



Bài 1

Tổng quan về Java

Biến và kiểu dữ liệu

Toán tử và nhập xuất trong Java

- Tổng quan về ngôn ngữ lập trình Java
- Cài đặt môi trường, IDE và phát triển ứng Java
- Các bước giải quyết vấn đề và lưu đồ
- Biến và kiểu dữ liệu
- Toán tử
- Nhập và xuất trong Java



- **JAVA LÀ GÌ?**

- Ngôn ngữ lập trình hướng đối tượng của Sun Microsystem
- Tiêu chí **“Write Once, Run Anywhere”**
- Ứng dụng Java chạy trên máy ảo (JVM), có thể chạy trên các nền tảng khác nhau, không phụ thuộc vào phần cứng, hệ điều hành...

- **TẠI SAO LÀ JAVA?**

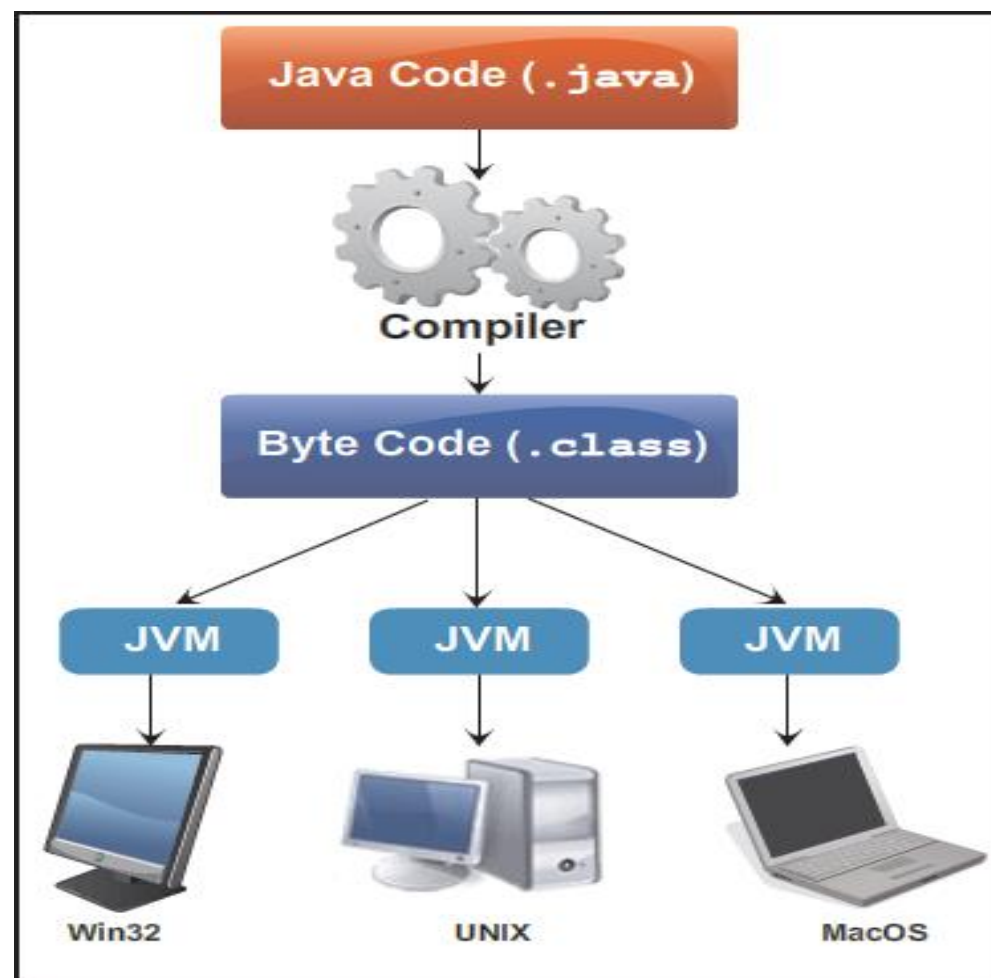
- Ngôn ngữ lập trình hướng đối tượng mã nguồn mở dễ tiếp cận
- Tài liệu phong phú đa dạng: API, Source, Documents, Article, ebook, video...
- Nhiều công cụ (IDE) lập trình: Netbeans, Eclipse, IntelliJ IDEA, Jdeveloper...
- Hoàn toàn miễn phí và có cộng đồng phát triển rộng lớn

- **JAVA CÓ THỂ LÀM GÌ?**

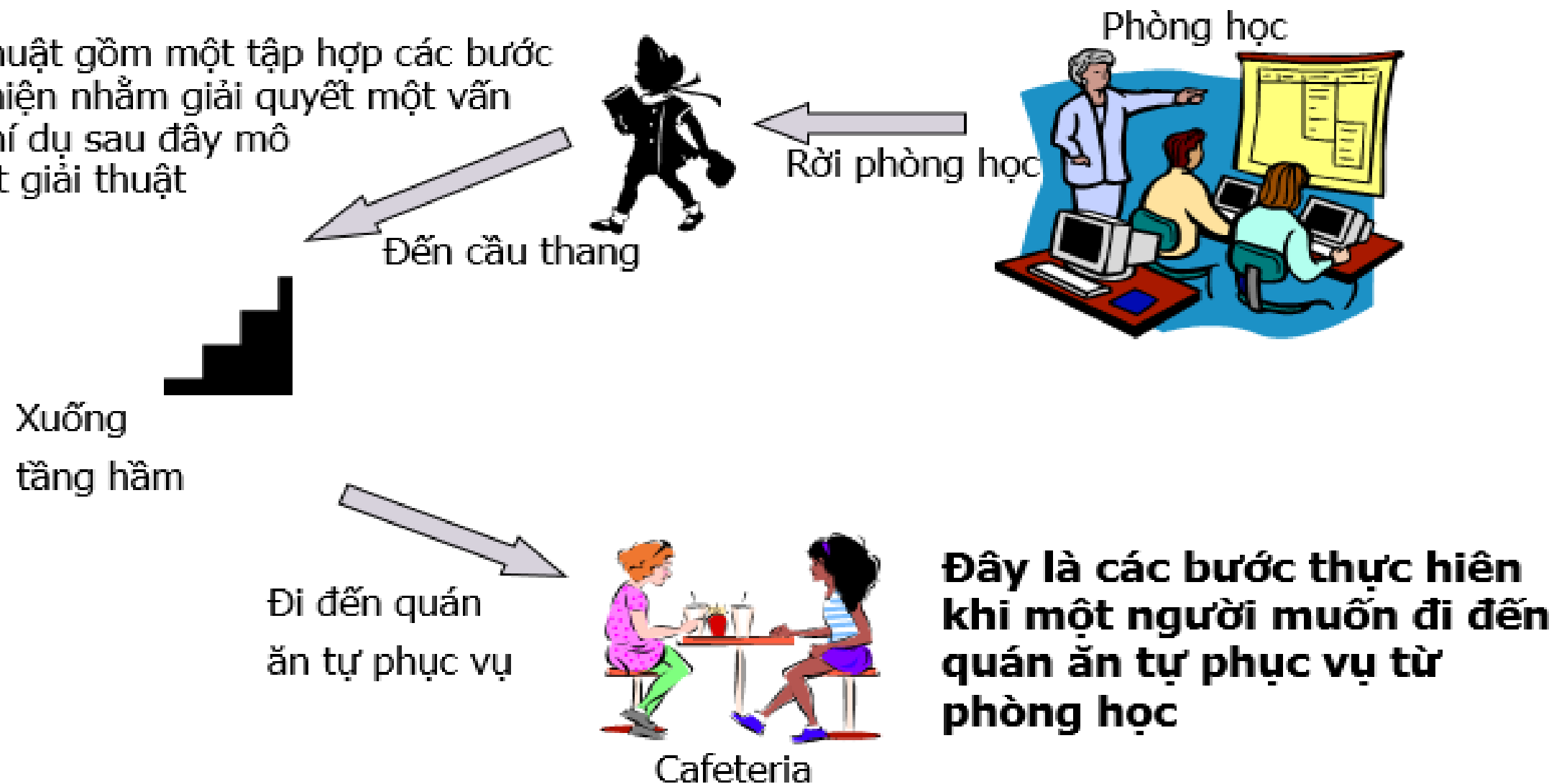
- Website
- Ứng dụng di động
- Games
- Điện toán đám mây
- Lập trình nhúng
- AI, IOT, Big Data...

1. Dowload và cài đặt JDK – Java Development Kit
2. Thiết lập biến môi trường và cấu hình cho java
3. Download và cài đặt công cụ phát triển – IDE: Netbeans, eclipse,...
4. Phát triển ứng dụng “**Hello Word**”


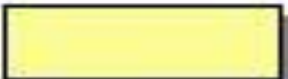

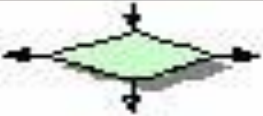


- Thực thi chương trình Java



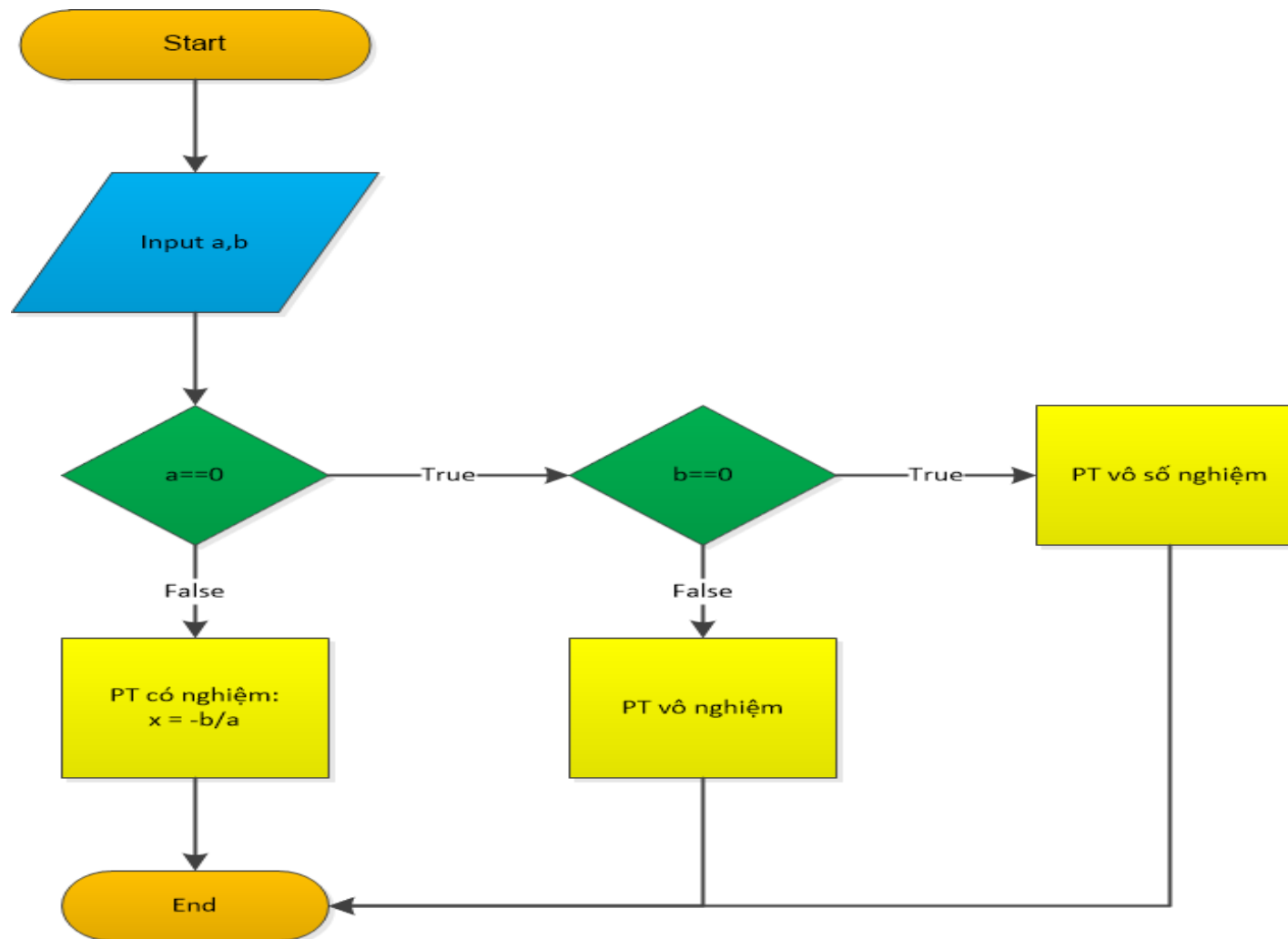
Giải thuật gồm một tập hợp các bước thực hiện nhằm giải quyết một vấn đề. Thí dụ sau đây mô tả một giải thuật



- Hình ảnh minh họa cho giải thuật, thể hiện rõ các bước để giải quyết vấn đề

| Biểu Tượng | Mô Tả |
|---|--|
|  | Bắt đầu hay kết thúc chương trình |
|  | Những bước tính toán |
|  | Các lệnh xuất hay nhập |
|  | Quyết định và rẽ nhánh |
|  | Bộ nối hai phần trong chương trình (đầu nối) |
|  | Dòng chảy |

- VD: Sử dụng lưu đồ thể hiện các bước của bài toán giải phương trình $ax + b = 0$



- Tên có ý nghĩa cho mỗi vị trí nhớ
- Biến giúp ta truy cập vào bộ nhớ mà không cần dùng địa chỉ của chúng
- Tham chiếu đến giá trị cụ thể trong bộ nhớ thông qua tên biến
- Cấu trúc khai báo biến

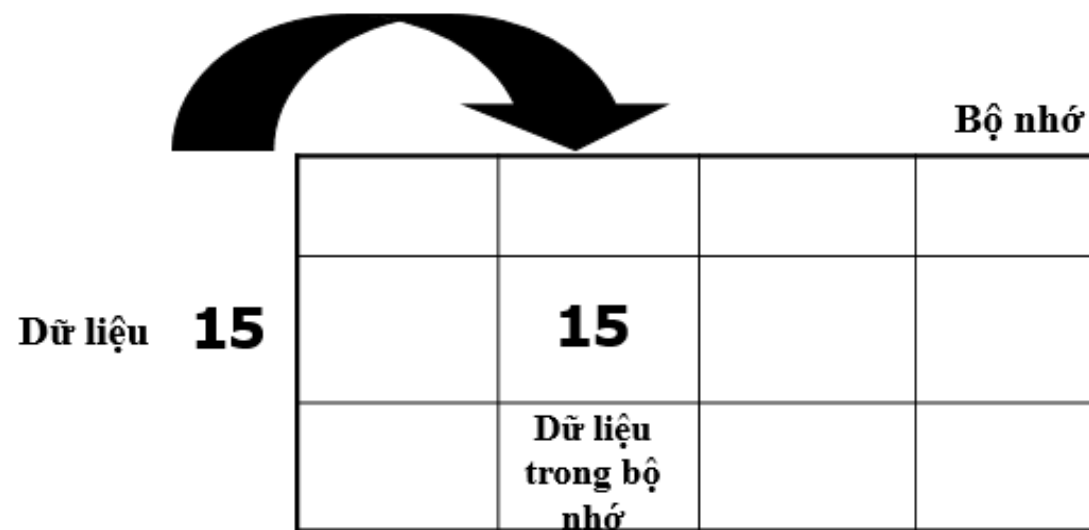
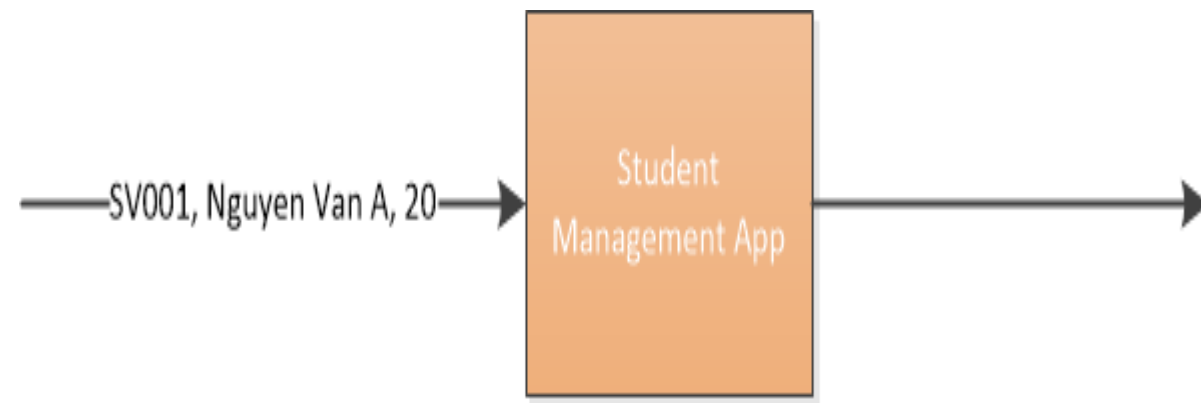
DATATYPE **VARIABLE_NAME**

- Khởi tạo giá trị cho biến

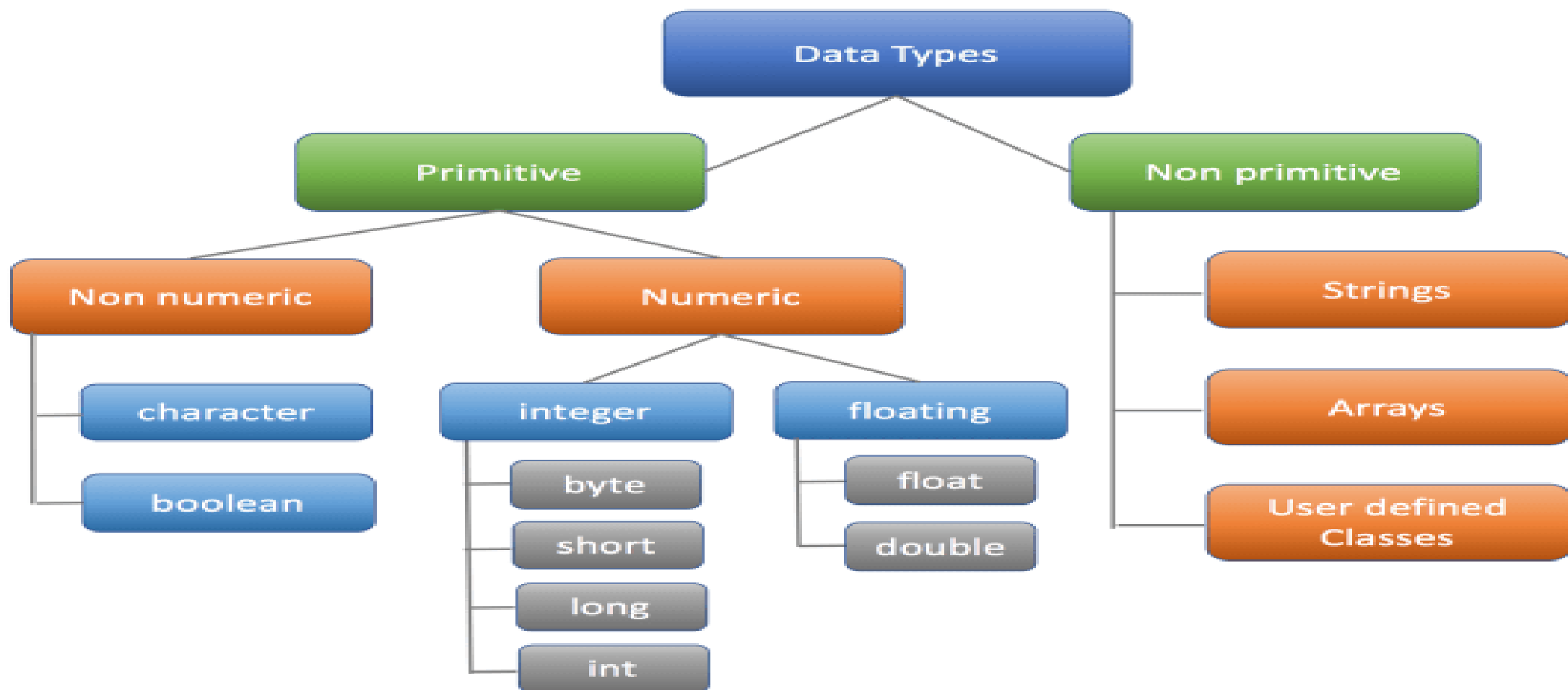
VARIABLE_NAME = **VALUE**

- Khai báo và khởi tạo giá trị cho biến

DATATYPE **VARIABLE_NAME** = **VALUE**



Mỗi vị trí trong bộ nhớ là duy nhất



- **Chứa** ký tự và số, **không** chứa các ký tự đặc biệt **trừ** ký tự “_” và “\$”
- **Không** trùng từ khóa java
- **Bắt đầu** là ký tự viết thường
- **Phân biệt** hoa thường
- **Nên** là danh từ, rõ ràng và có ý nghĩa
- Theo quy tắc **Camel-case**

student...Name **X**

student_status **V**

int **X**

Age **X**

1sttest **X**

class4Name **V**

\$mark **V**

- Cấu trúc khai báo:

final **DATATYPE** **CONSTANT_VARIABLENAME** = **VALUE**

- Hằng số

- Giá trị không thay đổi
- Tên hằng số viết hoa

- Ví dụ: **final float BASIC_SALARY** = 1300000F

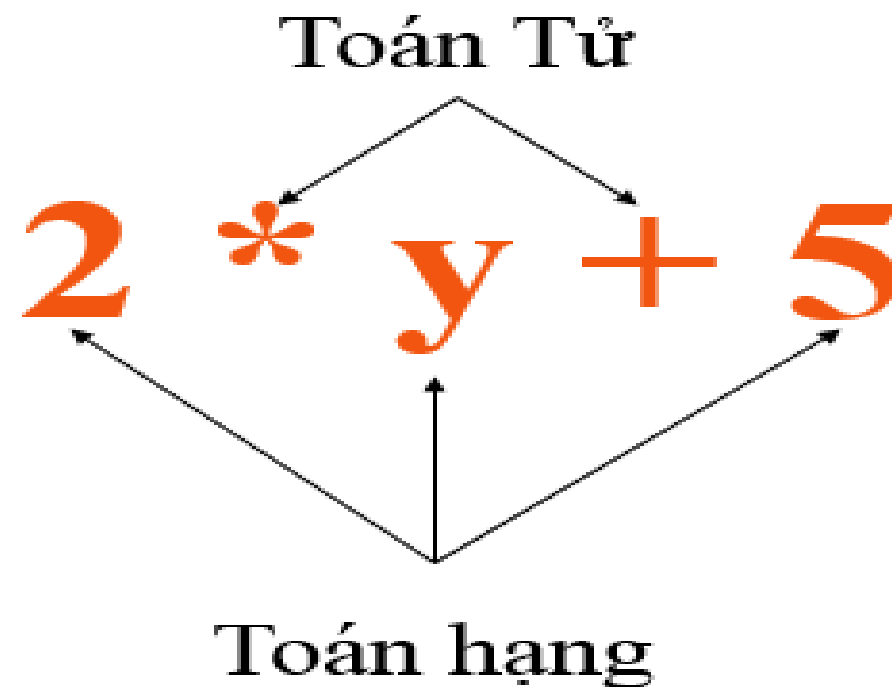
Toán
học

Quan hệ

Logic

Thao tác
Bit

Gán và
hỗn hợp



| Toán tử | Mô Tả | Ví dụ | Kết quả |
|---------|--|------------------------|----------|
| + | Trả về giá trị là tổng của hai toán hạng | $10 + 5$ | 15 |
| - | Trả về giá trị là hiệu của hai toán hạng | $10 - 5$ | 5 |
| * | Trả về giá trị là tích của hai toán hạng | $10 * 5$ | 50 |
| / | Trả về giá trị là thương của hai toán hạng | $10 / 5$ | 2 |
| % | Trả về giá trị là số dư trong phép chia | $10 \% 5$ | 0 |
| ++ | Tăng giá trị lên 1 đơn vị | $10 ++$ | 11 |
| -- | Giảm giá trị xuống 1 đơn vị | $10 --$ | 9 |
| += | Cộng giá trị toán hạng bên phải vào toán hạng bên trái và gán giá trị tổng về toán hạng bên trái | $a = 10$ $a += 5$ | $a = 15$ |
| -= | Trừ giá trị toán hạng bên phải cho toán hạng bên trái và gán giá trị hiệu về toán hạng bên trái | $a = 10$ $a -= 5$ | $a = 5$ |
| *= | Nhân giá trị toán hạng bên phải với toán hạng bên trái và gán giá trị tích cho toán hạng bên trái | $a = 10$ $a *= 5$ | $a = 50$ |
| /= | Chia giá trị toán hạng bên phải cho toán hạng bên trái và gán giá trị thương cho toán hạng bên trái | $a = 10$ $a /= 5$ | $a = 2$ |
| %= | Chia giá trị toán hạng bên phải cho toán hạng bên trái và gán giá trị số dư trong phép chia cho toán hạng bên trái | $a = 10$ $a \% = 5$ | $A = 0$ |

| Toán tử | Mô Tả | Ví dụ | Kết quả |
|--------------|---|---------|---------|
| == | So sánh hai toán hạng có giá trị bằng nhau không | 10 == 5 | false |
| != | So sánh hai toán hạng có giá trị khác nhau không | 10 != 5 | true |
| > | Kiểm tra toán hạng bên trái có lớn hơn toán hạng bên phải không | 10 > 5 | true |
| < | Kiểm tra toán hạng bên trái có nhỏ hơn toán hạng bên phải không | 10 < 5 | false |
| >= | Kiểm tra toán hạng bên trái có lớn hơn hoặc bằng toán hạng bên phải không | 10 >= 5 | true |
| <= | Kiểm tra toán hạng bên trái có nhỏ hơn hoặc bằng toán hạng bên phải không | 10 <= 5 | false |

| Toán tử | Mô Tả | Ví dụ | Kết quả |
|-------------------|--|---|------------------------|
| && | Toán tử Và (AND): Trả về giá trị true nếu cả hai toán hạng là true | true && true true && false false && false | true false false |
| | Toán tử Hoặc (OR): Trả về giá trị true nếu ít nhất một toán hạng là true | true true true false false false | true false false |
| ^ | Toán tử XOR: trả về giá trị true nếu chỉ một và duy nhất một toán hạng là true , còn lại trả về giá trị false | true ^ false true ^ true false ^ false | true false false |
| ! | Toán tử phủ định (NOT): phủ định giá trị | ! true ! false | false true |

| Toán tử | Mô Tả |
|---------|---|
| ~ | Toán tử NOT: Trả về giá trị phủ định của một bit |
| & | Toán tử AND: Trả về giá trị là 1 nếu các toán hạng là 1, các trường hợp khác trả về 0 |
| | Toán tử OR: Trả về giá trị là 1 nếu một trong các toán hạng là 1, các trường hợp khác trả về 0 |
| ^ | Toán tử Exclusive OR: Trả về giá trị là 1 nếu chỉ duy nhất một toán hạng là 1, các trường hợp khác trả về 0 |
| >> | Dịch phải: Chuyển toàn bộ các bit sang phải một vị trí, giữ nguyên dấu. Toán hạng bên trái là số bị dịch, toán hạng bên phải là số vị trí mà các bit cần dịch |
| << | Dịch trái: Chuyển toàn bộ các bit sang trái một vị trí, giữ nguyên dấu. Toán hạng bên trái là số bị dịch, toán hạng bên phải là số vị trí mà các bit cần dịch |

- Toán tử gán (=) : Gán giá trị vào biến

- Toán tử hỗn hợp:

- Cấu trúc

<Expression 1> ? <Expression 2> : <Expression 3>

- Trong đó

Expression 1: Biểu thức logic → Trả về giá trị **true** hoặc **false**

Expression 2: Giá trị trả về nếu Expression 1 là true

Expression 3: Giá trị trả về nếu Expression 2 là false

- Ví dụ

int a = (10%5==0) ? 4 : 2 → a = 4

- Thay đổi thứ tự ưu tiên: Sử dụng dấu ngoặc đơn ()

In một chuỗi không xuống dòng:
System.out.print("String")

In một chuỗi rồi xuống dòng:
System.out.println("String")

In chuỗi có định dạng:
System.out.printf("Control String", values)

| Kiểu dữ liệu | Định dạng dữ liệu |
|--------------|-------------------|
| Int | %d |
| Float | %f |
| Double | %f |
| Char | %c |
| String | %s |
| Boolean | %b |

| Ký tự đặc biệt | Mô tả |
|----------------|---|
| \n | Xuống dòng – đưa con trỏ đến đầu dòng kế tiếp |
| \r | Đưa con trỏ về đầu dòng |
| \t | Đưa con trỏ vào 1 tab |
| \\ | In ký tự ‘ \ ’ |
| \' | In ký tự ‘ ’ ’ |
| \” | In ký tự ‘ ” ’ |

Bước 1

- Khởi tạo đối tượng Scanner (`java.util.Scanner`)
- **`Scanner sc = new Scanner(System.in)`**

Bước 2

- Thông báo nhập dữ liệu

Bước 3

- Nhận giá trị nhập vào từ bàn phím theo định dạng dữ liệu nhập
- Gán giá trị nhập vào biến

| Phương thức | Mô tả |
|------------------------------|--|
| public String next() | Trả về kết quả nội dung trước khoảng trắng |
| public String nextLine() | Trả về kết quả nội dung của chuỗi nhập vào |
| public int nextInt() | Trả về kết quả dữ liệu số nguyên nhập vào |
| public float nextFloat() | Trả về kết quả dữ liệu số thực nhập vào |
| public double nextDouble() | Trả về kết quả dữ liệu số thực nhập vào |
| public boolean nextBoolean() | Trả về kết quả dữ liệu kiểu boolean nhập vào |

| Phương thức | Mô tả |
|--------------------------------|---|
| Integer.parseInt("String") | Chuyển đổi từ chuỗi(String) sang kiểu số nguyên(Int) |
| Float.parseFloat("String") | Chuyển đổi từ chuỗi(String) sang kiểu số thực(Float) |
| Double.parseDouble("String") | Chuyển đổi từ chuỗi(String) sang kiểu số thực(Double) |
| Boolean.parseBoolean("String") | Chuyển đổi từ chuỗi(String) sang kiểu boolean |

HỎI ĐÁP





TRẢI NGHIỆM THỰC HÀNH

HỆ THỐNG ĐÀO TẠO CNTT QUỐC TẾ BACHKHOA - APTECH



TRÂN TRỌNG CẢM ƠN!



238 Hoàng Quốc Việt, Bắc Từ Liêm, Hà Nội



0968.27.6996



tuyensinh@bachkhoa-aptech.edu.vn



www.bachkhoa-aptech.edu.vn