ví dụ, nếu hoạt động cơ bản là làm phần mềm thì quản trị dự án phần mềm sẽ là điều phối nhân lực, thiết bị, tài chính, giám sát tiến độ và chất lượng để làm ra phần mềm tốt, chi phí hợp lí và đúng thời hạn.

Nếu hoạt động cơ bản là khai thác các CSDL phục vụ ứng dụng tin học thì theo em, hoạt động quản trị CSDL tương ứng gồm những công việc gì?

Quản trị CSDL có mục đích đảm bảo cho các hoạt động liên quan đến CSDL thông suốt, an toàn và có hiệu quả cao. Như vậy, có thể nói vai trò quan trọng nhất liên quan tới công việc quản trị CSDL của một tổ chức là việc đảm bảo để các CSDL trong tổ chức ấy luôn ở trạng thái sẵn sàng khi cần. Nhiệm vụ này thường liên quan đến việc giám sát định kì các hoạt động và xử lí sự cố đối với các CSDL. Theo quy định về ngành nghề, người thực thi hoạt động quản trị CSDL được gọi là *nhà quản trị CSDL* mà trong thực tế thường được gọi đơn giản là người quản trị CSDL. Nhà quản trị CSDL có những nhiệm vụ chính như sau:

- **Cài đặt và cập nhật các hệ QTCSDL**

Nói chung, các hệ QTCSDL thường được các nhà cung cấp cập nhật thường xuyên để bổ sung tính năng và vá lỗi. Bên cạnh đó, theo thời gian, các công nghệ

CSDL có thể lạc hậu, cần được nâng cấp. Thực tế, nhiều cơ quan, tổ chức đã buộc phải loại bỏ các hệ QTCSDL không còn thích hợp để thay thế bằng các hệ QTCSDL có khả năng làm việc tốt hơn. Việc nâng cấp không chỉ là việc cài đặt phần mềm quản trị CSDL mới mà còn phải chuyển đổi CSDL, bao gồm cả chuyển đổi dữ liệu.

- **Tạo lập và điều chỉnh CSDL**

Thiết kế CSDL là một nghề khác có liên quan mật thiết đến quản trị CSDL. Nhà quản trị CSDL có thể phải tham gia tạo lập và điều chỉnh các CSDL phục vụ cho các ứng dụng; lập hồ sơ về CSDL để phục vụ cho hoạt động duy trì và nâng cấp hệ thống sau này.

- **Đảm bảo tài nguyên cho các hoạt động CSDL**

Cần cung cấp đầy đủ tài nguyên như máy tính có công suất thích hợp, có bộ nhớ lưu trữ đủ lớn cùng các phương tiện kĩ thuật kèm theo. Việc này không chỉ được thực hiện khi khởi tạo CSDL mà cần được thực hiện thường xuyên theo mức tăng trưởng, mức độ khai thác dữ liệu và tiến triển của công nghệ CSDL.

Việc đảm bảo tài nguyên là một phần của hoạt động lập kế hoạch phát triển CSDL.

- **Đảm bảo an toàn, bảo mật**

Ở các tổ chức đã tin học hoá, dữ liệu có tầm quan trọng đặc biệt, cần được bảo vệ tốt. Khi hỏng thiết bị, ta có thể sửa chữa hay mua mới; khi phần mềm có lỗi, ta có thể nâng cấp, cài đặt lại; nhưng mất dữ liệu do hỏng thiết bị hay dữ liệu bị đánh cắp, bị sửa với mục đích xấu thì có thể làm tê liệt hoạt động, thậm chí có thể làm sụp đổ toàn bộ tổ chức sở hữu dữ liệu. Do vậy, việc đảm bảo an toàn, bảo mật cho dữ liệu là một hoạt động quan trọng hàng đầu trong quản trị CSDL.

Nhà quản trị CSDL phải phối hợp với những người thiết kế hệ thống và các bộ phận nghiệp vụ để phân quyền sử dụng dữ liệu đến từng nhóm người dùng, sau đó sử dụng hệ QTCSDL để thiết lập quyền truy cập dữ liệu; điều chỉnh quyền truy cập dữ liệu khi có biến động về nhân sự hay nghiệp vụ.

Nhà quản trị CSDL phải thường xuyên tạo các bản sao lưu dữ liệu dự phòng để khi gặp sự cố có thể khôi phục dữ liệu, nhất là các sự cố liên quan đến phần cứng. Thường xuyên giám sát phát hiện sớm các sự cố dữ liệu để khắc phục kịp thời.

Ngoài các công việc chính đã nêu trên, các nhà quản trị CSDL có thể tham gia các hoạt động *nghiên cứu, xây dựng các quy trình và tư vấn* về các vấn đề có liên quan đến CSDL.

Quản trị CSDL là hoạt động nhằm đảm bảo cho việc sử dụng CSDL thông suốt và hiệu quả. Quản trị CSDL có các nhiệm vụ:

- Cài đặt và nâng cấp các hệ QTCSDL;
- Tạo lập, điều chỉnh CSDL;
- Đảm bảo tài nguyên cho các hoạt động CSDL;
- Đảm bảo an toàn và bảo mật cho dữ liệu.



1. Kể ra các nội dung quản trị CSDL.
2. Trong ba nhóm đối tượng những người làm phần mềm, sử dụng phần mềm quản lí và quản trị CSDL, ai là người chịu trách nhiệm chính mỗi công việc sau đây: *Cập nhật dữ liệu*, *Thiết kế dữ liệu*, *Sao lưu dữ liệu*?

## 2. PHẨM CHẤT VÀ NĂNG LỰC CỦA NHÀ QUẢN TRỊ CSDL

**Hoạt động 2** **Tìm hiểu các kiến thức và kĩ năng cần có của nhà quản trị CSDL**

1. Căn cứ vào các công việc cần thực hiện để quản trị CSDL, em hãy đề xuất những kiến thức, kĩ năng và phẩm chất cần có của nhà quản trị CSDL.
2. Có thể học kiến thức và rèn luyện kĩ năng quản trị CSDL ở đâu?

Nhà quản trị CSDL cần có nền tảng tốt về CSDL, hiểu được các mô hình CSDL. Tuy nhiên, kĩ năng làm việc trên các hệ thống cụ thể đặc biệt cần thiết.

Để quản trị CSDL trên thực tế, điều quan trọng nhất là phải làm chủ được một hệ QTCSDL, biết cài đặt, biết thiết lập các tham số cần thiết, thành thạo ngôn ngữ quản trị, trong đó có ngôn ngữ truy vấn SQL - ngôn ngữ có thể làm việc được với tất cả các hệ QTCSDL hiện đại.

Nhà quản trị CSDL có thể phải tham gia thiết kế hoặc điều chỉnh cấu trúc của CSDL khi nâng cấp phần mềm. Cho dù họ không nhất thiết phải tham gia thiết kế CSDL nhưng biết thiết kế là điều rất quan trọng để quản trị tốt các CSDL.

Việc cài đặt hệ QTCSDL bao giờ cũng được thực hiện trên một hệ điều hành cụ thể, có liên quan đến mạng máy tính nên có hiểu biết tốt về hệ điều hành và mạng máy tính là một điểm cộng cho nhà quản trị CSDL.

Nhà quản trị CSDL cần có một số phẩm chất cần thiết sau:

– Tính cẩn thận, tỉ mỉ, kiên trì. Cùng với khả năng phân tích, các phẩm chất này giúp nhà quản trị CSDL phát hiện và xử lí sự cố tốt hơn.

– Tinh thần ham học. Cùng với kĩ năng ngoại ngữ và tìm kiếm thông tin, phẩm chất này giúp nhà quản trị CSDL có khả năng tự học để nắm bắt được những công nghệ mới liên quan đến quản trị CSDL.

Ở bậc đại học, các chuyên ngành có liên quan nhiều đến nghề quản trị CSDL là *Các hệ thống thông tin* và *Công nghệ phần mềm*, ở đó sinh viên được đào tạo chuyên sâu về CSDL cũng như phân tích và thiết kế hệ thống. Hầu hết các trường đại học có đào tạo công nghệ thông tin đều có các chuyên ngành trên. Các kiến thức khác như hệ điều hành và mạng máy tính được học ở mức cơ bản. Một vài trường đại học mở cả chuyên ngành tin học quản lí. Một số trường đại học đào tạo chuyên ngành CSDL có thể không dạy các hệ QTCSDL cụ thể nhưng sinh viên có thể tiếp cận chúng qua các bài thực hành và các dự án nhỏ.

Có nhiều cơ hội để bổ sung kiến thức và rèn luyện kĩ năng quản trị CSDL như tham gia các khoá học của các trung tâm đào tạo, lấy chứng chỉ quản trị CSDL của các hãng nổi tiếng như IBM, Microsoft, Oracle,... hoặc thực tập ở các công ti tin học.

- Nhà quản trị CSDL cần có kiến thức cơ bản về CSDL, biết thiết kế CSDL và sử dụng thành thạo hệ QTCSDL.
- Tính cách tỉ mỉ, cẩn thận, kiên nhẫn, kĩ năng phân tích rất cần thiết với nhà quản trị CSDL khi phải xử lí tình huống.
- Khả năng học tập suốt đời, giúp nhà quản trị CSDL nâng cao bản lĩnh nghề nghiệp.
- Để trở thành nhà quản trị CSDL tốt, cần được học kiến thức một cách bài bản ở các trường đại học chuyên ngành, rèn luyện kĩ năng trong các khoá học nghề nghiệp về quản trị CSDL, về các hệ QTCSDL cụ thể và rèn luyện trong công việc thực tế.

1. Hãy kể ra các phẩm chất và năng lực cần có của một nhà quản trị CSDL.
2. Cần làm gì để có các kiến thức và kĩ năng cần thiết về quản trị CSDL?

# 3. CƠ HỘI VIỆC LÀM

**Hoạt động 3** **Nhu cầu tuyển dụng nhân lực quản trị CSDL**

Em hãy sử dụng cụm từ khoá *tuyển dụng quản trị cơ sở dữ liệu* để tìm kiếm thông tin trên mạng về nhu cầu tuyển dụng liên quan tới công việc quản trị CSDL.

Đã từ lâu, ở những nơi có mức độ tin học hoá cao như các tổ chức tài chính, ngân hàng, bảo hiểm hay các cơ quan nhà nước cung cấp các dịch vụ công, các công ti tin học lớn,... đều cần các nhà quản trị CSDL để đảm bảo các hệ thống ứng dụng hoạt động thông suốt.

Với sự phát triển nhanh chóng của Internet, rất nhiều ứng dụng trực tuyến có quy mô lớn đã và đang được triển khai như các sàn thương mại điện tử, các mạng xã hội, các hệ thống học tập, giải trí có các CSDL lớn và phức tạp. Việc đảm bảo cho các CSDL của các ứng dụng đó luôn hoạt động tốt là một trong những yêu cầu cốt lõi để cung cấp dịch vụ một cách tin cậy. Những nhà quản trị CSDL ở các tổ chức đó rất được trọng dụng.

Cuộc Cách mạng công nghiệp 4.0 có một nội dung quan trọng là chuyển đổi số, theo đó toàn bộ các quy trình nghiệp vụ của tổ chức chuyển đổi số được tin học hoá trên nền tảng những CSDL tích hợp, nhất quán và chia sẻ được. Nhu cầu tuyển dụng nhà quản trị CSDL sẽ tăng nhanh cùng với quá trình chuyển đổi số.

Sở hữu bằng đại học đúng chuyên ngành, có trong tay các chứng chỉ về quản trị CSDL của các hãng danh tiếng, có kinh nghiệm thực tế sử dụng các hệ QTCSDL chắc chắn sẽ mang lại những cơ hội làm việc tại những nơi có điều kiện tốt và thu nhập cao. Các hiểu biết và kĩ năng quản trị CSDL cũng sẽ hết sức hữu ích cho tất cả những ai có nhu cầu phát triển năng lực chuyên môn ở những bậc học cao hơn liên quan tới tin học như Khoa học máy tính, Hệ thống thông tin, Khoa học dữ liệu,...

Nhu cầu nhân lực quản trị CSDL tăng theo nhu cầu phát triển các ứng dụng tin học sử dụng CSDL.

Có thể tìm được rất nhiều địa chỉ tuyển dụng nhờ công cụ tìm kiếm trên Internet.

Tại sao nhu cầu nhân lực về quản trị CSDL ngày một tăng?

## LUYỆN TẬP

1. Hãy tìm danh sách ít nhất 5 trường đại học có đào tạo quản trị CSDL hay tin học quản lí.
2. Hãy tìm một số địa chỉ cung cấp các dịch vụ đào tạo để cấp chứng chỉ về CSDL của các công ti như IBM, Oracle, Microsoft.

## VẬN DỤNG

1. Hãy tìm ba địa chỉ tuyển dụng nhà quản trị CSDL trong ngành ngân hàng.
2. Hãy tìm ba địa chỉ tuyển dụng nhà quản trị CSDL trong các công ti tin học.

Chủ đề 6

BÀI 17

# Kĩ thuật lập trình

# DỮ LIỆU MẢNG MỘT CHIỀU VÀ HAI CHIỀU

SAU BÀI HỌC NÀY EM SẼ:

- Biết cách thiết lập và làm việc với cấu trúc dữ liệu mảng một chiều và hai chiều.

Em đã biết thiết lập cấu trúc dữ liệu đóng vai trò quan trọng khi giải quyết trong các bài toán thực tế trên máy tính. Trong các bài toán thực tế sau em sẽ thiết lập cấu trúc dữ liệu như thế nào?

– Lập danh sách họ tên các bạn lớp em. Sau đó có thể tìm kiếm, sắp xếp và thực hiện các bài toán quản lí khác.

– Giả sử lớp em cần khảo sát ý kiến về một vấn đề nào đó. Mỗi học sinh cần có đánh giá theo 4 mức, kí hiệu lần lượt là Đồng ý (2); Không phản đối (1); Không ý kiến (0); Phản đối (−1). Em sẽ tổ chức dữ liệu khảo sát như thế nào để có thể dễ dàng cập nhật và tính toán theo dữ liệu khảo sát.

– Em được giao nhiệm vụ thiết lập và lưu trữ một danh sách các địa điểm là nơi các bạn trong lớp sẽ thường xuyên đến tham quan và trải nghiệm thực tế. Mỗi địa điểm như vậy cần nhiều thông tin, nhưng thông tin quan trọng nhất là toạ độ (x, y) của thông tin đó trên bản đồ. Em sẽ dùng cấu trúc dữ liệu gì để mô tả danh sách các địa điểm này?

## 1. CẤU TRÚC DỮ LIỆU MẢNG MỘT CHIỀU

Trong Tin học, dữ liệu mảng một chiều, thường được gọi là cấu trúc dữ liệu tuyến tính, là cấu trúc dữ liệu bao gồm một dãy các phần tử dữ liệu có cùng kiểu. Các phần tử của mảng sẽ được truy cập theo chỉ số, cho phép các thao tác đơn giản như truy cập và thay đổi giá trị.

Trong Python, dữ liệu mảng một chiều có thể được biểu diễn bằng danh sách (list) với các phần tử có cùng kiểu dữ liệu.

### Hoạt động 1 Tìm hiểu cấu trúc dữ liệu mảng một chiều trong Python

Thực hiện các thao tác sau, quan sát, trao đổi và thảo luận, từ đó nhận biết về cấu trúc dữ liệu mảng một chiều trong Python.

a) Thiết lập các mảng một chiều, thông qua kiểu dữ liệu list trong Python.

```
>>> A = [1, 3, 5, 7, 9, 11]
>>> B = ["Hà","Bình","Ngọc","Anh"]
>>> C = [9.5, 8.0, 10, 7.2]
```

b) Có thể truy cập từng phần tử của mảng một chiều theo chỉ số. Toán tử **in** kiểm tra một đối tượng có nằm trong mảng hay không. Trong Python, chỉ số bắt đầu từ 0.

```
>>> 7 in A
True
```

c) Có thể dễ dàng duyệt từng phần tử của mảng một chiều bằng lệnh **for**.

```
>>> for hs in B:
    print(hs, end = " ")
Hà Bình Ngọc Anh
```

d) Các lệnh cơ bản làm việc với danh sách trong Python đã được học trong chương trình Tin học 10, ví dụ: **append** (bổ sung phần tử vào cuối), **remove** (xoá một phần tử), **insert** (bổ sung phần tử vào vị trí bất kì), **clear** (xoá toàn bộ các phần tử của danh sách).

*Lưu ý*: Mặc dù mảng được biểu diễn bằng dữ liệu list, nhưng không được đồng nhất khái niệm mảng của khoa học máy tính với kiểu dữ liệu list trong Python.

Cấu trúc dữ liệu mảng một chiều trong Python có thể biểu diễn bằng kiểu dữ liệu danh sách (list) với chức năng truy cập dễ dàng từng phần tử theo chỉ số, truy cập theo vùng chỉ số. Toán tử **in** và lệnh **for .. in** sẽ hỗ trợ duyệt từng phần tử của mảng.

1. Sử dụng hàm sum() tính tổng các số của một dãy, hãy viết câu lệnh tính giá trị trung bình của dãy số A cho trước.
2. Có thể duyệt các phần tử của mảng theo chiều ngược lại, từ cuối về đầu, được không?

## 2. CẤU TRÚC DỮ LIỆU MẢNG HAI CHIỀU

**Hoạt động 2** **Mô hình danh sách trong danh sách của Python**

Thực hiện các lệnh sau, quan sát, trao đổi và thảo luận, từ đó nhận biết về mô hình danh sách trong danh sách của Python.

a) Xét dữ liệu biểu diễn điểm của học sinh trong lớp, mỗi phần tử sẽ bao gồm hai thông tin là tên học sinh và điểm số. Lệnh sau sẽ tạo một bộ dữ liệu như vậy. Ta thấy mỗi phần tử của dãy này lại là một danh sách bao gồm hai phần tử.

```
>>> DS_diem = [["Quang",7.5],["Hà",8.0],["Bình",9.5]]
```

b) Nếu truy cập một phần tử của danh sách, ta nhận được một dãy bao gồm tên và điểm của học sinh tương ứng.

```
>>> DS_diem[1]
['Hà', 8.0]
```

c) Muốn truy cập một giá trị điểm cụ thể, ví dụ điểm của học sinh tên "Quang", cần thực hiện lệnh sau:

```
>>> DS_diem[0][1]
7.5
```

d) Lệnh duyệt các phần tử của dãy gốc có dạng mở rộng như sau:

```
>>> for hs,diem in DS_diem:
    print(hs,diem)
Quang 7.5
Hà 8.0
Bình 9.5
```

Python hỗ trợ mô hình dữ liệu danh sách trong danh sách, tức là mỗi phần tử của danh sách là một đối tượng dạng danh sách khác.



1. Thiết lập mảng bao gồm dữ liệu là toạ độ các điểm trên mặt phẳng, mỗi điểm p được cho bởi hai toạ độ (px, py).
2. Thiết lập mảng bao gồm dãy các thông tin là danh sách học sinh và thông tin ba điểm thi của học sinh tương ứng các bài thi số 1, 2, 3. Viết đoạn lệnh nhập bộ dữ liệu trên và chương trình in ra danh sách học sinh cùng với điểm trung bình của các bài thi.

## Hoạt động 3 Mô hình mảng hai chiều của Python

Đọc, trao đổi, thảo luận về cấu trúc dữ liệu mảng hai chiều trong Python.



a) Trong tin học, một cấu trúc dữ liệu hai chiều được hiểu là một bảng hay còn gọi là ma trận, bao gồm các hàng và cột dữ liệu. Bảng có thể có kích thước vuông n × n hoặc kích thước bất kì m × n.

Ví dụ một bảng (hay ma trận) vuông bậc ba có dạng như sau:

$$A = \begin{pmatrix} 12 & 10 & 91 \\ 11 & 45 & 20 \\ 15 & 34 & 55 \end{pmatrix}$$

Tổng quát mô hình bảng bậc m × n có dạng như sau:

$$A = \begin{pmatrix} a_{11} & a_{12} \dots & a_{1n} \\ a_{21} & a_{22} \dots & a_{2n} \\ \dots\dots\dots\dots\dots & & \\ a_{m1} & a_{m2} \dots & a_{mn} \end{pmatrix}$$

Phần tử nằm tại hàng i và cột j sẽ kí hiệu là $a_{ij}$. Như vậy $a_{ij}$ là phần tử của ma trận tại vị trí hàng i, cột j.

Ma trận kích thước m × n sẽ bao gồm m × n phần tử, có cấu trúc bao gồm m hàng, mỗi hàng có n cột. Trong Python, mô hình ma trận m × n sẽ dễ dàng được biểu diễn bởi cấu trúc danh sách trong danh sách với một mảng có m phần tử, mỗi phần tử đều có dạng danh sách có n phần tử. Ví dụ ma trận vuông A bậc 3 ở trên có thể khai báo trong Python như sau:

```
>>> A = [[12,10,91],[11,45,20],[15,34,55]]
```

Vì mảng trong Python được đánh chỉ số từ 0 nên muốn truy cập phần tử tại hàng 3 cột 2, gõ lệnh:

```
>>> A[2][1]
34
```

b) Với cấu trúc dữ liệu hai chiều, ví dụ như ma trận A bậc 3 ở trên, thao tác duyệt theo từng phần tử sẽ được thực hiện bằng hai lệnh duyệt for lồng nhau. Ví dụ sau cho biết cách duyệt ma trận A có 3 × 3 = 9 phần tử, kết quả thể hiện ma trận A theo đúng khuôn mẫu chuẩn là một bảng gồm 3 hàng, 3 cột.

```
>>> for i in range(3):
        for j in range(3):
            print(A[i][j], end = " ")
        print()
12 10 91
11 45 20
15 34 55
```

Trong Python cấu trúc dữ liệu hai chiều có thể được biểu diễn bằng mô hình danh sách trong danh sách. Có thể truy cập từng phần tử và duyệt cấu trúc dữ liệu hai chiều bằng hai lệnh for lồng nhau.



**1.** Câu lệnh sau sẽ tạo dữ liệu thuộc loại gì?

```
temp = [1,2,3,4]
A = [temp,temp]
```

**2.** Nếu A là bảng (ma trận) kích thước m × n thì đoạn chương trình sau sẽ in ra thông tin gì trên màn hình?

```
>>> for i in range(m):
        for j in range(n):
            print(A[i][j], end = " ")
        print()
```

## LUYỆN TẬP

**1.** Giả sử số đo chiều cao các bạn trong lớp được cho trong dãy số A. Hãy viết đoạn chương trình tính:

- Số đo chiều cao trung bình của cả lớp.
- Số bạn có chiều cao lớn hơn chiều cao trung bình của cả lớp.

**2.** Viết chương trình nhập từ bàn phím số tự nhiên m, sau đó lần lượt nhập m dòng, mỗi dòng bao gồm n số cách nhau bởi dấu cách, đưa dữ liệu đã nhập vào ma trận A, sau đó in ma trận A ra màn hình.

## VẬN DỤNG

**1.** Viết hàm số UnitMatrix(n) với n là số tự nhiên cho trước, hàm trả lại giá trị là ma trận bậc n như Hình 17.1.

$$E = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ & & \dots & & \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix}$$

*Hình 17.1*

**2.** Viết chương trình cho phép người dùng nhập từ bàn phím một dãy số tự nhiên, hãy đếm với mỗi giá trị của dãy có bao nhiêu số lặp lại. Ví dụ nếu dãy ban đầu là:

```
0 1 5 7 0 2 5 1 1 2
```

thì chương trình cần thông báo như Hình 17.2.

```
Số 0 lặp lại 2 lần
Số 1 lặp lại 3 lần
Số 5 lặp lại 2 lần
Số 7 lặp lại 1 lần
Số 2 lặp lại 2 lần
```

*Hình 17.2*

**3.** Em ghi số tiền điện gia đình em theo từng tháng vào một danh sách gồm 12 số. Mỗi năm lại ghi lại số tiền điện vào một danh sách và ghép với danh sách các năm trước. Như vậy em thu được một bảng kích thước n × 12, trong đó hàng thứ k là số tiền điện của năm thứ k, cột tương ứng số tiền điện theo tháng.

a) Thiết lập mảng mới tính số tiền điện trung bình của các năm, mỗi năm ghi một số.

b) Tính số tiền điện trung bình của tất cả các năm đã được ghi dữ liệu trong bảng.

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

# BÀI 18 THỰC HÀNH DỮ LIỆU MẢNG MỘT CHIỀU VÀ HAI CHIỀU

SAU BÀI HỌC NÀY EM SẼ:

- Sử dụng được mảng một chiều và hai chiều trong lập trình.
- Sử dụng được kiểu dữ liệu list trong một vài bài toán cụ thể.

Trong bài học trước, em đã biết cấu trúc dữ liệu mảng có thể giúp giải quyết tốt các bài toán quản lí danh sách trong thực tế. Việc quản lí danh sách điểm kiểm tra của một môn học là quản lí danh sách học sinh và mỗi học sinh có một danh sách điểm kiểm tra. Bảng điểm môn học cả lớp được coi là danh sách của danh sách điểm của các học sinh. Theo em, người ta dùng mảng một chiều hay hai chiều để quản lí danh sách điểm một môn học của mỗi học sinh? để quản lí điểm một môn học của cả lớp?

## Nhiệm vụ 1

Viết chương trình quản lí điểm kiểm tra một môn học của một học sinh trong một học kì.

Chương trình được thực hiện như sau:

– Nhập điểm: yêu cầu người dùng nhập các đầu điểm kiểm tra (từ hai đầu điểm trở lên).

– Thống kê điểm: chương trình duyệt qua các đầu điểm rồi tính và in ra điểm trung bình kiểm tra, điểm thấp nhất, cao nhất.

### Hướng dẫn:

*Phân tích:* Nhiệm vụ này có thể được thực hiện bằng cách sử dụng mảng một chiều, cụ thể là sử dụng danh sách trong Python. Việc nhập điểm được thực hiện ngay trên màn hình. Sau khi đã có danh sách các đầu điểm, dùng cấu trúc for để duyệt qua các phần tử của mảng để tính tổng điểm, thống kê điểm cao nhất, thấp nhất.

**qldiem.py**

```
1  marks = []
2  line = input("Hãy nhập các điểm kiểm tra cách nhau bởi dấu cách: ")
3  marks = [float(x) for x in line.split()]
4  total = 0
5  Min = marks[0]
6  Max = marks[0]
7  for m in marks:
8      total = total + m
9      if Min > m:
10         Min = m
11     if Max < m:
```

```
12          Max = m
13 print("Điểm trung bình: ", total/len(marks))
14 print("Điểm cao nhất: ", Max)
15 print("Điểm thấp nhất: ", Min)
```



## Nhiệm vụ 2

Viết chương trình quản lí điểm kiểm tra một môn học trong một học kì của tất cả học sinh trong lớp.

Chương trình được thực hiện như sau:

– Nhập dữ liệu: yêu cầu người dùng nhập số học sinh trong lớp, sau đó với mỗi học sinh hỏi người dùng nhập tên học sinh rồi nhập các đầu điểm của học sinh đó.

– Thống kê dữ liệu: chương trình in ra danh sách các học sinh với điểm trung bình kiểm tra của họ, tên học sinh có điểm trung bình cao nhất và điểm kiểm tra thấp nhất trong tất cả các đầu điểm.

**Hướng dẫn:**

*Phân tích:* Nhiệm vụ này có thể được thực hiện bằng cách sử dụng một mảng hai chiều để lưu tất cả điểm của học sinh trong lớp, mỗi hàng là điểm của một học sinh, điểm cụ thể là các phần tử của hàng. Để thống kê, cần dùng vòng lặp để duyệt qua từng hàng, tính trung bình từng hàng và xét từng điểm để tìm ra đầu điểm thấp nhất. Trong khi tính, tìm điểm trung bình kiểm tra của từng học sinh, cần lưu lại số thứ tự của học sinh có điểm trung bình cao nhất để tham chiếu in ra tên của học sinh đó.

**qldiem_ca_lop.py**

```
1  names = []
2  marks_all = []
3  n = int(input("Hãy nhập số lượng học sinh: "))
4
5  for i in range(n):
6      name = input("Tên học sinh thứ " + str(i+1) + ": ")
7      names.append(name)
8      line = input("Điểm kiểm tra của học sinh thứ "+str(i+1)+": ")
9      marks = [float(x) for x in line.split()]
10     marks_all.append(marks)
11
12# Định nghĩa các biến để thống kê
13 max_avg = 0
14 index_max_avg = 0
15 Min = marks_all[0][0]
16
17#Duyệt qua mảng 2 chiều
18 for i in range(n):
19     total = 0
20     for j in range(len(marks_all[i])):
21         total = total + marks_all[i][j]
```

```
22          if Min > marks_all[i][j]:
23              Min = marks_all[i][j]
24      avg = total/len(marks_all[i])
25      print("Điểm trung bình của",names[i],":",avg)
26      if max_avg < avg:
27          max_avg = avg
28          index_max_avg = i
29
30 #In ra kết quả
31 print(names[index_max_avg], "đạt điểm trung bình", max_avg,"cao nhất lớp.")
32 print("Điểm thấp nhất:",Min)
```



## LUYỆN TẬP

**1.** Chỉnh sửa lại chương trình của Nhiệm vụ 1 để bổ sung chức năng:

a) Thông báo điểm đầu tiên và điểm cuối cùng trong danh sách.

b) Cho phép người dùng tra cứu đầu điểm thứ n với quy ước n bắt đầu từ 1 ứng với điểm đầu tiên. Nếu n lớn hơn tổng số đầu điểm hoặc nhỏ hơn 1, cần thông báo không hợp lệ và yêu cầu người dùng nhập lại.

**2.** Chỉnh sửa lại chương trình để người dùng có thể:

a) Tra cứu các đầu điểm kiểm tra theo STT (số thứ tự) của học sinh. Quy ước số thứ tự bắt đầu từ 1. Nếu người dùng nhập STT lớn hơn số lượng học sinh thì chương trình thông báo STT không hợp lệ và yêu cầu nhập lại.

b) Tra cứu điểm kiểm tra cụ thể lần thứ n của một học sinh theo STT. Nếu n và STT không hợp lệ chương trình cần thông báo và yêu cầu nhập lại.



## VẬN DỤNG

**1.** Viết chương trình nhập vào từ bàn phím danh sách tên (không gồm họ và đệm) học sinh cách nhau bởi dấu cách và lưu vào trong một mảng. Giả thiết rằng tên không gồm khoảng trắng. Sau đó hãy thống kê xem có bao nhiêu tên khác nhau và mỗi tên xuất hiện bao nhiêu lần trong danh sách.

**2.** Viết chương trình nhập từ bàn phím số tự nhiên m và n. Sau đó lần lượt nhập m dòng, mỗi dòng bao gồm n số cách nhau bởi dấu cách. Đưa dữ liệu đã nhập vào ma trận A, in ma trận A ra màn hình. Sau đó:

a) Tính tổng các phần tử ma trận A.

b) In ra dòng có tổng các phần tử lớn nhất (nếu có nhiều dòng bằng nhau thì in tất cả các dòng).

c) In ra giá trị các phần tử phân biệt trong ma trận tức là nếu có các giá trị xuất hiện nhiều lần trong ma trận A thì chỉ in ra một lần.

d) Cho phép người dùng tìm số lần xuất hiện của một số bất kì trong ma trận A, ví dụ người dùng nhập vào số 3, chương trình thông báo số 3 xuất hiện x lần trong ma trận tại các vị trí (i, j) cụ thể.

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

# BÀI 19 BÀI TOÁN TÌM KIẾM

SAU BÀI HỌC NÀY EM SẼ:

- Biết được ý nghĩa của bài toán tìm kiếm trên thực tế.
- Biết và thực hiện được chương trình tìm kiếm tuần tự và tìm kiếm nhị phân.

**Trò chơi lật thẻ**

Giả sử có một bộ thẻ, trên mỗi thẻ in một số bất kì. Các thẻ được xếp úp mặt xuống bàn theo thứ tự tăng dần của các số ghi trên thẻ. Người chơi mỗi lần chỉ được lật một thẻ để xem giá trị số in trên đó. Nếu giá trị số in trên thẻ lật lên bằng số K cho trước thì trò chơi kết thúc. Bạn An đã chơi bằng cách lật lần lượt từng thẻ từ đầu đến cuối. Theo em, An có chắc chắn xác định được thẻ nào in số K không? Em có cách nào xác định được thẻ in số K nhanh hơn An không?

## 1. BÀI TOÁN TÌM KIẾM TRÊN THỰC TẾ

**Hoạt động 1** Bài toán tìm kiếm

Với các bài toán tìm kiếm sau, hãy thảo luận về miền dữ liệu và khả năng các kết quả có thể tìm được của bài toán:

Bài toán 1. Em cần tìm hình ảnh các cây hoa hồng đẹp trên Internet để đưa vào bài trình bày về cách trồng hoa.

Bài toán 2. Em cần tìm một tệp văn bản có tên bai-hoc-1.docx trên máy tính của em nhưng đã lâu rồi chưa sử dụng lại.

Bài toán 3. Em cần tìm 5 bạn học sinh có điểm trung bình các bài thi cao nhất trong kì thi Olympic Tin học năm 2020 của thành phố Hà Nội.

– Với bài toán 1, miền dữ liệu là tất cả các ảnh có trên các máy tính kết nối mạng Internet. Kết quả là các ảnh có hình hoa hồng.

– Với bài toán 2, miền dữ liệu là các tệp văn bản có trên đĩa cứng máy tính của em. Kết quả là tệp có tên bai-hoc-1.docx.

– Với bài toán 3, miền dữ liệu là danh sách học sinh và điểm các bài dự thi của kì thi Olympic Tin học thành phố. Kết quả là danh sách 5 bạn có thành tích cao nhất tính theo điểm trung bình.

Có thể nói tìm kiếm là một trong những bài toán quan trọng nhất của Tin học. Việc thiết kế thuật toán tìm kiếm sẽ phụ thuộc vào cấu trúc của miền dữ liệu cần tìm kiếm và tiêu chí cụ thể của bài toán tìm kiếm.

Em hãy xác định miền dữ liệu và nghiệm có thể của các bài toán tìm kiếm sau.

1. Bài toán tìm đường đi từ nhà em đến trường học dựa trên bản đồ số.
2. Bài toán tìm tất cả các trường trung học phổ thông (tên trường, địa chỉ) ở quận (huyện) em đang cư trú.

## 2. TÌM KIẾM TUẦN TỰ

Cách An lần lượt lật các thẻ từ đầu đến cuối trong phần khởi động chính là việc thực hiện tìm kiếm tuần tự trong các thẻ bài. Đây là một trường hợp riêng của việc tìm kiếm trên mô hình dữ liệu là một dãy các đối tượng (danh sách) và thuật toán được thực hiện một cách tuần tự. Để cho đơn giản, ta xét danh sách tìm kiếm là một dãy các số. Bài toán tìm kiếm được mô tả như sau:

*Đầu vào:* Cho trước dãy số A[0], A[1],..., A[n-1] và giá trị K.

*Đầu ra:* Cần tìm ra chỉ số i mà phần tử A[i] có giá trị bằng K. Nếu không thấy thì trả về giá trị -1.

**Hoạt động 2** **Thuật toán tìm kiếm tuần tự**

Quan sát cách thực hiện thuật toán tìm kiếm tuần tự trên ví dụ cụ thể sau. Hãy trao đổi, thảo luận để hiểu và mô tả được thuật toán trong trường hợp tổng quát.

Cho dãy số A = [1, 4, 7, 8, 3, 9, 10] và cần tìm kiếm phần tử có giá trị bằng 9.

Có thể thực hiện tìm kiếm tuần tự như sau:

*Bước 1.* i = 0: A[0] = 1 không bằng 9.

*Bước 2.* i = 1: A[1] = 4 không bằng 9.

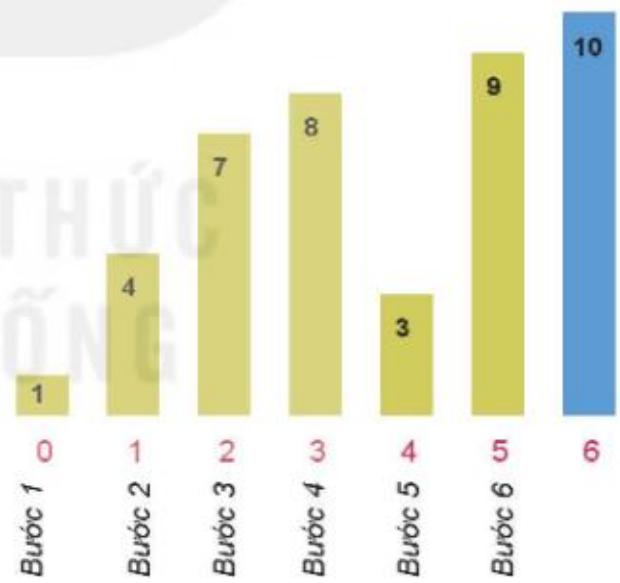
*Bước 3.* i = 2: A[2] = 7 không bằng 9.

*Bước 4.* i = 3: A[3] = 8 không bằng 9.

*Bước 5.* i = 4: A[4] = 3 không bằng 9.

*Bước 6.* i = 5: A[5] = 9 là phần tử cần tìm.

Như vậy phần tử cần tìm có chỉ số 5.




Các bước thực hiện như trên chính là các bước của thuật toán tìm kiếm tuần tự.

*Thuật toán tìm kiếm tuần tự:* Duyệt lần lượt các phần tử của dãy để tìm phần tử có giá trị bằng K. Nếu tìm thấy, trả về chỉ số của phần tử bằng K; Ngược lại, thông báo không tìm thấy và trả về giá trị –1. Thuật toán có thể duyệt từ đầu dãy hoặc từ cuối dãy.

Thuật toán tìm kiếm tuần tự có thể viết như sau:

```
1 def LinearSeach(A,K):
2     for i in range(len(A)):
3         if A[i] == K:
4             return i
5     return -1
```

**Thuật toán tìm kiếm tuần tự** được thực hiện bằng cách duyệt lần lượt các phần tử của dãy từ đầu đến cuối để tìm phần tử có giá trị bằng giá trị cần tìm.



**1.** Cho dãy A = [1, 91, 45, 23, 67, 9, 10, 47, 90, 46, 86].

Thuật toán tìm kiếm tuần tự cần thực hiện bao nhiêu lần duyệt để tìm ra phần tử có giá trị bằng 47 trong dãy?

**2.** Khi nào thì tìm kiếm tuần tự sẽ tìm được ngay kết quả, cần ít bước nhất?

**3.** Khi nào thì tìm kiếm tuần tự sẽ cần nhiều bước nhất? Cho ví dụ.

## 3. TÌM KIẾM NHỊ PHÂN

**Hoạt động 3** **Thuật toán tìm kiếm nhị phân**

Cho trước một dãy số đã được sắp xếp theo thứ tự tăng dần. Hãy đọc, quan sát và thảo luận cách làm sau đây để hiểu được thuật toán tìm kiếm nhị phân, biết được tính ưu việt của thuật toán này so với thuật toán tìm kiếm tuần tự trên một dãy các phần tử đã sắp xếp.



### a) Phân tích bài toán

Khác với bài toán tìm kiếm tuần tự, bài toán tìm kiếm nhị phân tìm kiếm với dãy số đã được sắp xếp. Khi duyệt một phần tử bất kì của dãy số, em có thể xác định được phần tử cần tìm sẽ nằm ở bên trái hay bên phải phần tử đang duyệt, từ đó quyết định tìm tiếp theo hướng nào mà không cần duyệt tất cả các phần tử của dãy số.

### b) Thuật toán tìm kiếm nhị phân

Thuật toán tìm kiếm nhị phân được thực hiện bằng cách liên tục thu hẹp phạm vi tìm kiếm. Nếu giá trị của phần tử ở giữa bằng K thì thông báo tìm thấy. Nếu giá trị K nhỏ hơn giá trị của phần tử ở giữa thì thu hẹp phạm vi tìm kiếm là nửa đầu của dãy tăng A (ngược lại thì phạm vi tìm kiếm là nửa sau). Cứ tiếp tục thu hẹp phạm vi như vậy cho đến khi tìm thấy hoặc đã duyệt hết thì thông báo không tìm thấy. Thuật toán tìm kiếm nhị phân được thực hiện như sau:

– Thiết lập các giá trị left, right là chỉ số phần tử đầu và cuối của dãy cần tìm. Như vậy cần tìm K trong dãy A[left..right]. Ban đầu đặt left = 0, right = n – 1.

– So sánh K với phần tử giữa dãy A[mid], với mid là phần nguyên của phép chia (left + right) cho 2, có ba trường hợp có thể xảy ra:

- Nếu K = A[mid] thì trả về chỉ số mid và kết thúc chương trình.
- Nếu K < A[mid] thì phần tử cần tìm sẽ nằm ở dãy con bên trái của phần tử A[mid], cập nhật giá trị right = mid – 1, giữ nguyên giá trị left.

• Nếu K > A[mid] thì phần tử cần tìm sẽ nằm ở dãy con bên phải của phần tử A[mid], cập nhật giá trị left = mid + 1, giữ nguyên giá trị right.

Lặp lại bước trên cho đến khi tìm thấy phần tử bằng K hoặc phạm vi tìm kiếm bằng rỗng (right < left).

## c) Minh hoạ các bước của thuật toán tìm kiếm nhị phân

Giả sử dãy số đã sắp xếp là A = [1, 3, 4, 7, 8, 9, 10]. Giá trị cần tìm là K = 9.

*Bước 1.* Phạm vi tìm kiếm là các phần tử được in đậm [**1**, **3**, **4**, **7**, **8**, **9**, **10**].

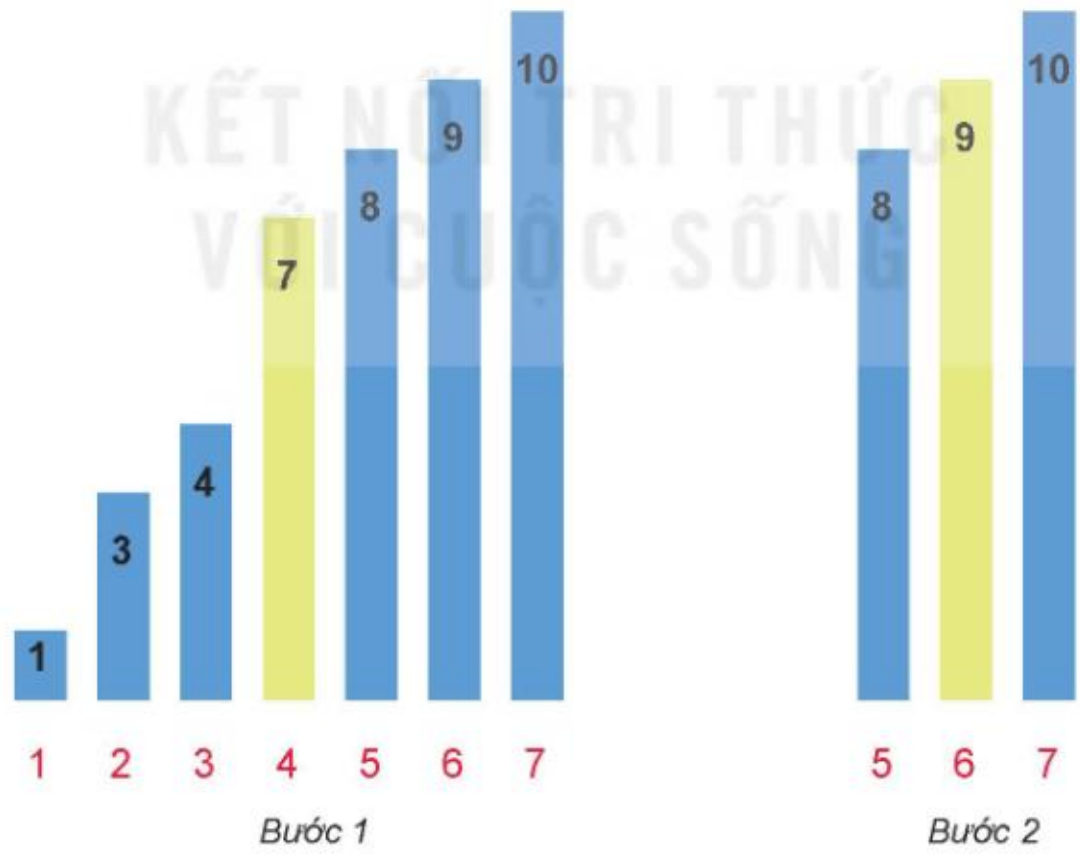
```
1  left = 0, right = 6
2  mid = (0 + 6) // 2 = 3
3  A[mid] = A[3] = 7 < K # phần tử cần tìm nằm ở dãy con bên phải
```

Cập nhật chỉ số `left = mid + 1 = 3 + 1 = 4`.

*Bước 2.* Phạm vi tìm kiếm là các phần tử được in đậm [1, 3, 4, 7, **8**, **9**, **10**].

```
1  left = 4, right = 6
2  mid = (4+6)//2 = 5
3  A[mid] = A[5] = 9 = K # phần tử cần tìm có chỉ số 5. Kết thúc chương trình.
```

Ví dụ trên cho thấy thời gian tìm với thuật toán tìm kiếm nhị phân nhanh hơn so với thuật toán tìm kiếm tuần tự, do số phần tử cần duyệt giảm một nửa sau mỗi vòng lặp. Cùng dãy số có số phần tử như nhau (A = [1, 3, 4, 7, 8, 9, 10]) và cùng giá trị tìm kiếm (K = 9) thì thuật toán tìm kiếm tuần tự cần 6 bước, nhưng nếu dãy số được sắp xếp và dùng thuật toán tìm kiếm nhị phân thì chỉ cần hai bước.




Thuật toán tìm kiếm nhị phân trên dãy số đã sắp xếp tăng dần có thể như sau, trong đó hàm BinarySearch(A,K) trả lại chỉ số i nếu tìm thấy A[i] = K và trả lại giá trị –1 nếu không tìm thấy K trong dãy A.

```python
def BinarySearch(A,K):
    left = 0
    right = len(A) - 1
    while left <= right:
        mid = (left + right)//2
        if A[mid] == K:
            return mid
        elif A[mid] < K:
            left = mid + 1
        else:
            right = mid - 1
    return -1
```

**Thuật toán tìm kiếm nhị phân** được áp dụng cho các dãy được sắp xếp theo thứ tự xác định. Sau mỗi bước lặp của thuật toán, phạm vi tìm kiếm được thu hẹp dần. Ví dụ với dãy tăng dần, nếu giá trị cần tìm nhỏ hơn giá trị của phần tử ở giữa của dãy thì phạm vi tìm kiếm thu hẹp vào nửa đầu của dãy, ngược lại, phạm vi tìm kiếm là nửa cuối của dãy. Cứ tiếp tục như vậy cho đến khi tìm thấy hoặc phạm vi tìm kiếm bằng rỗng.

Cho dãy A = [0, 4, 9, 10, 12, 14, 17, 18, 20, 31, 34, 67].

1. Với thuật toán tìm kiếm tuần tự, cần duyệt bao nhiêu phần tử để tìm ra phần tử có giá trị bằng 34?
2. Với thuật toán tìm kiếm nhị phân, cần duyệt bao nhiêu phần tử để tìm ra phần tử có giá trị bằng 34?
3. Thay vì lần lượt lật các thẻ từ đầu đến cuối, bạn Minh đã chơi như sau: Đầu tiên Minh lật thẻ ở giữa, sau đó tuỳ theo số ghi trên thẻ là lớn hơn hay nhỏ hơn số K mà lật tiếp thẻ ở ngay bên trái, hoặc ngay bên phải thẻ ở giữa. Trong trường hợp này, số lần nhiều nhất mà Minh phải lật để tìm ra thẻ in số K là bao nhiêu?

## LUYỆN TẬP

1. Em hãy chỉnh sửa thuật toán tìm kiếm tuần tự để tìm ra tất cả các phần tử trong dãy bằng giá trị cần tìm, biết dãy đó có nhiều phần tử bằng giá trị cần tìm.
2. Viết chương trình của thuật toán tìm kiếm nhị phân với dãy sắp xếp giảm dần.

## VẬN DỤNG

1. Cho A là danh sách tên các học sinh trong lớp, viết chương trình tìm kiếm tuần tự để tìm ra các học sinh có tên là Hoàn.
2. Cho A là danh sách tên các học sinh trong lớp được sắp xếp theo thứ tự bảng chữ cái, viết chương trình tìm kiếm nhị phân để tìm ra các học sinh có tên là Minh.

BÀI 20

# THỰC HÀNH BÀI TOÁN TÌM KIẾM

SAU BÀI HỌC NÀY EM SẼ:

- Biết được cách đọc dữ liệu từ tệp dữ liệu trong máy tính.
- Áp dụng được thuật toán tìm kiếm trong một vài bài toán cụ thể.



Trong bài học trước, các em đã được học cách triển khai thuật toán tìm kiếm tuần tự và tìm kiếm nhị phân; vậy chúng ta có thể áp dụng các thuật toán tìm kiếm đã học vào thực tế như thế nào? Trong các bài toán thực tế, các dữ liệu thường không được viết trực tiếp vào chương trình mà thường được lưu trong các tệp chứa dữ liệu. Ở bài học này, chúng ta sẽ được học cách đọc dữ liệu từ tệp chứa dữ liệu trong máy tính.



## Nhiệm vụ 1. Đọc và ghi tệp trong Python

**Hướng dẫn:**

Để thực hiện được các bài tập thực hành trong sách, em cần biết một số lệnh và thao tác đọc, ghi dữ liệu từ tệp văn bản.

### a) Đối tượng tệp

Python cho phép đọc, ghi dữ liệu với tệp văn bản (text file) rất dễ dàng. Các tệp đều được mở để đọc và ghi dưới dạng văn bản. Đối tượng tệp được tạo ra bằng lệnh open() với các cách như sau:

– Lệnh mở tệp để đọc:

```
f = open(<file name>,"r",encoding = "UTF-8")
```

Chú ý, tham số "r" có thể bỏ qua. Tham số encoding dùng khi mở tệp văn bản có mã Unicode, ví dụ tiếng Việt. Nếu không có tham số encoding thì chỉ đọc được văn bản có mã ASCII.

– Lệnh mở tệp để ghi dữ liệu từ đầu:

```
f = open(<file name>,"w",encoding = "UTF-8")
```

Chú ý, tham số "w" là bắt buộc.

– Lệnh mở tệp để ghi tiếp dữ liệu:

```
f = open(<file name>,"a",encoding = "UTF-8")
```

Chú ý, tham số "a" là bắt buộc.

Sau khi mở tệp thì có thể tiến hành đọc (hoặc ghi) dữ liệu. Sau khi thực hiện xong các thao tác đọc, ghi thì cần đóng đối tượng tệp bằng lệnh sau:

```
f.close()
```

## b) Các lệnh đọc dữ liệu từ tệp văn bản

Dữ liệu trong các tệp văn bản được lưu dưới dạng các dòng, mỗi dòng là một xâu kí tự. Sau khi mở tệp bằng lệnh open(), con trỏ đọc mặc định ở đầu của tệp sẵn sàng chờ lệnh đọc.

Bảng 20.1 mô tả và minh hoạ kết quả của các lệnh đọc dữ liệu từ đối tượng f trong Python với tệp Data.inp.

**Data.inp**

```
Hà 9.6
Bình 8.5
Quang 7.2
```

**Bảng 20.1.** *Các lệnh đọc dữ liệu từ đối tượng f trong Python*

| Lệnh đọc dữ liệu | Ý nghĩa | Kết quả dữ liệu |
|---|---|---|
| `f.readline()` | Đọc một dòng tiếp theo từ f. | Kết quả là từng dòng của tệp f. |
| `f.readlines()` | Đọc toàn bộ dữ liệu tệp, đưa kết quả vào một danh sách (list), mỗi phần tử là một dòng. | ['Hà 9.6\n', 'Bình 8.5\n', 'Quang 7.2'] |
| `L = list(f)` | Đưa toàn bộ nội dung f vào list L, mỗi dòng là một phần tử. | ['Hà 9.6\n', 'Bình 8.5\n', 'Quang 7.2'] |
| `for line in f:` | Duyệt từng dòng của f. | Lệnh này tương đương nhóm lệnh:<br>`L = list(f)`<br>`for line in L:` |

**Ví dụ 1.** Với bộ dữ liệu trên, cần đọc và đưa vào hai list là TenHS và DiemHS. Chương trình sau đọc dữ liệu từ tệp Data.inp, đưa dữ liệu vào hai mảng trên và hiển thị kết quả ra màn hình.

```
1  fname = "Data.inp"
2  def NhapDL(fname):
3      f = open(fname,encoding = "UTF-8")
4      TenHS = []
5      DiemHS = []
6      for line in f:
7          L = line.split()
8          TenHS.append(L[0])
9          DiemHS.append(float(L[1]))
10     f.close()
11     return TenHS,DiemHS
12
13 TenHS,DiemHS = NhapDL(fname)
14 for i in range(len(TenHS)):
15     print(TenHS[i],DiemHS[i])
```

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

## c) Ghi dữ liệu ra tệp văn bản

Việc ghi dữ liệu ra tệp văn bản đơn giản nhất là sử dụng lệnh print(). Cú pháp lệnh print() như sau:

```
print(<v1>, <v2>, …, file = <đối tượng file>)
```

Chú ý, tham số file = <đối tượng file> là bắt buộc khi cần đưa dữ liệu ra tệp. Tất cả các tham số khác của lệnh print() vẫn có giá trị khi đưa dữ liệu ra tệp.

**Ví dụ 2.** Giả sử có hai mảng TenHS và DiemHS tương ứng với dữ liệu tên và điểm của các học sinh trong lớp. Chương trình sau ghi những thông tin này ra tệp Data.out có định dạng tương tự như Data.inp.

```
1 fname = "Data.out"
2 TenHS = ['Hà','Bình','Quang']
3 DiemHS = [9.6,8.5,7.2]
4
5 f = open(fname,"w",encoding = "UTF-8")
6 for i in range(len(TenHS)):
7     print(TenHS[i],DiemHS[i],file = f)
8 f.close()
```



## Nhiệm vụ 2. Viết chương trình tra cứu điểm thi

*Yêu cầu:* Viết chương trình tra cứu điểm thi theo tên các học sinh trong lớp. Chương trình cho phép người dùng nhập tên của học sinh cần tra cứu, sau đó kiểm tra và thông báo điểm số của học sinh cần tìm.

*Nhập dữ liệu:* Yêu cầu người dùng nhập dữ liệu điểm học sinh từ tệp trên máy tính. Tệp bao gồm nhiều hàng, mỗi hàng gồm tên học sinh và điểm cách nhau bởi dấu cách. Ví dụ tệp có cấu trúc như Hình 20.1.

**diem.inp**

```
Nam 7.8
Sơn 5.6
Hương 8.9
Huyền 7.4
Hà 9.5
Hùng 8.4
```

*Hình 20.1*

*Tra cứu dữ liệu:* Cho phép người dùng nhập tên học sinh cần tra cứu, tìm và in ra màn hình kết quả điểm thi của học sinh đó. Nếu không tìm thấy tên học sinh trong danh sách đã nhập, thông báo "không tìm thấy dữ liệu của học sinh".

### Hướng dẫn:

*Phân tích bài toán:* Nhiệm vụ này có thể được thực hiện thông qua thuật toán tìm kiếm. Với dữ liệu đầu vào là tên của học sinh, chương trình cần sử dụng thuật toán tìm kiếm để kiểm tra xem tên của học sinh có tồn tại trong danh sách không. Nếu tồn tại thì học sinh cần tìm nằm ở vị trí nào trong danh sách. Do tên học sinh và điểm số được nhập vào danh sách theo cùng một thứ tự, từ vị trí tên học sinh trong danh sách tên, chúng ta có thể suy ra điểm số của học sinh đó trong danh sách điểm.

Chương trình có thể như sau:

**Tracuuten.py**

```
1 input_file = open("diem.inp", encoding="UTF-8") #encoding = utf8 để đọc
được tiếng Việt
```

```
2  ten_list = []
3  diem_list = []
4  for line in input_file.readlines():# đọc lần lượt từng dòng trong tệp
5      ten,diem =line.split() # tách tên và điểm ở từng dòng
6      ten_list.append(ten)
7      diem_list.append(float(diem))
8  ten=''
9  while(ten!='end'):
10     ten = input('Nhập tên học sinh cần tra cứu, nhập từ khoá end để kết thúc:')
11     if (ten!='end'):
12         vitri=-1
13         for i in range(0,len(ten_list)):
14             if ten==ten_list[i]: # kiểm tra xem tên có tồn tại ở vị trí i
15                 vitri=i
16         if vitri==-1:
17             print('Không tìm thấy học sinh trong danh sách')
18         else:
19             print('Điểm số của học sinh là:',diem_list[vitri])
```



## Nhiệm vụ 3. Viết chương trình kiểm tra điểm thi

*Yêu cầu:* Viết chương trình kiểm tra điểm thi của các học sinh trong một lớp học. Điểm thi của học sinh được ghi trong tệp diemthi_sx.inp, trong đó mỗi điểm thi của các học sinh được viết trong một hàng và được sắp xếp theo thứ tự tăng dần. Chương trình đọc dữ liệu điểm thi từ tệp, sau đó cho phép người dùng nhập một điểm số cần kiểm tra. Nếu điểm số có tồn tại thì in ra vị trí mà điểm số đó xuất hiện trong tệp, nếu điểm số không tồn tại thì in ra thông báo điểm số không tồn tại. Ví dụ tệp có cấu trúc như Hình 20.2.

**diemthi_sx.inp**

```
5.6
5.8
6.8
7.4
7.5
7.9
```

*Hình 20.2*

**Hướng dẫn:**

*Phân tích bài toán:* Tương tự như Nhiệm vụ 2, nhiệm vụ này vẫn được thực hiện bằng cách sử dụng thuật toán tìm kiếm. Trong nhiệm vụ này, dữ liệu điểm số đã được sắp xếp theo thứ tự tăng dần do đó chúng ta có thể áp dụng thuật toán tìm kiếm nhị phân để gia tăng tốc độ tìm kiếm.

Chương trình có thể như sau:

**Kiemtradiem_sx.py**

```
1  def BinarySearch(A,K):
2      left = 0
3      right = len(A) - 1
4      while left <= right:
5          mid = (left + right)//2
6          if A[mid] == K:
7              return mid
8          elif A[mid] < K:
```

```
            left = mid + 1
        else:
            right = mid - 1
    return -1
input_file = open("diemthi_sx.inp") # đọc dữ liệu từ file diemthi_sx.inp
ds_diem = [] #khởi tạo danh sách để chứa điểm số đọc từ file
for line in input_file.readlines():
    ds_diem.append(float(line)) # lần lượt đọc các dòng của file và đưa vào danh sách điểm thi
input_file.close()
diem = float(input('Nhập điểm số cần kiểm tra:'))
vitri = BinarySearch(ds_diem,diem) # gọi đến hàm BinarySearch kiểm tra xem có phần tử 'diem' trong danh sách 'ds_diem' hay không
if vitri == -1:
    print('Không tồn tại điểm số cần tìm trong danh sách')
else:
    print('Điểm cần tìm nằm ở hàng thứ', vitri, 'trong danh sách')
```

## LUYỆN TẬP

Chỉnh sửa lại chương trình của Nhiệm vụ 3 để cho phép chương trình có thể tìm kiếm điểm số trên danh sách điểm số được sắp xếp theo thứ tự giảm dần.

## VẬN DỤNG

Viết chương trình tra cứu tên theo điểm thi của học sinh trong lớp. Chương trình cho phép người dùng nhập vào khoảng điểm số cần tìm kiếm (ví dụ từ 6 đến 8). Chương trình kiểm tra và thông báo tên của học sinh có điểm số nằm trong khoảng tương ứng. Giải bài toán trong hai trường hợp: điểm được sắp xếp theo thứ tự ngẫu nhiên như trong Nhiệm vụ 1 hoặc điểm được sắp xếp theo thứ tự tăng dần như sau:

Sơn 5.6

Huyền 7.4

Nam 7.8

Hùng 8.4

Hương 8.9

Hà 9.5

# BÀI 21 CÁC THUẬT TOÁN SẮP XẾP ĐƠN GIẢN

SAU BÀI HỌC NÀY EM SẼ:

- Biết và thực hiện được một số thuật toán sắp xếp đơn giản.

Bài học trước cho em thấy việc tìm kiếm trên một dãy đã sắp xếp nhanh hơn so với việc tìm kiếm tuần tự. Vì vậy bài toán tìm kiếm liên quan mật thiết đến bài toán sắp xếp. Bài toán sắp xếp cơ bản có dạng như sau:

Cho dãy A gồm n phần tử:

`A[0], A[1], ..., A[n-1]` (1)

Cần xếp dãy A theo thứ tự tăng dần:

`A[0] ≤ A[1] ≤ ... ≤ A[n-1]` (2)

Em hãy trình bày ý tưởng của mình để giải bài toán sắp xếp với dãy có bốn phần tử.

## 1. THUẬT TOÁN SẮP XẾP CHÈN

**Hoạt động 1** Tìm hiểu ý tưởng thuật toán sắp xếp chèn

Quan sát sơ đồ mô phỏng, trao đổi, thảo luận về ý tưởng chính của thuật toán sắp xếp chèn.




Cho dãy A = [5, 3, 9, 7, 2]. Sơ đồ sau mô phỏng các bước của thuật toán sắp xếp chèn.

| Chỉ số của dãy | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| Trước vòng lặp | 5 | 3 | 9 | 7 | 2 |
| Vòng lặp 1, i = 1 | Duyệt phần tử thứ hai, vì 3 nhỏ hơn 5 nên chèn 3 vào trước vị trí số 5 | | | | |
| Sau vòng lặp | 3 | 5 | 9 | 7 | 2 |
| Vòng lặp 2, i = 2 | Duyệt phần tử thứ ba, vì 9 đã lớn hơn 3 và 5 nên giữ nguyên vị trí | | | | |
| Sau vòng lặp | 3 | 5 | 9 | 7 | 2 |
| Vòng lặp 3, i = 3 | Duyệt phần tử thứ tư, vì 5 < 7 < 9 nên chèn 7 vào giữa vị trí của 5 và 9 | | | | |
| Sau vòng lặp | 3 | 5 | 7 | 9 | 2 |
| Vòng lặp 4, i = 4 | Duyệt phần tử thứ năm, vì 2 < 3 nên chèn 2 vào trước vị trí số 3 | | | | |
| Kết thúc | 2 | 3 | 5 | 7 | 9 |

*Hình 21.1. Mô phỏng thuật toán sắp xếp chèn*

Quan sát sơ đồ mô phỏng và trả lời các câu hỏi sau:

1) So sánh số bước lặp với độ dài của dãy số ban đầu.

2) Vị trí xuất phát của mũi tên màu đỏ có quan hệ gì với chỉ số bước lặp?

3) Khi kết thúc lặp ta thu được kết quả gì?

Ý tưởng của thuật toán sắp xếp chèn là cho chỉ số i chạy từ 1 (phần tử thứ hai của dãy) đến n – 1 (phần tử cuối của dãy). Mỗi vòng như vậy sẽ "chèn" phần tử A[i] vào vị trí đúng của dãy con đã sắp xếp A[0], A[1],..., A[i-1]. Như vậy sau n – 1 bước lặp thì dãy được sắp xếp xong.

Thao tác "chèn" A[i] vào vị trí đúng trong dãy con A[0], A[1],..., A[i-1] có thể được mô tả bằng cách "nhấc" A[i] lên và chuyển các phần tử bên trái A[i] nhưng lớn hơn A[i] sang phải, sau đó đặt A[i] vào vị trí đúng.

```
value = A[i]
j = i-1
while j >= 0 and A[j] > value:
    Dịch A[j] sang vị trí liền kề bên phải
    j = j - 1
A[j+1] = value
```

Thuật toán sắp xếp chèn có thể mô tả bằng hàm **InsertionSort(A)** như sau:

```
def InsertionSort(A):
    n = len(A)
    for i in range(1,n):
        value = A[i]
        j = i-1
        while j >= 0 and A[j] > value:
            A[j+1] = A[j]
            j = j - 1
        A[j+1] = value
```

Ý tưởng của thuật toán sắp xếp chèn là thực hiện vòng lặp duyệt từ phần tử thứ hai đến cuối dãy. Sau mỗi bước lặp phần tử tương ứng sẽ được chèn vào vị trí đúng của dãy con đã sắp xếp là các phần tử phía trước vị trí đang duyệt.

**1.** Mô phỏng chi tiết các bước lặp sắp xếp chèn dãy A = [5, 0, 4, 2, 3].

**2.** Nếu dãy ban đầu đã được sắp xếp thì thuật toán sắp xếp chèn sẽ thực hiện như thế nào?

## 2. THUẬT TOÁN SẮP XẾP CHỌN

**Hoạt động 2** **Tìm hiểu ý tưởng thuật toán sắp xếp chọn**

Quan sát sơ đồ mô phỏng, trao đổi, thảo luận về ý tưởng chính của thuật toán sắp xếp chọn.

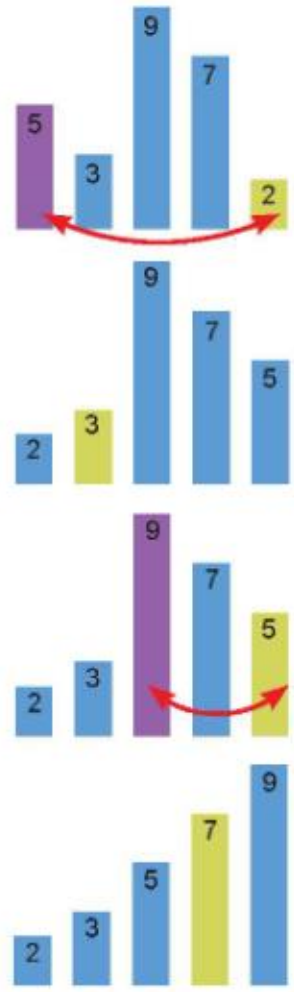
Xét dãy A = [5, 3, 9, 7, 2]. Sơ đồ sau mô phỏng các bước thực hiện thuật toán sắp xếp chọn. Quan sát sơ đồ và trả lời các câu hỏi sau:

1) Có bao nhiêu vòng lặp? Chỉ số i bắt đầu bằng bao nhiêu?

2) Tại mỗi vòng lặp đều có một thao tác đổi chỗ hai phần tử, đó là các phần tử nào?

3) Khi kết thúc vòng lặp ta thu được kết quả gì?

| Chỉ số của dãy | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| Trước vòng lặp | 5 | 3 | 9 | 7 | 2 |
| Vòng lặp 1, i = 0 | 2 là phần tử nhỏ nhất, đổi chỗ 2 với 5 | | | | |
| Sau vòng lặp | 2 | 3 | 9 | 7 | 5 |
| Vòng lặp 2, i = 1 | 3 là phần tử nhỏ nhất không tính phần tử đầu tiên, giữ nguyên vị trí dãy số | | | | |
| Sau vòng lặp | 2 | 3 | 9 | 7 | 5 |
| Vòng lặp 3, i = 2 | 5 là phần tử nhỏ nhất không tính hai phần tử đầu tiên, đổi chỗ 5 và 9 | | | | |
| Sau vòng lặp | 2 | 3 | 5 | 7 | 9 |
| Vòng lặp 4, i = 3 | 7 là phần tử nhỏ nhất không tính 3 phần tử đầu tiên, giữ nguyên vị trí dãy số | | | | |
| Kết thúc | 2 | 3 | 5 | 7 | 9 |




*Hình 21.2. Mô phỏng thuật toán sắp xếp chọn*

Ý tưởng của thuật toán sắp xếp chọn là cho chỉ số i chạy từ 0 (phần tử đầu tiên) đến n – 2 (phần tử gần cuối). Tại mỗi bước lặp, cần tìm phần tử nhỏ nhất nằm trong dãy A[i], A[i+1], ..., A[n-1] và đổi chỗ phần tử nhỏ nhất này với A[i]. Mô tả thuật toán chọn như sau:

```
1 for i in range(n-1):
2     Chọn phần tử nhỏ nhất trong dãy A[i], A[i+1], ..., A[n-1]
3     Đổi chỗ phần tử này với A[i]
```

Bước chọn phần tử tại dòng 2 và đổi chỗ tại dòng lệnh 3 có thể viết như sau:

```
1 iMin = i
2 for j in range(i+1,n):
3     if A[j] < A[iMin]:
4         iMin = j
5 Đổi chỗ A[i] và A[iMin]
```

Ở đây sử dụng biến iMin để lưu chỉ số phần tử nhỏ nhất của dãy A[i], A[i+1], ..., A[n-1]. Thuật toán sắp xếp chọn có thể được mô tả bằng hàm SelectionSort(A) như sau:

```
1 def SelectionSort(A):
2     n = len(A)
3     for i in range(n-1):
4         iMin = i
5         for j in range(i+1,n):
6             if A[j] < A[iMin]:
7                 iMin = j
8         A[i],A[iMin] = A[iMin],A[i]
```

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

**Thuật toán sắp xếp chọn** thực hiện một vòng lặp với chỉ số i chạy từ 0 (phần tử đầu tiên) đến n – 2 (phần tử gần cuối). Tại mỗi bước lặp, chọn phần tử nhỏ nhất nằm trong dãy A[i], A[i+1],..., A[n-1] và đổi chỗ phần tử này với A[i].

1. Thực hiện mô phỏng sắp xếp theo thuật toán sắp xếp chọn dãy sau: 4, 5, 2, 1, 3.
2. Theo thuật toán sắp xếp chọn, sau mỗi bước thứ i thì các phần tử A[0], A[1],..., A[i] đã được sắp xếp đúng. Đúng hay sai?

## 3. THUẬT TOÁN SẮP XẾP NỔI BỌT

**Hoạt động 3** **Tìm hiểu các ý tưởng thuật toán sắp xếp nổi bọt**

Cùng trao đổi, thảo luận về các ý tưởng của thuật toán sắp xếp nổi bọt.

Thuật toán sắp xếp nổi bọt lấy ý tưởng từ hiện tượng "nổi bọt" của không khí dưới nước. Các bọt khí nổi dần lên mặt nước. Ý tưởng của thuật toán nổi bọt là liên tục đổi chỗ hai phần tử cạnh nhau nếu chúng chưa được sắp thứ tự đúng.

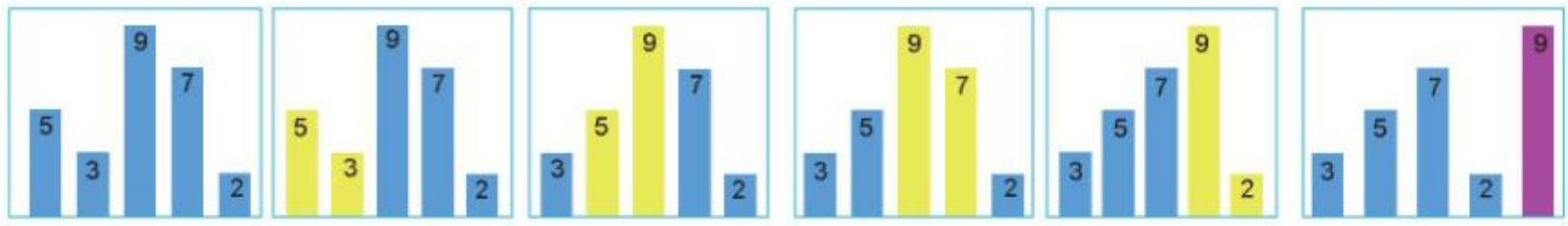
Thuật toán này sẽ thực hiện nhiều vòng lặp. Quan sát vòng lặp sau để hiểu cách thực hiện: chỉ số j chạy từ 0 đến n – 2 và kiểm tra hai phần tử liền nhau A[j], A[j+1], nếu chúng chưa sắp thứ tự đúng thì đổi chỗ.

```
for j in range(n-1):
    if A[j] > A[j+1]:
        Đổi chỗ A[j], A[j+1]
```

Quan sát vòng lặp đầu tiên của thuật toán nổi bọt để thấy sau vòng lặp này, phần tử lớn nhất được chuyển về cuối dãy A.

| Chỉ số của dãy: | 0 | 1 | 2 | 3 | 4 | |
|---|---|---|---|---|---|---|
| Trước vòng lặp | 5 | 3 | 9 | 7 | 2 | |
| Bước lặp 1, j = 0 | 5 | 3 | 9 | 7 | 2 | So sánh phần tử thứ nhất và phần tử thứ hai |
| Bước lặp 2, j = 1 | 3 | 5 | 9 | 7 | 2 | So sánh phần tử thứ hai và phần tử thứ ba |
| Bước lặp 3, j = 2 | 3 | 5 | 9 | 7 | 2 | So sánh phần tử thứ ba và phần tử thứ tư |
| Bước lặp 4, j = 3 | 3 | 5 | 7 | 9 | 2 | So sánh phần tử thứ tư và phần tử thứ năm |
| Kết thúc vòng 1 | 3 | 5 | 7 | 2 | 9 | |

Trong sơ đồ trên, mũi tên màu đỏ cho biết có đổi chỗ hai phần tử, mũi tên màu tím cho biết không thay đổi vị trí.




*Hình 21.3. Mô phỏng thuật toán sắp xếp nổi bọt*

Sau vòng lặp thứ nhất, phần tử lớn nhất được chuyển về cuối dãy. Sau vòng lặp thứ hai, phần tử lớn thứ hai được chuyển về đúng vị trí ở cuối dãy. Cứ như vậy, sau n – 1 vòng lặp thì dãy được sắp xếp. Có thể mô tả thuật toán nổi bọt như sau:

```
1 for i in range(n-1):
2     for j in range(n-1):
3         if A[j] > A[j+1]:
4             Đổi chỗ A[j],A[j+1]
```

Quan sát dòng 2, ta thấy không cần phải có đủ n – 1 bước lặp vì sau i vòng lặp thì i phần tử lớn nhất đã chuyển về đúng vị trí ở cuối dãy, nên với chỉ số i thì vòng lặp ở dòng 2 chỉ cần n – 1 – i bước lặp. Thuật toán sắp xếp nổi bọt được mô tả bằng hàm BubbleSort(A) như sau:

```
1 def BubbleSort(A):
2     n = len(A)
3     for i in range(n-1):
4         for j in range(n-1-i):
5             if A[j] > A[j+1]:
6                 A[j],A[j+1]=A[j+1],A[j]
```

**Thuật toán sắp xếp nổi bọt** thực hiện nhiều vòng lặp, kiểm tra hai phần tử cạnh nhau, nếu chúng chưa sắp xếp đúng thì đổi chỗ. Có nhiều cách thể hiện thuật toán này, nhưng cách thường dùng là sử dụng hai vòng lặp lồng nhau, vòng lặp trong thực hiện thao tác đổi chỗ hai phần tử cạnh nhau cho đến khi dãy được sắp xếp xong.

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

1. Mô tả các bước thuật toán sắp xếp nổi bọt của dãy A = [4, 3, 1, 2].
2. Khi nào thì các mũi tên ở tất cả các bước trong sơ đồ mô phỏng thuật toán sắp xếp nổi bọt đều có màu đỏ?

## LUYỆN TẬP

1. Cho dãy A = [5, 8, 1, 0, 10, 4, 3]. Viết các chương trình sắp xếp dãy A theo thứ tự tăng dần theo các thuật toán sắp xếp chèn, sắp xếp chọn và sắp xếp nổi bọt.
2. Viết chương trình nhập một dãy số từ bàn phím, các số cách nhau bởi dấu cách, thực hiện sắp xếp dãy đã nhập theo một trong các thuật toán sắp xếp rồi in kết quả ra màn hình.

## VẬN DỤNG

1. Viết lại các thuật toán sắp xếp trong bài theo thứ tự giảm dần.
2. Nêu ý nghĩa thực tế của các thuật toán sắp xếp đã học, chẳng hạn sắp xếp các học sinh trong lớp theo chiều cao tăng dần.

# BÀI 22 THỰC HÀNH BÀI TOÁN SẮP XẾP

SAU BÀI HỌC NÀY EM SẼ:

- Áp dụng được thuật toán sắp xếp trong một vài bài toán cụ thể.

Trong Bài 21, em đã được học cách triển khai thuật toán sắp xếp để sắp xếp các phần tử trong danh sách theo thứ tự tăng dần. Nếu cần sắp xếp theo thứ tự ngược lại thì câu lệnh so sánh tương ứng trong vòng lặp sẽ cần thay đổi như thế nào?

## Nhiệm vụ 1

Cho danh sách số lượng mỗi mặt hàng trong kho của một cửa hàng. Người quản lí kho cần xem các mặt hàng theo thứ tự số lượng tăng dần. Em hãy viết chương trình sắp xếp các mặt hàng trong kho theo thứ tự số lượng tăng dần, sử dụng thuật toán sắp xếp chèn, sau đó in ra màn hình dãy số vừa sắp xếp. Danh sách số lượng các mặt hàng được đọc từ tệp văn bản kho.inp, mỗi dòng bao gồm số lượng một mặt hàng.

**kho.inp**

```
5
3
10
4
8
2
```

**Hướng dẫn:**

*Phân tích:*

Chúng ta có thể sử dụng thuật toán sắp xếp chèn đã học ở bài học trước để sắp xếp các phần tử trong danh sách số lượng mặt hàng theo thứ tự tăng dần. Bài toán này bao gồm các bước sau: Đầu tiên, chúng ta đọc số lượng các mặt hàng trong kho từ tệp văn bản; Sau đó sử dụng thuật toán sắp xếp chèn để sắp xếp số lượng các mặt hàng; Cuối cùng là in số lượng các mặt hàng đã sắp xếp ra màn hình.

**quanlykho.py**

```
1  def InsertionSort(A):
2      n = len(A)
3      for i in range(1,n):
4          value = A[i]
5          j = i-1
6          while j >= 0 and A[j] > value:
7              A[j+1] = A[j]
8              j = j - 1
9          A[j+1] = value
10 input_file = open("kho.inp", encoding="UTF-8")
11 soluong_ds = []
12 for line in input_file.readlines():
13     soluong_ds.append(int(line))
14 InsertionSort(soluong_ds)
15 print('Số lượng các mặt hàng trong kho theo thứ tự tăng dần là:')
16 for i in range(len(soluong_ds)):
17     print(soluong_ds[i])
```

*Đọc số lượng các mặt hàng trong tệp và đưa vào danh sách soluong_ds*

## Nhiệm vụ 2

Cho danh sách điểm trung bình môn Tin học của các học sinh. Em hãy sử dụng thuật toán sắp xếp chọn để sắp xếp danh sách này theo thứ tự điểm trung bình giảm dần, sau đó in danh sách đã sắp xếp ra màn hình. Danh sách điểm trung bình được đọc từ tệp văn bản diem.inp, mỗi dòng bao gồm điểm trung bình của một học sinh.

**diem.inp**

```
7.8
5.6
8.9
7.4
9.5
8.4
9.1
```

**Hướng dẫn:**

*Phân tích:*

Chúng ta có thể sử dụng thuật toán sắp xếp chọn đã học ở bài học trước để sắp xếp danh sách điểm số. Chú ý đề bài yêu cầu sắp xếp danh sách theo thứ tự điểm trung bình giảm dần. Do đó ở mỗi vòng lặp của thuật toán sắp xếp chọn, chúng ta phải tìm kiếm phần tử có giá trị cao nhất, thay vì tìm phần tử có giá trị bé nhất. Bài toán này bao gồm các bước sau: Đầu tiên, chúng ta đọc điểm trung bình từ tệp văn bản diem.inp; Sau đó sử dụng thuật toán sắp xếp chọn đã học ở bài trước để sắp xếp điểm trung bình theo thứ tự giảm dần; Cuối cùng là in danh sách điểm trung bình đã sắp xếp ra màn hình.

**sapxepdiem.py**

```
def SelectionSort(A):
    n = len(A)
    for i in range(n-1):
        iMax = i
        for j in range(i+1,n):
            if A[j] > A[iMax]:
                iMax = j
        A[i],A[iMax] = A[iMax],A[i]
input_file = open("diem.inp", encoding="UTF-8")
diem_ds = []
for line in input_file.readlines():
    diem_ds.append(float(line))
SelectionSort(diem_ds)
print('Danh sách điểm theo thứ tự giảm dần là:')
for i in range(len(diem_ds)):
    print(diem_ds [i])
```

## LUYỆN TẬP

1. Sử dụng thuật toán sắp xếp chọn viết lại chương trình trong Nhiệm vụ 1.
2. Sử dụng thuật toán sắp xếp nổi bọt viết lại chương trình trong Nhiệm vụ 2.

## VẬN DỤNG

Một người đi mua hàng với danh sách các mặt hàng cần mua, đơn giá từng mặt hàng và số lượng hàng cần mua được lưu trong tệp văn bản muahang.inp.

Hãy sử dụng thuật toán nổi bọt để sắp xếp các mặt hàng theo thứ tự thành tiền của các mặt hàng tăng dần rồi in ra tên các mặt hàng và thành tiền tương ứng.

# BÀI 23 KIỂM THỬ VÀ ĐÁNH GIÁ CHƯƠNG TRÌNH

SAU BÀI HỌC NÀY EM SẼ:

- Biết được vai trò của kiểm thử làm tăng độ tin cậy của chương trình nhưng chưa chứng minh được tính đúng của chương trình.
- Biết được các tiêu chí đánh giá hiệu quả và tính đúng của chương trình.

Trong các bài trước em đã học cách thiết kế thuật toán cho một số bài toán như bài toán tìm kiếm, bài toán sắp xếp và thiết lập chương trình thực hiện thuật toán đó. Một bài toán có thể có nhiều thuật toán khác nhau và do đó có thể có nhiều chương trình khác nhau cùng giải quyết một bài toán. Hãy thảo luận và trả lời các câu hỏi sau:

– Làm thế nào để biết trong các thuật toán giải cùng một bài toán thì thuật toán nào là tốt nhất?

– Có những tiêu chí nào để đánh giá tính "tối ưu" của một thuật toán?

## 1. VAI TRÒ CỦA KIỂM THỬ CHƯƠNG TRÌNH

**Hoạt động 1** Tìm hiểu ý nghĩa của kiểm thử chương trình

Ở lớp 10, em đã học một số phương pháp kiểm thử chương trình. Em hãy thảo luận với các bạn về các phương pháp kiểm thử sau, nêu ý nghĩa của chúng trong việc đánh giá độ tin cậy và chứng minh tính đúng của chương trình:

1. Tạo các bộ dữ liệu kiểm thử (test) để kiểm tra dữ liệu đầu ra có chính xác hay không.
2. Thiết lập điểm dừng hoặc cho chương trình chạy theo từng lệnh để kiểm tra và tìm ra lỗi (bug) của chương trình.
3. Thực hiện in dữ liệu trung gian trong quá trình kiểm thử để tìm ra lỗi của chương trình (nếu có).

– Phương pháp 1 sử dụng các bộ dữ liệu kiểm thử để kiểm tra tính đúng của chương trình. Nếu phát hiện lỗi không chính xác của dữ liệu đầu ra thì kết luận ngay thuật toán và chương trình không đúng. Nếu với các bộ dữ liệu kiểm thử, dữ liệu đầu ra đều đúng thì điều đó chưa chứng minh được thuật toán hay chương trình đúng. Kết quả đó chỉ làm tăng khả năng đúng của chương trình.

– Mục đích của các phương pháp 2 và 3 là tìm ra lỗi và sửa lỗi của chương trình. Như vậy, với việc tìm ra và sửa lỗi chứng tỏ chương trình trở nên tin cậy hơn, tốt hơn. Tuy nhiên, điều đó không chứng minh được tính đúng của thuật toán và chương trình.

Một thuật toán nếu được thiết kế đúng thì với mọi bộ dữ liệu đầu vào sẽ nhận được bộ dữ liệu đầu ra đúng tương ứng. Các phương pháp kiểm thử không có tính năng chứng minh được tính đúng của một thuật toán.

Kiểm thử sẽ giúp tăng độ tin cậy của chương trình nhưng chưa chứng minh được tính đúng của thuật toán và chương trình.



1. Giả sử em thiết lập chương trình giải bài toán nào đó. Em đã kiểm thử với 10 bộ dữ liệu và tất cả các kết quả đều đúng. Khi đó có thể kết luận chương trình đó đúng hay chưa?
2. Giả sử một chương trình kiểm thử với 10 bộ dữ liệu cho kết quả 9 lần đúng, 1 lần sai. Chương trình đó là sai hay đúng?

## 2. KIỂM TRA TÍNH ĐÚNG CỦA CHƯƠNG TRÌNH

**Hoạt động 2** **Tìm hiểu cách kiểm tra tính đúng của chương trình**

Quan sát chương trình mô tả thuật toán sắp xếp chèn. Hãy thảo luận và đưa ra các lập luận để kiểm tra tính đúng của thuật toán sắp xếp chèn.



Chương trình thực hiện thuật toán sắp xếp chèn để sắp xếp dãy A cho trước theo thứ tự tăng dần như sau:

```
1 def InsertionSort(A,n):
2   for i in range(1,n):
3       value = A[i]
4       k = i-1
5       # di chuyển các phần tử A[k] > value qua phải
6       while k >= 0 and A[k] > value:
7         A[k+1] = A[k]
8         k = k - 1
9       A[k+1] = value
```

**Trao đổi 1.** Cài đặt chương trình sắp xếp chèn, nhập một số bộ dữ liệu đầu vào bất kì cho dãy A và kiểm tra kết quả đầu ra. Ta sẽ nhận thấy các kết quả đầu ra của chương trình đều đúng là dãy A được sắp xếp theo thứ tự tăng dần. Từ đó kết luận thuật toán sắp xếp chèn của chương trình trên là đúng.

**Trao đổi 2.** Giả sử dãy A ban đầu là A[0], A[1], ..., A[n–1] với n là độ dài dãy A. Dòng lệnh 2 là một vòng lặp gồm n – 1 bước, mỗi bước gán i chạy từ 1 đến n – 1. Với mỗi i (từ 1 đến n – 1), các dòng lệnh từ 3 đến 9 thực hiện việc "chèn" phần tử A[i] vào một vị trí trong dãy con phía trước A[0], A[1], ..., A[i – 1]. Giá trị A[i] sẽ được chèn vào vị trí giữa hai phần tử A[k] và A[k+1] nếu $A[k] \le A[i] < A[k+1]$. Do đó các dòng lệnh từ 3 đến 9 sẽ chèn phần tử A[i] vào đúng vị trí ở dãy đầu ra được sắp xếp tăng dần. Từ đó suy ra đến bước cuối cùng dãy A đã được sắp xếp đúng theo thứ tự.

**Trao đổi 3.** Trong khoa học máy tính, việc kiểm tra tính đúng của thuật toán đều phải chứng minh bằng lôgic toán học. Một phương pháp chứng minh thường dùng là *bất biến vòng lặp*. Ví dụ với thuật toán sắp xếp chèn ở trên cần chứng minh mệnh đề sau bằng bất biến vòng lặp:

*Mệnh đề:* Sau mỗi bước lặp i của vòng lặp, dãy con A[0], A[1], ..., A[i – 1], A[i] đã được sắp xếp đúng theo thứ tự tăng dần.

Nếu mệnh đề trên được chứng minh bằng toán học thì áp dụng với i = n – 1 ta sẽ kết luận thuật toán sắp xếp chèn là đúng.

Qua các trao đổi trên ta có nhận xét:

– Trong trao đổi 1, việc sử dụng các bộ dữ liệu kiểm thử chưa chứng minh được tính đúng của thuật toán và chương trình. Tuy nhiên, nếu thử được càng nhiều bộ dữ liệu kiểm thử thì độ tin cậy của chương trình càng cao.

– Trao đổi 2 là một suy luận lôgic vì nó bắt nguồn từ ý tưởng chính của thuật toán sắp xếp chèn. Đây là một cách lập luận đơn giản thường được sử dụng để chứng minh tính đúng của thuật toán.

– Trao đổi 3 cho biết việc chứng minh tính đúng của chương trình thường bằng lập luận toán học, sử dụng phương pháp quy nạp toán học. Đây là cách tốt nhất để chứng minh tính đúng của một thuật toán.

Tính đúng của thuật toán cần được chứng minh bằng lập luận toán học. Sử dụng các bộ dữ liệu kiểm thử có thể làm tăng độ tin cậy của chương trình nhưng chưa chứng minh được tính đúng của thuật toán.

**1.** Chương trình sau giải bài toán: Yêu cầu nhập số tự nhiên n và tính tổng 1 + 2 + ... + n.

```
1  n = int(input("Nhập số tự nhiên n: "))
2  S = 0
3  for i in range(n+1):
4      S = S + i
5  print(S)
```

Chương trình trên có đúng không?

**2.** Chương trình sau giải bài toán đếm số các ước số thực sự của số tự nhiên n.

```
1  def dem(n):
2      count = 0
3      k = 2
4      while k < n:
5          if n%k == 0:
6              count = count + 1
7          k = k + 1
8      return count
```

Chương trình trên là đúng hay sai?

## 3. ĐÁNH GIÁ HIỆU QUẢ CHƯƠNG TRÌNH

**Hoạt động 3** **Những tiêu chí đánh giá tính hiệu quả của chương trình**

Thảo luận về các tiêu chí đánh giá tính hiệu quả của thuật toán hay chương trình giải một bài toán.

1. Tiêu chí quan trọng nhất là thời gian chạy chương trình phải nhanh, không cần quan tâm đến không gian bộ nhớ sử dụng của chương trình.
2. Tiêu chí tiết kiệm bộ nhớ là quan trọng nhất, sau đó mới đến thời gian chạy chương trình.
3. Các tiêu chí 1 và 2 không quan trọng mà quan trọng là chương trình được viết một cách đơn giản, rõ ràng, dễ hiểu và áp dụng.

Hiệu quả hay tính tối ưu của chương trình thường được xem xét trên cơ sở *đánh giá độ phức tạp tính toán* (computational complexity) là lượng tài nguyên (amounts of resources) cần thiết để thực hiện chương trình. Hai loại độ phức tạp tính toán phổ biến nhất đó là:

– **Độ phức tạp thời gian** (time complexity) được xác định là thời gian thực hiện chương trình/thuật toán. Thời gian này phụ thuộc vào khối lượng của dữ liệu cần phải lưu trữ trong quá trình thực hiện chương trình/thuật toán, *đặc biệt liên quan tới các bước giải quyết một vấn đề cụ thể* đưa ra trong chương trình/thuật toán.

– **Độ phức tạp không gian** (space complexity) được xác định là tài nguyên của máy tính trong đó có phần bộ nhớ được sử dụng để thực hiện chương trình.

Một chương trình/thuật toán là ***hiệu quả*** (efficient) nếu độ phức tạp của thuật toán này là thấp, nghĩa là *tốn ít thời gian* và *tốn ít bộ nhớ cần thiết* để thực hiện chương trình/thuật toán.

Ngoài ra để đánh giá hiệu quả chương trình đôi khi người ta còn quan tâm tới các tiêu chí như tính dễ hiểu, rõ ràng, ngắn gọn, dễ bảo trì, dễ cài đặt,... của chương trình.

Độ phức tạp thời gian thường bị ảnh hưởng bởi số lần thực hiện các phép toán/câu lệnh có trong chương trình/thuật toán. Với các bài toán kĩ thuật, thiết kế, nghiên cứu khoa học đòi hỏi khối lượng tính toán rất lớn thì việc thiết kế các chương trình/thuật toán có độ phức tạp thời gian thấp luôn là mối quan tâm hàng đầu. Còn đối với những bài toán chạy liên tục như các dịch vụ trực tuyến thì độ tin cậy của phần mềm, tính dễ cài đặt, dễ duy trì, dễ sử dụng lại có vai trò quan trọng.

Trong phạm vi kiến thức phổ thông, ta sẽ chỉ quan tâm tới độ phức tạp thời gian của chương trình/thuật toán.

Trong các bài học sau em sẽ làm quen với việc *xác định độ phức tạp thời gian dựa trên cơ sở ước lượng thời gian thực hiện các bước (các câu lệnh)* trong chương trình/thuật toán.

Chương trình sau cho em biết một cách đơn giản đo thời gian chạy của chương trình. Hàm chính mô tả thuật toán sắp xếp chèn là **InsertionSort(A)** với A là dãy số đầu vào. Hàm **perf_counter()** đo nhịp xung CPU tại thời điểm thực hiện, đơn vị tính thời gian là giây.

Chương trình sau tính được thời gian chạy của hàm **InsertionSort(A)** hay chính là thời gian chạy của thuật toán sắp xếp chèn. Hàm tính thời gian tại thời điểm trước và sau khi chạy hàm chính tại các dòng 10, 12. Lệnh tại dòng 13 sẽ in ra thời gian chạy của hàm chính **InsertionSort(A)**.

```
1  from time import perf_counter
2  def InsertionSort(A):
3      n = len(A)
4      for i in range(1,n):
5          value = A[i]
6          j = i-1
7          while j >= 0 and A[j] > value:
8              A[j+1] = A[j]
9              j = j -1
10         A[j+1] = value
11 A = [3,0,1,10,7,9,5]
```

```
t1 = perf_counter()
InsertionSort(A)
t2 = perf_counter()
print(t2-t1)
```

Tính hiệu quả của chương trình/thuật toán được xem xét trên cơ sở đánh giá độ phức tạp tính toán. Độ phức tạp tính toán quan trọng nhất là độ phức tạp thời gian có liên quan trực tiếp tới các câu lệnh được thực hiện trong chương trình/thuật toán.

Hai tiêu chí đánh giá độ phức tạp tính toán quan trọng nhất là gì?

## LUYỆN TẬP

**1.** Hãy xây dựng các bộ dữ liệu kiểm thử để tìm lỗi cho chương trình tính n! với n là một số nguyên dương nhập từ bàn phím.

```
n = int(input("Nhập số n: "))
if n > 0:
    giaithua=1
    for i in range(1,n+1):
        giaithua = giaithua*i
    print(n,"giai thừa bằng:",giaithua)
```

**2.** Xét hàm mô tả thuật toán tính tổng các số chẵn của một dãy số cho trước.

```
def tongchan(A):
    S = 0
    for i in range(len(A)):
        if A[i] % 2 == 0:
            S = S + A[i]
    return S
```

Tìm hai bộ dữ liệu đầu vào có cùng kích thước của thuật toán trên nhưng có thời gian chạy khác nhau.

## VẬN DỤNG

**1.** Cho dãy các số A = [3, 1, 0, 10, 13, 16, 9, 7, 5, 11].

a) Viết chương trình mô tả thuật toán tìm kiếm phần tử C = 9 của dãy trên. Tính thời gian chính xác thực hiện công việc tìm kiếm này.

b) Giả sử dãy A ở trên đã được sắp xếp theo thứ tự tăng dần: A = [0, 1, 3, 5, 7, 9, 10, 11, 13, 16]. Viết chương trình tìm kiếm phần tử C = 9 theo thuật toán tìm kiếm nhị phân. Tính thời gian thực hiện thuật toán. So sánh với kết quả tìm kiếm ở câu a.

**2.** Viết ba chương trình mô phỏng các thuật toán sắp xếp chèn, sắp xếp chọn và sắp xếp nổi bọt mà em đã biết. Cho biết thời gian thực hiện các chương trình trên với bộ dữ liệu đầu vào là dãy A = [3, 1, 0, 10, 13, 16, 9, 7, 5, 11].

# BÀI 24 ĐÁNH GIÁ ĐỘ PHỨC TẠP THỜI GIAN THUẬT TOÁN

SAU BÀI HỌC NÀY EM SẼ:

- Biết cách phân tích độ phức tạp thời gian thuật toán.
- Nhận biết được phép toán tích cực trong chương trình.
- Biết và thực hiện được tính toán độ phức tạp thời gian của một số thuật toán đã biết.

Quan sát Hình 24.1 chúng ta dễ thấy phép nhân hai số có n chữ số sẽ cần $n^2$ phép nhân và 2n phép cộng, vậy tổng số các phép tính đơn của phép nhân này là $n^2 + 2n$, chúng ta nói độ phức tạp thời gian của phép nhân này có bậc $n^2$.

```
    3 1 2
  × 2 3 1
  -------
    3 1 2
  9 3 6
6 2 4
-------
7 2 0 7 2
```

*Hình 24.1*

Năm 1960, trong một tiết dạy về công nghệ thông tin, nhà toán học Nga, Viện sĩ Kolmogorov đã hỏi các sinh viên của mình là có ai tìm được cách tính phép nhân trên với thời gian tốt hơn bậc $n^2$ được không? Khi đó đây là một bài toán chưa có lời giải. Đúng một tuần sau, một sinh viên tên là Karatsuba đã đưa cho Viện sĩ Kolmogorov một lời giải tốt hơn về phép tính nhân trên chỉ với độ phức tạp thời gian bậc $n^{1,58496}$.

Quan sát và ước lượng thời gian thực hiện các đoạn chương trình 1 và 2 trong Hình 24.2. Chương trình nào chạy nhanh hơn? Vì sao?

*Chương trình 1*

```
1 n = 100
2 C = 0
3 for k in range(n):
4         C = C + 1
5 print(C)
```

*Chương trình 2*

```
1 n = 100
2 C = 0
3 for i in range(n):
4     for j in range(n):
5         C = C + 1
6 print(C)
```

*Hình 24.2*

## 1. ĐÁNH GIÁ THỜI GIAN THỰC HIỆN CHƯƠNG TRÌNH

Có thể không cần cài đặt và chạy chương trình mà vẫn ước lượng được thời gian chạy dựa trên việc tính tổng thời gian các phép tính đơn và các lệnh đơn của chương trình. Cách tính này có thể không chính xác hoàn toàn như thời gian thực nhưng có thể dùng để so sánh và ước lượng thời gian chạy chương trình khá chính xác. Khi tính thời gian chạy chương trình, có thể coi tất cả các lệnh đơn (ví dụ lệnh gán) và các phép tính đơn (ví dụ phép tính số học, phép so sánh) có thời gian chạy như nhau, được gọi chung là một đơn vị thời gian. Cách tính này sẽ làm đơn giản hoá cách phân tích thời gian tính toán nhưng vẫn bảo đảm độ chính xác của tính toán.

## Hoạt động 1 Tìm hiểu cách đánh giá thời gian thực hiện chương trình

Quan sát và thực hiện đánh giá thời gian chạy của các chương trình 1 và 2 trong Hình 24.2. Từ đó biết và hiểu được cách đánh giá thời gian thực hiện chương trình.

**Chương trình 1.** Gọi $T_1$ là thời gian chạy của chương trình này.

– Mỗi lệnh tại dòng 1 và 2 cần 1 đơn vị thời gian để thực hiện.

– Vòng lặp tại dòng 3 có n bước lặp, mỗi bước của vòng lặp sẽ thực hiện lệnh tại dòng 4, lệnh này cần 1 đơn vị thời gian. Vậy suy ra tổng thời gian của vòng lặp 3 là n thời gian.

– Lệnh cuối tại dòng 5 cần 1 đơn vị thời gian.

Vậy để thực hiện toàn bộ chương trình 1 cần: $T_1 = T_1(n) = 2 + n + 1 = n + 3$ đơn vị thời gian.

**Chương trình 2.** Gọi $T_2$ là thời gian chạy của chương trình này.

– Mỗi lệnh tại dòng 1 và 2 cần 1 đơn vị thời gian.

– Các lệnh tại dòng 3, 4 là hai vòng lặp lồng nhau. Mỗi vòng lặp có n bước như vậy thực chất có $n^2$ bước lặp của hai lệnh này. Mỗi bước lặp sẽ thực hiện các lệnh tại dòng 5, lệnh này cần 1 đơn vị thời gian. Vậy suy ra tổng thời gian của vòng lặp 3, 4 là $n^2$ đơn vị thời gian.

– Lệnh cuối tại dòng 6 cần 1 đơn vị thời gian.

Vậy tổng hợp toàn bộ chương trình 2 ta có: $T_2 = T_2(n) = 2 + n^2 + 1 = n^2 + 3$ đơn vị thời gian.

Cách đánh giá thời gian chạy chương trình được dựa trên một bộ khung các nguyên tắc dùng làm căn cứ để tính toán. Các nguyên tắc khung như sau:

1. Các phép toán đơn giản như phép tính số học +, –, *, /, phép lấy thương nguyên và số dư, các phép so sánh sẽ tính là 1 đơn vị thời gian.
2. Các phép toán lôgic cơ bản như AND, OR, NOT sẽ tính là 1 đơn vị thời gian.
3. Các lệnh đơn như lệnh gán, lệnh in, đọc dữ liệu,... tính là 1 đơn vị thời gian.
4. Vòng lặp for hoặc while sẽ được tính thời gian bằng tổng đơn vị thời gian thực hiện của mỗi bước lặp.
5. Lệnh if với nhiều trường hợp rẽ nhánh sẽ được tính thời gian bằng đơn vị thời gian lớn nhất của các lệnh nhánh.

Áp dụng các nguyên tắc tính khung thời gian trên chúng ta có thể tính được gần chính xác thời gian thực hiện chương trình mà không cần cài đặt và chạy chương trình trên máy tính.

*Lưu ý*: Trong một chương trình, phép toán được thực hiện nhiều nhất và đóng vai trò chính khi thực hiện tính thời gian, được gọi là **phép toán tích cực**. Ví dụ trong chương trình 1 thì phép toán tích cực là phép cộng C = C + 1 tại dòng 4. Với chương trình 2 thì phép cộng C = C + 1 tại dòng 6 chính là phép toán tích cực.

**1.** Các lệnh và đoạn chương trình sau cần chạy trong bao nhiêu đơn vị thời gian?

(a)

```
1  n = 1000000
2  for k in range(n):
3      if k%3 == 0:
4          print(k)
```

(b)

```
1  n = 1000000
2  b = 3
3  for k in range(0,n,b):
4      print(k)
```

**2.** Khẳng định "Trong mọi chương trình chỉ có đúng một phép toán tích cực" là đúng hay sai?

## 2. PHÂN TÍCH ĐỘ PHỨC TẠP THỜI GIAN THUẬT TOÁN

**Hoạt động 2** **Tìm hiểu khái niệm độ phức tạp thời gian thuật toán**

Cùng trao đổi và tìm hiểu cách phân loại thuật toán dựa trên độ phức tạp thời gian thuật toán.

Trong phạm vi kiến thức phổ thông có thể hiểu độ phức tạp thời gian thuật toán là khối lượng thời gian cần thiết để chạy chương trình thể hiện thuật toán. Một trong các cách phân loại thuật toán đó là dựa trên việc ước lượng độ phức tạp thời gian thuật toán. Độ phức tạp thời gian, trong trường hợp tổng quát, có thể coi là một hàm số T(n), với n là một số tự nhiên được xác định tuỳ thuộc từng bài toán cụ thể liên quan tới dữ liệu đầu vào. Giá trị của T(n) thường được xác định trên cơ sở số lượng các phép toán/câu lệnh cần thực hiện trong chương trình/thuật toán. Khi n càng lớn thì thời gian T(n) sẽ tăng lên nhưng tốc độ tăng khác nhau. Để phân loại được các hàm thời gian này, các nhà khoa học đã đưa vào định nghĩa **O-lớn**. Kí hiệu **O-lớn** (big-O) dùng để so sánh và phân tích bậc của hàm thời gian T(n) khi n tăng lên vô cùng. Ví dụ chương trình 1 ở Hình 24.2 có độ phức tạp thời gian bậc n và viết là $T_1(n) = O(n)$, ý nghĩa là khi n tiến tới vô cùng, $T_1(n)$ sẽ tăng nhưng không quá bậc của n. Tương tự, chương trình 2 có độ phức tạp thời gian bậc $n^2$, và viết là $T_2(n) = O(n^2)$, ý nghĩa là khi n tăng lên thì $T_2(n)$ sẽ tăng không vượt quá bậc của $n^2$.

Định nghĩa kí hiệu **O-lớn**:

Cho f(n) và g(n) là hai hàm có đối số tự nhiên. Ta viết $f(n) = O(g(n))$ và nói f(n) có bậc **O-lớn** của g(n) nếu tồn tại hằng số $c > 0$ và số tự nhiên $n_0 \geq 1$ sao cho với mọi $n \geq n_0$ ta có $f(n) \leq c.g(n)$. Nếu f(n) là **O-lớn** của g(n) thì có thể viết: $f(n) = O(g(n))$.

Xét một số ví dụ.

– Chương trình 1 ở Hình 24.2 có hàm thời gian $T_1(n) = n + 3$.

Chọn $c = 2$, $n_0 = 3$. Khi đó với $n \geq n_0$ ta có: $T_1(n) = n + 3 \leq n + n = c.n$. Do đó $T_1(n) = O(n)$. Chúng ta nói chương trình 1 có độ phức tạp thời gian O(n) – tuyến tính.

– Chương trình 2 ở Hình 24.2 có hàm thời gian $T_2(n) = n^2 + 3$.

Chọn $c = 2$, $n_0 = 2$. Khi đó với $n \geq n_0$, ta có:

$T_2(n) = n^2 + 3 < n^2 + n_0^2 \leq n^2 + n^2 = 2n^2 = c.n^2$.

Vậy suy ra $T_2(n) = O(n^2)$. Ta nói chương trình 2 ở trên có độ phức tạp thời gian $O(n^2)$ – bình phương.

Kí hiệu **O-lớn** dùng để đánh giá và phân loại độ phức tạp thời gian của thuật toán khi kích thước đầu vào của bài toán tăng lên vô cùng. Các thuật toán sẽ được đánh giá qua độ phức tạp của một số hàm chuẩn như: O(1) – hằng số, O(logn) – logarit, O(n) – tuyến tính, O(nlogn) – tuyến tính logarit, $O(n^2)$ – bình phương, $O(n^k)$ – đa thức, $O(a^n)$ – luỹ thừa, O(n!) – giai thừa.

Tính độ phức tạp của các hàm thời gian sau:

a) $T(n) = 2n(n-2) + 4$.

b) $T(n) = n^3 + 5n - 3$.

## 3. MỘT SỐ QUY TẮC THỰC HÀNH TÍNH ĐỘ PHỨC TẠP THỜI GIAN THUẬT TOÁN

**Hoạt động 3** **Tìm hiểu một số quy tắc đơn giản tính độ phức tạp thời gian thuật toán**

Đọc, quan sát, thảo luận để biết một số quy tắc đơn giản tính độ phức tạp thời gian thuật toán.

Một số quy tắc tính đơn giản để tính độ phức tạp thời gian thuật toán.

**QT1.** *Quy tắc cộng:* $O(f(n) + g(n)) = O(\max(f(n), g(n)))$. Quy tắc này được áp dụng khi tính độ phức tạp thời gian cho hai chương trình được thực hiện nối tiếp nhau.

**QT2.** *Quy tắc nhân:* Phép nhân với hằng số: $O(C \cdot f(n)) = O(f(n))$, với C là hằng số bất kì.

Phép nhân với hàm số: $O(f(n) \cdot g(n)) = O(f(n) \cdot O(g(n)))$. Quy tắc này được áp dụng tính độ phức tạp thời gian cho chương trình có hai vòng lặp lồng nhau.

Ví dụ:

$T(n) = 10n^2 = O(n^2)$ (*Quy tắc nhân với hằng số*).

$T(n) = 3n^2 + n\log n = O(\max(3n^2, n\log n))$ (*Quy tắc cộng*) $= O(3n^2) = O(n^2)$ (*Quy tắc nhân với hằng số*).

Áp dụng các quy tắc trên để tính độ phức tạp của các hàm thời gian sau:

a) $T(n) = n^3 + n\log n + 2n + 1$.

b) $T(n) = 3n^4 + 2n^2\log n + 10$.

## LUYỆN TẬP

**1.** Xác định độ phức tạp thời gian tính toán cho chương trình sau:

```
1 n = 1000
2 S = 0
3 for i in range(n):
4     S = S + i(i+1)
5 print(S)
```

**2.** Xác định độ phức tạp thời gian tính toán cho chương trình sau:

```
1 n = 1000
2 Sum=0
3 i = 1
4 while i < n:
5     i = i*2
6     Sum = Sum + i
7 print(Sum)
```

## VẬN DỤNG

**1.** Xác định độ phức tạp thời gian của thuật toán sắp xếp chọn đã được học trong Bài 21.

**2.** Em hãy thiết lập chương trình và tính thời gian chạy thực tế trên máy tính của các chương trình 1 và 2 ở Hình 24.2 với các giá trị n khác nhau, từ đó thấy được ý nghĩa sự khác biệt độ phức tạp thời gian của hai chương trình này.

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

# BÀI 25 THỰC HÀNH XÁC ĐỊNH ĐỘ PHỨC TẠP THỜI GIAN THUẬT TOÁN

SAU BÀI HỌC NÀY EM SẼ:

- Thực hành xác định độ phức tạp (O-lớn) của hàm thời gian.



Biết cách phân tích, đánh giá độ phức tạp thuật toán là kĩ năng quan trọng của người thiết kế thuật toán và chương trình. Các quy tắc đơn giản tính độ phức tạp thời gian mang lại cho em điều gì khi đánh giá thuật toán?



## Nhiệm vụ 1

Xác định độ phức tạp thời gian tính toán của thuật toán tìm kiếm tuần tự được thể hiện bằng chương trình sau:

```
1  def LinearSearch(A,K):
2      for i in range(len(A)):
3          if A[i] == K:
4              return i
5      return -1
```

**Hướng dẫn:**

*Bước 1.* Phân tích thời gian tính toán của thuật toán.

Gọi n là kích thước của mảng đầu vào, T(n) là thời gian thực hiện thuật toán.

– Đối với mã lệnh trên, chương trình sẽ thực hiện duyệt mảng và với mỗi bước lặp sẽ kiểm tra phần tử thứ i có bằng với phần tử cần tìm kiếm không (dòng 3). Nếu bằng thì chương trình sẽ trả về chỉ số của phần tử tìm thấy và kết thúc (dòng 4). Như vậy, chương trình có thể kết thúc khi chưa duyệt hết mảng (trường hợp đã tìm thấy phần tử) và tối đa là duyệt hết mảng. Như vậy, trong trường hợp tồi nhất vòng lặp ở dòng 2 sẽ thực hiện n bước lặp, mỗi bước lặp sẽ thực hiện lệnh so sánh ở dòng 3 tốn 1 đơn vị thời gian.

– Lệnh trả về sẽ được thực hiện duy nhất một lần ở dòng 4 (trường hợp tìm thấy phần tử trong mảng) hoặc dòng 5 (trường hợp không tìm thấy phần tử trong mảng) mất 1 đơn vị thời gian.

Do đó, tổng số phép tính cơ bản của chương trình trong trường hợp tồi nhất là

$$T(n) = n+1.$$

*Bước 2.* Xác định độ phức tạp O-lớn của thuật toán

$$T(n) = n + 1 = O(n+1) = O(\max(n, 1)) = O(n).$$

Vậy thuật toán tìm kiếm tuần tự có độ phức tạp tuyến tính.

## Nhiệm vụ 2

Xác định độ phức tạp của thuật toán sắp xếp chọn được thể hiện bằng chương trình sau:

```
1  def SelectionSort(A):
2      n = len(A)
3      for i in range(n-1):
4          iMin = i
5          for j in range(i+1,n):
6              if A[j] < A[iMin]:
7                  iMin = j
8          A[i],A[iMin] = A[iMin],A[i]
```

**Hướng dẫn:**

Ý tưởng của thuật toán sắp xếp chọn là tại mỗi bước thứ i của vòng lặp sẽ tìm chính xác phần tử tại vị trí thứ i, tức là sẽ tìm phần tử nhỏ nhất trong dãy từ A[i], A[i + 1],…, A[n – 1] và đổi chỗ phần tử nhỏ nhất này với A[i].

Gọi n là kích thước của mảng A, T(n) là thời gian chạy của thuật toán. Thời gian chạy của thuật toán được phân tích như sau:

– Lệnh gán ở dòng 2 tốn 1 đơn vị thời gian.

– Vòng lặp tại dòng 3 biến i sẽ chạy từ 0 đến n – 2, vậy vòng lặp này có n – 1 bước lặp.

– Tại mỗi bước lặp của lệnh **for** tại dòng 3 chương trình sẽ thực hiện các lệnh sau:

+ Lệnh gán tại dòng 4 tốn 1 đơn vị thời gian

+ Vòng lặp **for** tại dòng 5, biến j sẽ chạy từ i + 1 đến n – 1, nên vòng lặp này có n – i – 1 bước lặp.

+ Với mỗi bước lặp tại dòng 5 chương trình sẽ thực hiện:

1 lệnh so sánh tại dòng 6 tốn 1 đơn vị thời gian và một lệnh gán tại dòng 7 tốn 1 đơn vị thời gian (nếu điều kiện thoả mãn).

Như vậy mỗi bước lặp tại dòng 5 sẽ tốn tối đa 2 đơn vị thời gian.

+ 1 lệnh đổi chỗ tại dòng 8 tốn 3 đơn vị thời gian.

Tổng hợp lại ta thấy thời gian chạy chương trình trên là:

$$T(n) = 1 + \sum_{i=0}^{n-2}\left(1 + 2(n-i-1) + 3\right)$$

$$T(n) = 1 + 4(n-1) + 2\sum_{i=0}^{n-2}(n-i-1)$$

$$T(n) = 1 + 4(n-1) + 2\sum_{k=1}^{n-1}k$$

$$T(n) = 1 + 4(n-1) + n(n-1)$$

$T(n) = n^2 + 3n - 3$

Xác định độ phức tạp O-lớn của thuật toán:

$T(n) = O(max(n^2, 3n, -3)) = O(n^2)$

Vậy thuật toán sắp xếp chọn có độ phức tạp thời gian bình phương.

## LUYỆN TẬP

**1.** Xác định độ phức tạp của thuật toán sắp xếp nổi bọt sau:

```
1 def BubbleSort(A):
2     n = len(A)
3     for i in range(n-1):
4         for j in range(n-1-i):
5             if A[j] > A[j+1]:
6                 A[j],A[j+1] = A[j+1],A[j]
```

**2.** Cho biết hàm sau sẽ trả về giá trị là bao nhiêu? Xác định độ phức tạp thời gian O-lớn của chương trình.

```
1 def Mystery(n):
2     r = 0
3     for i in range(n-1):
4         for j in range(i+1,n):
5             for k in range(1,j):
6                 r = r+1
7     return r
```

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

## VẬN DỤNG

**1.** Giả sử rằng mỗi phép tính đơn được thực hiện trong 1 micro giây (1 µs = một phần triệu giây). Hãy xác định giá trị lớn nhất của n trong các thuật toán tìm kiếm tuần tự, sắp xếp chèn và sắp xếp chọn nếu thời gian thực thi các thuật toán là 1 giây, 1 phút và 1 giờ.

**2.** Hãy cho biết hàm sau thực hiện công việc gì. Xác định độ phức tạp thời gian của thuật toán.

```
1 def func(A):
2     n = len(A)
3     for i in range(n-1):
4         for j in range(i+1,n):
5             if A[i] > A[j]:
6                 A[i],A[j] = A[j],A[i]
```

# BÀI 26 PHƯƠNG PHÁP LÀM MỊN DẦN TRONG THIẾT KẾ CHƯƠNG TRÌNH

SAU BÀI HỌC NÀY EM SẼ:

- Biết và giải thích được phương pháp làm mịn dần trong lập trình.
- Vận dụng được phương pháp làm mịn dần để thiết kế chương trình.

Em đã biết thiết kế một số thuật toán và chương trình: tìm kiếm tuần tự, tìm kiếm nhị phân, sắp xếp chèn, sắp xếp chọn, sắp xếp nổi bọt. Tất cả các thiết kế chương trình đó có điểm nào chung?

Theo em, để thiết kế một thuật toán đúng giải một bài toán cho trước cần trải qua các bước như thế nào? Nêu quan điểm của riêng em và trao đổi với các bạn.

## 1. PHƯƠNG PHÁP THIẾT KẾ LÀM MỊN DẦN

**Hoạt động 1** Tìm hiểu các bước thiết kế làm mịn dần

Cùng trao đổi, thảo luận các bước thiết kế chương trình theo thuật toán sắp xếp chèn, từ đó đưa ra phương pháp chính khi thiết kế chương trình. Sau mỗi bước thiết kế cần trao đổi và trả lời các câu hỏi sau:

1. Bước này đã thực hiện được công việc gì?
2. Kết quả vừa thực hiện với kết quả của bước trước đó khác nhau như thế nào?

**Bài toán gốc.** Cho trước dãy số A: A[0], A[1], ..., A[n-1]. Cần tiến hành sắp xếp dãy trên theo thứ tự tăng dần. Kết quả phải nhận được:

`A[0] ≤ A[1] ≤ ... ≤ A[n-1]`

Ví dụ với bộ dữ liệu đầu vào là dãy [2,1,7,10,4] thì kết quả thu được dãy [1,2,4,7,10].

Quá trình phân tích, thiết kế được mô tả theo các bước sau.

### a) Tìm hiểu bài toán

Bài toán gốc là cho trước dãy A, cần sắp xếp lại dãy này theo thứ tự tăng dần.

### b) Thiết kế chương trình giải bài toán

Việc thiết kế chương trình giải bài toán được chia thành nhiều bước, trong đó các hành động ở bước sau là sự cụ thể hoá hơn ý tưởng, thao tác được nêu trong bước trước.

**Bước 1.** Thiết lập ý tưởng thiết kế ban đầu.

Ý tưởng ban đầu của thuật toán đơn giản như sau: Cần duyệt một lượt từ phần tử thứ hai đến phần tử cuối của dãy sao cho khi kết thúc thì dãy cũng được sắp xếp xong. Như vậy phần chính của thuật toán là một vòng lặp với biến i chạy từ chỉ số 1 đến n – 1. Với mỗi giá trị i, cần thực hiện một số thao tác để bổ sung A[i] vào dãy các phần tử đã được sắp xếp A[0], A[1],..., A[i-1] sao cho dãy mới thu được từ A[0] đến A[i] được sắp xếp đúng.

Như vậy, thuật toán ban đầu có thể được mô tả như sau:

```
1 for i in range(1,n):
2     <Đặt A[i] vào đúng vị trí của dãy A[0],A[1],...,A[i-1]>
```

Tại dòng 2 của sơ đồ trên, bài toán được đặt ra là: "Chèn phần tử A[i] vào đúng vị trí của dãy A[0], A[1], ..., A[i-1]".

**Bước 2.** Làm chi tiết hơn, thực hiện việc "Chèn A[i] vào đúng vị trí."

Vì các phần tử bên trái của A[i] là A[0], A[1], ..., A[i-1] đã được sắp xếp đúng nên thao tác "chèn" phần tử A[i] sẽ được thực hiện như sau:

```
<Lấy phần tử A[i] ra và lần lượt chuyển các phần tử bên trái A[i] nhưng có giá trị lớn hơn A[i] sang phải. Sau đó đặt A[i] vào vị trí trống>
```

Theo mô tả trên, việc "Chèn A[i] vào đúng vị trí" có thể được thực hiện như sau:

Chèn A[i] vào đúng vị trí

```
1 Nhấc phần tử A[i] lên.
2 Chuyển các phần tử bên trái A[i] và lớn hơn A[i] sang phải.
3 Chèn A[i] vào vị trí trống.
```

Các bước tiếp theo sẽ làm mịn hơn, chi tiết hơn các thao tác trên.

**Bước 3.** Nhấc A[i] lên.

Thao tác này sẽ được thực hiện đơn giản bằng việc tạo ra một biến mới **value** để lưu trữ giá trị A[i].

```
value = A[i]
```

**Bước 4.** Chuyển các phần tử bên trái A[i] và lớn hơn A[i] sang phải.

Thao tác này có thể được thực hiện như sau: Thiết lập biến j = i – 1 là chỉ số của phần tử ngay bên trái A[i]. Sau đó liên tục so sánh A[j] với value. Nếu A[j] > value thì chuyển A[j] sang phải một vị trí bằng lệnh A[j+1] = A[j] và giảm j = j – 1. Quá trình sẽ kết thúc khi đi hết bên trái của dãy hoặc A[j] <= value. Tất cả các công việc này được thể hiện bằng đoạn chương trình sau:

```
1 j = i-1
2 while j >= 0 and A[j] > value:
3     A[j+1] = A[j]
4     j = j - 1
```

**Bước 5.** Chèn A[i] vào đúng vị trí trống.

Từ bước 4 chúng ta đã biết quá trình chuyển sang phải của các phần tử A[j] sẽ kết thúc khi A[j] <= A[i], do đó vị trí j+1 chính là vị trí trống cần chèn. Việc chèn phần tử A[i] (giá trị được lưu trong value) vào vị trí j + 1 được thực hiện bằng câu lệnh:

```
A[j+1] = value
```

Như vậy ba thao tác đã nêu ở bước 2 trên có thể được thực hiện bằng các câu lệnh chương trình như sau:

Chèn A[i] vào đúng vị trí

```
1 value = A[i]
2 j = i-1
3 while j >= 0 and A[j] > value:
4     A[j+1] = A[j]
5     j = j - 1
6 A[j+1] = value
```

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

Tới đây quá trình thiết kế kết thúc vì chúng ta đã chi tiết hoá bằng các câu lệnh tất cả các thao tác được mô tả trong các bước trên.

### c) Chương trình hoàn chỉnh

Chương trình giải bài toán đặt ra được thiết kế hoàn chỉnh dưới dạng hàm InsertionSort(A). Tổng hợp các bước trên chúng ta có chương trình hoàn chỉnh như sau.

```
1 def InsertionSort(A):
2     n = len(A)
3     for i in range(1,n):
4         value = A[i]
5         j = i-1
6         while j >= 0 and A[j] > value:
7            A[j+1] = A[j]
8            j = j -1
9         A[j+1] = value
```

Như vậy quá trình thiết kế chương trình theo thuật toán sắp xếp chèn đã trải qua một số bước, mỗi bước sẽ thực hiện chi tiết hoá hay còn gọi là làm mịn dần các phân tích của bước trước đó.

> Phương pháp làm mịn dần trong thiết kế chương trình là quá trình chi tiết hoá từ ý tưởng của các bước trước thành những hành động cụ thể hơn ở các bước sau. Ở bước cuối cùng, các hành động tương ứng với các câu lệnh của ngôn ngữ lập trình để viết chương trình hoàn chỉnh.



1. Trong các bước đã thực hiện của bài toán sắp xếp chèn ở trên, bước nào là đơn giản nhất theo nghĩa có thể thực hiện ngay bằng các lệnh lập trình?
2. Nếu bài toán đặt ra là sắp xếp dãy A theo thứ tự giảm dần thì các bước thiết kế như trên có cần thay đổi không? Thay đổi như thế nào?

## 2. THIẾT KẾ CHƯƠNG TRÌNH BẰNG PHƯƠNG PHÁP LÀM MỊN DẦN

**Hoạt động 2** **Thiết kế chương trình bằng phương pháp làm mịn dần**

Thực hiện thiết kế thuật toán và chương trình bằng phương pháp làm mịn dần theo các bài toán sau. Trao đổi, thảo luận với bạn bè để thiết lập được lời giải tốt hơn.



**Bài toán.** Cho trước dãy số A: A[0], A[1], ..., A[n-1]. Cặp phần tử A[i], A[j] được gọi là nghịch đảo nếu i < j nhưng A[i] > A[j]. Cần viết chương trình đếm số các cặp nghịch đảo của dãy A. Ví dụ dãy 3, 4, 2, 1 sẽ có 5 cặp nghịch đảo là (3,2), (3,1), (4,2), (4,1), (2,1).

*Thiết kế theo phương pháp làm mịn dần*

### a) Tìm hiểu bài toán

Bài toán gốc là cho trước dãy số A có n phần tử, cần đếm số các cặp phần tử nghịch đảo của A.

## b) Thiết kế chương trình giải bài toán

Chúng ta sẽ thiết kế lời giải bài toán theo phương pháp làm mịn dần.

**Bước 1.** Thiết lập ý tưởng thiết kế ban đầu.

Bài toán có yêu cầu chính là đếm tất cả các cặp chỉ số nghịch đảo của dãy A. Vậy phần khung chính của chương trình sẽ là:

```
<Đếm số lượng các cặp số nghịch đảo (A[i],A[j]) của dãy A, trả về giá trị này>
```

Như vậy để thực hiện được yêu cầu trên chúng ta cần thực hiện hai công việc: cần tìm tất cả các cặp chỉ số (i, j) có thể tạo cặp nghịch đảo A[i], A[j], sau đó kiểm tra xem cặp này có là nghịch đảo không, nếu có thì tăng biến đếm lên 1 đơn vị.

Lược đồ thuật toán ban đầu có thể được mô tả như sau:

```
count = 0
Tìm tất cả các cặp chỉ số (i,j) có thể tạo ra cặp phần tử nghịch đảo
Kiểm tra nếu cặp A[i], A[j] là nghịch đảo thì tăng count lên 1 đơn vị.
return count
```

**Bước 2.** Tìm tất cả các cặp chỉ số (i,j)

Cách tìm tự nhiên tất cả các cặp nghịch đảo là cần duyệt trên tất cả các cặp (i, j) trong đó i, j chạy từ 0 đến n – 1. Như vậy có thể thiết lập hai vòng lặp theo i, j để tìm. Chú ý để tiết kiệm thời gian chúng ta sẽ chỉ tìm các chỉ số i chạy từ 0 đến n-2, chỉ số j tính từ i+1 đến n – 1. Kết quả bước làm mịn này là đoạn chương trình sau:

```
for i in range(n-1):
    for j in range(i+1,n):
```

Như vậy tới bước này, thuật toán gốc có thể được mô tả như sau:

```
count = 0
for i in range(n-1):
    for j in range(i+1,n):
        if <cặp (i,j) là nghịch đảo>:
            tăng count lên 1 đơn vị
return count
```

**Bước 3.** Kiểm tra tính nghịch đảo của cặp (i,j).

Chúng ta đã biết cặp (i, j) sẽ là nghịch đảo khi và chỉ khi i < j và A[i] > A[j]. Tuy nhiên tại bước 2 chúng ta đã thiết lập được tất cả các cặp (i, j) với điều kiện i < j do đó việc kiểm tra nghịch đảo chỉ còn một điều kiện là A[i] > A[j]. Kết quả làm mịn của bước 3 như sau:

```
if A[i] > A[j]:
    count = count + 1
```

Tới bước này các thao tác chi tiết cần thực hiện để giải bài toán đã gần hoàn thành như sau:

```
count = 0
for i in range(n-1):
    for j in range(1,n):
        if A[i] > A[j]:
            count = count + 1
return count
```

## c) Chương trình hoàn chỉnh

Trên cơ sở các phân tích trên chúng ta có thể thiết lập hàm Nghichdao(A) để đếm số các cặp nghịch đảo của dãy A cụ thể như sau:

```
1  def Nghichdao(A):
2      n = len(A)
3      count = 0
4      for i in range(n-1):
5          for j in range(i+1,n):
6              if A[i] > A[j]:
7                  count = count + 1
8      return count
```

Phương pháp làm mịn dần trong thiết kế chương trình phải tuân thủ các quy trình và nguyên tắc sau:

- Chia việc thiết kế thành từng bước và thực hiện lần lượt các bước.
- Mỗi bước lớn có thể được chia thành nhiều bước nhỏ hơn để giải quyết độc lập.
- Tiếp cận bài toán từ tổng quan đến chi tiết, mỗi bước tiếp theo sẽ phải là thiết kế chi tiết hơn bước trước đó. Quá trình như vậy sẽ tiếp tục cho đến khi viết xong toàn bộ các câu lệnh của chương trình giải bài toán đã cho.

**1.** Với Bài toán ở Hoạt động 1, có thể tách các dòng lệnh từ 4 đến 9 thành một hàm con độc lập được không?

**2.** Trong thiết kế bài toán tìm các cặp phần tử nghịch đảo, các bước sau đã thực hiện những thay đổi quan trọng nào so với bước trước đó?

## LUYỆN TẬP

**1.** Phát biểu sau đúng hay sai?

Khi thiết kế chương trình thì việc đầu tiên là tìm hiểu yêu cầu chung của bài toán, xác định đầu vào, đầu ra của bài toán, sau đó mới đi cụ thể vào chi tiết.

**2.** Sử dụng thiết kế của Bài toán ở Hoạt động 2, tìm tất cả các cặp nghịch đảo của dãy: 3, 2, 1, 5, 4.

## VẬN DỤNG

**1.** Sử dụng phương pháp làm mịn dần để giải bài toán sau: Cho trước số tự nhiên không âm n, viết chương trình kiểm tra xem số n có phải là số nguyên tố hay không. Chương trình cần thông báo "CÓ" nếu n là số nguyên tố, ngược lại thông báo "KHÔNG".

**2.** Với thuật toán sắp xếp chèn, chứng minh rằng nếu thay toàn bộ phần

`<Chèn A[i] vào vị trí đúng của dãy con A[0], A[1], ..., A[i - 1]>`

bằng các lệnh sau thì chương trình vẫn đúng:

```
1  j = i
2  while j > 0 and A[j] < A[j-1]:
3      Đổi chỗ A[j] và A[j-1]
4      j = j - 1
```

# BÀI 27 THỰC HÀNH THIẾT KẾ CHƯƠNG TRÌNH THEO PHƯƠNG PHÁP LÀM MỊN DẦN

SAU BÀI HỌC NÀY EM SẼ:

- Thực hành thiết kế chương trình theo phương pháp làm mịn dần.

Phương pháp làm mịn dần là một trong các cách tiếp cận tổng quát khi giải quyết các bài toán cụ thể. Em có thể sử dụng sơ đồ hình cây để mô tả phương pháp này không?

## Nhiệm vụ 1. Kiểm tra hoán vị

Cho trước một dãy n số, các số được kí hiệu A[0], A[1], ..., A[n-1]. Cần thiết kế chương trình kiểm tra xem dãy trên có phải là một hoán vị của dãy số 1, 2, ..., n hay không. Chương trình cần thông báo kết quả là CÓ hoặc KHÔNG.

Hướng dẫn: *Phân tích thiết kế*

### a) Tìm hiểu bài toán

Bài toán gốc: Cho trước dãy số A gồm n phần tử, cần kiểm tra xem A có phải là một hoán vị của dãy số 1, 2,..., n hay không.

### b) Thiết kế theo phương pháp làm mịn dần

**Bước 1.** Thiết lập ý tưởng thiết kế ban đầu.

Ý tưởng ban đầu để giải bài toán này khá đơn giản: Sắp xếp A theo thứ tự tăng dần. Sau đó chỉ cần so sánh A có trùng khít với dãy [1, 2, ..., n] hay không. Như vậy sơ đồ khung ban đầu của lời giải sẽ như sau:

```
Sắp xếp dãy A theo thứ tự tăng dần.
Kiểm tra A có phải là dãy 1, 2, ..., n hay không
```

**Bước 2.** Sắp xếp dãy A

Việc sắp xếp dãy A theo thứ tự tăng dần được mô tả bằng hàm sapxep(A) theo một trong các thuật toán sắp xếp mà chúng ta đã biết. Ví dụ bằng thuật toán sau:

```
def sapxep(A):
    for i in range(len(A)):
        j = i
        while j > 0 and A[j] < A[j-1]:
            A[j],A[j-1] = A[j-1],A[j]
            j = j - 1
```

**Bước 3.** Với dãy A đã được sắp xếp, kiểm tra A có phải là dãy [1, 2, ..., n]?

Việc kiểm tra này có thể được thực hiện đơn giản như sau:

```
Kiểm tra lần lượt các phần tử của A với các phần tử tương ứng của dãy
1, 2,..., n. Nếu tất cả các so sánh đều bằng nhau thì trả về kết quả True,
ngược lại trả về False.
```

Sử dụng biến kq để trả lại kết quả của việc so sánh A và dãy các số 1, 2, ..., n, ta có thể viết đoạn chương trình chi tiết thực hiện công việc được mô tả trên như sau:

```
    sapxep(A)
    kq = True
    for i in range(len(A)):
        if A[i] != i+1:
            kq = False
            break
    return kq
```

Kết quả của toàn bộ bước này có thể viết dưới dạng hàm kt_hoanvi(A), trong đó A là dãy số ban đầu. Hàm sẽ trả về True nếu A là hoán vị của [1, 2, ..., n], ngược lại trả về False.

```
def kt_hoanvi(A):
    sapxep(A)
    kq = True
    for i in range(len(A)):
        if A[i] != i+1:
            kq = False
            break
    return kq
```

## c) Chương trình hoàn chỉnh

Tổng hợp các bước làm mịn trên chúng ta thu được chương trình hoàn chỉnh cho bài toán. Chương trình hoàn chỉnh sẽ có hai chương trình con là hàm sapxep() và kt_hoanvi().

```
def sapxep(A):
    for i in range(len(A)):
        j = i
        while j > 0 and A[j] < A[j-1]:
            A[j],A[j-1] = A[j-1],A[j]
            j = j - 1

def kt_hoanvi(A):
    sapxep(A)
    kq = True
    for i in range(len(A)):
        if A[i] != i+1:
            kq = False
            break
    return kq

# Chương trình chính
A = [2,1,9,10,8,6,5,2,3,1]
```

```
19 if kt_hoanvi(A):
20     print("CÓ")
21 else:
22     print("KHÔNG")
```



## Nhiệm vụ 2. Đếm số lần lặp

Thiết kế và viết chương trình theo phương pháp làm mịn dần cho bài toán sau: Cho trước dãy số A[0], A[1], ..., A[n-1]. Cần tính được mỗi giá trị của các phần tử của dãy trên được lặp lại bao nhiêu lần trong dãy đó. Kết quả cần được đưa ra dãy B. Như vậy dãy B sẽ có ý nghĩa như sau: B[k] = số lần lặp của phần tử A[k] trong dãy A.

Ví dụ nếu A = [2, 1, 1, 3, 5, 10, 2, 5] thì B = [2, 2, 2, 1, 2, 1, 2, 2].

Hướng dẫn: *Phân tích thiết kế*

### a) Tìm hiểu bài toán

Bài toán gốc: cho trước dãy số A có n phần tử. Cần tạo ra một dãy mới là số các lần lặp của các phần tử tương ứng trong A.

### b) Thiết kế theo phương pháp làm mịn dần

**Bước 1.** Thiết lập ý tưởng thiết kế ban đầu.

Theo yêu cầu chúng ta cần xây dựng dãy B có cùng kích thước với dãy A và các phần tử của B liên hệ với dãy A như sau: B[k] = số lần lặp của phần tử A[k] trong dãy A. Do vậy thiết kế lời giải sơ lược ban đầu của bài toán như sau:

```
Thiết lập dãy B rỗng
for i in range(len(A)):
    Tính số lần lặp của A[i] trong dãy A
    Bổ sung giá trị này vào dãy B
Trả về dãy B
```

**Bước 2.** Thiết lập dãy B rỗng.

Việc này được thực hiện đơn giản bằng lệnh B = [].

**Bước 3.** Tính số lần lặp của A[i] trong dãy A.

Công việc này có thể được thực hiện thông qua lời gọi hàm lap(x,A) tính số lần lặp của một giá trị bất kì trong dãy A. Hàm lap(x,A) có thể được viết như sau:

```
def lap(x,A):
    S = 0
    for a in A:
        if a == x:
            S = S + 1
    return S
```

**Bước 4.** Bổ sung số lần lặp vào dãy B.

Tổng hợp kết quả của các bước trên, chúng ta thu được đoạn chương trình hoàn chỉnh đã nêu ở bước 1 như sau:

```
    B = []
    for i in range(len(A)):
        B.append(lap(A[i],A))
    return B
```

**Bước 5.** Trả về dãy B cần tìm của bài toán.

Để hoàn thiện toàn bộ chương trình chúng ta sẽ thiết lập hàm tinh_lap(A) mô tả đoạn chương trình đã nêu trong bước 4. Hàm tinh_lap(A) trả về dãy B cần tìm:

```
def tinh_lap(A):
    B = []
    for a in A:
        B.append(lap(a,A))
    return B
```

### c) Chương trình hoàn chỉnh

Tới đây việc thiết kế theo phương pháp làm mịn dần kết thúc. Chương trình hoàn chỉnh được viết như sau:

```
def lap(x,A):
    S = 0
    for a in A:
        if a == x:
            S = S + 1
    return S

def tinh_lap(A):
    B = []
    for a in A:
        B.append(lap(a,A))
    return B
# Chương trình chính
A = [2, 1, 1, 3, 5, 10, 2, 5, 2]
B = tinh_lap(A)
print(B)
```

## LUYỆN TẬP

1. Thiết kế thuật toán cho Nhiệm vụ 1 với ý tưởng khác như sau: Dãy A là một hoán vị của dãy các số từ 1 đến n khi và chỉ khi dãy A có độ dài n và mọi số i từ 1 đến n đều nằm trong A.
2. Trong Nhiệm vụ 2, nếu dãy A đã được sắp xếp theo thứ tự tăng dần thì có thể cải tiến thuật toán tốt hơn được không?

## VẬN DỤNG

1. Cho dãy số A = A[0], A[1], ..., A[n – 1]. Thiết kế và viết chương trình kiểm tra trong dãy A có hai phần tử nào trùng nhau hay không. Cần đưa ra câu trả lời là "có" hay "không". Yêu cầu đưa ra quy trình thiết kế theo phương pháp làm mịn dần.
2. Xâu kí tự được gọi là đối xứng nếu thay đổi thứ tự ngược lại các kí tự của xâu thì vẫn nhận được dãy ban đầu. Ví dụ xâu "abcdcba" là đối xứng, còn xâu "1011" không là đối xứng.

   Thiết kế và viết chương trình kiểm tra một xâu kí tự cho trước có là đối xứng hay không. Yêu cầu đưa ra quy trình thiết kế theo phương pháp làm mịn dần.

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

# BÀI 28 THIẾT KẾ CHƯƠNG TRÌNH THEO MÔ ĐUN

SAU BÀI HỌC NÀY EM SẼ:

- Biết và vận dụng được cách thiết kế chương trình theo mô đun cho một số bài toán cụ thể.
- Nhận biết được lợi ích của phương pháp thiết kế chương trình theo mô đun.

Em được giao việc quản lí cho cửa hàng bán thực phẩm của gia đình. Hằng ngày, em phải nhập danh sách các mặt hàng và doanh số bán hàng. Cuối ngày, em cần báo cáo ba mặt hàng có doanh số cao nhất và ba mặt hàng có doanh số thấp nhất trong ngày.

Các mặt hàng mới nhập sẽ được lưu trong tệp văn bản bao gồm nhiều dòng, mỗi dòng là tên mặt hàng và doanh số được ngăn cách bằng dấu phẩy. Tên tệp đầu vào là Data.inp.

Báo cáo cuối ngày là tệp văn bản Data.out gồm 6 dòng, ba dòng đầu tiên là tên ba mặt hàng có doanh số cao nhất, ba dòng cuối là tên ba mặt hàng có doanh số thấp nhất, cần có cả tên hàng và doanh số.

**Data.inp**

Cà rốt, 1350
Khoai tây, 4400
Hành tươi, 1367.5
Bắp cải, 3400
Cà chua, 5609
Khoai lang, 2100
Gạo ST25, 19221
Gạo thường, 23124
Cam, 9800
Chuối, 7823

**Data.out**

Gạo thường 23124
Gạo ST25 19221
Cam 9800
Khoai lang 2100
Hành tươi 1367.5
Cà rốt 1350

Em sẽ thiết kế chương trình như thế nào? Trao đổi với bạn về cách thiết kế chương trình sao cho hợp lí nhất.

## 1. THIẾT KẾ CHƯƠNG TRÌNH THEO MÔ ĐUN

**Hoạt động 1** **Thiết kế chương trình theo mô đun**

Thực hiện các bước thiết kế giải bài toán trên theo phương pháp làm mịn dần, trao đổi và thảo luận để biết được cách thiết kế chương trình theo mô đun.

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

## Bước 1. Thiết kế chung

Từ yêu cầu của bài toán, ta thấy có thể chia bài toán đã cho thành ba công việc chính, các công việc này tương đối độc lập với nhau.

1) Công việc nhập dữ liệu: Dữ liệu được nhập vào tệp Data.inp và được đọc để đưa vào chương trình.

2) Công việc xử lí dữ liệu: Các công việc chuẩn bị tính toán dữ liệu theo yêu cầu của bài toán.

3) Báo cáo, đưa dữ liệu ra theo yêu cầu.

## Bước 2. Thiết lập công việc nhập dữ liệu

Thiết lập hàm NhapDL(fin) có tính năng đọc dữ liệu từ tệp Data.inp và đưa vào hai mảng P (mặt hàng) và S (doanh số). Hàm được mô tả tổng quát đọc dữ liệu từ tệp fin.

```
1  def NhapDL(fin):
2      f = open(fin,encoding="UTF-8")
3      P = []
4      S = []
5      for line in f:
6          A = line.split(",")
7          P.append(A[0])
8          S.append(float(A[1]))
9      f.close()
10     return P,S
```

## Bước 3. Thiết lập công việc xử lí dữ liệu

Yêu cầu chính của báo cáo là lấy được thông tin của các mặt hàng có doanh số cao nhất và thấp nhất. Do vậy công việc xử lí dữ liệu chính là cần sắp xếp lại các mảng P, S theo thứ tự tăng dần của S (doanh số). Hàm Sapxep(A,B) được thiết kế tổng quát sẽ sắp xếp lại hai mảng A, B nhưng theo thứ tự tăng dần của A.

```
1  def Sapxep(A,B):
2      n = len(A)
3      for i in range(1,n):
4          j = i
5          while j > 0 and A[j] < A[j-1]:
6              A[j],A[j-1] = A[j-1],A[j]
7              B[j],B[j-1]= B[j-1],B[j]
8              j = j-1
```

## Bước 4. Thiết lập báo cáo, đưa dữ liệu ra

Cuối cùng là chức năng đưa dữ liệu ra tệp Data.out sẽ được mô tả trong hàm GhiDL(P,S,fout). Hàm này lấy dữ liệu từ các mảng P, S và đưa dữ liệu ra tệp fout.

```
1  def GhiDL(P,S,fout):
2      f = open(fout,"w",encoding="UTF-8")
3      n = len(P)
4      for i in range(n-1,n-4,-1):
```

```
5           print(P[i],S[i],file = f)
6       for i in range(2,-1,-1):
7           print(P[i],S[i],file = f)
8       f.close()
```

Sau các bước trên, chúng ta đã thực hiện xong việc thiết kế các công việc chính được đặt ra ban đầu. Mỗi công việc đó được viết thành một hàm riêng biệt, độc lập với nhau. Các hàm này được gọi là các mô đun con của chương trình.

Chương trình chính sử dụng các chương trình con trên được mô tả đơn giản như sau:

```
1 fin = "Data.inp"
2 fout = "Data.out"
3 P,S = NhapDL(fin)
4 Sapxep(S,P)
5 GhiDL(P,S,fout)
```

Phương pháp thiết kế chương trình như trên được gọi là thiết kế theo mô đun. Mỗi mô đun thường là các chức năng độc lập, riêng biệt theo yêu cầu của chương trình. Tuỳ thuộc vào từng bài toán và quan điểm thiết kế có thể tạo các mô đun này theo nhiều cách khác nhau.

Phương pháp thiết kế chương trình theo mô đun sẽ tách bài toán lớn thành các bài toán nhỏ hơn, hay thành các mô đun, tương đối độc lập với nhau, sau đó tiến hành thiết kế thuật toán và chương trình cho từng mô đun con. Mỗi mô đun có thể là một số hàm hoặc thủ tục độc lập. Chương trình chính là một bản ghép nối các hàm và thủ tục con.



1. Chương trình trên được thiết kế có bao nhiêu mô đun?
2. Các mô đun của chương trình trên có quan hệ với nhau như thế nào?

## 2. LỢI ÍCH CỦA PHƯƠNG PHÁP THIẾT KẾ THEO MÔ ĐUN

**Hoạt động 2** **Tìm hiểu lợi ích của phương pháp thiết kế theo mô đun**

Với chương trình đã có trong Hoạt động 1, em sẽ làm gì nếu có các yêu cầu bổ sung như sau:

1. Yêu cầu thay đổi thông tin trong báo cáo: Ghi hai mặt hàng có doanh số cao nhất và bốn mặt hàng có doanh số thấp nhất.
2. Cập nhật, bổ sung các mặt hàng mới và doanh số trong ngày.
3. Yêu cầu làm thêm một báo cáo trong đó ghi doanh số bán trung bình trong ngày và danh sách các mặt hàng có doanh số lớn hơn doanh số trung bình này, kết quả đưa ra tệp Data2.out.

Em có nhận xét gì về việc thực hiện các công việc bổ sung này?

## Công việc bổ sung 1

Đây là công việc cần nâng cấp hàm GhiDL() và độc lập với các mô đun khác. Việc nâng cấp này rất đơn giản và được mô tả trong chương trình sau. Thay đổi chỉ ở hai lệnh tại dòng 4 và 6.

```
1  def GhiDL(P,S,fout):
2      f = open(fout,"w",encoding="UTF-8")
3      n = len(P)
4      for i in range(n-1,n-3,-1):
5          print(P[i],S[i],file = f)
6      for i in range(3,-1,-1):
7        print(P[i],S[i],file = f)
8      f.close()
```

So với mô đun gốc, bản nâng cấp này chỉ cần sửa hai dòng lệnh 4 và 6.

## Công việc bổ sung 2

Công việc này rất đơn giản là mở tệp Data.inp và bổ sung thêm thông tin các mặt hàng mới và doanh số, không cần phải sửa chương trình.

## Công việc bổ sung 3

Công việc này mới và độc lập hoàn toàn với các công việc khác của bài toán, nên có thể tách thành một hàm (mô đun) độc lập, có thể giao cho một nhóm khác thực hiện. Hàm mới sẽ đặt tên là BC2() và có nội dung đơn giản như sau:

```
1  def BC2(P,S,fout):
2      f = open(fout,"w",encoding="UTF-8")
3      n = len(P)
4      average = sum(S)/n
5      print("Doanh số trung bình:",average,file = f)
6      for i in range(n):
7          if S[i] > average:
8              print(P[i],S[i],file = f)
9      f.close()
```

Trong chương trình chính cần bổ sung lệnh sau đây để thực hiện báo cáo mới này.

```
BC2(P,S,"Data2.out")
```

Thiết kế thuật toán và chương trình theo mô đun có các ưu điểm sau:

- Chương trình ngắn gọn, sáng sủa, dễ hiểu.
- Các mô đun được thiết lập một lần và sử dụng nhiều lần.
- Dễ dàng nâng cấp, thay đổi, chỉnh sửa mà không mất công sửa lại toàn bộ chương trình.
- Dễ dàng bổ sung các mô đun mới.
- Có thể chia sẻ trong môi trường làm việc nhóm, ví dụ phân công mỗi người một công việc độc lập.

**1.** Phân loại các công việc bổ sung trên vào ba loại sau:

– Công việc mới hoàn toàn.

– Công việc nâng cấp một mô đun cũ.

– Công việc không liên quan đến thuật toán và lập trình.

**2.** Công việc sau đây, nếu có, sẽ thuộc nhóm công việc nào?

Nhập một giá trị số nào đó, ví dụ K, cần tìm trong danh sách các mặt hàng có doanh số xấp xỉ K (hơn kém nhau không quá hằng số C = 1).

## LUYỆN TẬP

**1.** Nếu công việc bổ sung 3 có thêm yêu cầu in ra số lượng mặt hàng đã bán trong ngày thì cần thêm hay sửa lệnh nào của chương trình BC2().

**2.** Viết thêm một chương trình cho công việc bổ sung 4 như sau: Cần in ra danh sách $\frac{1}{3}$ số mặt hàng có doanh số thấp nhất trong ngày.

## VẬN DỤNG

**1.** Thiết lập chương trình cho công việc thường làm vào cuối giờ bán hàng: Cho trước số K (một doanh số giả định), cần tìm ra mặt hàng có doanh số nhỏ hơn K nhưng gần với K nhất. Bài toán này có thể sử dụng thuật toán tìm kiếm nào để giải?

**2.** Thiết kế thuật toán và chương trình theo mô đun cho bài toán sau.

Một công ti du lịch có n địa điểm tham quan được đánh số theo thứ tự 0, 1, 2,..., n – 1. Công ti này luôn tổ chức các tour du lịch đi lần lượt từ vị trí 0, 1, 2,... và kết thúc tại vị trí cuối cùng n – 1. Để thuận tiện cho việc quảng bá du lịch công ti đã lấy ý kiến khách hàng đánh giá bằng điểm số cho từng địa điểm du lịch trên, các đánh giá có thể là các số dương, 0 hoặc số âm bất kì. Số lớn hơn 0 biểu thị đánh giá tốt, số nhỏ hơn 0 biểu thị đánh giá xấu về địa điểm đó. Mỗi khách hàng sẽ gửi lên công ti du lịch bảng đánh giá của mình, được biểu thị bằng một dãy n số, ví dụ như sau:

1, –3, 4, 10, 0, –5, –8, 2, –1, 7, 2

Công ti du lịch hứa sẽ tổ chức một tour riêng cho mỗi khách hàng, bảo đảm sự hài lòng cao nhất của khách hàng. Tour du lịch riêng của khách hàng sẽ là một dãy các vị trí liên tục các địa điểm, ví dụ từ vị trí i đến j, tức là xuất phát từ i, khách hàng sẽ lần lượt đi qua các vị trí i, i + 1, i + 2,... và kết thúc tại j. Công ti du lịch bảo đảm rằng tổng các đánh giá của khách hàng trên tour riêng của mình là lớn nhất.

Em hãy giúp công ti du lịch thiết lập tour du lịch tối ưu cho khách hàng nếu biết trước các đánh giá của khách hàng đó.

# BÀI 29 THỰC HÀNH THIẾT KẾ CHƯƠNG TRÌNH THEO MÔ ĐUN

SAU BÀI HỌC NÀY EM SẼ:

- Thực hành thiết kế một số chương trình hoàn chỉnh theo mô đun.

Trong Bài 28, em đã biết tầm quan trọng và ý nghĩa của việc thiết kế chương trình theo mô đun. Mỗi mô đun chính là một chương trình con được xác định khi phân tích thiết kế bài toán lớn. Mỗi mô đun được viết một cách độc lập, có thể sử dụng lại như các thư viện và có thể chia sẻ trong nhóm làm việc. Trong bài thực hành này em sẽ được tự mình thiết kế chương trình cho một bài toán hoàn chỉnh theo mô đun.

## Nhiệm vụ. Tính điểm tổng hợp của vận động viên

Trong một cuộc thi đấu thể thao khu vực dạng Sea Games, các cầu thủ tham gia thi đấu ở các bộ môn khác nhau và do sự phức tạp của truyền tin nên ban tổ chức (BTC) đã quyết định nhập thông tin kết quả thi đấu của các vận động viên vào các tệp văn bản, sau đó các tệp này được chuyển về trung tâm để xử lí. Mỗi bộ môn thể thao sẽ có số lượng ban giám khảo khác nhau. Mỗi tệp dữ liệu lưu thông tin sẽ có khuôn dạng gồm nhiều dòng, mỗi dòng là thông tin của một vận động viên thi đấu tại một bộ môn nào đó. Mỗi dòng của tệp thông tin sẽ bao gồm:

– Đầu tiên là mã số của vận động viên.

– Tiếp theo là điểm cho của các giám khảo, giữa các điểm số đều có dấu cách.

– Điểm số được cho có thể là số nguyên hoặc số thập phân, thang điểm 10.

Theo quy định của BTC, quy định tính điểm chung cho mỗi vận động viên như sau: Điểm tổng hợp của mỗi vận động viên là trung bình cộng điểm của ban giám khảo. Tuy nhiên trong mỗi ban giám khảo luôn có hai giám khảo đặc biệt, hai giám khảo này có hệ số tổng hợp là 2, trong khi các giám khảo khác có hệ số 1. Theo quy định của BTC thì các giám khảo đặc biệt sẽ ở vị trí đầu tiên và cuối cùng của danh sách. Ở tất cả các bộ môn thể thao số lượng thành viên ban giám khảo phải lớn hơn 2.

Nhiệm vụ của em là tính điểm tổng hợp của các vận động viên từ tệp văn bản SeaGames.inp. Kết quả ghi ra tệp ketqua.out có dạng như sau:

– Tệp sẽ có nhiều dòng, số dòng bằng đúng số dòng của tệp đầu vào.

– Trên mỗi dòng ghi mã của vận động viên và điểm tổng hợp tương ứng. Yêu cầu ghi điểm tổng hợp với hai chữ số sau dấu phẩy.

| SeaGames.inp |
|---|
| 101 7.5 8.0 9.0 9.5 7.1 6.8 |
| 200 8.5 9.1 9.5 8.6 9.9 |
| 003 6.6 7.0 7.5 6.8 5.9 8.1 |
| 045 8.5 7.9 9.3 9.0 8.9 |
| 901 9.2 9.7 8.6 |

| Ketqua.out |
|---|
| 101 7.77 |
| 200 9.14 |
| 003 7.08 |
| 045 8.71 |
| 901 9.06 |

Hướng dẫn:

## a) Phân tích tổng quát

Từ yêu cầu của bài toán, dễ thấy chương trình có thể chia thành ba công việc, hay mô đun con, tương đối độc lập như sau:

- Mô đun 1: Đọc dữ liệu từ tệp.
- Mô đun 2: Xử lí dữ liệu đã đọc ở mô đun 1.
- Mô đun 3: Ghi dữ liệu đã xử lí ra tệp theo yêu cầu.

Các mô đun sẽ được thiết kế một cách độc lập. Chương trình chính sẽ tổng hợp và kết nối các mô đun.

## b) Thiết kế mô đun nhập dữ liệu

Dữ liệu sẽ được nhập từ tệp SeaGames.inp và được đưa vào hai mảng:

- Mảng thứ nhất là danh sách mã số của các vận động viên.
- Mảng thứ hai có ý nghĩa như sau: Tương ứng với mỗi vận động viên từ mảng thứ nhất, mỗi phần tử của mảng thứ hai là dãy các điểm số mà ban giám khảo chấm cho vận động viên này.

Hai mảng này sẽ được đặt tên lần lượt là **DS** và **Diem**.

Mô đun này được thiết lập bằng hàm **nhapDL()** với tham số đầu vào là tên tệp dữ liệu. Hàm sẽ trả lại hai mảng DS và Diem. Chú ý: Khi đọc dữ liệu từ tệp, với mỗi dòng, phần tử đầu tiên chính là mã số của vận động viên, sẽ được đưa vào mảng DS, dãy các giá trị còn lại chính là dãy điểm và được đưa vào mảng Diem.

Hàm **nhapDL(finp)** được mô tả như sau:

```
1  def nhapDL(finp):
2      f = open(finp)
3       A = []
4       B = []
5       for line in f:
6           S = line.split()
7           A.append(S[0])
8           temp = S[1:len(S)]
9          temp = [float(x) for x in temp]
10          B.append(temp)
11      f.close()
12      return A,B
```

## c) Thiết kế mô đun xử lí dữ liệu chính

Mục đích của mô đun này là tính điểm tổng hợp cho mỗi vận động viên.

Mỗi vận động viên sẽ tương ứng với một dãy các điểm số được lấy từ mảng Diem. Từ dãy này sẽ tính được điểm tổng hợp theo cách tính đặc biệt của BTC. Mô đun này sẽ cần hai hàm.

- Hàm **diem_gk()** sẽ tính điểm tổng hợp các giám khảo từ dãy các điểm. Chú ý theo yêu cầu thì các giám khảo đầu tiên và cuối cùng có hệ số 2 nên cần tính điểm các giám khảo này hai lần. Điều này được mô tả tại dòng lệnh 2. Hàm này được thiết kế chi tiết như sau:

```
1 def diem_gk(d):
2     diem = sum(d) + d[0] + d[len(d)-1]
3     diem = diem/(len(d)+2)
4     return round(diem,2)
```

- Hàm **Xuly()** sẽ tính toán điểm tổng hợp cho tất cả các vận động viên với đầu vào là mảng **Diem**. Hàm này sẽ sử dụng hàm diem_gk() đã thiết lập ở phần trên. Kết quả tính toán là một dãy điểm tổng hợp và là giá trị trả lại của hàm này. Kết quả của hàm Xuly() là dãy **kq**.

```
1 def Xuly(B):
2     kq = []
3     for i in range(len(B)):
4         diem = diem_gk(B[i])
5         kq.append(diem)
6     return kq
```

## d) Thiết kế mô đun đưa kết quả ra

Cuối cùng là mô đun đưa kết quả ra tệp đầu ra theo yêu cầu của bài toán.

Hàm ghiDL() sẽ có ba tham số đầu vào là tên tệp dữ liệu ra, hai mảng DS và kq. Khai báo của hàm này là ghiDL(fout, A, B), trong đó **fout** là tên tệp dữ liệu đầu ra, A là mảng danh sách các mã số của vận động viên, B là mảng ghi kết quả điểm tổng hợp của các vận động viên. Kết quả đưa ra màn hình.

```
1 def ghiDL(fout,A,B):
2     f = open(fout,"w")
3     for i in range(len(A)):
4         print(A[i],B[i],file = f)
5     f.close()
```

## e) Tổng hợp chương trình chính

```
1 finp = "SeaGames.inp"
2 fout = "ketqua.out"
3 DS,Diem = nhapDL(finp)
4 kq = Xuly(Diem)
5 ghiDL(fout,DS,kq)
```

Sau đây là toàn bộ chương trình:

**SeaGames.py**

```
1  def nhapDL(finp):
2      f = open(finp)
3      A = []
4      B = []
5      for line in f:
6          S = line.split()
7          A.append(S[0])
8          temp = S[1:len(S)]
9          temp = [float(x) for x in temp]
10          B.append(temp)
11     f.close()
12     return A,B
13
14 def diem_gk(d):
15     diem = sum(d) + d[0] + d[len(d)-1]
16     diem = diem/(len(d)+2)
17     return round(diem,2)
18
19 def Xuly(B):
20     kq = []
21     for i in range(len(B)):
22         diem = diem_gk(B[i])
23         kq.append(diem)
24     return kq
25
26 def ghiDL(fout,A,B):
27     f = open(fout,"w")
28     for i in range(len(A)):
29         print(A[i],B[i],file = f)
30     f.close()
31
32 # Chương trình chính
33 finp = "SeaGames.inp"
34 fout = "ketqua.out"
35 DS,Diem = nhapDL(finp)
36 kq = Xuly(Diem)
37 ghiDL(fout,DS,kq)
```

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

## LUYỆN TẬP

1. Hãy chỉnh sửa lại chương trình trên nếu bổ sung thêm điều kiện sau vào nhiệm vụ: Trong tệp kết quả đầu ra, thứ tự các vận động viên được ghi theo thứ tự giảm dần của điểm đánh giá.

2. Trong nhiệm vụ trên, nếu công thức tính điểm tổng hợp của Sea Games thay đổi thì chúng ta có phải sửa lại toàn bộ chương trình hay không? Nếu cần thì chỉ phải sửa mô đun nào? Hàm nào?

## VẬN DỤNG

Thiết lập chương trình thiết kế theo mô đun cho các bài toán sau:

1. Cho trước số tự nhiên n, cần in ra trên màn hình dãy n số nguyên tố đầu tiên. Ví dụ nếu n = 5 thì dãy cần in ra sẽ là 2, 3, 5, 7, 11.

2. Trong một kì thi Tin học trẻ, mỗi học sinh sẽ phải làm 3 bài thi. Với mỗi bài, nếu học sinh làm sẽ được ban giám khảo chấm và cho điểm, nếu không làm thì sẽ không tính điểm. Sau khi thi, dữ liệu điểm thi của học sinh sẽ được lưu trong một tệp văn bản và gửi về ban tổ chức. Mẫu một tệp điểm thi có dạng sau:

**Diemthi.inp**

```
A12  12 -1  15
B123 9  14  -1
C11  10 12  18
A110 10 -1  -1
B01  12 10  4
```

Quy định ghi trong tệp trên như sau:

– Mỗi dòng sẽ bắt đầu bằng số báo danh của thí sinh, tiếp theo là ba giá trị điểm tương ứng với ba bài thi.

– Điểm thi sẽ là một số tự nhiên từ 0 đến 20.

– Nếu học sinh không làm thì bài đó ghi -1.

Em có nhiệm vụ tính toán tổng số điểm thi của các bạn học sinh và đưa dữ liệu ra tệp ketqua.out là danh sách ba bạn có tổng điểm cao nhất được sắp xếp giảm dần từ trên xuống dưới. Khuôn dạng dữ liệu đưa ra bao gồm: số báo danh, các điểm thi từng bài và cuối cùng là tổng điểm cả ba bài.

Ví dụ với dữ liệu trên thì kết quả như sau:

**ketqua.out**

```
C11  10 12 18 40
A12  12 -1 15 27
B01  12 10 4  26
```

BÀI 30

# THIẾT LẬP THƯ VIỆN CHO CHƯƠNG TRÌNH

SAU BÀI HỌC NÀY EM SẼ:

- Tạo được một thư viện nhỏ của người lập trình.
- Trình bày được cấu trúc danh sách liên kết.

Em đã học về cấu trúc mảng (một chiều hoặc hai chiều). Cấu trúc mảng là một danh sách các phần tử được đánh chỉ số và quan hệ với nhau thông qua hệ thống chỉ số này. Giả sử A[0], A[1], ..., A[n – 1] là mảng n phần tử, nếu thực hiện lệnh, ví dụ del A[1], xoá một phần tử của dãy trên, thì các phần tử còn lại sẽ tự động điều chỉnh lại chỉ số để đối tượng vẫn là mảng (nhưng có n – 1 phần tử).

Cấu trúc danh sách liên kết (hay danh sách móc nối, linked list) là đối tượng có cấu trúc gần giống với mảng nhưng có liên kết không chặt chẽ như mảng. Một ví dụ của cấu trúc danh sách liên kết là mô hình các trang web. Khi duyệt web, em không thể đánh chỉ số cho từng trang web đã duyệt, mà chỉ có thể di chuyển đến các trang trước và trang sau.

Em hãy tìm thêm các ví dụ thực tế của mô hình danh sách liên kết.

## 1. THIẾT LẬP THƯ VIỆN CHO CHƯƠNG TRÌNH

**Hoạt động 1** Tìm hiểu ý nghĩa của thư viện chương trình

Em hãy đọc, thảo luận và trả lời các câu hỏi sau:

1. Vì sao lại cần thư viện chương trình?
2. Ý nghĩa của các hàm trong thư viện chương trình là gì?

### a) Một số hàm của thư viện math

`math` là một thư viện các hàm chuẩn của Python liên quan đến các tính toán toán học. Ví dụ một số hàm thường dùng của thư viện math là hàm tính căn bậc hai `sqrt()`, làm tròn xuống `floor()` và làm tròn lên `ceil()`. Để đưa một thư viện vào bộ nhớ có thể dùng lệnh `import` hoặc `from` <thư viện> `import` <các hàm> như sau:

```
import math # đưa toàn bộ thư viện math vào bộ nhớ
```

hoặc:

```
from math import sqrt, floor, ceil # chỉ đưa vào bộ nhớ ba hàm
```

*Ví dụ:*

```
>>> from math import sqrt,floor,ceil
>>> sqrt(5)
2.23606797749979
>>> floor(8.7)
8
>>> ceil(7.1)
8
```

## b) Tự thiết lập thư viện

Cách thiết lập một thư viện rất đơn giản: Em đưa các hàm chuẩn vào một tệp chương trình và đặt tên của tệp này chính là tên thư viện muốn lưu trữ. Xét ví dụ sau, tệp chương trình **lib.py** đóng vai trò như một thư viện. Thư viện này có hai hàm như sau:

**lib.py**

```
1  def NhapDL():
2      S = input("Nhập dãy số nguyên cách nhau bởi dấu cách: ")
3      A = [int(x) for x in S.split()]
4      return A
5  def InsertionSort(A):
6      n = len(A)
7      for i in range(1,n):
8          value = A[i]
9          j = i-1
10         while j >= 0 and A[j] > value:
11             A[j+1] = A[j]
12             j = j -1
13         A[j+1] = value
```

Chương trình sử dụng thư viện có thể như sau (đặt tệp chương trình này cùng thư mục với tệp thư viện **lib.py**.

```
1 from lib import * # Đưa tất cả các hàm của thư viện lib vào bộ nhớ
2 A = NhapDL()
3 InsertionSort(A)
4 print(A)
```

Thư viện chương trình là tập hợp các hàm được đặt trong các mô đun độc lập để dùng chung cho nhiều chương trình khác nhau. Các thư viện này có thể được dùng nhiều lần và có thể cập nhật, nâng cấp bất cứ lúc nào. Trong Python, lệnh import có chức năng đưa thư viện vào bộ nhớ để sẵn sàng sử dụng.

Những câu nào sau đây là sai về ý nghĩa của việc sử dụng thư viện khi viết chương trình?

A. Chương trình sẽ ngắn hơn.

B. Các hàm thư viện được viết một lần và sử dụng nhiều lần.

C. Chương trình sáng sủa, dễ hiểu hơn.

D. Chương trình sẽ chạy nhanh hơn.

# 2. CẤU TRÚC DANH SÁCH LIÊN KẾT

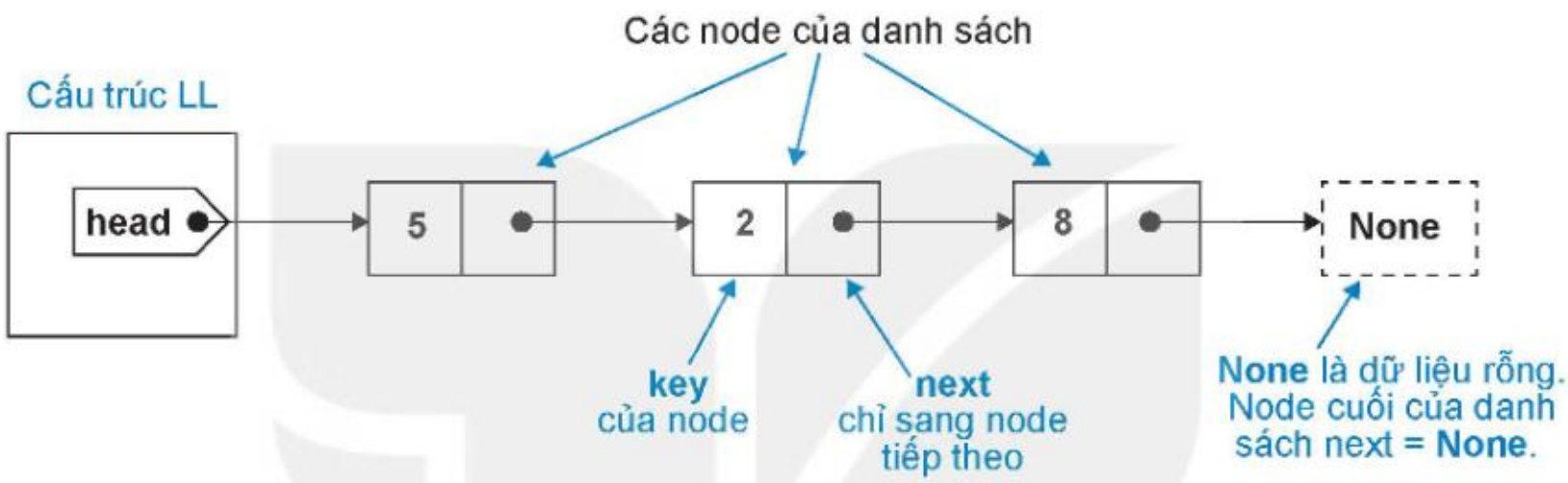
**Hoạt động 2** **Tìm hiểu cấu trúc danh sách liên kết**

Đọc, trao đổi và thảo luận để biết cấu trúc dữ liệu của danh sách liên kết và các thao tác dữ liệu cơ bản trên danh sách liên kết.

Mỗi danh sách liên kết sẽ bao gồm hai cấu trúc dữ liệu:

– Cấu trúc **Node** mô tả các phần tử độc lập của danh sách. Tối thiểu mỗi node cần có thông tin dữ liệu **key** (khoá) và thuộc tính **next** dùng để kết nối sang phần tử tiếp theo trong danh sách.

– Cấu trúc **LL** (linked list) sẽ có thông tin **head** (đầu) sẽ luôn chỉ vào node đầu tiên của danh sách liên kết.




Một số lệnh, thao tác chính với kiểu dữ liệu danh sách liên kết:

– Khởi tạo một danh sách liên kết mới.

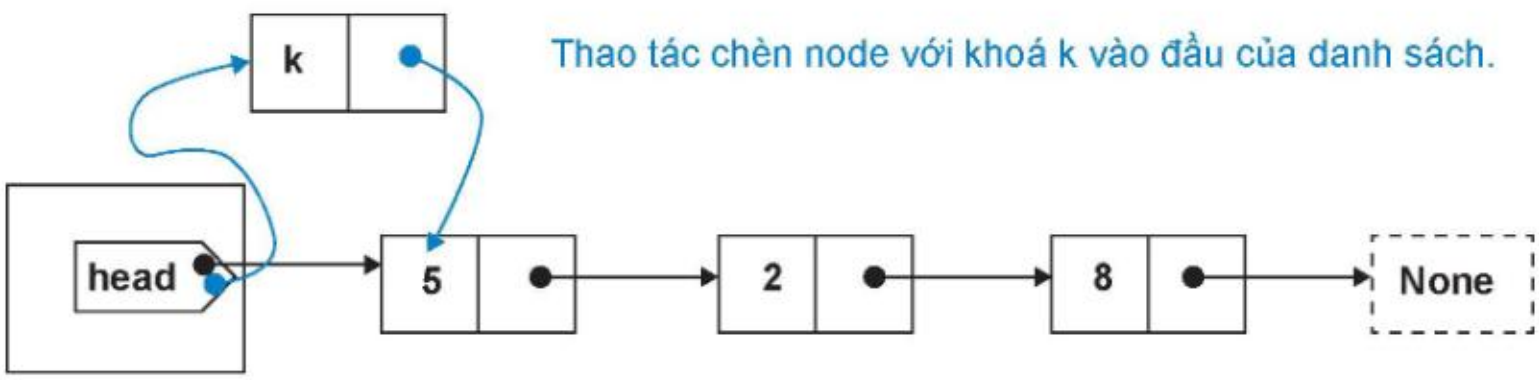
– Bổ sung một phần tử với khoá k cho trước vào danh sách.

– Tìm kiếm phần tử có khoá k trong danh sách cho trước.

– Xoá phần tử có khoá k trong danh sách.

Ta sẽ thiết lập một số hàm là các thao tác chuẩn trên dữ liệu danh sách liên kết.

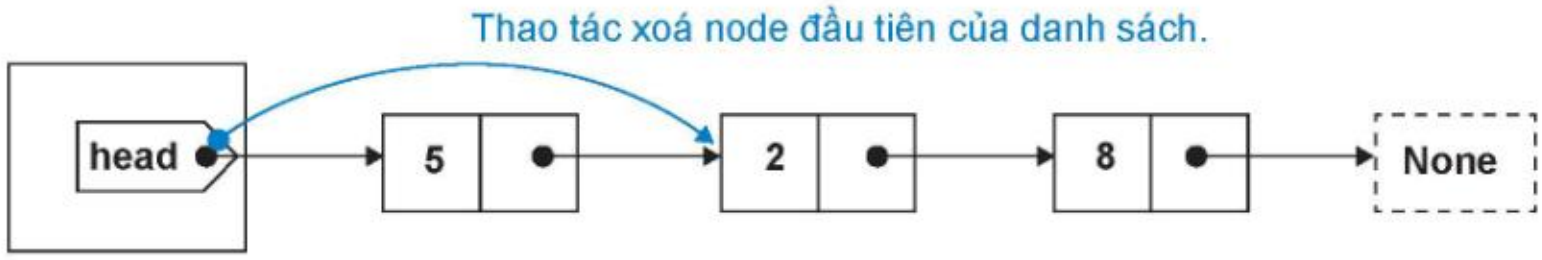
1. Hàm `insert(L,k)` sẽ bổ sung (chèn) node với khoá k vào đầu của danh sách L.




Chương trình như sau:

```
1  def insert(L,k):
2      node = Node(k)
3      node.next = L.head
4      L.head = node
```

2. Hàm **delete_first(L)** sẽ xoá node đầu tiên của danh sách (nếu danh sách không rỗng).




Chương trình như sau:

```
1  def delete_first(L):
2      if L.head != None:
3          L.head = L.head.next
```

3. Hàm tìm kiếm phần tử có khoá k trong danh sách L. Nếu tìm thấy sẽ trả về phần tử (node) tương ứng, nếu không tìm thấy trả về **None**.

Việc tìm kiếm bắt đầu từ node đầu tiên của danh sách (dòng 2). Lần lượt duyệt theo từng phần tử của danh sách cho đến khi nào tìm thấy phần tử có khoá k hoặc đi đến cuối danh sách thì dừng (các lệnh tại dòng 3, 4).

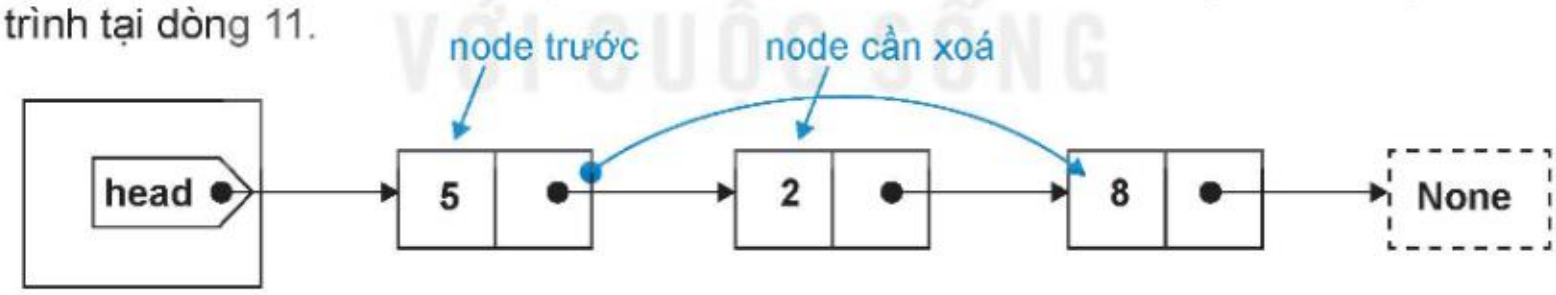
```
1  def search(L,k):
2      x = L.head
3      while x != None and x.key != k:
4          x = x.next
5      return x
```

4. Hàm xoá phần tử có khoá k trong danh sách L.

Cách thực hiện hàm này như sau:

– Nếu phần tử cần tìm là node đầu tiên của danh sách thì cách xoá giống hàm delete_first().

– Trong trường hợp tổng quát cần duyệt để tìm phần tử của danh sách có khoá k. Trong quá trình tìm luôn lưu trữ biến z là node trước của biến y cần xoá. Nếu tìm thấy thì lệnh xoá được mô tả trong sơ đồ sau, chính là lệnh x.next = y.next trong chương trình tại dòng 11.




Chương trình như sau:

```
1  def delete(L,k):
2      if L.head != None:
3          if L.head.key == k:
4              L.head = L.head.next
5          else:
6              z = y = L.head
7              while y != None and y.key != k:
8                  z = y
9                  y = y.next
10             if y != None:
11                 z.next = y.next
```

Nếu node đầu tiên có khoá k thì xoá node này.

Tìm kiếm node chứa khoá k. y là biến cần tìm, z luôn chỉ vào node trước của y. Nếu tìm thấy khoá k thì xoá node y.

5. Hàm **show(L)** có tính năng hiển thị toàn bộ thông tin của danh sách liên kết.

```
def show(L):
    x = L.head
    while x != None:
        print(x.key, end = " ")
        x = x.next
    print()
```

Toàn bộ thư viện chuẩn của cấu trúc danh sách liên kết được mô tả như sau:

**LinkedList.py**

```
class Node:
    def __init__(self,key):
        self.key = key
        self.next = None
class LL:
    def __init__(self):
        self.head = None
def insert(L,k):
    node = Node(k)
    node.next = L.head
    L.head = node
def delete_first(L):
    if L.head != None:
        L.head = L.head.next
def search(L,k):
    x = L.head
    while x != None and x.key != k:
        x = x.next
    return x
def delete(L,k):
    if L.head != None:
        if L.head.key == k:
            L.head = L.head.next
        else:
            z = y = L.head
            while y != None and y.key != k:
                z = y
                y = y.next
            if y != None:
                z.next = y.next
def show(L):
    x = L.head
    while x != None:
        print(x.key, end = " ")
        x = x.next
    print()
```

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

Một số ví dụ thiết lập cấu trúc dữ liệu Linked List:

a) Thiết lập một danh sách rỗng.

```
L = LL()
```

b) Thiết lập một danh sách bao gồm hai node có khoá là 5, 2.

```
L = LL()
insert(L,5)
insert(L,2)
```

c) Thiết lập một danh sách bao gồm các phần tử lấy từ dãy A cho trước.

```
A = [5,2,8,10,0,3]
L = LL()
for k in A:
    insert(L,k)
```

Danh sách liên kết là cấu trúc dữ liệu bao gồm:

- Cấu trúc node mô tả các phần tử của danh sách. Mỗi node sẽ có dữ liệu khoá (key) là thông tin chính và thông tin next để kết nối sang phần tử tiếp theo của danh sách.
- Cấu trúc head là đầu của mỗi danh sách liên kết. Head luôn chỉ vào node đầu tiên của danh sách.
- Node cuối cùng của danh sách sẽ có thông tin next = None (dữ liệu rỗng).

Có thể thiết lập các hàm tìm kiếm, bổ sung hoặc xoá thông tin trên danh sách liên kết.

**1.** Đoạn chương trình sau thực hiện công việc gì?

```
from LinkedList import *
L = LL()
insert(L,10)
insert(L,20)
show(L)
```

**2.** Viết đoạn chương trình ngắn sử dụng thư viện LinkedList để thiết lập một danh sách liên kết L và bổ sung các tên "Bình", "Hoa", "Hà" vào danh sách này.

## LUYỆN TẬP

**1.** Viết một thư viện bao gồm các hàm nhập dữ liệu là một dãy số và các hàm thư viện bao gồm sắp xếp chèn, sắp xếp chọn và sắp xếp nổi bọt.

**2.** Cho trước danh sách liên kết L với cấu trúc như đã mô tả trong bài học, muốn lấy ra khoá của node đầu tiên của danh sách thì dùng lệnh nào?

## VẬN DỤNG

**1.** Cho trước một danh sách liên kết L. Viết một hàm đếm số lượng phần tử của danh sách liên kết này.

**2.** Viết hàm delete_last(L) có chức năng xoá phần tử cuối cùng của danh sách liên kết L.

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

# BÀI 31 THỰC HÀNH THIẾT LẬP THƯ VIỆN CHƯƠNG TRÌNH

SAU BÀI HỌC NÀY EM SẼ:

- Viết được chương trình vận dụng những kiến thức tích hợp liên môn để giải quyết vấn đề.

Trong Bài 30, em đã tìm hiểu ý nghĩa và cách thiết lập thư viện chương trình. Em có thể thấy xung quanh em, đặc biệt là trong các lĩnh vực khoa học tự nhiên như Toán học, Vật lí và Hoá học, thường xuất hiện nhiều vấn đề hoặc nhiệm vụ tính toán mang tính tổng quát. Em hãy xây dựng thuật toán cho các vấn đề như vậy thành các thư viện để có thể sử dụng lại nhiều lần cũng như làm toàn bộ chương trình có cấu trúc trong sáng, rõ ràng, dễ phát triển, dễ bảo trì hơn.

## Nhiệm vụ 1

Viết thư viện hinh_tron gồm hai hàm để tính chu vi và diện tích của hình tròn với tham số của hàm số là bán kính. Trong thư viện này, hãy sử dụng hằng số math.pi là giá trị của số Pi được định nghĩa ở thư viện math. Sau đó, viết một tệp mã nguồn main.py để yêu cầu người dùng nhập bán kính đường tròn là một số dương rồi sử dụng thư viện trên để tính diện tích và chu vi hình tròn.

**Hướng dẫn:**

Nhiệm vụ này được thực hiện bằng cách định nghĩa hai hàm tinhChuVi và tinhDienTich được viết trong tệp circle.py và đoạn mã nguồn sử dụng thư viện hinh_tron trong tệp main.py.

**hinh_tron.py**

```
1 import math #Khai báo sử dụng thư viện math từ thư viện chuẩn của python
2 def tinhChuVi(r):
3     return 2*math.pi*r
4 def tinhDienTich(r):
5     return math.pi*r*r
```

**main.py**

```
1 import hinh_tron #Khai báo sử dụng thư viện hinh_tron vừa định nghĩa
2 r = float(input("Nhập bán kính hình tròn: "))
3 p = hinh_tron.tinhChuVi(r)
4 print("Chu vi hình tròn là", p)
5 s = hinh_tron.tinhDienTich(r)
6 print("Diện tích hình tròn là", s)
```

## Nhiệm vụ 2

Tạo thư viện cong_thuc_ly gồm hai hàm machSongSong(dsDienTro) và machNoiTiep(dsDienTro) để tính điện trở tương đương của mạch song song và nối tiếp gồm các điện trở được cho giá trị tính theo Ω trong mảng dsDienTro. Hãy viết chương trình trong tệp main.py sử dụng hai hàm vừa định nghĩa để tính điện trở tương đương của mạch gồm các điện trở với giá trị 3, 6 và 8 Ω.

### Hướng dẫn:

Áp dụng kiến thức vật lí: điện trở tương đương của mạch gồm các điện trở mắc nối tiếp bằng tổng các điện trở và mạch gồm các điện trở mắc song song bằng nghịch đảo của tổng các nghịch đảo giá trị điện trở thành phần. Ngoài ra kiểm tra nếu có một giá trị điện trở không hợp lệ (nhỏ hơn hoặc bằng 0) thì in thông báo và hàm trả về giá trị -1 với ý nghĩa dữ liệu không hợp lệ.

**cong_thuc_ly.py**

```
1  def machSongSong(dsDienTro):
2      for r in dsDienTro:
3          if r <= 0:
4              print("Dữ liệu không hợp lệ, tồn tại một điện trở <= 0")
5              return -1
6      return sum(dsDienTro)
7
8  def machNoiTiep(dsDienTro):
9      tongNghichDao = 0
10     for r in dsDienTro:
11         if r <= 0:
12             print("Dữ liệu không hợp lệ, tồn tại một điện trở <= 0")
13             return -1
14         tongNghichDao = tongNghichDao + 1/r
15     return round(1/tongNghichDao,2)
```

**main.py**

```
1  from cong_thuc_ly import *
2
3  dsDienTro = [3, 6, 8]
4  print("Điện trở tương đương của mạch mắc nối tiếp:", machSongSong(dsDienTro))
5  print("Điện trở tương đương của mạch mắc song song:", machNoiTiep(dsDienTro))
```

## Nhiệm vụ 3

Em hãy định nghĩa hàm tinhNtkTB(dsNtk, dstyLe) trong tệp cong_thuc_hoa.py để tính nguyên tử khối trung bình của một nguyên tố hoá học, trong đó tham số dsNtk là mảng giá trị các nguyên tử khối của các đồng vị và dstyLe là tỉ lệ phần trăm số nguyên tử của các đồng vị của nguyên tố đó. Sau đó, em hãy viết chương trình trong tệp main.py để sử dụng hàm tinhNtkTB tính nguyên tử khối trung bình của carbon biết carbon có hai đồng vị bền là $^{12}C$ chiếm 98,89% và $^{13}C$ chiếm 1,11%.

### Hướng dẫn:

Áp dụng kiến thức hoá học: nguyên tử khối trung bình của một nguyên tố gồm n đồng vị được tính theo công thức $ntkTB = \frac{\sum_{i=1}^{n} ntk_i * tyLe_i}{100}$ trong đó $ntk_i > 0$

và $tyLe_i \in (0,100)$ lần lượt là nguyên tử khối và tỉ lệ phần trăm của các đồng vị thứ i (i = 1,2,...,n) của nguyên tố đó.

**cong_thuc_hoa.py**

```
1 def tinhNtkTB(dsNtk,dstyLe):
2   if len(dstyLe) > 0 and len(dsNtk) == len(dstyLe):
3     tong = 0
4     for i in range(len(dsNtk)):
5       tong = tong + dsNtk[i] * dstyLe[i]
6     return tong/100
7   else:
8     return 0
```

**main.py**

```
1 from cong_thuc_hoa import tinhNtkTB
2
3 dsNtk = [12,13]
4 dsTyLe = [98.89, 1.11]
5 ntkTB = tinhNtkTB(dsNtk, dsTyLe)
6 print("Nguyên tử khối trung bình của Carbon là", ntkTB)
```

## LUYỆN TẬP

1. Đặt tất cả các tệp thư viện đã định nghĩa ở nhiệm vụ 1, 2 và 3 vào thư mục myLibs rồi viết mã nguồn ở tệp main.py (đặt cùng đường dẫn với thư mục myLibs) để sử dụng các hàm trong các thư viện đó.
2. Sửa lại thư viện hinh_tron ở nhiệm vụ 1 bằng cách không sử dụng thư viện chuẩn math mà hãy định nghĩa thư viện my_math trong đó có định nghĩa hằng số Pi.

## VẬN DỤNG

1. Tạo thư viện phuong_trinh gồm hàm phuongTrinhBac2(a, b, c) với a, b, c là các hệ số của phương trình $ax^2 + bx + c = 0$. Tuỳ vào các giá trị của các tham số, hàm sẽ in ra thông báo nghiệm của phương trình.
2. Viết chương trình quản lí các bài hát trong một đĩa CD hay một play list, sử dụng cấu trúc LinkedList (đã được định nghĩa ở bài trước). Chương trình gồm hai tệp:

   • Tệp quan_ly_cd.py gồm ba hàm:

   – Hàm nhapDL(): Yêu cầu người dùng nhập số lượng bài hát, rồi sau đó nhập lần lượt tên các bài hát và bổ sung vào đĩa CD (hay play list), trả lại biến kiểu LinkedList chứa các bài hát.

   – Hàm timBai(): Tham số gồm đối tượng LinkedList và tên bài hát <ten_bai>. Nếu có bài hát cần tìm, hàm in ra vị trí đầu tiên xuất hiện bài hát, nếu không in ra thông báo "Không tìm thấy bài hát <ten_bai>".

   – Hàm inTT(): tham số là đối tượng LinkedList, thực hiện in mỗi bài hát trên một dòng theo định dạng <Số thứ tự>.<Tên bài hát>.

   • Tệp main.py sử dụng thư viện quan_ly_cd.

# BẢNG GIẢI THÍCH THUẬT NGỮ

| | Thuật ngữ | Giải thích | Trang |
|---|---|---|---|
| A | An toàn dữ liệu | Việc đảm bảo sao cho dữ liệu trong các hệ thống thông tin chứa trong máy tính không bị truy cập hay sử dụng không được phép, không bị sửa đổi hay bị phá huỷ, không bị mất vì các sự cố bất thường. | 75 |
| B | Bảo mật | Giữ gìn để thông tin hay dữ liệu không bị truy cập hay sử dụng không được phép | 47 |
| G | Giao diện đồ hoạ | Giao diện giúp người dùng tương tác qua hình ảnh. Một số đối tượng phổ biến của giao diện đồ hoạ là cửa sổ (window), biểu tượng (icon), thanh cuộn (scroll bar), nút bấm (button), hộp chọn (check box),... Việc tương tác thực hiện nhờ các thiết bị chỉ định vị trí trên màn hình như chuột, bàn cảm ứng,... | 5 |
| | Giấy phép sử dụng phần mềm | Quy định về những điều kiện cần tuân thủ để được sử dụng phần mềm. | 17 |
| H | Hệ điều hành | Phần mềm hệ thống có chức năng quản lí, điều hành, phối hợp hoạt động tất cả phần cứng và phần mềm của hệ thống máy tính. Có vai trò trung gian cung cấp giao diện làm việc cho người dùng để tối ưu hoá hiệu suất sử dụng máy tính. | 5 |
| N | Ngôn ngữ truy vấn CSDL | Ngôn ngữ máy tính dùng để yêu cầu cập nhật, truy xuất dữ liệu,... trong một hệ QTCSDL.<br>Ngôn ngữ truy vấn chuẩn mực và được sử dụng rộng rãi nhất là ngôn ngữ truy vấn có cấu trúc SQL (Structured Query Language). | 69 |
| O | O-lớn | Kí hiệu dùng để biểu diễn độ phức tạp thời gian của thuật toán theo bậc của dữ liệu. | 113 |

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: PHẠM THỊ THANH NAM – NGUYỄN THỊ THANH XUÂN
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 THỊ HUẾ
Sửa bản in: PHẠM THỊ TÌNH – TẠ THỊ HƯỜNG
Chế bản: CÔNG TY CỔ PHẦN MĨ THUẬT VÀ TRUYỀN THÔNG

**Bản quyền © (2023) 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 11 – ĐỊNH HƯỚNG KHOA HỌC MÁY TÍNH**
**Mã số: G1HHYI001h25**
In ..... bản, (QĐ .....) khổ 19 x 26,5 cm.
Đơn vị in:.....
Cơ sở in:.....
Số ĐKXB: 6-2025/CXBIPH/272-11/GD
Số QĐXB: .../..... ngày ... tháng ... năm .....
In xong và nộp lưu chiểu quý ..... năm .....
Mã số ISBN: 978-604-0-35058-9



HUÂN CHƯƠNG HỒ CHÍ MINH

## BỘ SÁCH GIÁO KHOA LỚP 11 – KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

1. Ngữ văn 11, tập một
2. Ngữ văn 11, tập hai
3. Chuyên đề học tập Ngữ văn 11
4. Toán 11, tập một
5. Toán 11, tập hai
6. Chuyên đề học tập Toán 11
7. Lịch sử 11
8. Chuyên đề học tập Lịch sử 11
9. Địa lí 11
10. Chuyên đề học tập Địa lí 11
11. Giáo dục Kinh tế và Pháp luật 11
12. Chuyên đề học tập Giáo dục Kinh tế và Pháp luật 11
13. Vật lí 11
14. Chuyên đề học tập Vật lí 11
15. Hoá học 11
16. Chuyên đề học tập Hoá học 11
17. Sinh học 11
18. Chuyên đề học tập Sinh học 11
19. Công nghệ 11 – Công nghệ cơ khí
20. Chuyên đề học tập Công nghệ 11 – Công nghệ cơ khí
21. Công nghệ 11 – Công nghệ chăn nuôi
22. Chuyên đề học tập Công nghệ 11 – Công nghệ chăn nuôi
23. Tin học 11 – Định hướng Khoa học máy tính
24. Tin học 11 – Định hướng Tin học ứng dụng
25. Chuyên đề học tập Tin học 11 – Định hướng Tin học ứng dụng
26. Chuyên đề học tập Tin học 11 – Định hướng Khoa học máy tính
27. Mĩ thuật 11 – Thiết kế mĩ thuật đa phương tiện
28. Mĩ thuật 11 – Thiết kế đồ hoạ
29. Mĩ thuật 11 – Thiết kế thời trang
30. Mĩ thuật 11 – Thiết kế mĩ thuật sân khấu, điện ảnh
31. Mĩ thuật 11 – Lí luận và lịch sử mĩ thuật
32. Mĩ thuật 11 – Điêu khắc
33. Mĩ thuật 11 – Kiến trúc
34. Mĩ thuật 11 – Hội hoạ
35. Mĩ thuật 11 – Đồ hoạ (tranh in)
36. Mĩ thuật 11 – Thiết kế công nghiệp
37. Chuyên đề học tập Mĩ thuật 11
38. Âm nhạc 11
39. Chuyên đề học tập Âm nhạc 11
40. Hoạt động trải nghiệm, hướng nghiệp 11
41. Giáo dục thể chất 11 – Bóng chuyền
42. Giáo dục thể chất 11 – Bóng đá
43. Giáo dục thể chất 11 – Cầu lông
44. Giáo dục thể chất 11 – Bóng rổ
45. Giáo dục quốc phòng và an ninh 11
46. Tiếng Anh 11 – Global Success – Sách học sinh

**Website:** https://www.nxbgd.vn

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

ISBN 978-604-0-35058-9

9 786040 350589

*Giá: 20.000 đ*