***Câu 1: Lập trình là gì? Quy trình giải quyết một bài toán trên máy tính?***

*Lập trình* là việc sử dụng các ngôn ngữ lập trình, các đoạn mã lệnh và các tiện ích có sẵn để xây dựng các chương trình phần mềm, trò chơi ứng dụng của hệ thống xử lý các trang web… giúp người dùng có thể thực hiện các mệnh lệnh của máy tính cũng như tương tác qua lại với nhau thông qua các thiết bị điện tử.

*Quá trình* giải một bài toán trên máy tính gồm các bước:

Bước 1: Xác định bài toán

Bước 2: Tìm thuật toán giải bài toán và cách tổ chức dữ liệu.

Bước 3: Viết chương trình

Bước 4: Kiểm thử, chạy và hiệu chỉnh chương trình

***Câu 2: Khái niệm thuật toán? Tìm hiểu và vẽ sơ đồ thuật toán?***

*Thuật toán*: Là một tập hợp hữu hạn các hướng dẫn được xác định rõ ràng, có thể thực hiện được bằng máy tính, thường để giải quyết một lớp vấn đề hoặc để thực hiện một phép tính.

*Khái niệm của Sơ đồ khối thuật toán*

Sơ đồ khối thuật toán (algorithm flowchart) là một biểu đồ đồ họa sử dụng các hình dạng hình học như hình chữ nhật, hình tròn, hình bầu dục, và các mũi tên để biểu diễn các bước cụ thể trong một thuật toán hoặc quy trình logic. Sơ đồ khối thuật toán giúp mô tả một cách rõ ràng và trực quan các bước cần thực hiện để giải quyết một vấn đề hoặc thực hiện một nhiệm vụ trong lập trình và khoa học máy tính.

*Các khái niệm chính của sơ đồ khối thuật toán bao gồm:*

1. Hình dạng hình học: Sơ đồ khối sử dụng các hình dạng đặc biệt để biểu diễn các loại hành động khác nhau, chẳng hạn như hình chữ nhật để biểu thị các bước thực hiện, hình tròn cho các quyết định hoặc điều kiện, và mũi tên để kết nối các bước với nhau.

2. Bước thực hiện: Mỗi hình dạng trong sơ đồ khối đại diện cho một bước cụ thể trong thuật toán hoặc quy trình. Các bước này có thể là các thao tác, tính toán, gọi hàm, quyết định, hoặc các hoạt động khác.

3. Luồng điều khiển: Sơ đồ khối cung cấp một hình thức đồ họa để biểu thị luồng điều khiển của chương trình hoặc thuật toán. Điều này cho phép người đọc dễ dàng nhận thức về cách các bước tương tác với nhau và làm thế nào chúng quyết định luồng thực hiện.

4. Mũi tên và liên kết: Các mũi tên và liên kết kết nối các hình dạng với nhau, chỉ ra luồng thực hiện của thuật toán. Mũi tên thường đi từ trên xuống, biểu thị thứ tự thực hiện từ trên xuống dưới.

Sơ đồ khối thuật toán đóng vai trò quan trọng trong việc phát triển, hiểu và trình bày các thuật toán và quy trình logic phức tạp một cách dễ dàng hiểu.

***Câu 3: Các đặc trưng của một số thuật toán?***

*Các đặc trưng của thuật toán*

- Đầu vào (Input): Một thuật toán có các giá trị đầu vào từ một tập đã được chỉ rõ.

- Đầu ra(Output): Từ mỗi tập các giá trị đầu vào, thuật toán sẽ tạo ra các giá trị đầu ra. Các giá trị đầu ra chính là nghiệm của bài toán.

- Tính dừng: Sau một số hữu hạn bước thuật toán phải dừng.

- Tính xác định: Ở mỗi bước, các bước thao tác phải hết sức rõ ràng, không gây nên sự nhập nhằng. Nói rõ hơn, trong cùng một điều kiện hai bộ xử lý cùng thực hiện một bước của thuật toán phải cho những kết quả như nhau.

- Tính đúng đắn (hiệu quả): Trước hết thuật toán cần đúng đắn, nghĩa là sau khi đưa dữ liệu vào thuật toán hoạt động và đưa ra kết quả như ý muốn.

- Tính phổ dụng: Thuật toán có thể giải bất kỳ một bài toán nào trong lớp các bài toán. Cụ thể là thuật toán có thể có các đầu vào là các bộ dữ liệu khác nhau trong một miền xác định.

***Câu 4: Biểu diễn thuật toán bằng ngôn ngữ tự nhiên và sơ đồ khối?***

*Mô tả thuật toán bằng ngôn ngữ tự nhiên*

1. Thông báo “Mình cùng làm toán nhé” trong 2 giây

2. Nhập số thứ nhất

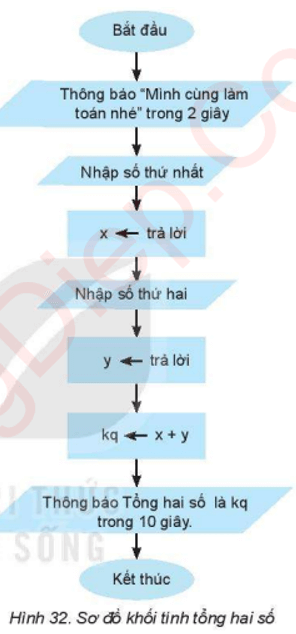
3. Gán giá trị trả lời vào biến x

4. Nhập số thứ hai

5. Gán giá trị trả lời vào biến y

6. Gán tổng x + y vào biến kq

7. Thông báo Tổng hai số trong 10 giây



***Câu 5: Giới thiệu về ngôn ngữ Python? Các thành phần cơ bản của ngôn ngữ Python?***

*Python là* một loại ngôn ngữ lập trình hướng đối tượng bậc cao, dùng để phát triển website và nhiều ứng dụng công nghệ khác nhau. So với các ngôn ngữ lập trình khác Python có cú pháp đơn giản, rõ ràng, cho phép người dùng viết mã lệnh với số lần gõ phím ít nhất. Vì thế đây là lựa chọn tối ưu nhất cho người lần đầu học lập trình.

*Các thành phần cơ bản cuả ngôn ngữ lập trình Python*:

## Chú thích

Các dòng chú thích được dùng để ghi chú cho người viết code dễ đọc các đoạn code khác, chú thích không được dịch bởi trình thông dịch. Để biến một dòng thành chú thích thì chúng ta thêm dấu # vào trước dòng đó.

## Biến

Biến là một cái tên dùng để lưu trữ một giá trị nào đó, có thể là một số, một chuỗi kí tự… Khác với các ngôn ngữ lập trình như C, Java… bạn phải khai báo cả kiểu dữ liệu đi kèm, thì trong Python bạn không phải khai báo kiểu dữ liệu, trình thông dịch Python sẽ tự phát hiện kiểu dữ liệu khi bạn gán giá trị cho biến đó.

Biến trong Python được đặt tên bằng chữ cái, chữ số và dấu gạch dưới \_, tuy nhiên tên biến không được phép bắt đầu bằng chữ số, còn lại chữ cái và dấu \_ thì được phép.

## Giá trị

Giá trị là các ký tự đại diện cho một đại lượng gắn liền với một biến nào đó. Nếu chúng ta không gán các giá trị vào biến thì chúng sẽ bị hủy bỏ.

## Toán tử

Toán tử chính là các phép tính toán.

|  |  |
| --- | --- |
| 1  2  3  4  5 | +    -    ~    \*    \*\*    /    %  << >>    &    |    ^  and    or    not    in    not in  is    is not    < >    !=    <>  ==    <= >= |

Trên đây là danh sách các toán tử trong ngôn ngữ Python. Chúng ta sẽ nói chi tiết về các toán tử ở bài sau.

## Khối lệnh

Trong các ngôn ngữ lập trình khác như Pascal, Java… một khối lệnh được nằm trong cặp dấu {} hoặc cặp từ khóa BEGIN–END. Còn Python thì dùng khoảng trắng, bạn có thể dùng dấu tab hoặc dấu cách (space) đều được. Trên hướng dẫn của Python khuyến khích dùng 4 dấu cách.

## Ký hiệu

Các ký hiệu dùng để bao bọc lấy các biểu thức, điều kiện… Trong các ví dụ trước có hàm print chúng ta phải dùng cặp ký hiệu () để bao bọc lấy nội dung cần được in ra màn hình. Thực ra đối với Python 2.x thì lệnh print không cần cặp ký tự này, nhưng Python 3.x thì phải có cặp ký tự này, lý do là vì trong phiên bản 2.x, print là một câu lệnh nhưng trong 3.x thì print đã được phát triển thành một hàm riêng.

## Từ khóa

Từ khóa là các từ dành riêng cho Python. Từ khóa thường được dùng để thi hành một lệnh nào đó, hoặc tên của một thành phần trong Python. Chẳng hạn từ khóa if thi hành lệnh so sánh. Từ khóa for để bắt đầu một vòng lặp, từ khóa and là tên của toán tử and… Bạn không được đặt tên biến hay tên hàm trùng với tên của các từ khóa.

Câu 6: Giới thiệu ngôn ngữ lập trình Python – Spyder?

Spyder là một IDE nguồn mở được nhiều nhà khoa học và nhà phân tích dữ liệu sử dụng. Nó cung cấp trải nghiệm phát triển toàn diện với các tính năng phân tích dữ liệu nâng cao, trực quan hóa dữ liệu và gỡ lỗi. Spyder cũng bao gồm các tính năng sau:

* Trình chỉnh sửa mã hỗ trợ nhiều ngôn ngữ
* Bảng điều khiển IPython tương tác
* Trình gỡ lỗi cơ bản
* Thư viện khoa học như Matplotlib, SciPy và NumPy
* Khả năng khám phá các biến trong mã
* Khả năng xem tài liệu trong thời gian thực

***Câu 7: Tạo biến và các phép tính toán đơn giản?***

Python cung cấp các phép toán đơn giản và dễ sử dụng cho các phép tính cộng, trừ, nhân và chia. Bạn có thể tạo các biến để lưu trữ giá trị và thực hiện các phép tính trên các biến đó.

Dưới đây là một ví dụ về cách tạo biến và thực hiện các phép tính toán đơn giản trong Python:

Python

# Tạo biến

age = 30

name = "Alice"

is\_working = True

# Phép tính cộng, trừ, nhân, chia

sum = age + 5

difference = age - 10

product = age \* 2

division = age / 3

# In kết quả

print("Age:", age)

print("Name:", name)

print("Working:", is\_working)

print("Sum:", sum)

print("Difference:", difference)

print("Product:", product)

print("Division:", division)

***Câu 8: Sơ lược trình biên dịch trong Python?***

Có thể bạn đã nghe về các công cụ hoặc khái niệm như:

**PyInstaller:** Công cụ này đóng gói các ứng dụng Python thành các file thực thi độc lập, có thể chạy trên các máy khác mà không cần cài đặt Python. Quá trình này thường được gọi là "biên dịch" nhưng thực chất nó chỉ là quá trình đóng gói và liên kết các thư viện cần thiết.

**Cython:** Đây là một ngôn ngữ lập trình mở rộng từ Python, cho phép bạn viết mã với kiểu tĩnh và các tính năng hiệu năng cao hơn. Mã Cython có thể được biên dịch thành mã C và sau đó được liên kết vào các chương trình Python.

**Numba:** Một công cụ JIT (Just-in-Time) compiler cho phép bạn biên dịch các đoạn mã NumPy thành mã máy, giúp tăng tốc đáng kể các tính toán số.

**Những công cụ này không thực sự "biên dịch" toàn bộ chương trình Python thành mã máy như các trình biên dịch truyền thống.** Chúng chỉ thực hiện các quá trình tối ưu hóa hoặc chuyển đổi một phần mã Python sang dạng hiệu quả hơn để tăng tốc độ thực thi.

***Câu 9: Sơ lược các kiểu dữ liệu trong Python?***

*Sơ Lược Các Kiểu Dữ Liệu Trong Python*

Python cung cấp một hệ thống các kiểu dữ liệu đa dạng và linh hoạt, giúp bạn biểu diễn và thao tác với nhiều loại thông tin khác nhau trong quá trình lập trình. Dưới đây là một số kiểu dữ liệu cơ bản và thường gặp nhất:

1. Kiểu Số (Numbers)

**int:** Số nguyên (ví dụ: 42, -10)

**float:** Số thực (số thập phân) (ví dụ: 3.14, -2.5)

**complex:** Số phức (ví dụ: 3+2j)

2. Kiểu Chuỗi (Strings)

**str:** Một chuỗi các ký tự (ví dụ: "Hello, World!", 'Python')

3. Kiểu Boolean

**bool:** Chỉ có hai giá trị: True (đúng) hoặc False (sai)

4. Kiểu Tập Hợp (Sequences)

**list:** Một danh sách có thứ tự, có thể thay đổi được các phần tử (ví dụ: [1, 2, 3, "apple"])

**tuple:** Một danh sách có thứ tự, nhưng không thể thay đổi được các phần tử (ví dụ: (1, 2, 3))

**range:** Một dãy số có thứ tự (ví dụ: range(5))

5. Kiểu Mapping

**dict:** Một từ điển, lưu trữ các cặp khóa-giá trị (ví dụ: {"name": "Alice", "age": 30})

6. Kiểu Tập (Sets)

**set:** Một tập hợp các phần tử không có thứ tự và không trùng lặp (ví dụ: {1, 2, 3})

**frozenset:** Một tập hợp không thể thay đổi được

7. Kiểu Khác

**NoneType:** Biểu thị giá trị null hoặc không có gì

**bytes:** Chuỗi các byte

**bytearray:** Một biến thể của bytes, có thể thay đổi được

**Ví dụ:**

Python

x = 10 # Số nguyên

y = 3.14 # Số thực

name = "Python" # Chuỗi

is\_python\_fun = True # Boolean

numbers = [1, 2, 3] # List

person = {"name": "Alice", "age": 30} # Dictionary