

ĐỀ CƯƠNG MÔN TIN HỌC 10

SÁCH KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

Hướng dẫn học sinh tự học Môn Tin học để bổ sung kiến thức và dự kiểm tra chuyển đổi môn học lựa chọn.

1. Học sinh tự nghiên cứu Bài học trong Sách Tin học 10 của bộ sách Kết nối tri thức với cuộc sống.
2. Học sinh hoàn thành các Yêu cầu cần đạt và nắm Nội dung Bài học trong Đề cương.
3. Học sinh hoàn thành trả lời các câu hỏi ở phần Củng cố - Vận dụng của từng Bài học trong Đề cương vào vở và nộp lại vở cho Nhà trường vào cuối kì tự học.
4. Kết quả đánh giá các câu trả lời phần Củng cố - Vận dụng trong vở của học sinh được dùng để đánh giá việc hoàn thành chương trình học của học sinh.

CHỦ ĐỀ 1: MÁY TÍNH VÀ XÃ HỘI TRI THỨC

BÀI 1: THÔNG TIN VÀ XỬ LÝ THÔNG TIN

A. YÊU CẦU CẦN ĐẠT:

- * Phân biệt được thông tin và dữ liệu.
- * Nêu được sự ưu việt của việc sử dụng lưu trữ, xử lý và truyền thông số.
- * Chuyển đổi được các đơn vị lưu trữ dữ liệu.

B. NỘI DUNG BÀI HỌC

1. Thông tin và dữ liệu

a) Quá trình xử lý thông tin

- Thông tin là tất cả những gì mang lại cho chúng ta hiểu biết.
- Quá trình xử lý thông tin của máy tính gồm các bước sau:
 - + Bước 1. Tiếp nhận dữ liệu: Máy tính tiếp nhận dữ liệu thường theo hai cách:
 - Cách 1. Từ thiết bị
 - Cách 2. Từ bàn phím do con người nhập
 - + Bước 2. Xử lý dữ liệu: Biến đổi dữ liệu trong bộ nhớ máy tính để tạo ra dữ liệu mới.
 - + Bước 3. Đưa ra kết quả: Máy tính có thể đưa ra kết quả theo hai cách:
 - Cách 1. Dữ liệu được thể hiện dưới dạng văn bản, âm thanh, hình ảnh,... mà con người có thể hiểu được. Như vậy dữ liệu đã được chuyển thành thông tin.
 - Cách 2. Lưu dữ liệu lên một vật mang tin như thẻ nhớ hoặc chuyển thành dữ liệu đầu vào cho một hoạt động xử lý khác.

b) Phân biệt dữ liệu và thông tin

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

Kết luận:

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

2. Đơn vị lưu trữ dữ liệu

- Máy tính không truy cập trong bộ nhớ tới từng bit mà truy cập theo từng nhóm bit. Nghĩa gốc của “byte” là một đơn vị dữ liệu dưới dạng một dãy các bit có độ dài nhỏ nhất có thể truy cập được.
- Các máy tính ngày nay đều tổ chức bộ nhớ trong thành những đơn vị lưu trữ có độ dài bằng bội của byte như 2, 4 hay 8 byte.
- Byte là đơn vị đo lường lưu trữ dữ liệu (thường được gọi là đơn vị lưu trữ thông tin)
- Các đơn vị đo dữ liệu hơn kém nhau $2^{10} = 1024$ lần

- Bảng các đơn vị lưu trữ dữ liệu

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

3. Lưu trữ, xử lí và truyền thông bằng thiết bị số

- Thẻ nhớ, bộ thu phát wifi, máy tính xách tay là các thiết bị số.

Về lưu trữ:

- Có thể lưu trữ một lượng thông tin rất lớn trong một thiết bị nhớ gọn nhẹ với chi phí thấp.

Ví dụ: Một đĩa cứng khoảng 2TB, có thể chứa 1 khối lượng thông tin ngang với 1 thư viện sách của 1 trường đại học.

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

Về xử lí:

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

- Máy tính thực hiện tính toán nhanh, cho kết quả chính xác và ổn định.

Về truyền thông.

- Xem phim qua Internet, tương tác với nhau qua mạng xã hội “một cách tức thời”.

- Các gia đình có thể sở hữu các đường cáp quang với tốc độ vài chục Mb/s, tương đương với vài triệu kí tự một giây

=> Thiết bị số có các ưu điểm:

- Giúp xử lí thông tin với năng suất rất cao và ổn định
- Có khả năng lưu trữ với dung lượng lớn, giá thành rẻ, tìm kiếm nhanh và dễ dàng.
- Có khả năng truyền tin với tốc độ rất lớn
- Giúp thực hiện tự động, chính xác, chi phí thấp và tiện lợi hơn một số việc.

C. CÙNG CỐ - VẬN DỤNG

(Học sinh trả lời các câu hỏi vào vở – nộp vở vào cuối kì)

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

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

CHỦ ĐỀ 1: MÁY TÍNH VÀ XÃ HỘI TRI THỨC

BÀI 2: VAI TRÒ CỦA THIẾT BỊ THÔNG MINH VÀ TIN HỌC ĐỐI VỚI XÃ HỘI

A. MỤC TIÊU CẦN ĐẠT:

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

B. NỘI DUNG BÀI HỌC:

1. THIẾT BỊ THÔNG MINH

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

- Thiết bị thông minh là các thiết bị điện tử có thể hoạt động một cách tự chủ trong một mức độ nhất định nhờ các phần mềm điều khiển được cài đặt sẵn.

Ví dụ:

- + Đồng hồ lịch vạn niên không có khả năng kết nối, máy ảnh số không hoạt động tự chủ => không phải là thiết bị thông minh
- + Camera kết nối internet để truyền dữ liệu một cách tự động và có khả năng chọn lọc chi ghi hình khi phát hiện chuyển động => là một thiết bị thông minh
- + Thiết bị thông minh thường gặp: điện thoại thông minh, máy tính bảng
- + Một số thiết bị thông minh hiện nay còn được tích hợp thêm khả năng “bắt chước” một vài hành vi hay cách tư duy của con người ở các mức độ khác nhau.

Ví dụ, người máy có thể hiểu và giao tiếp ngôn ngữ tự nhiên với con người; xe tự hành có thể dự đoán khả năng va chạm, từ đó giảm tốc độ và tránh để giữ an toàn,..... Các khả năng “bắt chước” đó của thiết bị thông minh nói riêng và của máy móc nói chung, tuy còn hạn chế, được gọi chung là trí tuệ nhân tạo (AI-artificial intelligence)

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

- Cuộc cách mạng công nghiệp lần thứ tư (gọi tắt là cách mạng công nghiệp 4.0) là cuộc cách mạng công nghiệp dựa trên nền tảng công nghệ số và tích hợp với các công nghệ thông minh để tạo ra quy trình và phương thức sản xuất mới.

- Cuộc cách mạng công nghiệp 4.0 đã thúc đẩy mạnh mẽ nền kinh tế tri thức.

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

⇒ Thiết bị thông minh là thiết bị điện tử có thể hoạt động tự chủ không cần sự can thiệp của con người, tự thích ứng với hoàn cảnh và có khả năng kết nối với các thiết bị khác để trao đổi dữ liệu

Thiết bị thông minh đóng vai trò chủ chốt trong các hệ thống IoT – một nội dung cơ bản của cuộc cách mạng công nghiệp 4.0

2. CÁC THÀNH TỰU CỦA TIN HỌC

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

- Các thành tựu về ứng dụng.
- Các thành tựu liên quan đến sự phát triển của chính ngành Tin học.

a) *Đóng góp của tin học với xã hội*

- *Quản lí.* Dùng máy tính quản lí các quy trình nghiệp vụ giúp xử lí công việc nhanh chóng, chính xác, hiệu quả và tiện lợi, tiết kiệm chi phí.

- *Tự động hóa.* Nhờ máy tính, tự động hóa đã thay đổi căn bản.

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

- Thay đổi cách thức làm việc của nhiều ngành nghề. Nhiều công việc có thể thực hiện trực tuyến như dạy học, mua hàng...

- Giao tiếp cộng đồng. Tin học giúp trao đổi thông tin nhanh chóng và hiệu quả qua các ứng dụng như

thư điện tử, các diễn đàn trên các trang web và các mạng xã hội như Youtube, Facebook, Twitter, Zalo...

b) Một số thành tựu phát triển của Tin học

- Hệ điều hành. Hệ điều hành giúp quản lý thông tin, quản lý phần cứng, quản lý các tiến trình xử lý của máy tính và cung cấp giao diện làm việc với người dùng.

- Mạng và Internet:

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

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

- Các ngôn ngữ lập trình bậc cao. Các chương trình máy tính phải được viết trong một ngôn ngữ lập trình. Ví dụ: FORTRAN, Cobol, C, Pascal, Python,...

- Các hệ quản trị cơ sở dữ liệu: cung cấp các công cụ để tổ chức, cập nhật, truy cập dữ liệu không phụ thuộc vào các bài toán cụ thể. Ví dụ DB2, MS/SQL, Oracle, MySQL.

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

⇒ Ngày nay, Tin học đã đem lại nhiều thay đổi trong mọi lĩnh vực của xã hội, từ quản lý điều hành, tự động hoá các quy trình sản xuất, giải quyết các bài toán cụ thể trong khoa học và kỹ thuật cho tới việc thay đổi cách thức làm việc của nhiều ngành nghề cũng như thói quen giao tiếp cộng đồng,...

Sự phát triển mạnh mẽ của các hệ thống phần cứng, các thiết bị số cùng các phần mềm hệ thống, phần mềm công cụ, phần mềm ứng dụng, các hệ quản trị cơ sở dữ liệu,... và sự phát triển mạng tính bùng nổ của mạng máy tính và Internet là những yếu tố quyết định để máy tính trở thành một phần không thể thiếu trong xã hội hiện đại.

C. CÙNG CÓ VÀ DẶN DÒ

(Học sinh trả lời các câu hỏi vào tập – nộp tập vào cuối kì)

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

CHỦ ĐỀ 1: MÁY TÍNH VÀ XÃ HỘI TRI THỨC

BÀI 7: THỰC HÀNH SỬ DỤNG THIẾT BỊ SỐ THÔNG DỤNG

A. YÊU CẦU CẦN ĐẠT:

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

B. NỘI DUNG BÀI HỌC:

1. TRỢ THỦ SỐ CÁ NHÂN

- Trợ thủ số cá nhân (Personal Digital Assistant - PDA) là các thiết bị số trong đó tích hợp một số chức năng hữu ích cho người dùng trong đời sống hàng ngày.



Hình 7.2. Màn hình PDA

- Các PDA phổ biến là điện thoại thông minh, máy tính bảng, đồng hồ thông minh, máy đọc sách... Phần lớn các PDA dạng di động và máy tính bảng hiện nay (Hình 7.2) đều chạy trên 2 hệ điều hành phổ biến là IOS của hãng Apple và Android của hãng Google.

Ghi nhớ

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

2. THỰC HÀNH SỬ DỤNG THIẾT BỊ SỐ CÁ NHÂN

Tìm hiểu về cách thức sử dụng điện thoại thông minh.

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

Hướng dẫn

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



Hình 7.4. Các nút trên điện thoại thông minh

- Nút khóa: Dùng để bật máy hoặc tắt màn hình
- Nút tăng/giảm âm lượng. Một số máy có nút bật tắt âm thanh.

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

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

Hướng dẫn

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

- Thanh trạng thái: hiển thị tình trạng kết nối, thời gian hiện tại, tỉ lệ % pin 16:10 còn lại....
- Các biểu tượng ứng dụng (application – gọi tắt app) cài trên máy. Các ứng dụng được nhà sản xuất cài đặt sẵn hoặc do người dùng cài đều được liệt kê ở đây. Với kích thước hữu hạn của màn hình chính, sau một thời gian, màn hình sẽ hết chỗ, khi đó sẽ có thêm các trang để chứa các biểu tượng của các ứng dụng mới
- Thanh truy cập nhanh chứa các ứng dụng hay dùng, sẽ được lập lại ở cuối tất cả các trang của màn hình chính.

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

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

- Bấm nút Home.
- Vuốt màn hình cảm ứng theo các chiều trái, phải, lên.
- Bấm vào phím **Quay lại** và phím **Tổng quan** (nếu dùng điện thoại có hệ điều hành Android)

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

Hướng dẫn

Bước 1. Quan sát các biểu tượng Hình 7.6 và cho biết những ứng dụng mà em biết Các chức năng và các ứng dụng có sẵn hoặc được cài đặt sau này đều được thể hiện bởi các biểu tượng trên màn hình.

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

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

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

Bước 2:

- Mở một ứng dụng hỗ trợ học tập trực tuyến như Zoom, Google meets tham gia buổi học trực tuyến do thầy/cô giáo thiết lập
- Mở và đăng ký dịch vụ lưu trữ đám mây như OneDrive, Google Drive.

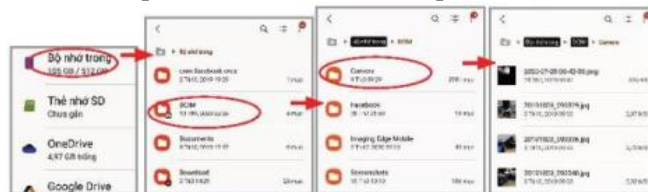
Nhiệm vụ 4. Hãy tìm xem trên điện thoại của bạn một ứng dụng quản lý tệp: Mở một tệp ảnh bất kì để xem thông tin, xóa tệp trên đám mây.

Hướng dẫn.

Bước 1: Mở ứng dụng quản lý tệp

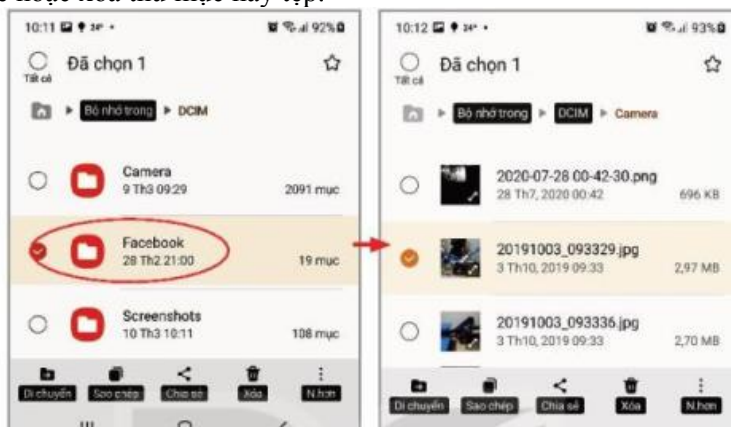
Bước 2: Thao tác mở, chọn, xem, sao chép, di chuyển các tệp tin trên điện thoại.

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



Hình 7.7. Mở để xem các tệp ảnh chụp

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



Hình 7.8. Mở tệp để di chuyển, sao chép, chia sẻ và xóa

C. CÙNG CỐ - VẬN DỤNG

(Học sinh trả lời các câu hỏi vào tập – nộp tập vào cuối kì)

Câu 1: Em hãy kết nối điện thoại thông minh với máy tính để sao chép ảnh chụp vào thư mục trên máy tính. Hãy thực hành lưu trữ các ảnh đó trên dịch vụ lưu trữ đám mây.

CHỦ ĐỀ 2: MẠNG MÁY TÍNH VÀ INTERNET

BÀI 8: MẠNG MÁY TÍNH TRONG CUỘC SỐNG HIỆN ĐẠI

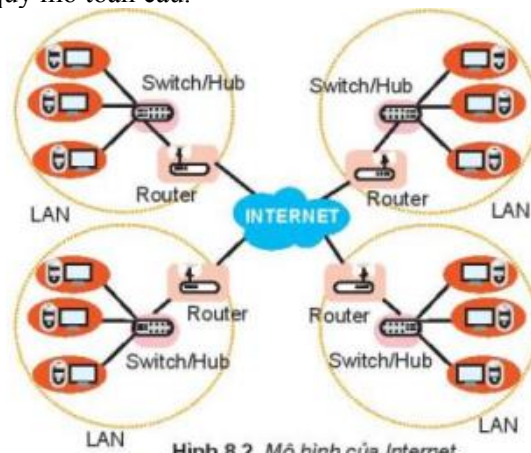
A. YÊU CẦU CẦN ĐẠT

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

B. NỘI DUNG BÀI HỌC

1. MẠNG LAN VÀ INTERNET

- Theo phạm vi địa lí, các mạng máy tính có thể chia thành hai loại là mạng cục bộ (Local Area Network, viết tắt là LAN) và mạng diện rộng (Wide Area Network, viết tắt là WAN).
- + Mạng LAN có phạm vi địa lí nhỏ như gia đình, trường học hay công ty.
- + Mạng diện rộng được hình thành bằng cách liên kết các LAN hay các máy tính đơn lẻ.
- Internet là mạng diện rộng có quy mô toàn cầu.



Hình 8.2. Mô hình của Internet

- + Switch hay HUB chỉ chuyển tiếp dữ liệu trong nội bộ mạng LAN.
- + Nguyên lí hoạt động của Router là khi phát hiện thấy dữ liệu gửi cho thiết bị không có trong LAN thì nó sẽ gửi qua công Internet. Người ta dùng router để kết nối các Lan với nhau.

2. Vai trò của Internet

- Internet ngày càng được sử dụng rộng rãi trên toàn thế giới và có ảnh hưởng đến hầu hết các lĩnh vực hoạt động của con người.
- + **Trong giao tiếp cộng đồng:** Internet đã thay đổi cách mọi người tương tác với nhau.
- + **Trong giáo dục:** Internet đã giúp hoạt động giáo dục hiệu quả hơn, Internet là một nguồn thông tin khổng lồ về mọi lĩnh vực.

Ghi nhớ

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

3. ĐIỆN TOÁN ĐÁM MÂY

a) Khái niệm về điện toán đám mây

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

b) Các dịch vụ điện toán đám mây cơ bản

- Các dịch vụ đám mây cơ bản nói chung đều chủ yếu liên quan tới việc cho thuê các tài nguyên phần mềm và phần cứng
- Phần mềm được chia thành 2 nhóm: nhóm các phần mềm ứng dụng và nhóm các phần mềm nền tảng (platform)
- Ví dụ: phần mềm ứng dụng: Google Docs, Zoom
- Việc cho thuê phần mềm ứng dụng được viết tắt là SaaS (Software as a service – phần mềm như là dịch vụ)
- Việc cho thuê nền tảng được viết tắt là PaaS (Platform as a service – nền tảng như là dịch vụ)

- Phần cứng như máy chủ, thiết bị lưu trữ, .. – những cấu thành quan trọng của hạ tầng công nghệ thông tin cũng có thể cho thuê qua Internet.
 - Lưu trữ thông tin trên Internet thông qua Dropbox hay Google Drive là một trong các ví dụ thuê phần cứng đơn giản nhất.
 - Việc cho thuê nền tảng được viết tắt là IaaS (Infrastructure as a service – hạ tầng như là dịch vụ)
- => SaaS, PaaS, IaaS là các dịch vụ chủ yếu của điện toán đám mây

c) Lợi ích của dịch vụ đám mây

- *Tính mềm dẻo và độ sẵn sàng cao:* Người dùng không bị phụ thuộc vào phương tiện cá nhân, thời gian và địa điểm là việc miễn là có kết nối Internet.
- *Chất lượng cao:* Các nhà cung cấp dịch vụ đám mây thường đầu tư chuyên nghiệp. Phần mềm được kiểm tra định nhờ số lượng người dùng lớn. Hạ tầng có công suất dự phòng lớn, ổn định và an toàn.
- *Kinh tế hơn:* Do chia sẻ cho nhiều người, dịch vụ đám mây có thể phân tải các dịch vụ và người dùng để không bị lãng phí. Chính người dùng cũng chỉ trả tiền theo mức sử dụng. Rất nhiều dịch vụ đám mây miễn phí đối với người dùng cá nhân (chỉ thu phí với người dùng là tổ chức) như Gmail để gửi thư, Google maps để tìm đường.

Ghi nhớ:

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

4. Kết nối vạn vật

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

- Một số lợi ích của IoT:

- + Có thể thu thập dữ liệu trên diện rộng nhờ mạng máy tính. Có thể làm việc ở những nơi có điều kiện bất lợi mà con người không làm được, như ghi dữ liệu giám sát trong lò phản ứng hạt nhân.
- + Có thể hoạt động liên tục, tự động, cung cấp dữ liệu tức thời - điều này đặc biệt quan trọng đối với hệ thống thời gian thực (real time) mà một quyết định chậm trễ có thể gây thảm họa, ví dụ điều khiển lò phản ứng hạt nhân hay là xe tự động.
- + Tiết kiệm chi phí do giảm bớt lao động thu thập và xử lý thông tin mang tính thủ công.

Ghi nhớ

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

C. Củng cố - Vận dụng

(Học sinh trả lời các câu hỏi vào tập – nộp tập vào cuối kì)

Câu 1: Tìm qua Internet một ứng dụng điện toán đám mây của một doanh nghiệp Việt Nam.

CHỦ ĐỀ 2: MẠNG MÁY TÍNH VÀ INTERNET

BÀI 9: AN TOÀN TRÊN KHÔNG GIAN MẠNG

A. YÊU CẦU CẦN ĐẠT

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

B. NỘI DUNG BÀI HỌC

1. MỘT SỐ NGUY CƠ TRÊN MẠNG

- Tin giả và tin phản văn hóa.
- Lừa đảo trên mạng.
- Lộ thông tin cá nhân.

Các biện pháp bảo vệ thông tin cá nhân:

- + Không ghi chép thông tin cá nhân ở những nơi mà người khác có thể đọc.
- + Giữ cho máy tính không bị nhiễm các phần mềm gián điệp.
- + Cẩn trọng khi truy cập mạng qua wifi công cộng vì hầu hết những trạm wifi công cộng không mã hoá thông tin khi truyền.
- Bắt nạt trên không gian mạng. Hành vi bắt nạt trên mạng ảnh hưởng nghiêm trọng tới tâm lí của nạn nhân vì:
 - Việc bắt nạt có thể xảy ra dai dẳng, bất cứ lúc nào;
 - Người bắt nạt có thể ẩn danh, không biết là ai để đối phó;
 - Số người theo dõi, bình luận có thể rất đông gây áp lực nặng nề, khiến nạn nhân có nguy cơ tự cô lập;
 - Nhiều người không tự giải quyết được nhưng không dám nói ra, dẫn đến trầm cảm và có các hành vi tiêu cực. Bắt nạt là một kiểu khủng bố trên không gian mạng.

Một số biện pháp phòng chống hành vi bắt nạt:

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

- **Nghiện mạng.**

Ghi nhớ

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

2. PHẦN MỀM ĐỘC HẠI

- Một đối tượng gây mất an toàn là phần mềm độc hại (malicious software, viết tắt là malware), những phần mềm được viết ra với ý đồ xấu, gây hại cho người dùng.
- Theo cơ chế lây nhiễm, có hai loại phần mềm độc hại là virus và worm. Còn một loại phần mềm độc hại khác là trojan chỉ nhằm chiếm đoạt thông tin hay chiếm quyền sử dụng máy tính sẽ ít chú trọng đến tính năng lây nhiễm.

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

- **Virus:** chỉ là các đoạn mã độc và phải gắn với một phần mềm mới phát tác và lây lan được. Khi chạy một phần mềm đã nhiễm virus, đoạn mã độc sẽ được đưa vào bộ nhớ, chờ khi thi hành một phần mềm khác sẽ chèn vào để hoàn thành một chu kì lây lan.
- **Worm, sâu máy tính:** là một phần mềm hoàn chỉnh. Để lây worm lợi dụng những lỗ hổng bảo mật của

hệ điều hành hoặc dẫn dụ lừa người dùng chạy để cài đặt vào máy của nạn nhân. Cách lừa thông thường là để một liên kết ngầm trong email hoặc tin nhắn với vỏ bọc là một nội dung lành mạnh, ví dụ “bấm vào đây để nhận tin” nhưng khi bấm vào, ngoài bản tin thì chính phần mềm độc hại cũng được tải vào máy.

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

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

b) Tác hại của phần mềm độc hại

- Virus hay worm: lây lan và gây ra các tác động không mong muốn
- Trojan: thực hiện các hoạt động nội gián.
- Các virus hay worm "dữ" có thể làm hỏng các phần mềm khác trong máy xóa dữ liệu hay làm tê liệt hệ thống máy tính.
- Virus có thể bị phát hiện theo hành vi, nhưng các worm (sâu) thường do chính nạn nhân bị lừa cài đặt nên rất khó phát hiện. Nhiều sâu đã gây ra những thảm họa.

c) Phòng chống phần mềm độc hại

- Cần thận trọng khi chép các tệp chương trình hay dữ liệu vào máy từ ổ cứng rời, thẻ nhớ hoặc tải về từ mạng.
- Không mở các liên kết trong email hay tin nhắn mà không biết rõ có an toàn hay không.
- Đừng để lộ mật khẩu các tài khoản của mình để tránh bị kẻ xấu chiếm quyền mạo danh.
- Ngoài ra, hãy sử dụng các phần mềm phòng chống các phần mềm độc hại.

Ghi nhớ:

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

3. THỰC HÀNH

Dùng phần mềm phòng chống virus Windows Defender

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

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

Hướng dẫn.

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



Hình 9.1. Truy cập chức năng bảo vệ chống virus

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



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

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

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

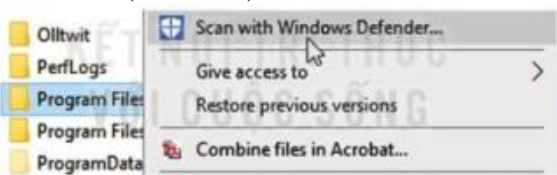
Bước 3: Quét virus. Ta có thể nháy vào nút Quick scan hoặc vào lựa chọn Scan options để lựa chọn kiểu quét và quét.

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

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

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

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



Hình 9.4. Truy cập nhanh lệnh quét trên thư mục

C. Củng cố - vận dụng

(Học sinh trả lời các câu hỏi vào tập – nộp tập vào cuối kì)

Câu 1: Em hãy tìm hiểu qua Internet các cách thức tấn công từ chối dịch vụ.

Câu 2: Em hãy tìm trên mạng thông tin về worm, kể một worm với tác hại của nó.

CHỦ ĐỀ 2: MẠNG MÁY TÍNH VÀ INTERNET

BÀI 10: THỰC HÀNH KHAI THÁC TÀI NGUYÊN TRÊN INTERNET

A. YÊU CẦU CẦN ĐẠT

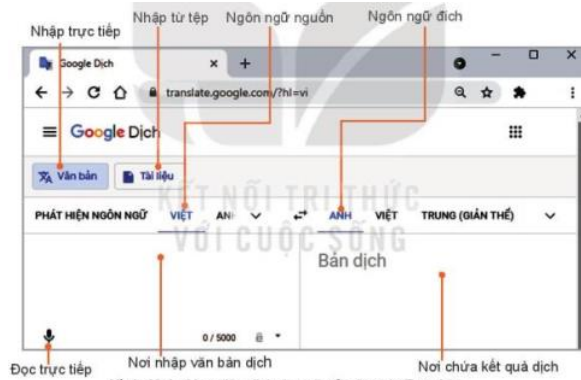
* Khai thác được một số dịch vụ và tài nguyên trên internet phục vụ học tập gồm: phần mềm dịch, kho học liệu mở.

B. NỘI DUNG BÀI HỌC

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

Hướng dẫn

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

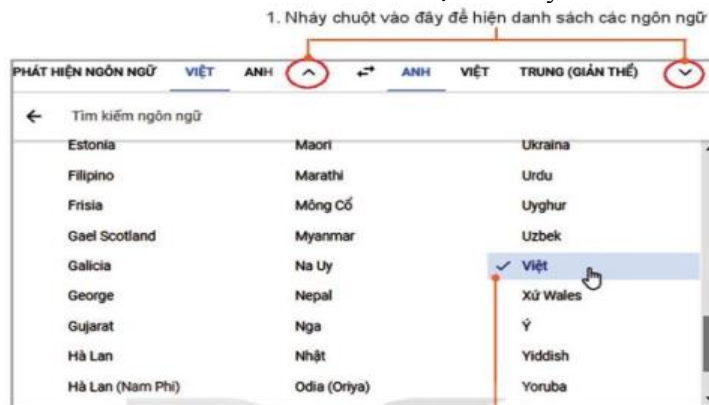


Hình 10.1. Giao diện dịch đa ngữ của Google Translate

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

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

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

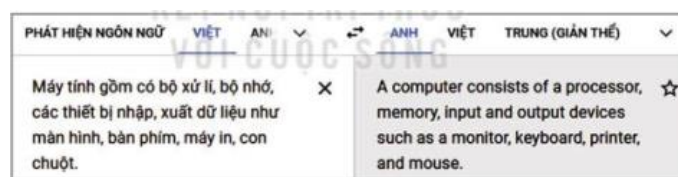


Hình 10.2. Chọn ngôn ngữ làm việc

Bước 3: Nhập văn bản để dịch.

Có ba cách nhập:

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

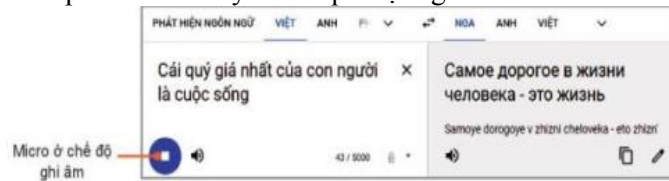


Hình 10.3. Nhập văn bản trực tiếp vào khung ngôn ngữ nguồn

2. Nhập bằng giọng nói. Trong trường hợp này, máy tính phải có micro để thu âm. Trước khi nói, phải chọn biểu

tượng micro. Khi biểu tượng micro đổi thành màu xanh thì em hãy đọc đoạn văn bản cần dịch. Nếu nháy chuột vào biểu tượng micro một lần nữa thì chế độ nhập bằng lời dừng, chuyển sang chế độ gõ trực tiếp văn bản cần dịch.

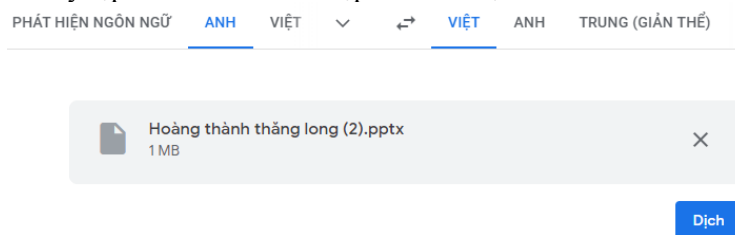
Hình 10.4 là giao diện dịch từ tiếng Việt sang tiếng Nga một câu trong một đoạn văn nổi tiếng của nhà văn Ostrovski trong tiểu thuyết “Thép đã tôi thế đấy”. Kết quả dịch gần chính xác so với bản gốc trong tác phẩm.



Hình 10.4. Nhập văn bản bằng giọng nói

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

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



Hình 10.5. Nhập văn bản từ tệp

Sau đó nháy vào nút lệnh **Dịch để dịch**.

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

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

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

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

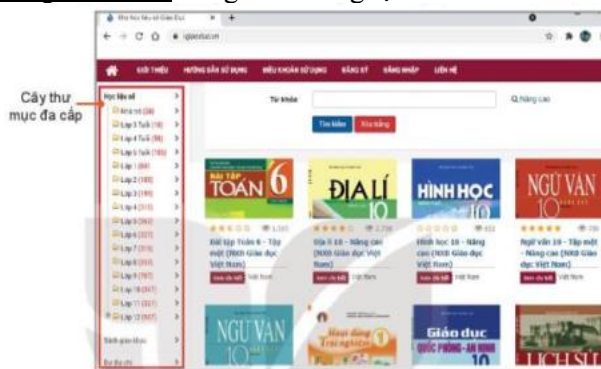


Hình 10.6. Kết quả dịch bảng tính

Hình 10.6. Kết quả dịch bảng tính

Nhiệm vụ 2: Khai thác một nguồn học liệu mở trên Internet để tìm các nguồn tài liệu phục vụ học tập, Hướng dẫn

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



Hình 10.7. Trang chủ kho học liệu số

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

Bước 2: Tìm kiếm, truy cập các học liệu

- Thư mục cấp một gồm ba loại là Học liệu số, Sách giáo khoa và Địa chỉ. Muốn xem loại nào nhấp chuột vào biểu tượng > ở mục tương ứng để mở ra thư mục cấp hai.
- Nhấp chuột vào một mục của thư mục cấp hai, ví dụ Học liệu số để mở ra thư mục cấp ba.
- Thư mục cấp ba của Học liệu số và Sách giáo khoa trải ra theo các môn học. Chỉ cần chọn môn để xem tất cả các học liệu của môn đó.



Hình 10.8. Các bài giảng môn Đại số lớp 10

Bước 3: Xem bài giảng.

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



Hình 10.9. Giới thiệu học liệu

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

CHỦ ĐỀ 3: ĐẠO ĐỨC, PHÁP LUẬT VÀ VĂN HÓA TRONG MÔI TRƯỜNG SỐ

BÀI 11: ỨNG XỬ TRÊN MÔI TRƯỜNG SỐ. NGHĨA VỤ TÔN TRỌNG BẢN QUYỀN

A. YÊU CẦU CẦN ĐẠT

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

B. NỘI DUNG BÀI HỌC

1. Những vấn đề đạo đức, pháp luật và văn hóa

- Đưa tin không phù hợp lên mạng (bao gồm cả đang chia sẻ tin bài). Tùy theo nội dung thông tin và hậu quả của việc đăng tin, mà những hành vi đó có thể là vi phạm pháp luật hay vi phạm đạo đức.
 - Công bố thông tin cá nhân hay tổ chức mà không được phép, gây ảnh hưởng đến uy tín và danh dự của cá nhân hay tổ chức là hành vi vi phạm pháp luật.
 - Gửi thư rác hay tin nhắn rác. Những thư hay tin nhắn nhằm mục đích quảng cáo mà người nhận không muốn nhận hoặc không bắt buộc phải nhận theo quy định pháp luật được gọi là thư hay tin nhắn rác. Về bản chất, quảng cáo bằng tin nhắn không phải là một hành vi xấu và không vi phạm pháp luật, nhưng nếu gửi nhiều mà người nhận đã có phản ứng không muốn tiếp nhận thì lại trở thành hành vi quấy nhiễu.
 - Vi phạm bản quyền khi sử dụng dữ liệu và phần mềm. Vấn đề bản quyền không chỉ đặt ra khi giao dịch trên mạng mà trong bất cứ hoàn cảnh nào, những sản phẩm được số hóa và đưa lên mạng rất dễ bị lấy, phát tán, sửa đổi gây thiệt hại cho chủ sở hữu.
 - Bắt nạt qua mạng.
 - Lừa đảo qua mạng. Các hình thức lừa đảo trên mạng khá phổ biến và tinh vi. Nhiều trường hợp lừa đảo qua mạng gây ra những thiệt hại rất lớn.
 - Ứng xử thiếu văn hóa. Trên các diễn đàn mạng hiện nay, có nhiều người tranh luận thiếu văn hóa, không tôn trọng người đối thoại, thậm chí chửi tục hay công kích sỉ nhục lẫn nhau.
- => Cần có những quy định pháp luật và các chuẩn mực đạo đức để đảm bảo lợi ích chung của cộng đồng người dùng mạng.

Ghi nhớ:

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

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

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

a) Các văn bản quy phạm pháp luật

- Các văn bản quy phạm pháp luật gồm nhiều loại trong đó có: các bộ luật do Quốc hội ban hành và các nghị định do Chính phủ ban hành để cụ thể hoá các điều khoản của luật.
- Quốc hội Việt Nam đã ban hành nhiều bộ luật liên quan đến Công nghệ Thông tin (CNTT) như: Luật Giao dịch điện tử (2005), Luật Công nghệ Thông tin (2006) và Luật an ninh mạng (2018).
- Cụ thể:
 - + Nghị định 90/2008/NĐ-CP ban hành ngày 13/8/2008 về chống thư rác;
 - + Nghị định 72/2013/NĐ-CP ban hành ngày 15/7/2013 về quản lí, cung cấp và sử dụng dịch vụ Internet;
 - + Nghị định 15/2020/NĐ-CP ban hành ngày 3/2/2020 quy định xử phạt vi phạm hành chính trong lĩnh vực bưu chính, viễn thông, tần số vô tuyến điện, CNTT và giao dịch điện tử.
 - + Tháng 6/2021, Bộ Thông tin và Truyền thông ban hành Bộ Quy tắc ứng xử trên mạng xã hội theo Quyết định số 874/QĐ-BTTTT ban hành ngày 17/6/2021.

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

- Điều 12 khoản 2 của Luật công nghệ thông tin quy định cấm "Cung cấp, trao đổi, truyền đưa, lưu trữ, sử dụng thông tin số" nhằm các mục đích sau đây:
 - a) Chống Nhà nước Cộng hoà hội chủ nghĩa Việt Nam, phá hoại khối đoàn kết toàn dân.
 - b) Kích động bạo lực, tuyên truyền chiến tranh xâm lược, gây hận thù giữa các dân tộc và nhân dân các nước, kích động dâm ô, đồi trụy, tội ác, tệ nạn xã hội, mê tín dị đoan, phá hoại thuần phong mỹ tục của dân tộc.

- c) Tiết lộ bí mật nhà nước, bí mật quân sự, an ninh, kinh tế, đối ngoại và những mặt khác đã được pháp luật quy định.
- d) Xuyên tạc, vu khống, xúc phạm uy tín của tổ chức, danh dự, nhân phẩm, uy tín của công dân.
- e) Quảng cáo, tuyên truyền hàng hoá, dịch vụ thuộc danh mục cấm đã được pháp luật quy định.
- Điều 8 khoản 1 trong Luật an ninh mạng cấm sử dụng không gian mạng để thực hiện hành vi sau đây:
- Hành vi quy định tại khoản 1 Điều 18 của Luật này.
 - Tổ chức, hoạt động, cấu kết, xúi giục, mua chuộc, lừa gạt, lôi kéo, đào tạo, huấn luyện người chống Nhà nước Cộng hoà xã hội chủ nghĩa Việt Nam.
 - Xuyên tạc lịch sử phủ nhận thành tựu cách mạng, phá hoại khối đại đoàn kết toàn dân tộc, xúc phạm tôn giáo, phân biệt đối xử về giới, phân biệt chủng tộc.
 - Thông tin sai sự thật gây hoang mang trong Nhân dân, gây thiệt hại cho hoạt động kinh tế - xã hội, gây khó khăn cho hoạt động của cơ quan Nhà nước hoặc người thi hành công vụ, xâm phạm quyền lợi và lợi ích hợp pháp của cơ quan, tổ chức, cá nhân khác.
 - Hoạt động mại dâm, tệ nạn xã hội, mua bán người, đăng tải thông tin dâm ô, đồi trụy, tội ác, phá hoại thuần phong, mỹ tục của dân tộc, đạo đức xã hội, sức khoẻ của cộng đồng.
 - Xúi giục, lôi kéo, kích động người khác phạm tội.
- Một số hành vi vi phạm pháp luật về đưa tin trên mạng xã hội được cụ thể hoá kèm theo mức phạt trong Điều 101, khoản 1 của Nghị định 15/2020/NĐ-CP
- Các nguyên tắc để nâng cao tính an toàn khi chia sẻ thông tin trên môi trường số:
- + Trước khi đăng tin, hãy kiểm tra tin bài có vi phạm pháp luật hay không
 - + Không chia sẻ tin bài vi phạm pháp luật. Ngoài ra cần phải biết tin tức có chính xác không. Ngày nay trên mạng có rất nhiều tin giả, việc chia sẻ một tin giả chính là tiếp tay cho hành vi tung tin giả.
 - + Ngay trong cả trường hợp việc đưa tin không vi phạm pháp luật thì cũng phải cân nhắc hậu quả, nhất là khía cạnh đạo đức. Một tin vô hại với người này có thể mang lại tai họa cho một người khác.

Ghi nhớ

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

3. Quyền tác giả và bản quyền

a) Quyền tác giả

- **Quyền tác giả** là quyền của tổ chức, cá nhân đối với tác phẩm do mình sáng tạo ra hoặc sở hữu.
- Trong luật sở hữu trí tuệ được Quốc Hội ban hành ngày 25 tháng 6 năm 2019, quy định quyền tác giả đối với tác phẩm (bài thơ, bài báo, bức tranh, hình vẽ, chương trình máy tính, sưu tầm dữ liệu,...) bao gồm quyền nhân thân và quyền tài sản. Trong đó:
- + *Quyền nhân thân* bao gồm các quyền: Đặt tên cho tác phẩm; Đứng tên thật hoặc bút danh trên tác phẩm; được nêu tên thật hoặc bút danh khi tác phẩm được công bố, sử dụng; Công bố tác phẩm hoặc cho phép người khác công bố tác phẩm; Bảo vệ sự toàn vẹn của tác phẩm, không cho người khác sửa chữa, cắt xén hoặc xuyên tạc tác phẩm dưới bất kì hình thức nào gây phương hại đến danh dự và uy tín của tác phẩm,...
 - + *Quyền tài sản* bao gồm các quyền: Làm tác phẩm phái sinh; Sao chép tác phẩm; Truyền đạt tác phẩm đến công chúng; Cho thuê bản gốc hoặc bản sao chương trình máy tính,...

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

- Vi phạm quyền tài sản hay quyền nhân thân đều là vi phạm bản quyền. Hành vi vi phạm bản quyền rất đa dạng. Sau đây là một số hành vi vi phạm bản quyền đối với các tác phẩm số:
- + Mạo danh tác giả.
 - + Công bố mà không được phép.
 - + Sửa chữa, chuyển thể phần mềm dữ liệu mà không được phép của tác giả.
 - + Sử dụng phần mềm lậu, không mua quyền sử dụng phần mềm đối với các phần mềm phải trả tiền
 - + Phá khoá phần mềm, vô hiệu hoá các biện pháp kỹ thuật do chủ sở hữu quyền tác giả thiết lập.
 - + Làm bản phái sinh, phân phối dữ liệu hay phần mềm, kể cả bản phái sinh mà không được phép.
 - + Chiếm đoạt mã phần mềm.
 - + Đăng tải các tác phẩm, kể cả bản phái sinh mà không có sự cho phép của chủ sở hữu.

c) Tôn trọng bản quyền trong tin học

- Trong tin học, khi mua phần mềm, cần phân biệt rõ việc mua bản quyền với mua quyền sử dụng (licence).
- + Khi mua bản quyền thì người mua có quyền thực hiện các hoạt động kinh doanh đối với tác phẩm đó giống như cách Google mua Youtube.
- + Còn nếu chỉ mua quyền sử dụng thì chỉ được sử dụng.
- Quyền sử dụng phần mềm máy tính theo số máy được cài đặt. Nếu mua quyền sử dụng cho một máy rồi cài lại

cho máy thứ hai là vi phạm bản quyền.

- Trong lĩnh vực tin học, vi phạm bản quyền có thể gây thiệt hại rất lớn cho người đầu tư vì sản phẩm có một số đặc tính:

+ Dễ sao chép với chi phí rất thấp.

+ Dễ phát tán trên quy mô lớn.

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

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

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

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

Ghi nhớ

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

C. Củng cố - vận dụng

(Học sinh trả lời các câu hỏi vào tập – nộp tập vào cuối kì)

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

- a) Vi phạm pháp luật
- b) Vi phạm đạo đức
- c) Tùy theo mức độ, có thể vi phạm đạo đức hay pháp luật.
- d) Không vi phạm gì

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

CHỦ ĐỀ 4: ỨNG DỤNG TIN HỌC

BÀI 12: PHẦN MỀM THIẾT KẾ ĐỒ HỌA

A. YÊU CẦU CẦN ĐẠT

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

B. NỘI DUNG BÀI HỌC

1. THIẾT KẾ ĐỒ HỌA

- Thiết kế đồ họa (graphic design) là quá trình thiết kế các thông điệp truyền thông bằng hình ảnh, giải quyết vấn đề thông qua sự kết hợp giữa hình ảnh; kiểu chữ với ý tưởng để truyền tải thông tin đến người xem.
- Có hai loại đồ họa cơ bản: đồ họa điểm ảnh (bitmap) và đồ họa vector (vector).
- + Trong đồ họa điểm ảnh, hình ảnh được tạo thành từ các điểm ảnh (pixel), mỗi điểm ảnh có màu riêng.
- + Trong đồ họa vector, hình ảnh được xác định theo đường nét, mỗi một đường có điểm đầu và điểm cuối, được tính bằng một phương trình toán học.
- So sánh giữa đồ họa điểm ảnh và đồ họa vector:

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

Ghi nhớ

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

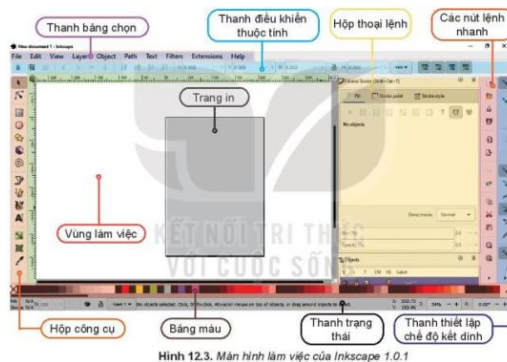
2. PHẦN MỀM ĐỒ HỌA

- Có hai loại phần mềm đồ họa:
- + Phần mềm tạo, chỉnh sửa hình vector: Adobe Illustrator, CorelDraw, Inkscape, ...
- + Phần mềm xử lý ảnh bitmap: Adobe Photoshop, GIMP, ...

a) Tải và cài đặt phần mềm

- Tải phần mềm tại địa chỉ: <https://inkscape.org/release/inkscape-1.0/>.
- Cài đặt theo hướng dẫn.

b) Giao diện của Inkscape



Hình 12.3. Màn hình làm việc của Inkscape 1.0.1

Ghi nhớ

- Có hai loại phần mềm đồ họa. Phần mềm tạo, chỉnh sửa hình vector và phần mềm xử lý ảnh bitmap
- Inkscape là phần mềm miễn phí để tạo, chỉnh sửa sản phẩm đồ họa vector

3. CÁC ĐỐI TƯỢNG ĐỒ HỌA CỦA HÌNH VẼ

- Tạo tệp mới bằng lệnh **File/New...** hoặc nhấn tổ hợp phím **Ctrl + N**
- Các đối tượng được thêm vào hình bằng cách:
 - + C1. Tạo ra các bản sao của các đối tượng đã có trong vùng làm việc
 - + C2. Chọn đối tượng từ hộp công cụ và vẽ vào vùng làm việc. Các đối tượng sẽ xuất hiện theo thứ tự lớp, đối tượng vẽ trước sẽ ở lớp dưới, đối tượng này có thể bị che khuất toàn bộ hay một

phần bởi các đối tượng vẽ sau ở lớp trên.

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

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



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

Bước 2: Chỉnh tùy chọn trong danh sách điều khiển thuộc tính nếu cần

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

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

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

- Lưu tệp bằng lệnh **File/Save** hoặc nhấn tổ hợp phím **Ctrl + S**. Phần mềm Inkscape lưu lại sản phẩm đã tạo dưới dạng tệp hình vecto có phần mở rộng là **svg**.

Ghi nhớ

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

CHỦ ĐỀ 4: ỨNG DỤNG TIN HỌC

BÀI 13: BỔ SUNG CÁC ĐỐI TƯỢNG ĐỒ HỌA

A. YÊU CẦU CẦN ĐẠT

* Biết và sử dụng được một số chức năng của các lệnh tạo, điều chỉnh các đối tượng đồ họa đơn giản.

B. NỘI DUNG BÀI HỌC

1. CÁC ĐỐI TƯỢNG HÌNH KHỐI

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

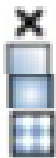
b. Bảng các thuộc tính cơ bản của một số hình có sẵn

Đối tượng	Thuộc tính	Ý nghĩa
Hình vuông, hình chữ nhật	W, H	Chiều rộng, chiều dài.
	Rx, Ry	Bán kính của góc bo (bằng 0 nếu góc của hình là góc vuông)
Hình tròn, hình elip	Rx, Ry	Bán kính theo phương ngang và thẳng đứng
	Start, End	Góc của điểm đầu và điểm cuối khi sử dụng chế độ vẽ (đơn vị: độ)
Hình đa giác, hình sao	Corners	Số đỉnh của đa giác hoặc số cánh của hình sao
	Rounded	Độ cong tại các đỉnh của hình
	Spoke ratio	Tỉ lệ bán kính từ tâm đến góc trong và từ tâm tới đỉnh đầu hình sao
	randomized	Tham số làm méo hình ngẫu nhiên

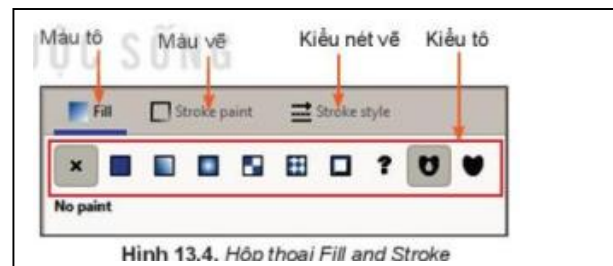
Ta có thể thay đổi thuộc tính của mỗi đối tượng được tạo từ công cụ có sẵn trong hình vẽ.

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

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



- Không màu (trong suốt)
- Màu chuyển giữa hai hoặc nhiều màu.
- Màu chuyển giữa hai hoặc nhiều màu từ trung tâm đối tượng.
- Họa văn (đối tượng được lấp đầy bởi một mẫu họa văn).



Hình 13.4. Hộp thoại Fill and Stroke

? Hủy đặt (đối tượng trở về trạng thái ban đầu, điều này cần thiết khi ta sao chép đối tượng và thay đổi thuộc tính cho đối tượng).

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


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

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

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

Lưu ý: Khi tô màu chuyển thì màu của đối tượng được tô sẽ chuyển dần từ màu này sang màu khác. Để tô

màu chuyển ta sử dụng hai kiểu tô là và . Nhấp vào biểu tượng cây bút phía dưới của hộp thoại Fill and Stroke để thay đổi thông số của gradient. Ta có thể thay đổi thông số của gradient bằng bảng điều khiển hoặc kéo thả chuột tại các vị trí điều khiển.

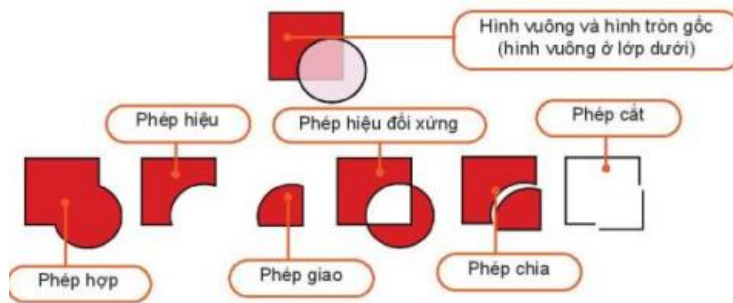
Khi sử dụng kiểu tô  ta có thể sao chép màu tại một vị trí nào đó trong vùng làm việc bằng cách sử dụng công cụ  phía dưới của hộp thoại **Fill and Stroke**.

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

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

3. CÁC PHÉP GHÉP ĐỐI TƯỢNG ĐỒ HOẠ

- Các hình phức tạp có thể thu được bằng cách ghép từ các hình đơn giản.
- Các phép ghép được sử dụng để ghép và cắt hình trong Inkscape gồm: hợp, hiệu, giao, hiệu đối xứng, chia, cắt của hai hay nhiều đối tượng đơn. Các phép ghép này được thực hiện bằng cách chọn lệnh trong bảng chọn **Path**. Phép hợp (Union, nhấn tổ hợp phím **Ctrl + +**) là tất cả các phần thuộc một trong các hình đơn. Phép hiệu (Difference, nhấn tổ hợp phím **Ctrl + -**) là phần thuộc hình lớp dưới nhưng không thuộc hình lớp trên. Phép giao (Intersection, nhấn tổ hợp phím **Ctrl + ***); là phần thuộc cả hai hình được chọn. Phép hiệu đối xứng (Exclusion, nhấn tổ hợp phím **Ctrl + ^**) là phần hình thuộc các hình trừ phần giao nhau. Phép chia (Division, nhấn tổ hợp phím **Ctrl + /**): Hình lớp dưới được chia thành các phần bởi đường nét của hình lớp trên. Phép cắt (Cut Path, nhấn tổ hợp phím **Ctrl + Alt + /**): Cắt hình lớp dưới thành các phần bởi hai điểm giao ở viền với hình lớp trên. Kết quả là các hình mới không có màu.



Hình 13.6. Các phép ghép hai hay nhiều hình

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

C. CỨNG CỐ - VẬN DỤNG

Học sinh thực hành 3 nhiệm vụ sau trên phần mềm thiết kế đồ họa

- Nhiệm vụ 1. Vẽ logo tương tự Hình 13.8
- Nhiệm vụ 2. Vẽ hình cây như Hình 13.10.
- Nhiệm vụ 3. Vẽ hình cầu như Hình 13.12.

CHỦ ĐỀ 4: ỨNG DỤNG TIN HỌC

BÀI 14: LÀM VIỆC VỚI ĐỐI TƯỢNG ĐƯỜNG VÀ VĂN BẢN

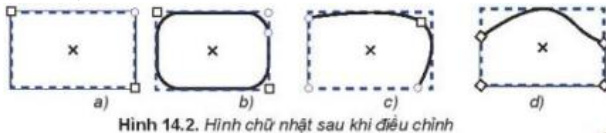
A. YÊU CẦU CẦN ĐẠT

- * Biết các thao tác chỉnh sửa hình
- * Thực hiện được việc vẽ, chỉnh sửa hình để được đường cong mong muốn

B. NỘI DUNG BÀI HỌC

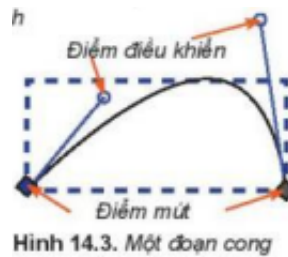
1. LÀM QUEN VỚI ĐỐI TƯỢNG DẠNG ĐƯỜNG

- Đối tượng tự do dạng đường do người dùng tạo ra, là tổ hợp của một hay nhiều đoạn cong hoặc thẳng nối lại với nhau
- Các đối tượng tự do dạng đường có thể điều chỉnh các đoạn độc lập với nhau để tạo ra hình dạng khác
- Ví dụ:



Hình 14.2. Hình chữ nhật sau khi điều chỉnh

- Đối tượng đường thường được biểu diễn bởi một chuỗi các đoạn thẳng hoặc đoạn cong nối với nhau. Mỗi đoạn cong biểu diễn bởi 4 điểm, hai điểm đầu mút và hai điểm điều khiển. Mỗi điểm điều khiển kết hợp với một điểm đầu mút tạo ra tiếp tuyến của đường cong tại điểm mút tương ứng (Hình 14.3)



Hình 14.3. Một đoạn cong

Các bước vẽ đối tượng đường

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

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

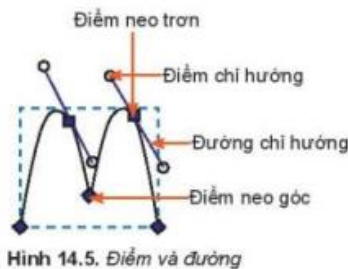
Bước 3. Nháy chuột để đặt các điểm neo trên hình vẽ (có thể kết hợp nháy chuột và kéo thả).

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

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


2. SỬ DỤNG CÔNG CỤ TÍNH CHỈNH ĐƯỜNG

- Khi nối các đoạn thẳng hoặc đoạn cong với nhau ta thu được đường cong phức tạp hơn.
- Các điểm nối giữa các đoạn có thể là điểm neo trơn (smooth nodes - thể hiện bởi hình vuông hay hình tròn) hoặc điểm neo góc (comer nodes - thể hiện bởi một hình thoi) (Hình 14.4)
- Khi xác định điểm neo trên bản vẽ, các đoạn ở giữa sẽ được tự động tạo ra để nối các điểm neo đã có. Hình vẽ ban đầu có thể chưa đúng với ý tưởng hoàn chỉnh, điều này sẽ được chỉnh sửa bằng công cụ tính chỉnh điểm



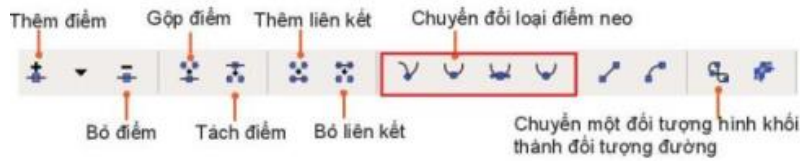
Hình 14.5. Điểm và đường

- Độ cong tại mỗi điểm neo phụ thuộc vào điểm chỉ hướng và đường chỉ hướng tại điểm đó (Hình 14.5). Các giá trị này được thay đổi bằng cách kéo thả điểm chỉ hướng. Khi thay đổi phương, chiều, độ lớn của các giá trị này tại một điểm, đoạn cong liên quan tới điểm đó sẽ thay đổi theo.
- Điểm neo trơn có hai đường chỉ hướng luôn cùng phương với nhau
- Các bước thực hiện việc chỉnh sửa điểm neo:

Bước 1. Chọn công cụ  trên hộp công cụ

Bước 2. Nháy chuột vào hình muốn chỉnh sửa


Bước 3. Nháy vào điểm neo cần sửa, chọn điểm neo hoặc điểm chỉ hướng rồi kéo thả chuột sang vị trí mới
 Nếu muốn loại bỏ, thêm mới hay chuyển đổi loại điểm neo, ... ta chọn vào biểu tượng tương ứng trên thanh điều khiển thuộc tính (Hình 14.6)



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

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

3. ĐỐI TƯỢNG VĂN BẢN

- Để bổ sung đối tượng văn bản, ta chọn biểu tượng  trên hộp công cụ.

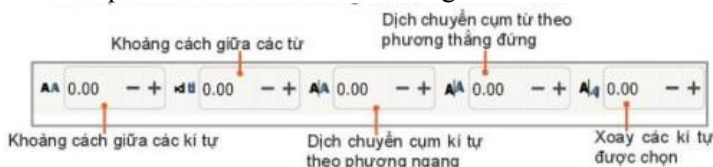
- Ta có thể tùy chỉnh từng phần trong một đối tượng văn bản bằng cách chọn đối tượng văn bản và bôi đen phần văn bản muốn căn chỉnh, sau đó điều chỉnh tham số trên thanh điều khiển thuộc tính hoặc lệnh trong bảng chọn **Text**.

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



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

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



Hình 14.10. Điều chỉnh thuộc tính từng cụm ký tự

Chú ý:

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

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

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

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

C.CỦNG CỐ - VẬN DỤNG

Thực hành

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

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

CHỦ ĐỀ 4: ỨNG DỤNG TIN HỌC

BÀI 15: HOÀN THIỆN HÌNH ẢNH ĐỒ HỌA

A. YÊU CẦU CẦN ĐẠT

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

B. NỘI DUNG BÀI HỌC

HS quan sát SGK Tin học 10/ trang 82 để tìm hiểu nội dung kiến thức theo yêu cầu

Hs hoàn thành tìm hiểu kiến thức

Học sinh thiết kế thiệp mời sinh nhật trên phần mềm thiết kế đồ họa Inkscape

CHỦ ĐỀ 5: GIẢI QUYẾT VẤN ĐỀ VỚI SỰ TRỢ GIÚP CỦA MÁY TÍNH

BÀI 16: NGÔN NGỮ LẬP TRÌNH BẬC CAO VÀ PYTHON

A. YÊU CẦU CẦN ĐẠT

- * Biết khái niệm ngôn ngữ lập trình bậc cao và ngôn ngữ lập trình bậc cao Python.
- * Phân biệt được chế độ gõ lệnh trực tiếp và chế độ soạn thảo chương trình trong môi trường lập trình Python
- * Biết cách tạo và thực hiện một chương trình Python.

B. NỘI DUNG BÀI HỌC

1. NGÔN NGỮ LẬP TRÌNH BẬC CAO

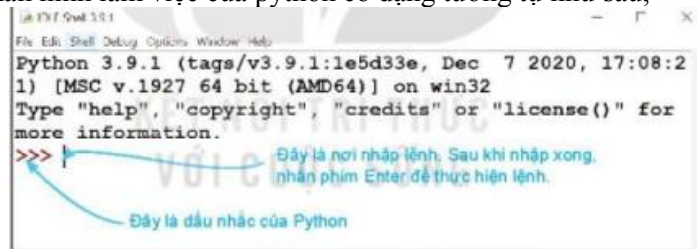
- Các lệnh viết bằng ngôn ngữ máy ở dạng mã nhị phân hay hợp ngữ sử dụng một số từ viết tắt (thường là tiếng Anh) không thuận tiện cho việc viết hoặc hiểu.
 - Lập trình bằng ngôn ngữ bậc cao: các câu lệnh được viết gần với ngôn ngữ tự nhiên. Tuy nhiên, để máy tính có thể hiểu và thực hiện, các chương trình đó cần được dịch sang ngôn ngữ máy nhờ một chương trình chuyên dụng được gọi là chương trình dịch
 - Các ngôn ngữ lập trình bậc cao như Java, C/C++, Python,... là những ngôn ngữ lập trình thông dụng nhất
 - Python là ngôn ngữ lập trình bậc cao do Guido van Rossum, người Hà Lan tạo ra và ra mắt lần đầu năm 1991.
- Ưu điểm:
- + Các câu lệnh của Python có cú pháp đơn giản. Môi trường lập trình Python dễ sử dụng, không phụ thuộc vào hệ điều hành, chạy trên nhiều loại máy tính, điện thoại thông minh, robot giáo dục,... + Python có mã nguồn mở nên thu hút nhiều nhà khoa học cùng phát triển.
 - + Các thư viện chương trình phong phú về trí tuệ nhân tạo, phân tích dữ liệu, kỹ thuật robot,... Python là ngôn ngữ lập trình được dùng phổ biến trong nghiên cứu và giáo dục

Ghi nhớ:

- Ngôn ngữ lập trình bậc cao có các câu lệnh được viết gần với ngôn ngữ tự nhiên giúp cho việc đọc, hiểu chương trình dễ dàng hơn
- Python là một ngôn ngữ lập trình bậc cao phổ biến trong nghiên cứu và giáo dục

2. MÔI TRƯỜNG LẬP TRÌNH PYTHON

Sau khi khởi động, màn hình làm việc của python có dạng tương tự như sau;



Hình 16.2. Màn hình làm việc của Python

Môi trường lập trình Python có hai chế độ:

- Chế độ gõ lệnh trực tiếp thường được dùng để tính toán và kiểm tra nhanh các lệnh.
- Chế độ soạn thảo dùng để viết các chương trình có nhiều dòng lệnh.

a) Chế độ gõ lệnh trực tiếp

- Gõ lệnh trực tiếp sau dấu nhắc >>> và nhấn phím Enter để thực hiện lệnh như sau:

>>> <lệnh python>

b) Chế độ soạn thảo

- Mở màn hình soạn thảo bằng cách vào File/NewFile



Hình 16.3. Màn hình soạn thảo trong môi trường Python

Chú ý: Có thể soạn thảo chương trình Python bằng phần mềm soạn thảo văn bản hoặc phần mềm lập trình python như Wingware, Pycharm, Thonny, VisualStudio, ...

Ghi nhớ:

=> Môi trường lập trình Python có 2 chế độ: chế độ gõ lệnh trực tiếp và chế độ soạn thảo.

3. MỘT SỐ LỆNH PYTHON ĐẦU TIÊN

Ví dụ 1. Các lệnh đầu tiên.

```
>>> 5
5
>>> 2.6
2.6
>>> "học sinh lớp 10"
'học sinh lớp 10'
```

Khi nhập 5, Python hiểu rằng đó là số nguyên 5.
Nếu nhập 2.6 thì Python tự động nhận ra đó là số thực 2.6.
Còn nếu gõ các chữ giữa cặp dấu nháy kép "" thì Python hiểu là xâu kí tự.

Python tự nhận biết kiểu dữ liệu và thực hiện các phép toán ngay trên dòng lệnh.

Ví dụ 2. Các lệnh với phép toán.

```
>>> 3 + 7
10
>>> 12*5
60
```

Các phép toán thông thường với số bao gồm phép cộng (+), trừ (-), nhân (*) và chia (/).

Ví dụ 3. Lệnh print().

```
>>> print(12)
12
>>> print(10, 3.4 + 4.1, "hoà bình")
10 7.5 hoà bình
>>> print("Dãy ba số chẵn:", 2, 4, 6)
Dãy ba số chẵn: 2 4 6
>>> print("3 + 7 =", 3+7)
3 + 7 = 10
```

Chú ý: nếu lệnh print() nhiều giá trị thì các dữ liệu này sẽ được đưa ra trên một dòng, giữa các dữ liệu sẽ có dấu cách.
Lệnh print() có thể tính toán và đưa ra kết quả của biểu thức.

- Trong Python, lệnh print() có chức năng đưa dữ liệu ra (xuất dữ liệu).

- Cú pháp lệnh print() như sau:

print(v1, v2, ..., vn) trong đó v1, v2, ..., vn là các giá trị cần đưa ra màn hình.

Ghi nhớ :

- Khi nhập giá trị số hoặc xâu kí tự từ dòng lệnh, Python tự nhận biết kiểu dữ liệu.
- Python có thể thực hiện các phép toán thông thường với số, phân biệt số thực và số nguyên.
- Lệnh print() có chức năng in dữ liệu ra màn hình, có thể in ra một hoặc nhiều giá trị đồng thời

C. Củng cố - Vận dụng

(Học sinh trả lời các câu hỏi vào tập – nộp tập vào cuối kì)

Câu 1: Hãy viết lệnh để tính giá trị các biểu thức sau trong chế độ gõ lệnh trực tiếp của Python:

a) $10+13$ b) $20-7$ c) $3 \times 10 - 16$ d) $12/5 + 13/6$

Câu 2: Các lệnh sau có lỗi không? Vì sao?

```
>>> 3 + * 5
```

```
>>> "Bạn là học sinh, bạn tên là "Nguyễn Việt Anh" "
```

Câu 3. Viết các lệnh in ra màn hình thông tin như sau:

a) $1 \times 3 \times 5 \times 7 = 105$

b) Bạn Hoa năm nay 16 tuổi

CHỦ ĐỀ 5: GIẢI QUYẾT VẤN ĐỀ VỚI SỰ TRỢ GIÚP CỦA MÁY TÍNH

BÀI 17: BIẾN VÀ LỆNH GÁN

A. YÊU CẦU CẦN ĐẠT

- Biết cách thiết lập biến. Phân biệt được biến và từ khóa.
- Biết sử dụng lệnh gán và thực hiện một số phép toán trên kiểu số nguyên, số thực và xâu kí tự.

B. NỘI DUNG BÀI HỌC

1. BIẾN VÀ LỆNH GÁN

- Biến là tên (định danh) của một vùng nhớ dùng để lưu trữ giá trị (dữ liệu) và giá trị đó có thể được thay đổi khi thực hiện chương trình.
- Biến trong Python được tạo ra khi thực hiện lệnh gán.
- Cú pháp của lệnh gán: <biến> = <giá trị>
- Khi thực hiện lệnh gán, <giá trị> bên phải sẽ được gán cho <biến>. Nếu biến chưa được khai báo thì nó sẽ được khởi tạo khi thực hiện câu lệnh gán.
- Biến trong Python được xác định kiểu dữ liệu tại thời điểm gán giá trị nên không cần khai báo trước kiểu dữ liệu cho biến.

Ví dụ:

```
>>> x = 5
>>> x      # x được gán 5 và có kiểu số nguyên tại đây.
5
>>> y = "Tin học 10"
>>> y      # y được gán "Tin học 10" có kiểu xâu kí tự tại đây.
'Tin học 10'
```

- Có thể thực hiện tất cả các phép toán thông thường như: +, -, *, /, ... trên các biến có cùng kiểu dữ liệu.

Ví dụ:

```
>>> x = y = 1      ← Có thể gán đồng thời nhiều biến với một giá trị.
>>> x = y + 1
>>> z = (x+y)**x   ← Phép tính lũy thừa: (x + y)x
>>> z
1
```

- Có thể gán giá trị biểu thức cho biến. Cú pháp: <biến> = <biểu thức>
- Khi thực hiện lệnh này, Python sẽ tính giá trị <biểu thức> và gán kết quả cho <biến> => mọi biến có trong <biểu thức> đều cần được xác định giá trị trước.

Ví dụ:

```
>>> x = 5          ← x là biến kiểu số nguyên có giá trị bằng 5.
>>> y = 2          ← y là biến kiểu số nguyên có giá trị bằng 2.
>>> z = x/y        ← z là biến kiểu số thực có giá trị bằng 2.5.
>>> z
2.5
```

- Tên biến thường được đặt sao cho dễ nhớ và có ý nghĩa.

Ví dụ:

```
>>> ten = "Hoài Nam"
>>> print("Xin chào",ten)
Xin chào Hoài Nam
```

- Có thể gán nhiều giá trị đồng thời cho nhiều biến. Cú pháp của lệnh gán đồng thời: <var1>, <var2>, ..., <varn> = <gt1>, <gt2>, ..., <gtn>

Ghi nhớ:

- Biến là tên của một vùng nhớ dùng để lưu trữ giá trị (dữ liệu) và giá trị đó có thể được thay đổi khi thực hiện chương trình.
- Cú pháp lệnh gán: <biến> = <biểu thức>
- Quy tắc đặt tên biến (định danh):
 - + Chỉ gồm các chữ cái tiếng Anh, các chữ số từ 0 đến 9 và kí tự gạch dưới “_”.
 - + Không bắt đầu bằng chữ số.
 - + Phân biệt chữ hoa và chữ thường.

2. CÁC PHÉP TOÁN TRÊN MỘT SỐ KIỂU DỮ LIỆU CƠ BẢN

```

Vi dụ 1. Các phép toán trên dữ liệu kiểu số.
>>> a, b = 10, 3          ← Sau lệnh này thì a = 10, b = 3.
>>> (a+b)**2 + (a-b)*10
239
>>> (a//b)*b + a%b
10
>>> c = b/2              ← Sau lệnh này, c là biến kiểu thực có giá trị 1.5.
>>> c
1.5
    
```

- Tất cả phép toán đều được thực hiện từ trái sang phải, riêng phép lũy thừa (**) thì thực hiện từ phải sang trái.
- Các phép toán cơ bản với dữ liệu kiểu số (số thực và số nguyên) trong Python là phép cộng "+", trừ "-", nhân "*", chia "/", lấy thương nguyên "//", lấy số dư "%" và phép lũy thừa "**"
- Thứ tự thực hiện các phép tính như sau: phép lũy thừa ** có ưu tiên cao nhất, sau đó là các phép toán /, *, //, %, cuối cùng là các phép toán +, -.

Ví dụ, lệnh sau :

```
>>> 3/2+4*2**4-5//2**2
```

tương đương với lệnh:

```
>>> 3/2+4 * (2**4) - 5//(2**2)
```

Chú ý. Nếu có ngoặc thì biểu thức trong ngoặc được ưu tiên thực hiện trước.

Vi dụ 2. Các phép toán với dữ liệu kiểu xâu kí tự

```
>>> s1 = "Hà Nội"
```

```
>>> s2 = "Việt Nam"
```

```
>>> s1 + s2          # Phép nối + nối hai xâu kí tự.
```

```
" Hà Nội Việt Nam"
```

```
>>> "123" * 5        # Phép * n lặp n lần xâu gốc.
```

```
"123123123123123"
```

```
>>> s*0             # Phép *n với số n ≤ 0 thì được kết quả là xâu rỗng.
```

Trong biểu thức có cả số thực và số nguyên thì kết quả sẽ có kiểu số thực

Ghi nhớ:

- Các phép toán trên dữ liệu kiểu số: +, -, *, /, //, %, **.
- Các phép toán trên dữ liệu kiểu xâu: + (nối xâu) và * (lặp)

3. TỪ KHOA

- Một tập hợp các từ tiếng Anh đặc biệt được sử dụng vào mục đích riêng của ngôn ngữ lập trình, được gọi là các từ khóa (keyword) của ngôn ngữ lập trình. Khi viết chương trình không được đặt tên biến hay các định danh trùng với từ khóa.

- Một số từ khóa trong Python phiên bản 3.x.

False	break	else	if	not	as	from
None	class	except	import	or	assert	global
True	continue	Finally	in	pass	del	lanbda
and	def	for	is	raise	elif	nonlocal
while	with	yleld	try	return		

Ghi nhớ

- Từ khóa là các từ đặc biệt tham gia vào cấu trúc của ngôn ngữ lập trình
- Không được phép đặt tên biến hay các định danh trùng với từ khóa

C. Củng cố - Vận dụng

(Học sinh trả lời các câu hỏi vào tập – nộp tập vào cuối kì)

THỰC HÀNH

Tạo và làm việc với biến, tính toán với các kiểu dữ liệu cơ bản trong Python.

Nhiệm vụ 1. Thực hiện các phép tính sau trong môi trường lập trình Python, so sánh kết quả với việc tính biểu thức toán học.

a) $(1+2+3+...+10)^3$

b) $1/2 + 1/3 + 1/4 + 1/5$

c) Thực hiện lệnh gán $x = 2$, $y = 5$ rồi tính giá trị biểu thức $(x + y)(x^2 + y^2 - 1)$

d) Thực hiện gán $a = 2$, $b = 3$, $c = 4$ rồi tính giá trị biểu thức $(a + b + c)(a + b - c)$

Hướng dẫn: Các phép tính trên có thể thực hiện trong môi trường lập trình Python như sau

```
>>> (1+2+3+4+5+6+7+8+9+10)**3
```

```
>>> x, y = 2, 5
```

```
>>> (x+y)*(x**2+y**2-1)
```

```
>>> 1/2 + 1/3 + 1/4 + 1/5
```

```
>>> a,b,c = 2,3,4
```

```
>>> (a+b+c) * (a+b-c)
```

Nhiệm vụ 2: Gán giá trị cho biến R là bán kính hình tròn rồi viết chương trình tính và in ra kết quả theo mẫu

Chu vi hình tròn là:

Diện tích hình tròn là:

Hướng dẫn: Soạn thảo chương trình sau trong môi trường lập trình Python

```
R = 4.5
```

```
Pi = 3.14
```

```
print("Chu vi hình tròn là:", 2*R*pi)
```

```
print("Diện tích hình tròn là:", pi*R*R)
```

Thực hiện chương trình và kiểm tra kết quả, so sánh với chế độ gõ lệnh trực tiếp

CHỦ ĐỀ 5: GIẢI QUYẾT VẤN ĐỀ VỚI SỰ TRỢ GIÚP CỦA MÁY TÍNH

BÀI 18: CÁC LỆNH VÀO RA ĐƠN GIẢN

A. YÊU CẦU CẦN ĐẠT

- Biết và thực hiện được một số lệnh vào ra đơn giản
- Thực hiện được một số chuyển đổi dữ liệu giữa các kiểu dữ liệu cơ bản

B. NỘI DUNG BÀI HỌC

1. CÁC CÂU LỆNH VÀO RA ĐƠN GIẢN

- Lệnh `print()` có chức năng đưa dữ liệu ra thiết bị chuẩn, thường là màn hình. Thông tin cần đưa ra có thể bao gồm một hay nhiều dữ liệu với kiểu khác nhau, cho phép cả biểu thức tính toán.
- Lệnh `input()` có chức năng nhập dữ liệu từ thiết bị vào chuẩn (thường là bàn phím). Nội dung nhập có thể là số, biểu thức hay xâu và cho kết quả là một xâu kí tự.

Cú pháp: `<biến> = input(<Dòng thông báo>)`

Ví dụ:

```
name = input("Nhập họ tên em: ")
print("Xin chào ", name)
```

Ghi nhớ:

- Các lệnh vào ra đơn giản của Python bao gồm lệnh `input()` và lệnh `print()`

2. CHUYỂN ĐỔI KIỂU DỮ LIỆU CƠ BẢN CỦA PYTHON

- Quan sát các lệnh sau để biết kiểu dữ liệu của mỗi biến.

```
>>> n,x,s = 10,1.8, "One"  ← Gán n = 10, x = 1.8, s = "One" đồng thời.
>>> n
10
>>> type(n)
<class 'int'> ← Biến n thuộc kiểu int – số nguyên.
>>> type(x)
<class 'float'> ← Biến x thuộc kiểu float – số thực.
>>> type(s)
<class 'str'> ← Biến s thuộc kiểu str – xâu kí tự.
```

- Kiểu dữ liệu logic cũng là kiểu dữ liệu cơ bản và dữ liệu kiểu này chỉ có hai giá trị là `True` (đúng) và `False` (sai).

Ví dụ dữ liệu kiểu logic là kết quả phép so sánh:

```
>>> 3 > 2
True
>>> 5 < 0
False
>>> b = 10 > 3
>>> type(b)
<class 'bool'> ← Tên kiểu dữ liệu logic là bool.
```

Ghi nhớ:

- Một số kiểu dữ liệu cơ bản của Python bao gồm: `int` (số nguyên), `float` (số thực), `str` (xâu kí tự), `bool` (logic).
- Lệnh `type()` dùng để nhận biết kiểu dữ liệu của biến trong Python.

Bài 1. Xác định kiểu và giá trị của các biểu thức sau:

- a) `"15 + 20 - 7"` b) `32 > 45`
- c) `13 != 8+5` d) `1 == 2`

- Lệnh `int()` có chức năng chuyển đổi số thực hoặc xâu chứa số nguyên thành số nguyên. Quan sát các lệnh sau:

```
>>> int(12.6)
12
>>> int("123")
123
>>> int("10.35") # Lệnh in không chuyển đổi được xâu chứa số thực
```

Traceback (most recent call last):

```
File "<pyshell#21>", line 1, in <module>
int("10.35")
```

ValueError: invalid literal for int() with base 10: "10.35"

- Lệnh float () dùng để chuyển đổi số nguyên và xâu kí tự thành số thực.

```
>>> float(8)
```

```
8.0
```

```
>>> float("10.23")
```

```
10.23
```

- Lệnh str () dùng để chuyển đổi các kiểu dữ liệu khác thành xâu kí tự.

```
>>> str(12+34)
```

```
'46'
```

```
>>> str(12.567)
```

```
'12.567'
```

```
>>> str(2>3)
```

```
'False'
```

Chú ý: Các lệnh int (), float () chỉ có thể chuyển đổi các xâu ghi giá trị số trực tiếp, không chuyển đổi xâu có công thức, ví dụ:

```
>>> int("12+45")
```

Traceback (most recent call last):

```
File "<pyshell#27>", line 1, in <module>
```

```
int("12+45")
```

ValueError: invalid literal for int() with base 10: "12+45"

Ghi nhớ

- Các lệnh int (), float (), str () có chức năng chuyển đổi dữ liệu từ các kiểu khác tương ứng về kiểu số nguyên, số thực và xâu kí tự.
- Các lệnh int (), float () không thực hiện xâu là biểu thức toán.

Bài 2.

1. Mỗi lệnh sau sẽ trả lại các giá trị nào?

a) str(150) b) int("1110") c) float("15,0")

2. Lệnh nào sau đây sẽ báo lỗi?

A. int("12,0") B. float(13+1) C. str(17,001)

- Cách nhập số nguyên, số thực:

```
<biến> = int(input())
```

```
<biến> = float(input())
```

Ví dụ:

```
>>> n = int(input("Nhập số tự nhiên: "))
```

```
Nhập số tự nhiên: 13
```

```
>>> x = float(input("Nhập số thực x: "))
```

C. Củng cố - Vận dụng

THỰC HÀNH. Nhập dữ liệu bàn phím từ lệnh input().

Nhiệm vụ 1. Viết chương trình cần nhập lần lượt ba số tự nhiên m, n, p, sau đó in ra tổng của ba số này.

Hướng dẫn. Cần thực hiện ba lệnh nhập lần lượt các số m, n, p. Chú ý cách nhập số nguyên cần dùng lệnh int() để chuyển đổi dữ liệu nhập từ bàn phím. Chương trình có thể viết như sau

```
m = int(input("Nhập số nguyên m: "))
```

```
n = int(input("Nhập số nguyên n: "))
```

```
p = int(input("Nhập số nguyên p: "))
```

```
print("Tổng ba số đã nhập là", m+n+p)
```

Nhiệm vụ 2. Viết chương trình nhập họ tên, sau đó nhập tuổi học sinh. Chương trình đưa ra thông báo, ví dụ: Bạn Nguyễn Hoà Bình 15 tuổi.

Hướng dẫn. Cần thực hiện hai lệnh nhập dữ liệu, một lệnh nhập tên học sinh, lệnh thứ hai nhập tuổi, sau đó thông báo ra màn hình. Chú ý khi nhập tuổi cần chuyển đổi dữ liệu.

```
ten = input("Nhập tên học sinh: ")
```

```
tuoi = int(input("Nhập tuổi: "))
```

```
print("Bạn", ten, tuoi, "tuổi")
```

LUYỆN TẬP

1. Những lệnh nào trong những lệnh sau sẽ bị báo lỗi?

a) int("12+45") b) float(123.56) c) float("123,5.5")

2. Vì sao khi nhập một số thực cần viết lệnh float(input())?

CHỦ ĐỀ 5: GIẢI QUYẾT VẤN ĐỀ VỚI SỰ TRỢ GIÚP CỦA MÁY TÍNH

BÀI 19: CẤU LỆNH RỄ NHÁNH IF

A. YÊU CẦU CẦN ĐẠT

- * Biết và trình bày được các phép toán với kiểu dữ liệu logic
- * Biết sử dụng được lệnh rẽ nhánh if trong lập trình

B. NỘI DUNG BÀI HỌC

1. BIỂU THỨC LOGIC

- Trong Python, biểu thức logic là biểu thức chỉ nhận giá trị True (đúng) hoặc False (sai). Biểu thức logic đơn giản nhất là các biểu thức so sánh số hoặc xâu kí tự.

- Quan sát các lệnh sau để nhận biết kiểu dữ liệu logic.

```
>>> a, b, s = 10, 2, "Number" # Gán a = 10, b = 2, s = "Number"
```

```
>>> a > 10
```

False # a > 10 là sai, b < 3 là đúng

```
>>> b < 3
```

True

```
>>> s == "number" # s và "number" là hai xâu có giá trị khác nhau
```

False

Các phép so sánh giá trị số trong Python

<	Nhỏ hơn	>	Lớn hơn	=	Bằng nhau
< =	Nhỏ hơn hoặc bằng	> =	Lớn hơn hoặc bằng	!=	Khác nhau

Chú ý: Với xâu kí tự cũng có đầy đủ các phép so sánh (sẽ học sau).

Các phép toán trên kiểu dữ liệu logic bao gồm phép and (và), or (hoặc) và not (phủ định). Bảng các phép toán logic như sau:

Phép toán and		
X	Y	X and Y
True	True	True
True	False	False
False	True	False
False	False	False

Phép toán or		
X	Y	X or Y
True	True	True
True	False	True
False	True	True
False	False	False

Phép toán not	
X	not X
True	False
False	True

Ví dụ: Cho các lệnh sau và dự đoán giá trị của các biến logic a, b, c

```
>>>x, y, z= 10, 5, 9
```

```
>>>b=x < 11 and z > 5
```

```
>>>c=x > 15 or y < 9
```

```
>>>a= not b
```

Giải thích: Ta có x = 10, z = 9 do x < 11 là đúng, z > 5 là đúng. Theo bảng phép toán **and** ta có b = x < 11 and z > 5 nhận giá trị đúng.

Ta lại có: x > 15 sai (vì x = 10) nhưng y < 9 đúng (vì y = 5). Theo bảng phép toán **or** suy ra c = x > 15 or y < 9 nhận giá trị đúng.

Cuối cùng, vì b là đúng nên a = **not** b sẽ nhận giá trị sai.

Ghi nhớ:

- Biểu thức logic là biểu thức chỉ nhận giá trị True hoặc False. Giá trị các biểu thức logic thuộc kiểu bool.
- Các phép toán trên kiểu dữ liệu logic là and (và), or (hoặc) và not (phủ định).

? Mỗi biểu thức sau có giá trị True hay False?

a) $100\%4 == 0$

b) $111//5 != 20$ or $20\%3 != 0$

2. LỆNH IF

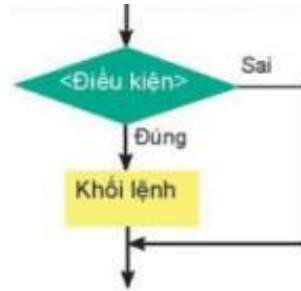
- Python cung cấp câu lệnh để mô tả cấu trúc rẽ nhánh:

+ *Câu lệnh điều kiện dạng thiếu:*

if <điều kiện>:

 <Khối lệnh>

Khi thực hiện lệnh, Python sẽ kiểm tra <điều kiện> nếu đúng thì thực hiện <khối lệnh>, ngược lại thì bỏ qua chuyển sang lệnh tiếp theo sau lệnh if.



Hình 19.2

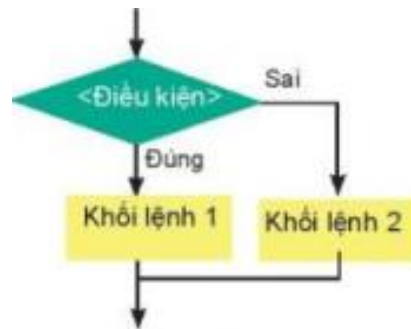
+ *Câu lệnh điều kiện dạng đủ:*

if <điều kiện>:

 <khối lệnh 1>

else:

 <khối lệnh 2>



Hình 19.3

Khi thực hiện lệnh, Python sẽ kiểm tra <điều kiện> nếu đúng thì thực hiện <khối lệnh 1>, ngược lại thì thực hiện <khối lệnh 2>.

- Ví dụ, nếu a, b là hai số đã được tạo thì lệnh sau sẽ in ra giá trị tuyệt đối của hiệu hai số.

if $a > b$:

 print(a – b)

else:

 print(b – a)

Chú ý:

- Từ khóa if và else cần viết thẳng lề trái.

- Các khối lệnh 1 và khối lệnh 2 cần viết lùi vào và thẳng hàng, mặc định là tab hay 4 dấu cách.

- Các khối lệnh trong Python đều cần viết sau dấu “:” Và lùi vào, thẳng hàng. Đây là điểm khác biệt của Python với các ngôn ngữ lập trình khác.

Ghi nhớ: Câu lệnh điều kiện if thể hiện cấu trúc rẽ nhánh trong Python. Khối lệnh rẽ nhánh của if được viết sau dấu “:”, cần viết lùi vào và thẳng hàng.

C. CƯỜNG CỐ - VẬN DỤNG

THỰC HÀNH

Các bài tập liên quan đến kiểu dữ liệu bool và lệnh if.

Nhiệm vụ 1. Viết chương trình nhập số tự nhiên n từ bàn phím. Sau đó thông báo số em đã nhập là số chẵn hay số lẻ phụ thuộc vào n là chẵn hay lẻ.

Hướng dẫn. Để kiểm tra một số tự nhiên n là chẵn hay lẻ, ta dùng phép toán lấy số dư $n\%2$. Nếu số dư bằng 0 thì n là số chẵn, ngược lại n là số lẻ. Chương trình có thể như sau:

```
n = int(input("Nhập số tự nhiên n: "))
if n%2 == 0:
    print("Số đã nhập là số chẵn.")
else:
    print("Số đã nhập là số lẻ.")
```

Nhiệm vụ 2. Giả sử giá điện sinh hoạt trong khu vực gia đình em ở được tính lũy kế theo từng tháng như sau (giá tính theo từng kWh điện tiêu thụ).

- Với mức điện tiêu thụ từ 0 đến 50 kWh, giá thành mỗi kWh là 1,578 nghìn đồng
- Với mức từ 51 đến 100, giá thành mỗi kWh là 1,734 nghìn đồng
- Từ mức 101 trở lên, giá thành mỗi kWh là 2,014 nghìn đồng.

Viết chương trình nhập số điện tiêu thụ trong tháng của gia đình em và tính số tiền điện phải trả
 Hướng dẫn. Gọi k là số kWh điện tiêu thụ của gia đình em. Khi đó theo cách tính lũy kế trên chúng ta cần tính dựa trên các điều kiện sau:

- Nếu $k \leq 50$ thì số tiền cần trả là $k \times 1,678$ nghìn đồng.
- Nếu $50 < k \leq 100$ thì số tiền cần trả là $50 \times 1,678 + (k - 50) \times 1,734$ nghìn đồng.
- Nếu $100 < k$ thì số tiền cần trả là $50 \times 1,678 + 50 \times 1,734 + (k - 100) \times 2,014$ nghìn đồng.

Chúng ta sử dụng lệnh round (t) để làm tròn số thực t. Chú ý trong máy tính dùng dấu "." để viết các số thập phân. Chương trình có thể như sau:

```
k = float(input("Nhập số kWh tiêu thụ điện nhà em: "))
if k <= 50: t=k*1.678
else:
    if k <= 100: t = 50*1.678 + (k-50)*1.734
    else: t = 50*1.678 + 50*1.734 + (k-100)*2.014
print("Số tiền điện phải trả là:",round(t), "nghìn đồng" )
```

CHỦ ĐỀ 5: GIẢI QUYẾT VẤN ĐỀ VỚI SỰ TRỢ GIÚP CỦA MÁY TÍNH

BÀI 20: CÂU LỆNH LẶP FOR

A. YÊU CẦU CẦN ĐẠT

- * Biết được ý nghĩa của vùng giá trị tạo bởi lệnh range().
- * Biết được chức năng của lện lặp for và cách dùng trong Python.

B. NỘI DUNG BÀI HỌC

1. LỆNH FOR

- Cú pháp của lệnh lặp với số lần biết trước for trong Python như sau:

for < i > **in** range(n):
 <khối lệnh lặp>

- Khi thực hiện, ở mỗi vòng lặp biến i sẽ được gán lần lượt các giá trị trong vùng giá trị của lệnh range() và thực hiện <khối lệnh lặp>

- Lệnh range(n) trả lại vùng giá trị gồm n số từ 0 đến n – 1.

Ví dụ 1. Tính tổng các số tự nhiên chẵn nhỏ hơn n, với n cho trước (n=10).

```
n = 10
S = 0
for k in range(n):
    if k%2 == 0:          # Điều kiện k là số chẵn là k%2 = 0
        S = S + k
print(S)
```

Ví dụ 2. Đếm các số nguyên nhỏ hơn n (n=20) và là bội của 3.

```
n = 20
C = 0
for k in range(n):
    if k%3 == 0:        # Điều kiện k là bội của 3 là k%3 = 0
        C = C + 1
print(C)
```

Ghi nhớ: for là lệnh lặp với số lần biết trước. Số lần lặp thường được xác định bởi vùng giá trị của lệnh range().

Câu hỏi:

? Với giá trị n cho trước, so sánh giá trị S trong đoạn chương trình sau với tổng $1+2+...+n$.

```
S = 0
for k in range(1, n+1):
    S = S + k
```

2. LỆNH RANGE

- Lệnh tạo vùng giá trị range() có dạng như sau :

+ range(stop) trả lại vùng giá trị từ 0 đến stop – 1.

+ range(start, stop) trả lại vùng giá trị từ start đến stop – 1.

- Ví dụ:

+ range(n) cho vùng gồm các số 0, 1, ..., n – 1.

+ range(1, n+1) cho vùng gồm các số 1, 2, ..., n.

+ range(0, 99) cho vùng giá trị gồm các số 0, 1, 2, ..., 98.

+ range(100,1) cho vùng rỗng.

Ghi nhớ:

- Lệnh tạo vùng giá trị có cú pháp range(start, stop) trả lại vùng giá trị gồm các số nguyên liên tiếp từ start đến stop -1.

? Hãy biểu diễn các dãy sau đây bằng lệnh range().

- a) 1,2,3,..., 50 b) 5, 6, 7, 8, 9, 10 c) 0,1 d) 10

C. CŨNG CỐ - VẬN DỤNG

THỰC HÀNH. Lệnh lặp for và lệnh range ()

Nhiệm vụ 1. Nhập số tự nhiên n từ bàn phím và in ra màn hình dãy các ước số của n theo chiều ngang màn hình.

Ví dụ nếu n=0 thì chương trình sẽ in ra dãy số 1,2,5,10.

Hướng dẫn. Các ước số của n là các số tự nhiên k thỏa mãn: $n\%k=0$. Muốn in các số trên một hàng ngang cần dùng thêm tham số `end = ""` trong lệnh `print ()`.

Chương trình có thể như sau:

```
n = int ( input (" Nhập số tự nhiên n: "))
for k in range (1, n+1):
    if n%k == 0:
        print (k, end = "")
```

Nhiệm vụ 2. Nhập số tự nhiên n từ bàn phím và đếm số các ước số thực sự của n . Ước số thực sự của n là số tự nhiên $k < n$ và là ước của n .

Hướng dẫn. Tương tự như chương trình ở nhiệm vụ 1, điểm khác là cần đếm số các ước số này và không tính n . Tạo một biến có tên `count` để đếm số các ước số thực sự của n .

```
n = int ( input (" Nhập số tự nhiên n: "))
count = 0
for k in range (1, n):
    if n%k == 0:
        count = count + 1
print (count)
```

CHỦ ĐỀ 5: GIẢI QUYẾT VẤN ĐỀ VỚI SỰ TRỢ GIÚP CỦA MÁY TÍNH

BÀI 21: CÂU LỆNH LẶP WHILE

A. YÊU CẦU CẦN ĐẠT

- * Biết và thực hành giải các bài toán sử dụng lệnh lặp while với số lần không biết trước
- * Biết ba cấu trúc lập trình cơ bản: tuần tự, rẽ nhánh, lặp,...

B. NỘI DUNG BÀI HỌC

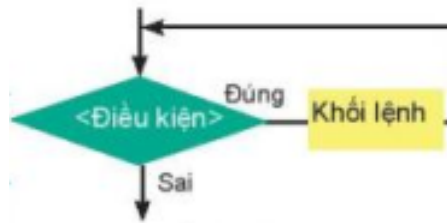
1. LỆNH WHILE

- Lệnh lặp **while** thực hiện khối lệnh với số lần lặp không biết trước. Khối lệnh lặp được thực hiện cho đến khi <điều kiện> = False

Cú pháp của lệnh **while** như sau:

```
while <điều kiện>:
    <khối lệnh lặp>
```

Chú ý: sau dấu ":" khối lệnh lặp cần được viết lùi vào và thẳng hàng. Mặc định các lệnh sẽ lùi vào 1 tab hoặc 4 dấu cách.



Hình 21.1

Trong đó <điều kiện> là biểu thức logic. Khi thực hiện lệnh, Python sẽ kiểm tra <điều kiện>, nếu đúng thì thực hiện khối lệnh lặp, nếu sai thì kết thúc lệnh **while**

Vi dụ 1. Quan sát đoạn chương trình sau và cho biết S là giá trị của biểu thức toán học nào?

```
S = 0
k = 1
while k*k < 100:
    S = S + k*k
    k = k + 1
```

Giải thích: Đoạn chương trình tính tổng $1^2 + 2^2 + \dots + k^2$ với điều kiện $k^2 < 100$. Vậy S chính là tổng bình phương các số tự nhiên nhỏ hơn 10.

Vi dụ 2. Thực hiện các lệnh sau. Kết quả sẽ in ra những số nào?

```
>>> k = 2
>>> while k < 50:
    print(k, end = " ")
    k = k + 3
```

Giải thích: Vòng lặp while sẽ dừng khi k vượt quá 50. Bắt đầu vòng lặp, k = 2. Sau mỗi bước lặp k tăng lên 3 đơn vị. Do vậy, kết quả sẽ phải in ra dãy sau:

2 5 8 11 14 17 20 23 26 29 32 35 38 41 44 47

Ghi nhớ: while là lệnh lặp với số lần không biết trước. Số lần lặp của lệnh while phụ thuộc vào điều kiện của lệnh

Câu hỏi:

1. Lệnh while kiểm tra điều kiện trước hay sau khi thực hiện khối lệnh lặp?
2. Viết đoạn chương trình tính tổng $2 + 4 + \dots + 100$ sử dụng lệnh while

Lưu ý:

1. Vì lệnh while không biết trước số lần lặp, mà phụ thuộc vào điều kiện. Do đó, cần chú ý đến điều kiện của lệnh while để tránh bị lặp vô hạn.
2. Trong trường hợp nếu muốn dừng và thoát ngay khỏi vòng lặp while hoặc for có thể dùng lệnh break

```
>>> for k in range(10):
    print(k, end = " ")
    if k == 5: break
0 1 2 3 4 5
```

2. CẤU TRÚC LẬP TRÌNH

- Với việc sử dụng câu lệnh if và câu lệnh lặp ta thấy một chương trình Python nói chung có thể chia ra thành các khối lệnh sau:

- + Khối gồm các lệnh được thực hiện theo trình tự từ trên xuống dưới. Khối này tương ứng với cấu trúc tuần tự trong chương trình và được thể hiện bằng các câu lệnh như gán giá trị, nhập/xuất dữ liệu, ...
- + Khối các câu lệnh chỉ được thực hiện tùy thuộc vào điều kiện nào đó là đúng hay sai. Khối lệnh này tương ứng với cấu trúc rẽ nhánh và được thể hiện bằng câu lệnh if
- + Khối các câu lệnh được thực hiện lặp đi lặp lại tùy theo điều kiện nào đó vẫn còn đúng hay sai. Khối lệnh này tương ứng với cấu trúc lặp và được thể hiện bằng các câu lệnh lặp for, while

Ghi nhớ: Ba cấu trúc lập trình cơ bản của các ngôn ngữ lập trình bậc cao gồm: cấu trúc tuần tự, cấu trúc rẽ nhánh, cấu trúc lặp.

C. CƯỜNG CỐ - VẬN DỤNG

THỰC HÀNH

Nhiệm vụ 1. Viết chương trình in toàn bộ dãy các số tự nhiên từ 1 đến 100 trên một hàng ngang

Hướng dẫn: Mở Python và nhập chương trình sau:

```
k = 0
```

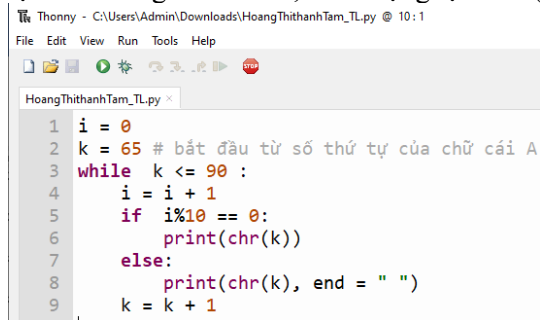
```
while k < 100 :
```

```
    k = k + 1
```

```
    print(k, end = " ")
```

Nhiệm vụ 2. Viết chương trình in ra màn hình dãy các chữ cái tiếng Anh từ “A” đến “Z” theo ba hàng ngang trên màn hình, hai hàng ngang đầu có 10 chữ cái, hàng thứ ba có 6 chữ cái.

Hướng dẫn: Do các chữ cái tiếng Anh từ A đến Z chiếm các vị trí từ 65 đến 90 trong bảng mã ASCII. Với số thứ tự k của bảng mã ASCII, ta sử dụng lệnh chr(k) trả lại kí tự tương ứng trong bảng mã này



```

1 i = 0
2 k = 65 # bắt đầu từ số thứ tự của chữ cái A
3 while k <= 90 :
4     i = i + 1
5     if i%10 == 0:
6         print(chr(k))
7     else:
8         print(chr(k), end = " ")
9     k = k + 1

```

CHỦ ĐỀ 5: GIẢI QUYẾT VẤN ĐỀ VỚI SỰ TRỢ GIÚP CỦA MÁY TÍNH

BÀI 22: KIỂU DỮ LIỆU DANH SÁCH

A. YÊU CẦU CẦN ĐẠT

- * Biết được kiểu dữ liệu danh sách (list), cách khởi tạo và truy cập từng phần tử của danh sách
- * Biết và thực hiện được cách duyệt các phần tử của danh sách bằng lệnh for
- * Thực hành được một số phương thức đơn giản trên dữ liệu danh sách

B. NỘI DUNG BÀI HỌC

1. KIỂU DỮ LIỆU DANH SÁCH

Ví dụ 1. Quan sát các lệnh sau để tìm hiểu kiểu dữ liệu danh sách.

```
>>> A = [1,2,3,4,5]
```

```
>>> B [1.5, 2, "Python", "List", 0]
```

```
>>> A[0]
```

```
1
```

```
>>> B[2]
```

```
"Python"
```

⇒ Có thể truy cập từng phần tử của danh sách thông qua chỉ số. Chỉ số của list đánh số từ 0

- Khởi tạo kiểu dữ liệu danh sách trong Python:

$$\langle \text{tên list} \rangle = [\langle v_1 \rangle, \langle v_2 \rangle, \dots, \langle v_n \rangle]$$

- Trong đó:

+ các giá trị $\langle v_k \rangle$ có thể có kiểu dữ liệu khác nhau (số nguyên, số thực, xâu kí tự....).

- Danh sách của Python có thể gồm các phần tử có kiểu dữ liệu khác nhau.

Ví dụ 2. Quan sát các lệnh sau để biết cách thay đổi hoặc xoá phần tử của danh sách

```
>>> A = [1,2,3,4,5]
```

```
>>> len(A) # tính độ dài danh sách
```

```
5
```

```
>>> A[1] = "One"
```

- Thay đổi giá trị của từng phần tử bằng lệnh gán

```
>>> A
```

```
[1, 'One', 3, 4, 5]
```

- Lệnh del để xoá một phần tử của danh sách

```
>>> del (A[4])
```

```
>>> A
```

```
[1, 'One', 3, 4]
```

Ví dụ 3. Quan sát các lệnh sau để biết cách tạo danh sách rỗng (có độ dài 0) và các phép toán ghép danh sách (phép +).

```
>>> a = [ ]
```

```
>>> len(a)
```

```
0
```

```
>>> [1,2] + [3,4,5,6] # ghép hai danh sách
```

```
[1, 2, 3, 4, 5, 6]
```

Ghi nhớ:

- List là kiểu dữ liệu danh sách (dãy, mảng) trong Python. Tạo list bằng lệnh gán với các phần tử trong cặp dấu ngoặc []. Các phần tử của danh sách có thể có các kiểu dữ liệu khác nhau. Truy cập hoặc thay đổi giá trị của từng phần tử thông qua chỉ số: < danh sách >[< chỉ số >]

- Chỉ số của danh sách bắt đầu từ 0 đến $\text{len}() - 1$, trong đó $\text{len}()$ là lệnh tính độ dài danh sách.

Câu hỏi

1. Cho danh sách $A = [1, 0, \text{"One"}, 9, 15, \text{"Two"}, \text{True}, \text{False}]$. Hãy cho biết giá trị các phần tử:

a) $A[0]$ b) $A[2]$ c) $A[7]$ d) $A[\text{len}(A)]$

2. Giả sử A là một danh sách các số, mỗi lệnh sau thực hiện gì?

a) $A = A + [10]$ b) $\text{del}(A[0])$ c) $A = [100] + A$ d) $A = A[1] * 25$

2. DUỖYỆT CÁC PHẦN TỬ CỦA DANH SÁCH

Ví dụ 1. Duyệt và in ra từng phần tử của danh sách.

```
>>> A = [1,2,3,4,5]
```



```
>>> for i in range(len(A)):
    print(A[i], end = " ")
```

Vi dụ 2. Duyệt và in một phần của danh sách.

```
>>> A = [3, 2, 1, 5, 6, 10, 7, 12, 18]
>>> for i in range(2,5):
    print(A[i], end = " ")
```

1 5 6

Ghi nhớ: Có thể duyệt lần lượt các phần tử của danh sách bằng lệnh for kết hợp với vùng giá trị của lệnh range().

Câu hỏi:

1. Giải thích các lệnh ở mỗi câu sau thực hiện công việc gì?

a)

```
>>> S = 0
>>> for i in range(len(A)):
    if A[i] > 0: S = S + A[i]
>>> print(S)
```

b)

```
>>> C = 0
>>> for i in range(len(A)):
    if A[i] > 0: C = C + 1
>>> print(C)
```

Cho dãy các số nguyên A, viết chương trình in ra các số chẵn của A.

3. THÊM PHẦN TỬ VÀO DANH SÁCH

Python có những lệnh đặc biệt để thêm phần tử vào một danh sách. Các lệnh này được thiết kế riêng cho kiểu dữ liệu danh sách và còn được gọi là phương thức (method) của danh sách.

Ví dụ. Thêm phần tử vào cuối danh sách

```
>>> A = [1,2]
>>> A.append(10)
>>> A
[1, 2, 10]
```

Ghi nhớ

- Python có một số lệnh dành riêng (phương thức) cho dữ liệu kiểu danh sách. Cú pháp các lệnh đó như sau: <danh sách>.<phương thức>
- Lệnh thêm phần tử vào cuối danh sách là <danh sách>.append()

Câu hỏi:

1. Sau khi thêm một phần tử vào danh sách A bằng lệnh append() thì độ dài danh sách A thay đổi như thế nào?

2. Danh sách A sẽ như thế nào sau các lệnh sau?

```
>>> A = [2,4,10,1,0]
>>> A.append(100)
>>> del(A[1])
```

C. Củng cố - Vận dụng

THỰC HÀNH

Khởi tạo, nhập dữ liệu, thêm phần tử cho danh sách

Nhiệm vụ 1. Nhập số n từ bàn phím, sau đó nhập danh sách n tên các bạn lớp em và in ra danh sách các tên đó, mỗi tên trên một dòng.

Hướng dẫn. Chương trình yêu cầu nhập số tự nhiên n, sau đó nhập từ tên trong danh sách, dùng phương thức append() để đưa dần vào danh sách.

Chú ý. Vì vùng giá trị của lệnh range(n) bắt đầu từ 0 nên trong thông báo nhập cần viết là str(i+1) để bắt đầu từ 1.

Chương trình có thể như sau:

```

HoangThiThanhTam_TL.py x
1 dslop = []
2 n = int(input("Nhập số học sinh trong lớp: "))
3 for i in range(n):
4     name = input("Nhập họ tên học sinh thứ "+str(i+1)+" : ")
5     dslop.append(name)
6 print("Danh sách học sinh đã nhập: ")
7 for i in range(len(dslop)):
8     print(dslop[i])
9
Shell x
Nhập số học sinh trong lớp: 5
Nhập họ tên học sinh thứ 1: Hoàng Thị Thanh Tâm
Nhập họ tên học sinh thứ 2: Trần Quang Anh
Nhập họ tên học sinh thứ 3: Nguyễn Thị Hồng
Nhập họ tên học sinh thứ 4: Lê Hải Nam
Nhập họ tên học sinh thứ 5: Mai Trung Hiếu
Danh sách học sinh đã nhập:
Hoàng Thị Thanh Tâm
Trần Quang Anh
Nguyễn Thị Hồng
Lê Hải Nam
Mai Trung Hiếu
    
```

Nhiệm vụ 2. Nhập một dãy số từ bàn phím. Tính tổng, trung bình của dãy và in dãy số trên một hàng ngang
Hướng dẫn. Tương tự nhiệm vụ 1, chỉ khác là nhập số nguyên nên dùng lệnh int() để chuyển đổi dữ liệu.

```

HoangThiThanhTam_TL.py x
1 A = []
2 T = 0
3 n = int(input("Nhập số tự nhiên n: "))
4 for i in range(n):
5     num = int(input("Nhập số thứ "+str(i+1)+" : "))
6     A.append(num)
7     T = T + num
8 print("Dãy số đã nhập: ")
9 for i in range(n):
10    print(A[i],end = " ")
11 print()
12 print("Tổng: ",T)
13 print("Trung bình: ",T/n)
    
```

CHỦ ĐỀ 5: GIẢI QUYẾT VẤN ĐỀ VỚI SỰ TRỢ GIÚP CỦA MÁY TÍNH

BÀI 23: MỘT SỐ LỆNH LÀM VIỆC VỚI DỮ LIỆU DANH SÁCH

A. YÊU CẦU CẦN ĐẠT

- * Biết cách duyệt danh sách bằng toán tử **In**
- * Biết và thực hiện được một số phương thức thường dùng với danh sách

B. NỘI DUNG BÀI HỌC

1. Duyệt danh sách với toán tử **IN**

Ví dụ 1. Dùng toán tử **in** để kiểm tra một giá trị có nằm trong danh sách hay không

```
>>> A = [1, 2, 3, 4, 5]
>>> 2 in A ← Số nguyên 2 nằm trong dãy A kết quả trả lại True.
True
>>> 10 in A ← Số 10 không nằm trong dãy A kết quả trả lại False
False
```

Ví dụ 2. Sử dụng toán tử **in** để duyệt từng phần tử của danh sách.

```
>>> A = [10, 11, 12, 13, 14, 15]
>>> for k in A ← khi thực hiện lệnh này, biến k sẽ lần lượt nhận các giá trị từ dãy A.
print(k, end = " ")
10 11 12 13 14 15
```

Ghi nhớ

- Tính toán tử **in** dùng để kiểm tra một phần tử có nằm trong danh sách đã cho không. Kết quả trả lại **True** (Đúng) hoặc **False** (Sai). <giá trị> **in** <danh sách>
- Có thể duyệt nhanh từng phần tử của danh sách bằng toán tử **in** và lệnh **for** mà không cần sử dụng lệnh **range()**.

2. MỘT SỐ LỆNH LÀM VIỆC VỚI DANH SÁCH

Ví dụ 1. Lệnh **clear()** xóa toàn bộ một danh sách

```
>> A = [1, 2, 3, 4, 5]
>> A.clear()
Sau khi thực hiện lệnh clear() danh sách gốc trở thành rỗng
>> A
[]
```

Ví dụ 2. Lệnh **remove(value)** sẽ xoá phần tử đầu tiên của danh sách có giá trị **value**. Nếu không có phần tử nào như vậy thì sẽ báo lỗi

```
>> A = [1, 2, 3, 4, 5]
>> A.remove(1)
>>> A
[2, 3, 4, 5]
>>> A.remove(10) # Lệnh lỗi vì giá trị không có trong danh sách
```

Ví dụ 3. Lệnh **insert(index, value)** có chức năng chèn giá trị **value** vào danh sách tại vị trí **index**

```
>> A = [1, 2, 6, 10]
>> A.insert(2, 5)
>>> A
[1, 2, 5, 6, 10]
- Chú ý: nếu k nằm ngoài phạm vi chỉ số của danh sách thì lệnh vẫn có tác dụng nếu: index < -len() thì chèn vào đầu danh sách, nếu index > len() thì chèn vào cuối danh sách.
```

```
>> A = []
>> A.insert(-10, 1)
>>> A.insert(100, 2)
>>> A
[1, 2]
```

Một số lệnh làm việc với dữ liệu danh sách:

A.append(x)	Bổ sung phần tử x vào cuối danh sách A
A.insert(k, x)	Chèn phần tử x vào vị trí k của danh sách A
A.clear()	Xóa toàn bộ dữ liệu của danh sách A
A.remove(x)	Xóa phần tử x từ danh sách

C. Củng cố - Vận dụng

Thực hành

Các lệnh làm việc với dữ liệu kiểu danh sách

Nhiệm vụ 1. Nhập số n từ bàn phím, sau đó nhập danh sách n tên học sinh trong lớp và in ra danh sách học sinh này, mỗi tên học sinh trên một dòng. Yêu cầu danh sách được in ra theo thứ tự ngược lại thứ tự đã nhập Hướng dẫn. Chương trình sẽ yêu cầu nhập số tự nhiên n, sau đó sẽ lần lượt yêu cầu nhập n tên học sinh. Tuy nhiên do yêu cầu in danh sách học sinh theo thứ tự ngược lại so với thứ tự nhập nên cần dùng lệnh insert() để chèn tên học sinh được nhập vào đầu danh sách. Chương trình có thể như sau:

```
HoangThiThanhTam_TL.py x
1 dsLop = []
2 n = int(input("Nhập số học sinh trong lớp: "))
3 for i in range(n):
4     name = input("Nhập tên học sinh thứ " + str(i+1) + ": ")
5     dsLop.insert(0, name)
6 print("Danh sách học sinh đã nhập:")
7 for name in dsLop:
8     print(name)
9
```

Nhiệm vụ 2. Cho trước dãy số A. Viết chương trình xoá đi các phần tử có giá trị nhỏ hơn 0 từ A Hướng dẫn. Duyệt từng phần tử của dãy số A, kiểm tra nếu phần tử này nhỏ hơn 0 thì xoá đi Dùng lệnh remove() để duyệt từng phần tử của A

```
HoangThiThanhTam_TL.py x
1 A = [0, 1, -3, -10, 5, 9, -20, 55]
2 i = 0
3 while i < len(A):
4     if A[i] < 0:
5         A.remove(A[i])
6     else:
7         i = i + 1
8 print(A)
9
```

Nhiệm vụ 3. Cho trước dãy số A. Viết phương trình tìm và chỉ ra vị trí đầu tiên của dãy số A mà ba số hạng liên tiếp có giá trị là 1, 2, 3. Nếu tìm thấy thì thông báo vị trí tìm thấy, nếu không thì thông báo “Không tìm thấy mẫu” Hướng dẫn. Soạn thảo chương trình sau rồi thực hiện và kiểm tra tính đúng đắn của chương trình.

```
HoangThiThanhTam_TL.py x
1 A = [0, 4, 0, 1, 2, 3, 8, 9, 0, 1, 2, 3, 17, -16, 0, 1, 2]
2 p = [1, 2, 3]
3 pkq = -1
4 i = 0
5 while i < len(A) - 3 and pkq == -1:
6     if A[i] == p[0] and A[i+1] == p[1] and A[i+2] == p[2]:
7         pkq = i
8     else:
9         i = i + 1
10 if pkq >= 0:
11     print("Tìm thấy mẫu ", p, " tại vị trí", pkq)
12 else:
13     print("Không tìm thấy mẫu", p)
14
```

CHỦ ĐỀ 5: GIẢI QUYẾT VẤN ĐỀ VỚI SỰ TRỢ GIÚP CỦA MÁY TÍNH

BÀI 24: XÂU KÝ TỰ

A. YÊU CẦU CẦN ĐẠT

- * Hiểu được xâu ký tự là kiểu dữ liệu cơ bản của Python
- * Biết và thực hiện được lệnh for để xử lý xâu ký tự

B. NỘI DUNG BÀI HỌC

1. XÂU LÀ MỘT DÃY KÝ TỰ

Ví dụ 1. Xâu ký tự và cách truy cập đến từng ký tự của xâu

```
>>> s = "Hoàng Thị Thanh Tâm"
>>> len(s)
19
>>> s[0]
'H'
>>> s[10]
'T'
```

- Một xâu ký tự được hiểu là một dãy các ký tự. Tương tự danh sách, ta có thể truy cập từng ký tự của xâu thông qua chỉ số, chỉ số bắt đầu từ 0.

Ví dụ 2. Quan sát các lệnh sau để thấy sự khác nhau giữa xâu và danh sách

```
>>> d=["t","â","m"]
>>> d[0]="T"
>>> s="tâm"
>>> s[0]="T"
Traceback (most recent call last):
  File "<pyshell>", line 1, in <module>
TypeError: 'str' object does not support item assignment
```

⇒ Báo lỗi

- Python không cho phép thay đổi từng ký tự của một xâu. Điều này khác với danh sách.
- Python không có kiểu dữ liệu ký tự. Ký tự chính là xâu có độ dài 1. Xâu rỗng được định nghĩa như sau: `empty = ""`

Ghi nhớ: Xâu ký tự trong Python là dãy các ký tự Unicode. Xâu có thể được coi là danh sách các ký tự nhưng không thay đổi từng ký tự của xâu. Truy cập từng ký tự của xâu qua chỉ số, chỉ số từ 0 đến độ dài `len() - 1`.

2. LỆNH DUYỆT KÝ TỰ CỦA XÂU

- Cách thứ nhất, biến `i` lần lượt chạy theo chỉ số của xâu ký tự `s`, từ 0 đến `len(s) - 1`. Ký tự tại chỉ số `i` là `s[i]`.
- Cách duyệt thứ hai duyệt theo từng ký tự của xâu `s`. Biến `ch` sẽ được gán lần lượt các ký tự của xâu `s` từ đầu đến cuối.

Chú ý: Từ khoá `in`, tùy trường hợp cụ thể, hoặc là toán tử logic dùng để kiểm tra một giá trị có mặt hay không trong một vùng giá trị/danh sách/xâu, hoặc để chọn lần lượt từng phần tử trong một vùng giá trị/danh sách/xâu.

```
>>> "a" in "abcd"
```

True

```
>>> "abc" in "abcd"
```

True

Ghi nhớ

- Có thể duyệt các ký tự của xâu bằng lệnh `for` tương tự với danh sách. `s1 in s2` trả lại giá trị True nếu `s1` là xâu con của `s2`

Câu hỏi

1. Sau khi thực hiện các lệnh sau, biến `skq` sẽ có giá trị bao nhiêu?

```
>>> s = "81723"
```

```
>>> skq = ""
```

```
>>> for ch in s:
```

```
    if int(ch) % 2 != 0:
```

```
        skq = skq + ch
```

2. Cho $s1 = \text{"abc"}$, $s2 = \text{"ababcabca"}$. Các biểu thức logic sau cho kết quả là đúng hay sai?

a) $s1 \text{ in } s2$

b) $s1 + s1 \text{ in } s2$

c) $\text{"abcabca" in } s2$

d) $\text{"abc123" in } s2$

C. Củng cố - Vận dụng

THỰC HÀNH

Các lệnh cơ bản làm việc với chuỗi ký tự

Nhiệm vụ 1. Viết chương trình nhập số tự nhiên n là số học sinh, sau đó nhập họ và tên học sinh. Lưu họ và tên học sinh vào một danh sách. In danh sách ra màn hình, mỗi họ tên trên một dòng.

Hướng dẫn. Chương trình có thể như sau:

```
HoangThiThanhTam_TL.py ×
1 n = int(input("Nhập số học sinh trong lớp: "))
2 ds_lop = []
3 for i in range(n):
4     hoten = input("Nhập họ tên học sinh thứ " + str(i+1)+": ")
5     ds_lop.append(hoten)
6 print("Danh sách lớp học:")
7 for i in range(n):
8     print(ds_lop[i])
```

Nhiệm vụ 2. Nhập một chuỗi ký tự S từ bàn phím rồi kiểm tra xem chuỗi S có chứa chuỗi con "10" không.

Hướng dẫn. Cách 1. Nếu chuỗi S chứa chuỗi con "10" thì sẽ có chỉ số k mà $S[k] = \text{"1"}$ và $S[k+1] = \text{"0"}$. Cách 2. Dùng toán tử in để kiểm tra chuỗi "10" có là chuỗi con của S .

Cách 1: Duyệt ký tự của chuỗi theo chỉ số.

```
HoangThiThanhTam_TL.py ×
1 S = input("Nhập chuỗi ký tự bất kì: ")
2 kq = False
3 for i in range(len(S)-1):
4     if S[i] == "1" and S[i+1] == "0" :
5         kq = True
6         break
7 if kq:
8     print("Chuỗi gốc có chứa chuỗi '10'")
9 else:
10    print("Chuỗi gốc không chứa chuỗi '10'")
11
```

Cách 2: Sử dụng toán tử in.

```
HoangThiThanhTam_TL.py ×
1 S = input("Nhập chuỗi ký tự bất kì: ")
2 s10 = "10"
3 if s10 in S:
4     print("Chuỗi gốc có chứa chuỗi '10'")
5 else:
6     print("Chuỗi gốc không chứa chuỗi '10'")
7
```

CHỦ ĐỀ 5: GIẢI QUYẾT VẤN ĐỀ VỚI SỰ TRỢ GIÚP CỦA MÁY TÍNH

BÀI 25: MỘT SỐ LỆNH LÀM VIỆC VỚI XÂU KÝ TỰ

A. YÊU CẦU CẦN ĐẠT

* Biết và thực hiện được một số lệnh thường dùng với chuỗi ký tự

B. NỘI DUNG BÀI HỌC

1. XÂU CON VÀ LỆNH TÌM VỊ TRÍ XÂU CON

Ví dụ 1: Dùng toán tử in để kiểm tra một chuỗi có là chuỗi con của chuỗi khác không.

```
>>> "abc" in "123abc"
```

```
True
```

```
>>> "010" in "1101"
```

```
False
```

- Biểu thức kiểm tra <chuỗi 1> nằm trong <chuỗi 2> là: <chuỗi 1> in <chuỗi 2>

Nếu đúng thì trả lại giá trị True, nếu sai trả lại giá trị False.

Ví dụ 2. Lệnh find () tìm vị trí xuất hiện của một chuỗi trong chuỗi khác.

```
>>> s = "ab bc cd 123 456 00"
```

```
>>> s.find("b")
```

1 □ Vị trí xuất hiện đầu tiên của "b" trong chuỗi s là chỉ số 1

```
>>> s.find("12")
```

9 □ Vị trí tìm thấy đầu tiên của "12" trong chuỗi s chỉ là số 9.

```
>>> s.find("AB")
```

-1 □ Không tìm thấy chuỗi "AB" trong chuỗi s nên trả về -1

- Một số lệnh đặc biệt dành riêng cho chuỗi ký tự (phương thức). Cách thực hiện phương thức là:

<chuỗi>. <phương thức>

- Cú pháp đơn của lệnh find (): <chuỗi mẹ>. find (<chuỗi con>)

Lệnh sẽ tìm vị trí đầu tiên của chuỗi con trong chuỗi mẹ và trả về vị trí đó. Nếu không tìm thấy thì trả về -1.

- Cú pháp đầy đủ của lệnh find (): <chuỗi mẹ>. find (<chuỗi con>, start)

Lệnh sẽ tìm chuỗi con bắt đầu từ vị trí start

Ví dụ 3

```
>>> sub = "Đà Nẵng"
```

```
>>> s = "Hà Nội – Đà Nẵng – Hồ Chí Minh"
```

```
>>> s.find(sub)
```

```
9
```

```
>>> s.find(sub,10)
```

```
-1
```

Ghi nhớ

Để tìm một chuỗi trong một chuỗi khác có thể dùng toán tử in hoặc lệnh find (). Lệnh find () trả về vị trí của chuỗi con trong chuỗi mẹ.

2. MỘT SỐ LỆNH THƯỜNG DÙNG VỚI XÂU KÝ TỰ:

Ví dụ 1: Lệnh split () tách một chuỗi thành danh sách các từ.

```
>>> s = "Tiên học lễ hậu học văn"
```

```
>>> s.split() #Tách chuỗi dùng dấu cách để phân biệt tách.
```

```
["Tiên", "học", "lễ", "hậu", "học", "văn"]
```

```
>>> st = "0, 1, 2, 3, 4, 5, 6, 10"
```

```
>>> s.split(",") #Tách chuỗi dùng dấu "," để phân biệt tách
```

```
["0", "1", "2", "3", "4", "5", "6", "10"]
```

- Cú pháp của lệnh split() **<chuỗi mẹ>.split(<ký tự cách>)**

Ví dụ 2. Lệnh join() nối danh sách gồm các từ thành một chuỗi.

```
>>> A = ['Tiên', 'học', 'lễ', 'hậu', 'học', 'văn']
```

```
>>> "".join(A) # Lệnh join() này sẽ nối các phần tử của danh sách A bởi dấu cách.
```

```
'Tiên học lễ hậu học văn'
```

```
>>> B = ['0', '1', '2', '3', '4', '5', '6', '10']
```

```
>>> ",".join(B) # Lệnh join() này sẽ nối các phần tử của danh sách B bởi dấu ",".
```

```
'0,1,2,3,4,5,6,10'
```

- Lệnh join() có tác dụng ngược với lệnh split(). Có chức năng nối các phần tử (là chuỗi) của một danh sách thành

một chuỗi. Cú pháp của lệnh join() là:

“kí tự nối”. join(< danh sách >)

Ghi chú: Python có các lệnh đặc biệt để xử lý chuỗi là split() dùng để tách chuỗi thành danh sách và lệnh join() dùng để nối danh sách các chuỗi thành một chuỗi.

Câu hỏi:

? Cho chuỗi kí tự: “gà,vịt,chó,lợn,ngựa,cá”. Em hãy trình bày cách làm để xóa các dấu,” và thay thế bằng dấu “ ” trong chuỗi này.

C. Củng cố - vận dụng

THỰC HÀNH

Một số bài toán liên quan đến chuỗi kí tự.

Nhiệm vụ 1. Viết chương trình nhập nhiều số nguyên từ bàn phím, các số cách nhau bởi dấu cách. Khi nhập xong thông báo số lượng các số đã nhập và in các số này thành hàng ngang.

Hướng dẫn. Dữ liệu nhập vào là một chuỗi. Dùng lệnh split() để tách thành danh sách. Chuyển các phần tử danh sách này thành số và in ra màn hình.

```

Thonny - C:\Users\Admin\Downloads\HoangThiThanhTam_TL.py @ 2:17
File Edit View Run Tools Help
HoangThiThanhTam_TL.py x
1 s = input("Nhập các số nguyên cách nhau bởi dấu cách:")
2 sline = s.split()
3 n = len(sline)
4 nline = []
5 for x in sline:
6     nline.append(int(x))
7 print("Bạn đã nhập",n,"số")
8 for k in nline:
9     print(k, end = " ")

```

Nhiệm vụ 2. Viết chương trình nhập một chuỗi kí tự có thể có nhiều dấu cách giữa các từ. Sau đó chỉnh sửa chuỗi kí tự đó sao cho giữa các từ chỉ có một dấu cách. In chuỗi kết quả ra màn hình.

Hướng dẫn. Chuyển chuỗi kí tự ban đầu thành danh sách các từ đơn bằng lệnh split(), sau đó nối các từ đơn này bằng lệnh join().

```

Thonny - C:\Users\Admin\Downloads\HoangThiThanhTam_TL.py @ 5:1
File Edit View Run Tools Help
HoangThiThanhTam_TL.py x
1 s = input ("Nhập đoạn văn bản: \n")
2 sline = s.split()
3 skq = " ".join(sline)
4 print(skq)
5

```

Nhiệm vụ 3. Viết chương trình nhập số tự nhiên n, rồi nhập họ tên của n học sinh. Sau đó in ra danh sách tên học sinh theo hai cột, cột 1 là tên, cột 2 là họ đệm.

Hướng dẫn. Họ tên ban đầu tách ra thành tên và họ đệm bằng lệnh split(). Các tên được đưa vào danh sách ten, các họ đệm được đưa vào danh sách hodem. Sau đó in ra danh sách theo yêu cầu.

```

Thonny - C:\Users\Admin\Downloads\HoangThiThanhTam_TL.py @ 5:42
File Edit View Run Tools Help
HoangThiThanhTam_TL.py x
1 n = int (input ("Nhập số học sinh trong lớp: "))
2 ten = [ ]
3 hodem = [ ]
4 for i in range (n):
5     s = input ("Nhập họ tên học sinh thứ "+str(1+i)+": ")
6     sline = s.split()
7     m = len(sline) -1
8     ten.append (sline[m])
9     del sline[m]
10    hodem.append (" ".join(sline))
11 print ("Danh sách học sinh:")
12 for i in range(n):
13    print(ten[i], hodem[i])
14

```


CHỦ ĐỀ 5: GIẢI QUYẾT VẤN ĐỀ VỚI SỰ TRỢ GIÚP CỦA MÁY TÍNH

BÀI 26: HÀM TRONG PYTHON

A. YÊU CẦU CẦN ĐẠT

- * Biết được chương trình con là hàm
- * Biết cách tạo hàm

B. NỘI DUNG BÀI HỌC

1. Một số hàm thiết kế sẵn của Python

Ví dụ: Ta có các lệnh như sau

```
# lệnh in chuỗi kí tự "Thời khóa biểu" trong dấu ngoặc ra màn hình
print("Thời khóa biểu")
```

```
# chuyển chuỗi "52" thành số nguyên 52
```

```
x = int("52")
```

```
type(y) # trả lại kiểu dữ liệu của biến y
```

```
x = input() # nhập một chuỗi bất kì từ bàn phím
```

=> Các lệnh trong Bảng 26.1 chính là các chương trình con được thiết kế sẵn của Python, cho phép người dùng tùy ý sử dụng trong các chương trình của riêng mình.

Trong các ví dụ trên, chuỗi kí tự bên trong ngoặc của các hàm int () và print() là tham số của hàm. Cú pháp câu lệnh gọi hàm trong Python có dạng chung như sau:

<tên hàm>(<danh sách tham số hàm>)

Ghi nhớ

Python cung cấp sẵn nhiều hàm thực hiện những công việc khác nhau cho phép người dùng được tùy ý sử dụng khi viết chương trình bằng các câu lệnh gọi hàm tương ứng.

Câu hỏi

? Mô tả tham số và giá trị trả lại của mỗi hàm sau: float(), str(), len(), list()

2. THIẾT LẬP CÁC HÀM TỰ ĐỊNH NGHĨA

Ví dụ 1. Cách viết hàm có trả lại giá trị

```
>>> def inc(n):
    return n+1
```

```
>>> inc(3)
```

```
4
```

Tên hàm: inc

Tham số hàm: số n

Giá trị trả lại: số n + 1

Ví dụ 2. Cách viết hàm không trả lại giá trị.

```
>>> def thong_bao(msg):
    print("Xin chào bạn", msg)
    return
```

```
>>> thong_bao("Trần Quang Minh")
```

```
Xin chào bạn Trần Quang Minh
```

Tên hàm: thong_bao

Tham số hàm: chuỗi kí tự msg

Giá trị trả lại: không có

Ghi nhớ

Hàm trong Python được định nghĩa bằng từ khóa **def**, theo sau là tên hàm (tên hàm sẽ theo quy tắc đặt tên định danh). Hàm có thể có hoặc không có tham số. Khối lệnh mô tả hàm được viết sau dấu ":" và viết lùi vào, thẳng hàng. Hàm có thể có hoặc không có giá trị trả lại sau từ khóa return.

- Cú pháp thiết lập hàm có trả lại giá trị

```
def <tên hàm> (<tham số>):
```

```
    <khối lệnh>
```

```
    return <giá trị>
```

Cần có lệnh return <giá trị>. Hàm sẽ kết thúc khi gặp lệnh return và trả lại <giá trị>

- Cú pháp thiết lập hàm không trả lại giá trị

```
def <tên hàm> (<tham số>):
```

```
    <khối lệnh>
```

return

Lệnh return không có giá trị trả lại. Hàm số kết thúc khi gặp lệnh return. Nếu hàm không trả lại giá trị thì có thể không cần lệnh return

Ghi nhớ: Để thiết lập hàm trả lại giá trị, câu lệnh return trong khai báo hàm cần có <giá trị> đi kèm.

Để thiết lập hàm không trả lại giá trị có thể dùng return không có <giá trị> hoặc không cần có return

C. Củng cố - Vận dụng

THỰC HÀNH

Thiết lập hàm trong Python

Nhiệm vụ 1. Viết hàm yêu cầu người dùng nhập họ tên rồi đưa lời chào ra màn hình

Hướng dẫn: Chương trình có thể như sau:

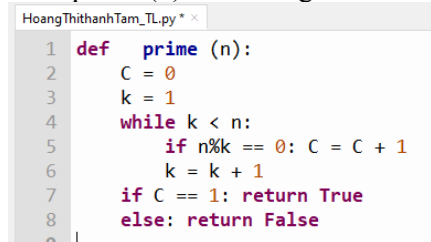
```
def meeting():
    ten = input("Nhập họ tên của em:")
    print("Xin chào", ten)
```

meeting()

Nhiệm vụ 2. Viết hàm prime (n) với tham số tự nhiên n và trả lại True nếu n là số nguyên tố, trả lại False nếu n không phải số nguyên tố

Hướng dẫn: Số nguyên tố là số tự nhiên lớn hơn 1, không có ước nào ngoài 1 và chính nó. Để thiết lập hàm prime (n) chúng ta cần tính số ước thật sự của n (từ 1 đến n-1). Biến C dùng để đếm số các ước thật sự của n. Khi đó, n sẽ là số nguyên tố khi và chỉ khi C = 1

Hàm prime (n) và chương trình có thể được thiết lập của như sau:



```
HoangThithanhTam_TL.py * x
1 def prime (n):
2     C = 0
3     k = 1
4     while k < n:
5         if n%k == 0: C = C + 1
6         k = k + 1
7     if C == 1: return True
8     else: return False
9
```

CHỦ ĐỀ 5: GIẢI QUYẾT VẤN ĐỀ VỚI SỰ TRỢ GIÚP CỦA MÁY TÍNH

BÀI 27: THAM SỐ CỦA HÀM

A. YÊU CẦU CẦN ĐẠT

- * - Biết cách thiết lập các tham số của hàm. Hiểu được cách truyền giá trị thông qua đối số hàm.
- * - Biết viết chương trình có sử dụng chương trình con .

B. NỘI DUNG BÀI HỌC

1. THAM SỐ VÀ ĐỐI SỐ CỦA HÀM

Ví dụ. Cách truyền dữ liệu qua tham số

```

1 >>> def f(a,b,c):      # Hàm f() có 3 tham số a, b, c
2     return a+b+c
3 >>> f(1,2,3)          # Hàm f() được gọi với ba giá trị cụ thể
4 6
5 >>> x,y,z = 10,20,5
6 >>> f(x,y,z)          # Hàm f() được gọi với ba biến đã có giá trị
7 35
8 >>> f(a,b,c)          # Lỗi gọi hàm bị lỗi nếu các tham số được truyền vào chưa có giá trị
9 Traceback (most recent call last):
10  File "<pyshell#6>", line 1, in <module>
11     f(a,b,c)
12 NameError: name 'a' is not defined
13 >>>

```

Ghi nhớ: Tham số của hàm được định nghĩa khi khai báo hàm và được dùng như biến trong định nghĩa hàm. **Đối số** là giá trị được truyền vào khi gọi hàm. Khi gọi hàm, các **tham số (parameter)** sẽ được truyền bằng giá trị thông qua **đối số (argument)** của hàm, số lượng giá trị được truyền vào hàm bằng với số tham số trong khai báo của hàm.

2. CÁCH SỬ DỤNG CHƯƠNG TRÌNH CON

Ví dụ 1. Việc kiểm tra một số có là số nguyên tố được lặp đi lặp lại từ 1 đến n và do đó nên sử dụng hàm prime(n) để kiểm tra sẽ giúp chương trình cấu trúc rõ ràng và dễ hiểu hơn.

Chương trình hoàn chỉnh giải bài toán trên có thể được viết như sau:

```

HoangThithanhTam_TL.py ×
1 def prime(n):
2     C = 0
3     k = 1
4     while k < n:
5         if n%k == 0: C = C + 1
6         k = k + 1
7     if C == 1: return True
8     else: return False
9 # Chương trình chính
10 n = int(input("Nhập số tự nhiên n: "))
11 for k in range(1, n+1) :
12     if prime(k):
13         print(k, end = " ")

```

Ví dụ 2. Chương trình sử dụng chương trình con.

Cho trước hai dãy số B, C, chương trình chính cần tính tổng các số hạng dương của mỗi dãy này. Chúng ta sẽ thiết lập hàm tongduong(A) để tính tổng các số hạng lớn hơn của một dãy A. Chương trình chính sẽ gọi hàm tongduong(A)

Chương trình có thể như sau:

```

HoangThithanhTam_TL.py ×
1 def tongduong(A) :
2     S=0
3     for k in A:
4         if k > 0: S = S + k
5     return S
6 # Chương trình chính
7 A = [0, 2, -1, 5, 10, -3]
8 B = [1, -10, -11, 8, 2, 0, -5]
9 # Sử dụng hàm tongduong tính tổng các số dương của dãy A
10 print("Tổng các số dương trong dãy A = ",tongduong(A))
11 # Sử dụng hàm tongduong tính tổng các số dương của dãy B
12 print("Tổng các số dương trong dãy B = ",tongduong(B))
13

```

Tóm lại:

Sử dụng chương trình con có thể giúp phân chia việc giải một bài toán lớn thành giải quyết các bài toán nhỏ và phát huy được tinh thần làm việc nhóm; Chương trình chính có cấu trúc rõ ràng, dễ hiểu hơn; Nếu cần hiệu chỉnh, phát triển và nâng cấp cũng thuận tiện hơn.

C. Củng cố - Vận dụng

THỰC HÀNH

Truyền giá trị cho đối số của hàm

Nhiệm vụ 1. Thiết lập hàm `f_sum(A, b)` có chức năng tính tổng các số của danh sách A theo quy định sau:

- Nếu `b = 0` thì tính tổng các số của danh sách A
- Nếu `b` khác 0 thì chỉ tính tổng các số dương của A

Hướng dẫn. Chương trình luôn kiểm tra giá trị của đối số `b` khi tính tổng các số của danh sách A

Chương trình có thể như sau:

```
HoangThithanhTam_TL.py * x
1 def f_sum(A, b):
2     S = 0
3     for x in A:
4         if b == 0:
5             S = S + x
6         else:
7             if x > 0:
8                 S = S + x
9     return S
```

Nhiệm vụ 2. Thiết lập hàm `f_dem(msg, sep)` có chức năng đếm số từ của một chuỗi `msg` với kí tự tách từ là `sep`

Ví dụ:

`f_dem("Mùa thu lịch sử", " ")`

trả lại giá trị 4

`f_dem("Mùa thu lịch sử", ". ")`

trả lại giá trị 1

Hướng dẫn. Để tách chuỗi `msg` thành các từ, ta dùng lệnh `split()`. Tham số `sep` chính là tham số của lệnh `split()`.

Chương trình có thể như sau:

```
HoangThithanhTam_TL.py * x
1 def f_dem(msg, sep):
2     xlist = msg.split(sep)
3     return len(xlist)
```

Nhiệm vụ 3. Thiết lập hàm `merge_str(s1, s2)` với `s1, s2` là hai chuỗi cần gộp .

Hàm này sẽ gộp hai chuỗi `s1, s2` theo cách, lấy lần lượt kí tự `s1, s2` đưa vào chuỗi kết quả. Nếu có một chuỗi hết kí tự thì đưa phần còn lại của chuỗi dài hơn vào chuỗi kết quả. Ví dụ nếu `s1 = "1111"`, `s2 = "0000"`, thì chuỗi kết quả là "10101010"

Hướng dẫn. Gọi `S` là chuỗi kết quả trước và sau khi gộp hai chuỗi `s1` và `s2`, chương trình có thể như sau:

```
HoangThithanhTam_TL.py x
1 def merge_str(s1, s2) :
2     S = ""
3     l1 = len(s1)
4     l2 = len(s2)
5     l = min(l1,l2)
6     for i in range (l):
7         S = S + s1[i] + s2[i]
8         if l1 < l2 :
9             for i in range(l, l2):
10                S = S + s2[i]
11            if l1 > l2:
12                for i in range (l, l1):
13                    S = S + s1[i]
14    return S
15 a=input("Nhập chuỗi a = ")
16 b=input("Nhập chuỗi b = ")
17 print(merge_str(a,b))
```

CHỦ ĐỀ 5: GIẢI QUYẾT VẤN ĐỀ VỚI SỰ TRỢ GIÚP CỦA MÁY TÍNH

BÀI 28: PHẠM VI CỦA BIẾN

A. YÊU CẦU CẦN ĐẠT

* Biết và trình bày được ý nghĩa của phạm vi hoạt động của biến trong chương trình và hàm.

B. NỘI DUNG BÀI HỌC

1. PHẠM VI CỦA BIẾN KHAI BÁO TRONG HÀM

Các biến được khai báo bên trong một hàm chỉ được sử dụng bên trong hàm đó. Chương trình chính không sử dụng được.

```
>>> def func(a,b) :
    n = 10
    a = a * 2
    b = a+b
    return a + b + n
```

```
>>> a = 1
```

```
>>> b = 2
```

```
>>> func(a,b)
```

```
16
```

```
>>> a,b
```

```
(1,2)
```

```
>>>n
```

```
Traceback (most recent call last): File "<pyshell#11>", line 1, in <module>
```

```
n
```

```
NameError: name 'n' is not defined
```

Như vậy:

Trong Python tất cả các biến khai báo bên trong hàm đều có tính địa phương (cục bộ), không có hiệu lực ở bên ngoài hàm.

2. PHẠM VI CỦA BIẾN KHAI BÁO NGOÀI HÀM

Ví dụ 1. Biến khai báo bên ngoài hàm không có tác dụng bên trong hàm.

```
>>> def f(n):
    t = n + 1
    return t
```

```
>>> t = 10
```

```
>>> f(5)
```

```
6
```

```
>>> t
```

```
10
```

Ví dụ 2. Bên trong hàm có thể truy cập để sử dụng giá trị của biến đã khai báo trước đó ở bên ngoài hàm.

```
>>> def f(a, b):
    return a + b + N
```

```
>>> N = 10
```

```
>>> f(1, 2)
```

```
13
```

Lưu ý: nếu muốn biến bên ngoài vẫn có tác dụng bên trong hàm thì chỉ cần khai báo lại biến này bên trong hàm với từ khóa **global**

```
>>> def f(n) :
    global t
    t = 2*n + 1
    return t
```

```
>>> t = 10
```

```
>>> f(1)
```

```
3
```

```
>>> t
```

```
3
```

Tóm lại: Biến đã khai báo bên ngoài hàm chỉ có thể truy cập giá trị để sử dụng bên trong hàm mà không làm

thay đổi được giá trị của biến đó (trừ trường hợp với từ khóa global)

Câu hỏi: Giả sử hàm f(x, y) được định nghĩa như sau:

```
>>> def f(x, y):
    a = 2*(x + y)
    print(a + n)
    n = 10
    f(1, 2)
```

Kết quả nào được in ra khi thực hiện các lệnh sau?

C. Củng cố - Vận dụng

Thực Hành Phạm vi của biến

Nhiệm vụ 1. Viết hàm với đầu vào là danh sách A chứa các số và số thực x. Hàm trả lại một danh sách kết quả B từ danh sách A bằng cách chỉ giữ lại các phần tử lớn hơn hoặc bằng x.

Hướng dẫn. Biến B kiểu danh sách cần được định nghĩa trong hàm và được bổ sung thêm các phần tử từ A nếu thỏa mãn điều kiện lớn hơn hoặc bằng x.

```
def Select(A, x):
    B = []
    for k in range(len(A)):
        if A[k] >= x: B.append(A[k])
    return B
```

Nhiệm vụ 2. Viết hàm với đầu vào là chuỗi ký tự Str và số c, đầu ra là danh sách các từ được tách ra từ chuỗi Str nhưng đã được chuyển thành chữ in hoa hoặc chữ in thường. Hoặc chỉ chuyển ký tự đầu các từ thành chữ in hoa tùy thuộc vào tham số đầu vào c như sau :

- Nếu c = 0, danh sách B là các từ được chuyển thành chữ in hoa.
- Nếu c = 1, danh sách B là các từ được chuyển thành chữ in thường.
- Nếu c = 2, danh sách B là các từ được chuyển viết chữ hoa ký tự đầu của mỗi từ.

Hướng dẫn. Chúng ta cần sử dụng các lệnh sau:

Str.upper() – chuyển ký tự của chuỗi thành chữ in hoa.

Str.lower() – chuyển ký tự của chuỗi thành chữ in thường.

Str.title() – chuyển ký tự đầu mỗi từ của chuỗi thành chữ in hoa, các ký tự khác chuyển về chữ thường

- Hàm được định nghĩa có dạng **Tach_tu(Str, c)**. Đầu tiên chuỗi Str cần được tách từ bằng lệnh split(). Sau đó danh sách kết quả sẽ được chuyển đổi chữ in hoa, in thường sử dụng một trong các lệnh trên tùy thuộc vào giá trị của đối số c.

```
HoangThithanhTam_TL.py
1 def Tach_tu(Str, c):
2     A = Str.split()
3     for k in range(len(A)):
4         if c == 0:
5             A[k] = A[k].upper()
6         if c == 1:
7             A[k] = A[k].lower()
8         if c == 2:
9             A[k] = A[k].title()
10    return A
11 s="hoàng thị thanh tâm"
12 c=int(input("nhập c = 0,1,2: "))
13 a=Tach_tu(s,c)
14 print(a)
```

Nhiệm vụ 3. Viết chương trình yêu cầu thực hiện lần lượt các việc sau, mỗi việc cần được thực hiện bởi một hàm:

1. Nhập từ bàn phím một dãy các số nguyên, mỗi số cách nhau bởi dấu cách. Chuyển các số này vào danh sách A và in danh sách A ra màn hình.
2. Trích từ danh sách A ra một danh sách B gồm các phần tử lớn hơn 0. In danh sách B ra màn hình.
3. Trích từ danh sách A ra một danh sách C gồm các phần tử nhỏ hơn 0. In danh sách C ra màn hình.

Hướng dẫn. Với mỗi việc trên được viết thành một hàm. Toàn bộ chương trình có thể như sau:

```
HoangThithanhTam_TL.py
1 t = 0
2 def Nhap_Dulieu():
3     s = input("Nhập các số nguyên cách nhau bởi dấu cách: ")
4     A = s.split()
5     for k in range(len(A)): A[k] = int(A[k])
6     return A
7 def getB(A):
8     B = []
9     for x in A:
10        if x > 0: B.append(x)
11    return B
12 def getC(A):
13    C = []
14    for x in A:
15        if x < 0: C.append(x)
16    return C
17 # Chương trình chính
18 A = Nhap_Dulieu()
19 print("Danh sách A:", A)
20 B = getB(A); C = getC(A)
21 print("Danh sách B:", B)
22 print("Danh sách C:", C)
```

CHỦ ĐỀ 5: GIẢI QUYẾT VẤN ĐỀ VỚI SỰ TRỢ GIÚP CỦA MÁY TÍNH BÀI 29: NHẬN BIẾT LỖI CHƯƠNG TRÌNH A. YÊU CẦU CẦN ĐẠT

- * Biết và phân biệt được một số loại lỗi chương trình
- * Biết được một vài lỗi ngoại lệ thường gặp

B. NỘI DUNG BÀI HỌC

1. NHẬN BIẾT LỖI CHƯƠNG TRÌNH

Trường hợp 1: Người lập trình viết sai cú pháp lệnh, chương trình lập tức dừng lại và thông báo lỗi cú pháp

```
>>> While true print("Hello")
```

SyntaxError: Invallid syntax

Trường hợp 2: Người dùng nhập dữ liệu sai, chương trình thông báo lỗi nhập dữ liệu không đúng khuôn dạng

```
>>> n = int( input("Nhập số nguyên n: "))
```

Nhập số nguyên n: 1.5

Traceback (most recent call last):

File "<pyshell#0>", line 1, in <module>

```
N= int( input("Nhập số nguyên n: "))
```

Trường hợp 3: Chương trình thông báo lỗi chỉ số vượt quá giới hạn cho phép

```
A = [1, 3, 10, 0]
```

```
for i in range(5):
```

```
    print(A[i], end = " ")
```

Khi chạy chương trình sẽ báo lỗi.

```
1 3 10 0 Traceback (most recent call last):
```

File "C:\Python\Array_b1.py", line 3, in <module>

```
print (A[1],end")
```

IndexError: list index out of range

Trường hợp 4. Chương trình thực hiện bình thường nhưng kết quả không đúng với yêu cầu của bài toán. Đây là lỗi logic bên trong chương trình.

```
# Tính tổng của ba số nguyên dương đầu tiên
```

```
>>> s = 0
```

```
>>> for i in range (3):
```

```
    s = s + i
```

```
>>> print(s)
```

```
3
```

Chương trình cho kết quả là 3 mà kết quả đúng là $1 + 2 + 3 = 6$. Lí do là hàm range(3) trả lại vùng giá trị là 0, 1, 2 chứ không phải là 1, 2, 3

Giải thích:

- Với trường hợp 1, chương trình dừng và thông báo lỗi Syntax Error (lỗi cú pháp)

- Với trường hợp 2, khi người dùng nhập dữ liệu sai, hàm int() không thể thực hiện được, chương trình dừng lại và báo lỗi. Mã lỗi là ValueError. Đây là lỗi Runtime (lỗi trong khi đang thực hiện) hay còn gọi là **lỗi ngoại lệ (Exceptions error)**

- Với trường hợp 3, chương trình phát hiện lỗi chỉ số vượt quá giới hạn tại dòng 3. Chương trình dừng và báo lỗi. Mã lỗi là IndexError. Đây là lỗi Runtime.

- Với trường hợp 4, chương trình không còn lỗi Runtime, nhưng kết quả đưa ra sai. Không có mã lỗi nào được trả lại. Đây là lỗi ngữ nghĩa hoặc lỗi logic bên trong chương trình.

Kết luận:

+ Tổng thể có thể phân biệt lỗi chương trình Python làm ba loại

1. Lỗi khi có lệnh viết sai cú pháp hoặc sai cấu trúc ngôn ngữ Python quy định. Chương trình sẽ lập tức dừng và thông báo lỗi Syntax Error

2. Lỗi khi không thể thực hiện một lệnh trong chương trình. Chương trình dừng lại và thông báo một mã lỗi. Lỗi này gọi là lỗi ngoại lệ (Exceptions Error), mã lỗi trả lại gọi là mã lỗi ngoại lệ.

3. Chương trình chạy không lỗi ngoại lệ, nhưng kết quả đưa ra sai không chính xác. Đây là lỗi logic bên trong chương trình.

+ Với mỗi loại lỗi trên, cách xử lí và kiểm soát lỗi sẽ khác nhau.

2. MỘT SỐ LỖI NGOẠI LỆ THƯỜNG GẶP

Chúng ta đã biết, nếu gặp lỗi ngoại lệ, chương trình Python sẽ dừng lại, báo lỗi. Một trong những vấn đề được đưa ra khi kiểm soát lỗi là làm thế nào để vẫn phát hiện lỗi, xử lý lỗi nhưng chương trình không bị dừng lại trong khi thực hiện.

Mã lỗi ngoại lệ	Mô tả lỗi
ZeroDivisionError	Lỗi này xảy ra khi lệnh thực hiện phép chia cho giá trị 0.
IndexError	Lỗi xảy ra khi lệnh cố gắng truy cập phần tử của danh sách nhưng chỉ số vượt quá giới hạn.
NameError	Lỗi xảy ra khi chương trình muốn tìm một tên nhưng không thấy. Ví dụ khi lệnh gọi một hàm nhưng không có hàm đó.
TypeError	Lỗi kiểu dữ liệu. Một số ví dụ lỗi loại này: - Lệnh truy cập một phần tử của danh sách nhưng chỉ số không là số nguyên - Lệnh tính biểu thức số nhưng lại có một toán hạng không phải là số
ValueError	Lỗi liên quan đến giá trị của đối tượng. Lỗi khi thực hiện lệnh chuyển đổi kiểu dữ liệu, đối số của hàm có giá trị mà hàm không hỗ trợ. Ví dụ khi thực hiện lệnh <code>int("1.55")</code> sẽ sinh lỗi loại này.
IndentationError	Lỗi khi các dòng lệnh thụt vào không thẳng hàng hoặc không đúng vị trí
SyntaxError	Lỗi cú pháp.

C. CƯỜNG CỐ - VẬN DỤNG

THỰC HÀNH

Lập trình và kiểm tra khả năng sinh lỗi khi chạy chương trình.

Nhiệm vụ 1. Viết chương trình nhập các số nguyên m, n từ bàn phím, cách nhau bởi dấu cách. Chương trình đưa ra tổng, hiệu, thương của hai số đã nhập.

Hướng dẫn. Chương trình chính là khối các lệnh nhập từ bàn phím hai số nguyên m, n. Các số này được nhập bằng lệnh `input()`, kết quả là một chuỗi ký tự. Chuỗi này sẽ được tách thành danh sách các chuỗi con bằng lệnh `split()`. Kết quả thu được sẽ chuyển đổi thành hai số m, n bằng lệnh `int()`. Nhập chương trình sau và kiểm tra khả năng sinh lỗi khi chạy chương trình.

```
HoangThithanhTam_TL.py x
1 s = input("Nhập hai số m, n cách nhau bởi dấu cách: ")
2 sline = s.split( )
3 m, n = int(sline[0]), int(sline[1])
4 print("Tổng, hiệu, thương 2 số đã nhập là :",m+n, m-n, m/n)
5
```

Gợi ý. Các khả năng sinh lỗi của chương trình:

- Các số m, n khi nhập vào không là số nguyên
- Giữa hai số m, n không có dấu cách
- Số n nhập vào là số 0

Nhiệm vụ 2. Viết chương trình nhập số tự nhiên n và nhập lần lượt n số nguyên đưa vào danh sách số A. Sau khi nhập xong in danh sách A ra màn hình.

Hướng dẫn. Nhập chương trình sau và kiểm tra khả năng sinh lỗi khi chạy chương trình.

```
HoangThithanhTam_TL.py x
1 n = int(input("Nhập số tự nhiên n : "))
2 A = []
3 for k in range(n):
4     num = int(input("Nhập số thứ "+str(k+1)+" : "))
5     A.append(num)
6 print("Dãy đã nhập:",A)
7
```

Gợi ý. Các khả năng sinh lỗi của chương trình:

- Số n được nhập không là số nguyên
- Mỗi số hạng của danh sách nhập vào không là số nguyên

CHỦ ĐỀ 5: GIẢI QUYẾT VẤN ĐỀ VỚI SỰ TRỢ GIÚP CỦA MÁY TÍNH

BÀI 30: KIỂM THỬ VÀ GỠ LỖI CHƯƠNG TRÌNH

A. YÊU CẦU CẦN ĐẠT

- Biết được một vài phương pháp đơn giản kiểm thử chương trình.
- Biết được một vài cách gỡ lỗi đơn giản một chương trình.

B. NỘI DUNG BÀI HỌC

1. MỘT VÀI PHƯƠNG PHÁP KIỂM THỬ CHƯƠNG TRÌNH

Có rất nhiều phương pháp và công cụ khác nhau để kiểm thử chương trình. Các công cụ này không những có mục đích *tìm ra lỗi* (hay bug) của chương trình mà còn có tác dụng *phòng ngừa* và *ngăn chặn* các lỗi phát sinh tiếp trong tương lai.

a) Quan sát mã lỗi Runtime và bắt lỗi ngoại lệ

Nếu chương trình có lỗi Runtime (tức là đang chạy bị dừng lại), cần quan sát các mã lỗi (mã lỗi ngoại lệ) để kiểm tra vị trí dòng lệnh sinh ra lỗi này. Từ đó phân tích, tìm và sửa lỗi.

b) Kiểm thử chương trình với các bộ dữ liệu test

Chương trình cần được thử với một số bộ dữ liệu test gồm đầu vào tiêu biểu phụ thuộc đặc thù của bài toán và kết quả đầu ra đã biết trước. Các bộ test có thể có đầu vào theo các tiêu chí khác nhau như độ lớn và tính đa dạng của dữ liệu. Cần chú ý một số điểm sau:

- Cần có nhiều bộ test (theo các tiêu chí khác nhau như độ lớn, tính đa dạng của dữ liệu...)
- Cần có bộ test ngẫu nhiên. Việc sinh ngẫu nhiên dữ liệu đầu vào trong miền xác định của chương trình làm tăng khả năng tìm lỗi nếu có.
- Cần có bộ test dữ liệu ở vùng biên. Ví dụ dữ liệu đầu vào là cặp (x, y) xác định trên miền $0 \leq x, y \leq 1$. Khi đó cần kiểm tra chương trình với bộ dữ liệu biên là $(0; 0)$, $(0; 1)$, $(1; 0)$ và $(1; 1)$. Thực tế cho thấy thường phát sinh lỗi tại các vùng biên hoặc lân cận của biên. Một ví dụ khác của dữ liệu biên là cần tìm các bộ test với n và các giá trị (x_1, x_2, \dots, x_n) rất lớn (vùng cận biên lớn)

c) In các thông số trung gian

Bổ sung vào giữa các dòng lệnh `print()` để in ra các biến trung gian, qua đó kiểm tra các quy trình hay thuật toán được viết có đúng không.

Giả sử chương trình có đầu vào là (x_1, x_2, \dots, x_n) , đầu ra là (a_1, a_2, \dots, a_m) nhưng có sử dụng các biến trung gian (y_1, y_2, \dots, y_k) . Khi đó với mỗi bộ test đầu vào, chúng ta sẽ bổ sung vào các dòng lệnh của chương trình để in ra các giá trị trung gian:

$$(x_1, x_2, \dots, x_n), (y_1, y_2, \dots, y_k), (a_1, a_2, \dots, a_m)$$

Thông qua các giá trị trung gian trong quá trình thực hiện chương trình, nếu kết quả cuối cùng có lỗi thì sẽ dễ tìm ra lỗi đó.

d) Sử dụng công cụ break point (điểm dừng)

Công cụ break point cho phép tạo ra các “điểm dừng” bên trong chương trình. Khi chạy, chương trình sẽ tạm dừng lại tại các “điểm dừng” cho phép người kiểm thử có thể quan sát các thông tin khác bên trong chương trình, qua đó kiểm tra tính đúng đắn của chương trình.

Trên thực tế sử dụng phương pháp điểm dừng thường kết hợp với phương pháp in các giá trị trung gian sẽ là hiệu quả hơn để kiểm thử chương trình.

Một số ghi nhớ:

- Sử dụng công cụ in các biến trung gian.
- Sử dụng công cụ sinh các bộ dữ liệu test.
- Sử dụng công cụ điểm dừng trong phần mềm soạn thảo lập trình.

Quan sát các mã lỗi của chương trình nếu phát sinh.

2. VÍ DỤ MINH HỌA: Xem SGK/ trang 146

**CHỦ ĐỀ 5: GIẢI QUYẾT VẤN ĐỀ
VỚI SỰ TRỢ GIÚP CỦA MÁY TÍNH**
BÀI 31: THỰC HÀNH VIẾT CHƯƠNG TRÌNH ĐƠN GIẢN
BÀI 32: ÔN TẬP LẬP TRÌNH PYTHON

A. YÊU CẦU CẦN ĐẠT

- * Thực hành viết chương trình đơn giản bằng ngôn ngữ Python
- * Thực hành được các bước gỡ rối chương trình bằng công cụ debug – thiết lập điểm dừng và chạy theo từng lệnh.

B. NỘI DUNG BÀI HỌC

Bài 31: Thực hành viết chương trình đơn giản/ SGK Tin học 10/ trang 149 – 152

Bài 32: Ôn tập lập trình Python/ SGK Tin học 10/ trang 143 – 155

HS quan sát SGK để tìm hiểu nội dung kiến thức theo yêu cầu

Hs hoàn thành tìm hiểu kiến thức

CHỦ ĐỀ 6: HƯỚNG NGHIỆP VỚI TIN HỌC

BÀI 33: NGHỀ THIẾT KẾ ĐỒ HỌA MÁY TÍNH

A. YÊU CẦU CẦN ĐẠT

- * Biết được khái niệm, kiến thức và kỹ năng cần có của nghề thiết kế đồ họa.
- * Biết các ngành học và nhu cầu nhân lực liên quan đến nghề thiết kế đồ họa.
- * Tự tìm kiếm và khai thác được thông tin hướng nghiệp về lĩnh vực thiết kế đồ họa, giao lưu và chia sẻ với bạn bè qua các kênh truyền thông tin số về thông tin nghề nghiệp.

B. NỘI DUNG BÀI HỌC

1. KHÁI NIỆM VỀ THIẾT KẾ ĐỒ HỌA

- Thiết kế đồ họa là việc dàn dựng bố cục, sắp xếp, chỉnh sửa hình ảnh, lựa chọn màu sắc để sáng tạo các thông điệp truyền thông hấp dẫn và thu hút, đáp ứng yêu cầu truyền đạt thông tin một cách hiệu quả nhằm phục vụ mục đích tuyên truyền hoặc kinh doanh. Tùy theo phương thức thể hiện, thông điệp truyền thông có thể là các ấn phẩm (tấm thiệp, tờ rơi, logo, biểu hiệu, áp phích, tài liệu quảng cáo/giới thiệu sản phẩm, bìa sách/tạp chí,...), các trang web,...

- Các hình ảnh đồ họa thường bao gồm nhiều thành phần như văn bản, các đối tượng hình ảnh như các đường, các hình cơ bản hay hình vẽ, ảnh chụp, màu sắc,... Nhiệm vụ của người thiết kế đồ họa là lựa chọn, vẽ, cắt, ghép, sắp xếp các thành phần trên để tạo thành sản phẩm hoàn chỉnh.

- Thiết kế đồ họa đem lại nhiều lợi ích cho mọi người, ngành nghề, lĩnh vực khác nhau:

- Giúp tạo dựng hình ảnh chuyên nghiệp của tổ chức hoặc cá nhân đối với mọi người thông qua các sản phẩm như logo, áp phích, danh thiếp, thẻ nhân viên, hình ảnh trên mạng xã hội...
- Mang lại trải nghiệm đặc biệt cho độc giả, người xem thông qua các hình ảnh truyền thông thu hút và hấp dẫn.
- Tăng hiệu quả tiếp thị và doanh thu nhờ các tờ rơi, quảng cáo...., với các hình ảnh sản phẩm bắt mắt ấn tượng.

Ghi nhớ

- Thiết kế đồ họa là sáng tạo các thông điệp truyền thông kết hợp giữa hình ảnh, kiểu chữ, màu sắc để truyền tải thông tin đến người xem.
- Thiết kế đồ họa đem lại nhiều lợi ích cho mọi người, cho mọi ngành nghề, lĩnh vực.

2. KIẾN THỨC, KỸ NĂNG CẦN CÓ CỦA NGƯỜI THIẾT KẾ ĐỒ HỌA.

- Bất cứ ngành nghề nào cũng cần phải có kiến thức, kỹ năng nhất định. Đối với ngành thiết kế đồ họa, ngoài kỹ năng vẽ, sắp xếp các đối tượng đồ họa thì còn đòi hỏi những yêu cầu sau:

+ Có kiến thức về công nghệ nói chung và thành thạo kỹ năng máy tính và các thiết bị thông minh nói riêng, đặc biệt là kiến thức và kỹ năng làm việc trên các phần mềm đồ họa máy tính như Adobe Photoshop, CorelDraw, GIMP, InDesign, Scribus, AutoCard, Corel Designer, Solid Works,.... Ngoài ra, kiến thức về công nghệ in ấn cũng là điểm cộng đối với những người thiết kế đồ họa.

+ Người làm đồ họa máy tính cần luôn học hỏi những điều mới, cần có kiến thức rộng về các lĩnh vực như toán học, vật lý, nghệ thuật, xã hội,... để có thể ứng dụng trong công việc của mình. Đồng thời, họ cần phải có kỹ năng nhận biết được xu hướng, nắm bắt được nhu cầu của xã hội, tìm kiếm thông tin, tra cứu tài liệu để học hỏi và theo kịp với xu thế của đời sống, xã hội.

+ Bên cạnh đó, người làm thiết kế đồ họa không thể thiếu được khả năng sáng tạo, sự yêu thích cái đẹp, kỹ năng đánh giá, phân biện, phân tích, cũng như tư duy với con số và khả năng ngoại ngữ.

Ghi nhớ

Người làm nghề thiết kế đồ họa cần có:

- Khả năng sáng tạo, yêu thích và cảm nhận cái đẹp.
- Kiến thức về công nghệ nói chung, công nghệ in ấn, công nghệ thông tin và truyền thông nói riêng.
- Kiến thức rộng về các lĩnh vực như toán học, vật lý, nghệ thuật, xã hội.
- Kỹ năng vẽ, sắp xếp các đối tượng đồ họa.
- Kỹ năng sử dụng máy tính thông minh, sử dụng thành thạo phần mềm thiết kế đồ họa.
- Kỹ năng học hỏi những điều mới, công nghệ mới, kỹ năng tìm kiếm thông tin
- Kỹ năng đánh giá, phân biện, phân tích cũng như tư duy với những con số

3. HỌC TẬP VÀ VIỆC LÀM TRONG NGÀNH THIẾT KẾ ĐỒ HỌA

- Để bắt đầu với lĩnh vực thiết kế đồ họa, có thể theo học tại các trung tâm, trường dạy nghề. Cũng có thể theo học bậc đại học, cao đẳng tại các trường về mỹ thuật kiến trúc, thiết kế hoặc nhiều trường đào tạo ngành Công nghệ thông tin cũng đào tạo chuyên ngành Thiết kế đồ họa trên máy tính.

- Có thể tìm kiếm thông tin về hướng nghiệp, việc làm trên Internet thông qua các công cụ tìm kiếm phổ biến

như Google Search, Bing... với các từ khoá về nghề như thiết kế đồ họa, thiết kế mỹ thuật, thiết kế 3D, thiết kế giao diện, nhận diện thương hiệu, thiết kế quảng cáo,... Cũng có thể truy cập vào các diễn đàn, dịch vụ tìm kiếm việc làm như LinkedIn, Vietnamworks,... để tìm kiếm cũng như trao đổi thông tin.

- Những cơ hội nghề nghiệp như: chuyên viên thiết kế, tư vấn thiết kế tại các công ty quảng cáo, công ty thiết kế, công ty truyền thông và tổ chức sự kiện, studio nghệ thuật, xưởng phim hoạt hình và truyện tranh, các toà soạn, các nhà xuất bản, cơ quan truyền hình, báo chí,...

- Ngoài ra, sau khi tốt nghiệp, có thể tự thành lập doanh nghiệp, các công ty thiết kế, dịch vụ studio hoặc tư vấn, giảng dạy tại các trường học, trung tâm, câu lạc bộ,...

- Cơ hội làm thêm tại nhà như thiết kế website, thiết kế logo, nhận diện thương hiệu,... khi đã có những kinh nghiệm cần thiết, em hoàn toàn có tự mở công ty riêng cho mình, nhận dự án của các công ty, tổ chức,...

Tóm lại

- Theo học lĩnh vực thiết kế đồ họa tại các trung tâm, trường dạy nghề, các trường đại học, cao đẳng có chuyên ngành thiết kế đồ họa, thiết kế đồ họa trên máy tính.
- Có thể tìm kiếm thông tin về hướng nghiệp trên Internet hay qua các diễn đàn nghề nghiệp.
- Nhu cầu nhân sự cao với nhiều công việc và cách thức làm việc đa dạng

C. CÙNG CỐ - VẬN DỤNG

(Học sinh trả lời các câu hỏi vào tập – nộp tập vào cuối kì)

Câu 1: Hãy tìm các kênh thông tin giới thiệu việc làm liên quan đến thiết kế đồ họa và chia sẻ với bạn bè về kênh thông tin đó.

Câu 2: Sử dụng công cụ tìm kiếm trên internet để biết các phần mềm công cụ đồ họa như Illustrator, Photoshop, Indesign, AutoCad,... được dùng để làm gì?

CHỦ ĐỀ 6: HƯỚNG NGHIỆP VỚI TIN HỌC

BÀI 34: NGHỀ PHÁT TRIỂN PHẦN MỀM

A. YÊU CẦU CẦN ĐẠT

- * Hiểu được khái niệm nghề phát triển phần mềm và một số kiến thức, kỹ năng cần có của người làm nghề phát triển phần mềm.
- * Biết các ngành học ở bậc đại học, cao đẳng liên quan đến phát triển phần mềm và cơ hội nghề nghiệp liên quan đến phát triển phần mềm.

B. NỘI DUNG BÀI HỌC

1. PHÁT TRIỂN PHẦN MỀM LÀ GÌ?

Các công việc cơ bản, cũng chính là các công đoạn cần thực hiện để sản xuất một phần mềm gồm có:

Điều tra khảo sát: Tiếp xúc với khách hàng, tìm hiểu về yêu cầu nghiệp vụ, xây dựng hồ sơ yêu cầu của hệ thống.

Phân tích hệ thống: Dựa trên các tài liệu điều tra khảo sát, chuyên viên phân tích sẽ tạo ra tài liệu mô tả đầy đủ yêu cầu của phần mềm.

Thiết kế hệ thống: Dựa vào tài liệu phân tích, chuyên viên thiết kế sẽ đưa ra thiết kế tổng thể, thiết kế dữ liệu và thiết kế chức năng và có thể cả giao diện chi tiết.

Lập trình: Dựa vào tài liệu thiết kế, các lập trình viên sẽ tiến hành tạo cơ sở dữ liệu nếu cần và viết các đoạn mã thực hiện các chức năng.

Kiểm thử: Phát hiện để loại bỏ các lỗi cũng như các bất hợp lý trong sử dụng chương trình nếu có; kiểm tra kết quả thực hiện theo chức năng đã thiết kế,...

Chuyển giao: Cài đặt, khởi tạo dữ liệu, hướng dẫn sử dụng và chuyển giao.

Bảo trì: nhằm khắc phục triệt để các lỗi, nâng cấp cả về tính năng và giao diện của phần mềm. Công việc này có thể là một vòng phát triển mới, liên quan tới tất cả các công việc sản xuất phần mềm nêu trên.

Hoạt động có tính bao trùm lên toàn bộ các công việc cơ bản của sản xuất phần mềm là *quản trị dự án phần mềm*, bao gồm lập kế hoạch, điều phối nhân sự, tài chính, phương tiện, kiểm soát chất lượng, để đảm bảo thành công của dự án.

Tất cả các công việc và hoạt động nêu trên được gọi chung là *phát triển phần mềm* mà lập trình chỉ là một hoạt động trong đó. Những người tham gia vào các công việc và hoạt động đó đều được gọi là *người phát triển phần mềm* (Software Developer).

Tóm lại

Phát triển phần mềm gồm các công việc và hoạt động sau: điều tra, khảo sát, phân tích và thiết kế hệ thống; lập trình; kiểm thử; chuyển giao; bảo trì và quản trị dự án.

2. KIẾN THỨC, KỸ NĂNG CỦA NGƯỜI PHÁT TRIỂN PHẦN MỀM

Có ba hoạt động chính trong phát triển phần mềm là:

- Lập trình.

- Tổ chức phát triển phần mềm bao gồm việc vận dụng các kiến thức, hiểu biết và kỹ thuật để tổ chức các hoạt động phân tích, thiết kế, lập trình, kiểm thử, bảo trì, đánh giá, chuyển giao.

- Quản trị dự án phát triển phần mềm.

Khởi đầu, lập trình viên chỉ cần có các hiểu biết cơ bản về một ngôn ngữ lập trình phù hợp để có thể bắt đầu phụ trách những đoạn mã ngắn, đơn giản theo thiết kế. Với kiến thức và kỹ năng có được, ngoài việc lập trình, họ có thể tham gia một số công đoạn khác như kiểm thử, chuyển giao hay bảo trì phần mềm.

Ở cấp độ cao hơn, lập trình viên được trang bị thêm các kiến thức về thuật toán, cấu trúc dữ liệu, trí tuệ nhân tạo, mật mã,... để có thể viết các chương trình phức tạp đòi hỏi hiểu biết chuyên sâu về toán học và khoa học máy tính.

Khái niệm kỹ sư phần mềm thường để chỉ những người tổ chức làm phần mềm. Họ có thể phụ trách các khâu quan trọng như phân tích, thiết kế hay trực tiếp tham gia hoặc chủ trì quản trị dự án phần mềm.

Sự khác biệt giữa các kỹ sư phần mềm và lập trình viên tương tự như các kiến trúc sư và thợ xây trong xây dựng công trình. Kỹ sư phần mềm không nhất thiết phải lập trình nhưng hiểu biết về lập trình rất quan trọng giúp họ có giải pháp thiết kế tốt. Trong thực tế, chuyên viên phân tích và thiết kế nói chung đều trải qua quá trình lập trình.

Người quản lý dự án cần có tầm nhìn, hiểu biết về quy trình làm phần mềm, hiểu biết xu hướng công nghệ, có khả năng tổ chức, lập kế hoạch, điều phối các nguồn lực, tổ chức giám sát.

Đối với các dự án phần mềm lớn, hoạt động quản trị dự án có vai trò cốt yếu cho sự thành công của dự án phần mềm.

Tóm lại

- Lập trình viên, kỹ sư phần mềm, người quản trị dự án là những người đảm nhận những công việc quan trọng nhất trong phát triển phần mềm
- Có những kiến thức nhất định về toán học, cấu trúc dữ liệu về giải thuật nói riêng và về khoa học máy tính nói chung ở các mức khác nhau cùng các khả năng vận dụng thuận phục các kiến thức ấy vào thực tế là những yêu cầu cần có đối với lập trình viên và các kỹ sư phần mềm – người đảm nhận những vị trí quan trọng trong tổ chức phát triển phần mềm.
- Quản trị dự án là công việc xuyên suốt quá trình sản xuất phần mềm có vai trò chủ chốt cho sự thành công của dự án phần mềm. Việc có tầm nhìn, hiểu biết về quá trình làm phần mềm, hiểu biết xu hướng công nghệ, có khả năng tổ chức, lập kế hoạch, điều phối các nguồn lực, tổ chức giám sát... là những yêu cầu không thể thiếu đối với người quản trị viên dự án phát triển phần mềm.

C. CUNG CỐ - VẬN DỤNG

(Học sinh trả lời các câu hỏi vào tập – nộp tập vào cuối kì)

Câu 1: Ở khu vực nơi em sinh sống hay các tỉnh thành phố lân cận, trường đại học nào đào tạo nghề phát triển phần mềm? Khối thi ngành liên quan đến phát triển phần mềm của trường đó là gì?

Câu 2: Ở tỉnh thành phố nơi em cư trú có Trung tâm dạy nghề phát triển phần mềm nào không? Liệt kê một vài khóa học tiêu biểu mà họ cung cấp. Chia sẻ thông tin em tìm hiểu được với các bạn.

Câu 3: Ở tỉnh thành phố nơi em cư trú có doanh nghiệp nào chuyển về phát triển phần mềm không? Họ có cung cấp các chương trình đào tạo cho người muốn trở thành người phát triển phần mềm của công ty hay không?