

# Chương 1

## Giới thiệu phương pháp lập trình hướng đối tượng

<b>Giáo viên</b>	: ThS. Trần Văn Thọ
<b>Đơn vị</b>	: Bộ môn KTHT & MMT

Xét một công việc sau

Bạn muốn xây một ngôi nhà?

# Chia nhỏ công việc



Đào móng

Tầng 1

Tầng 2

Tầng 3

Tường bao

San lấp mặt bằng

Đào hố

Cắm cọc

Dựng khung

Đổ móng

Nhổ cây

Dọn rác

Lấp đất

...

## Vấn đề ?

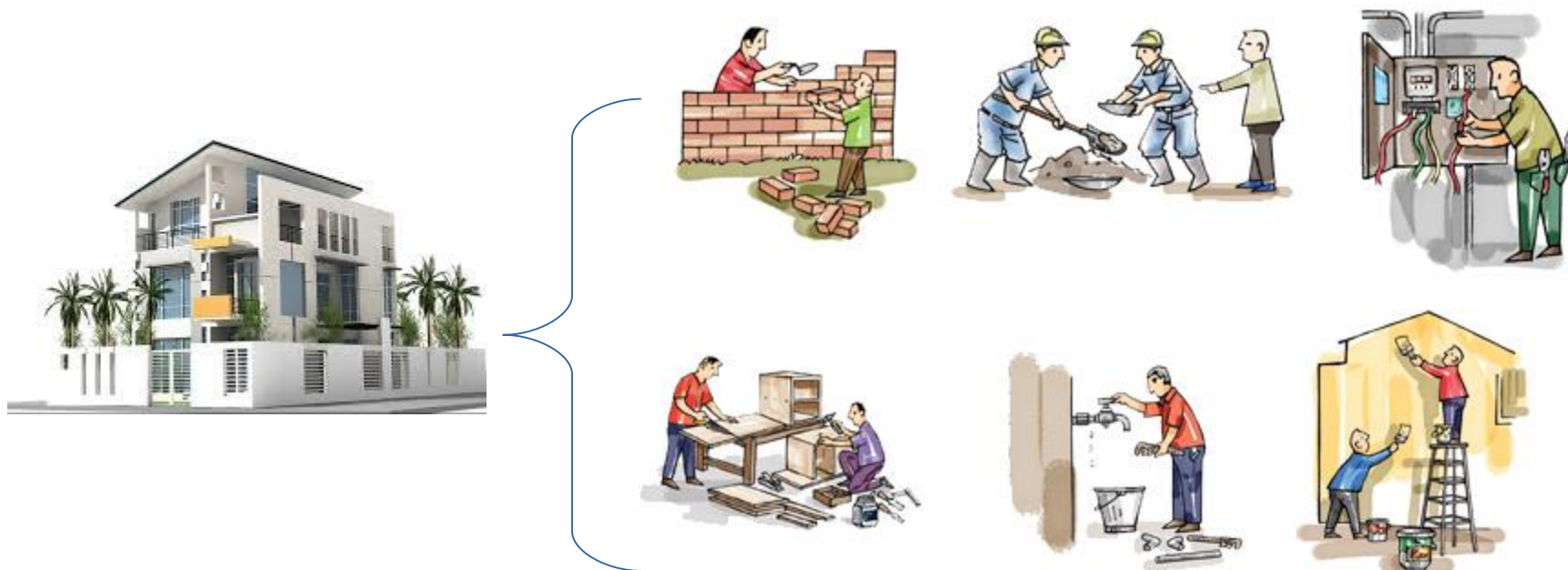
Công việc lớn, phức tạp

Cần tính chuyên môn hóa, bảo mật

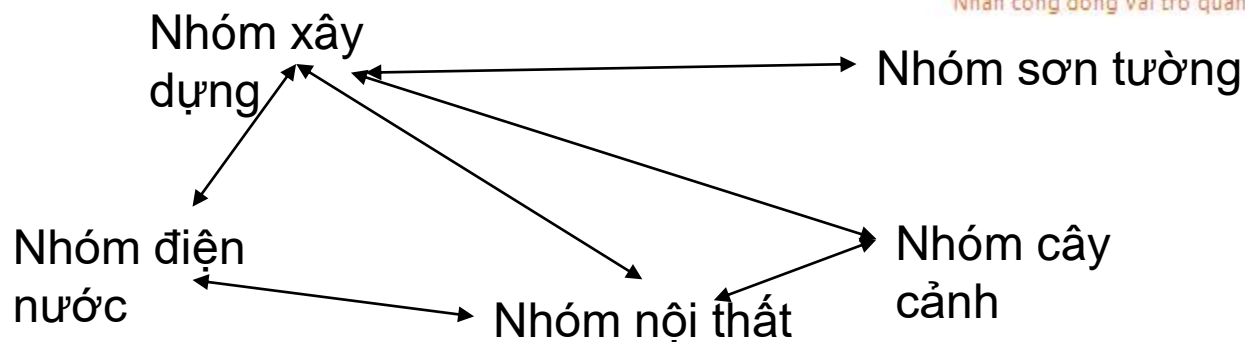
Cần sự kế thừa, tái sử dụng

Gần với thực tế

# Kết hợp của nhiều đối tượng



Nhân công đóng vai trò quan trọng trong công trình của bạn



**GIẢI QUYẾT VẤN ĐỀ BẰNG CÁCH  
KẾT HỢP HOẠT ĐỘNG CỦA CÁC ĐỐI  
TƯỢNG**

# 1. Phương pháp lập trình

## 1.1 Khái niệm:

*Phương pháp lập trình (programming paradigm) là một kiểu lập trình chương trình máy tính, nó bao gồm các cách thức xây dựng cấu trúc chương trình và các thành phần trong chương trình.*

- Cách nhìn nhận bản chất của bài toán
- Cách tư duy khi phân tích vấn đề
- Cách xây dựng các thành phần tạo nên chương trình

# 1. Phương pháp lập trình

## 1.2 Các phương pháp lập trình chính:

- Lập trình tuyến tính
- Lập trình thủ tục (Procedural programming)
- Lập trình hướng đối tượng (Object-oriented programming)
- Lập trình hàm (Functional programming)
- Lập trình logic (Logic programming)
- Lập trình ngữ nghĩa (Symbolic programming)

# 1. Phương pháp lập trình

## 1.2 Các phương pháp lập trình chính:

### a) Lập trình tuyến tính:

- Là phương pháp lập trình đầu tiên khi máy tính mới ra đời
- Chương trình chạy theo 1 chiều tuần tự từ trên xuống qua từng dòng lệnh, không có các lệnh có cấu trúc, chỉ có các lệnh goto.
- Không có cơ chế kiểm soát dữ liệu, mọi dữ liệu đều là toàn cục
- Chỉ phù hợp khi xây dựng các chương trình nhỏ, đơn giản

Ngôn ngữ lập trình tiêu biểu: Assembly



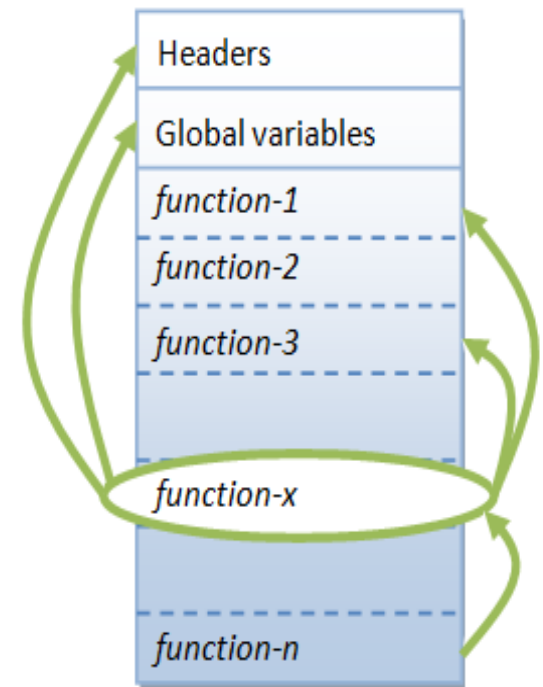
# 1. Phương pháp lập trình

## 1.2 Các phương pháp lập trình chính:

### b) Lập trình thủ tục:

- Chương trình được tổ chức thành các chương trình con (hàm, thủ tục), mỗi chương trình con đảm nhận một nhiệm vụ riêng và có thể được tạo thành bởi nhiều chương trình con nhỏ hơn.
- Các dữ liệu (biến, hằng) đã được quy định phạm vi truy cập khi khai báo
- Khả năng tái sử dụng không cao do các hàm rất khó để có thể dùng trong nhiều chương trình khác nhau (khả năng đóng gói không tốt)
- Khó mô tả các bài toán phức tạp trong thực tế.

Ngôn ngữ tiêu biểu: C, Pascal,...



A function (in C) is not well-encapsulated

# 1. Phương pháp lập trình

## 1.2 Các phương pháp lập trình chính:

### c ) Lập trình hướng đối tượng:

- Thiết kế chương trình xoay quanh dữ liệu của hệ thống:
  - Các dữ liệu được nhóm thành các thực thể hay đối tượng
  - Mỗi đối tượng chứa dữ liệu và các hành động của riêng nó.
  - Các chức năng của chương trình là các tương tác qua lại giữa các đối tượng
- Có tính đóng gói và khả năng sử dụng lại cao
- Phù hợp với mọi loại chương trình từ nhỏ đến rất lớn
- Có khả năng mô tả được các quan hệ phức tạp trong các bài toán thực tế
- Ngôn ngữ tiêu biểu: Java, C++,...

## 2. Phương pháp lập trình hướng đối tượng

### 2.1 Khái niệm:

*Lập trình hướng đối tượng là phương pháp lập trình dựa trên các khái niệm về “đối tượng”, là các cấu trúc chứa cả dữ liệu và phương thức xử lý*

## 2. Phương pháp lập trình hướng đối tượng

### 2.2 Đặc điểm của lập trình hướng đối tượng:

- Tập trung vào dữ liệu
- Chương trình được chia thành các đối tượng
- Các cấu trúc dữ liệu được thiết kế sao cho đặc tả được các đối tượng
- Các hàm thao tác trên các vùng dữ liệu của đối tượng được gắn với cấu trúc dữ liệu đó
- Dữ liệu được đóng gói lại, được che giấu và không cho phép các hàm ngoại lai truy cập tự do
- Các đối tượng tác động và trao đổi thông tin qua các hàm
- Có thể dễ dàng bổ sung dữ liệu và các hàm mới vào đối tượng nào đó khi cần
- Chương trình được thiết kế từ dưới lên.

## 2. Phương pháp lập trình hướng đối tượng

### 2.3 Một số khái niệm trong lập trình hướng đối tượng

- Đối tượng (Object): là cấu trúc kết hợp cả dữ liệu và thủ tục (phương thức) thao tác trên các dữ liệu đó.

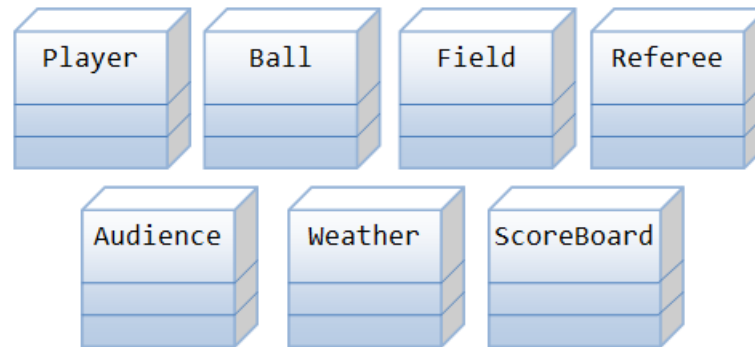
Đối tượng = dữ liệu + phương thức

- Lớp (class): là một tập các đối tượng cùng loại (cùng kiểu dữ liệu và phương thức). Có thể coi lớp là khuôn mẫu để “đúc” ra các đối tượng.
- Thừa kế: định nghĩa một lớp mới (lớp con) dựa trên các lớp đã tồn tại (lớp cha), có bổ sung thêm các dữ liệu và phương thức mới.
- Tính đa hình

## 2. Phương pháp lập trình hướng đối tượng

### 2.4 Các tính chất của lập trình hướng đối tượng

- Tính trừu tượng hóa dữ liệu:
  - Các thực thể trong bài toán thực tế có thể được mô tả thành các đối tượng với phần dữ liệu và phương thức, phản ánh được nội dung và hành động của đối tượng



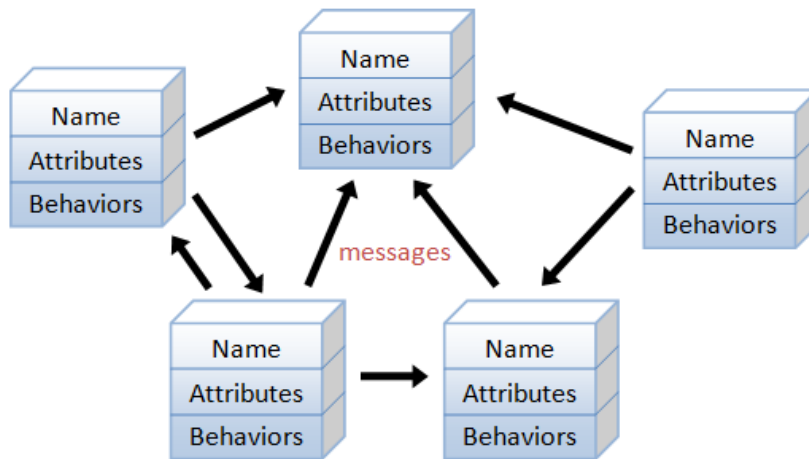
Classes (Entities) in a Computer Soccer Game

## 2. Phương pháp lập trình hướng đối tượng

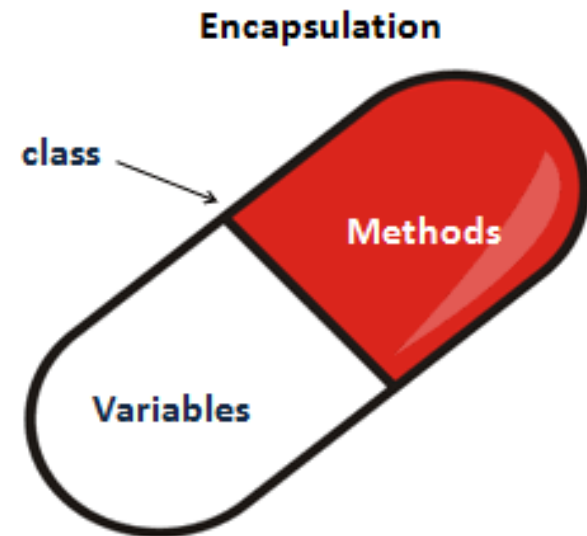
### 2.4 Các tính chất của lập trình hướng đối tượng

- Tính đóng gói dữ liệu:

- ✓ Các đối tượng có phần dữ liệu được bảo vệ, chỉ cho phép các phương thức trong đối tượng được tương tác. Các đối tượng muốn trao đổi dữ liệu phải thông qua các hàm.



An object-oriented program consists of many well-encapsulated objects and interacting with each other by sending messages



## 2. Phương pháp lập trình hướng đối tượng

### 2.4 Các tính chất của lập trình hướng đối tượng

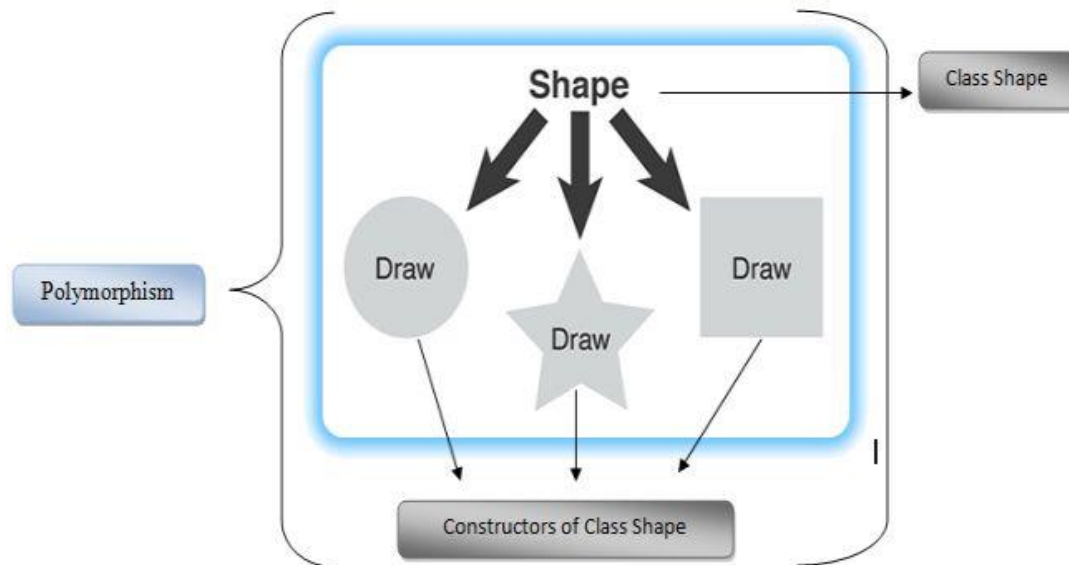
- Tính thừa kế:
  - ✓ Cho phép một đối tượng có thể có sẵn các đặc tính của đối tượng khác thông qua kế thừa, cho phép mở rộng các đối tượng sẵn có mà không cần định nghĩa lại



## 2. Phương pháp lập trình hướng đối tượng

### 2.4 Các tính chất của lập trình hướng đối tượng

- Tính đa hình:
  - ✓ Cho phép lớp con định nghĩa lại các hàm ở lớp cha, khi gọi các hàm này thì tùy theo các hoàn cảnh cụ thể sẽ gọi ra các hàm ở các lớp con tương ứng.



## 2. Phương pháp lập trình hướng đối tượng

### 2.5 Các ưu điểm của lập trình hướng đối tượng

- Tính đóng gói và kế thừa làm tăng khả năng sử dụng lại các thành phần của chương trình
- Cho phép xây dựng các quy trình cho việc phát triển phần mềm một cách hiệu quả.
- Tính đóng gói cũng cho phép bảo vệ các thành phần của chương trình một cách an toàn và không bị thay đổi bởi các đoạn chương trình khác.

#### **Mô tả tốt hơn các đối tượng trong thế giới thực**

- Dễ cài đặt hơn
- Dễ mở rộng, nâng cấp thành các hệ thống lớn
- Kỹ thuật trao đổi thông tin giữa các đối tượng làm việc mô tả giao diện với các hệ thống bên ngoài đơn giản hơn
- Cho phép quản lý độ phức tạp của các sản phẩm phần mềm một cách hiệu quả.

## 2. Phương pháp lập trình hướng đối tượng

### 2.6 Các nhược điểm của lập trình hướng đối tượng

- Có thể trở nên rất phức tạp về thiết kế và kiến trúc.
- Tốn nhiều thời gian hơn nếu phát triển từ đầu.
- Khó học hơn lập trình thủ tục.

### 3. Ngôn ngữ lập trình Java

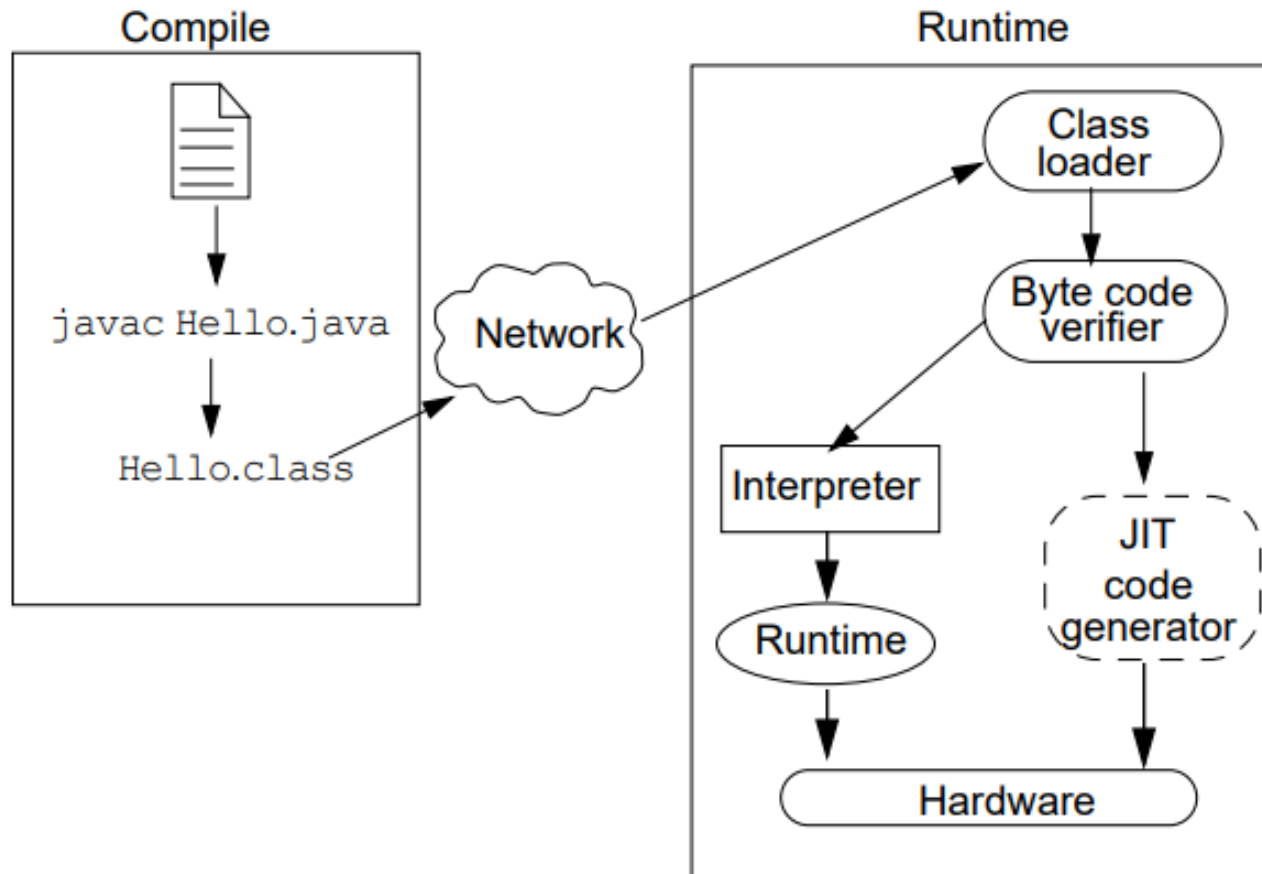
- Java là ngôn ngữ lập trình hướng đối tượng thuần túy
- Java được thiết kế để có thể chạy trên mọi nền tảng
- Ứng dụng của Java rất đa dạng:
  - ✓ Ứng dụng Desktop
  - ✓ Lập trình Web
  - ✓ Lập trình các hệ thống phần mềm doanh nghiệp
  - ✓ Lập trình di động
  - ✓ Lập trình nhúng
  - ✓ Game

## 3. Ngôn ngữ lập trình Java

### 3.1 Đặc điểm của Java

- Là ngôn ngữ thuần túy hướng đối tượng, bảo mật và đa năng
- Cú pháp của Java tương tự cú pháp C++
- Java loại bỏ nhiều đặc điểm khó hiểu và ít dùng như con trỏ, chồng toán tử...
- Java có sẵn cơ chế thu gom giải phóng bộ nhớ nên không cần phải hủy đối tượng sau khi dùng
- Java chạy trên một môi trường riêng (JVM) nên nó độc lập với nền tảng
- Java là ngôn ngữ thông dịch
- Mã nguồn .java được dịch ra mã bytecode (các file .class), rồi khi chạy, mã byte code được kiểm tra rồi thực thi trên JVM

### 3. Ngôn ngữ lập trình Java



Quá trình dịch và thực thi mã nguồn của Java

### 3. Ngôn ngữ lập trình Java

```
public class HelloWorldApp{  
    public static void main (String args[]) {  
        System.out.println("Hello World!");  
    }  
}
```

#### Chương trình Java cơ bản nhất

Làm thế nào để biên dịch và chạy chương trình java?

- Sử dụng các công cụ trong JDK :

**javac** HelloWorldApp.java

**java** HelloWorldApp

- Đơn giản hơn: sử dụng công cụ tích hợp trong IDE

## 3. Ngôn ngữ lập trình Java

### 3.2 Môi trường phát triển Java

- JVM (Java Virtual Machine): máy ảo Java là một môi trường riêng cho ứng dụng java hoạt động, có vai trò tải và thực thi bytecode với mỗi hệ điều hành.
- JRE (Java Runtime Environment): tập hợp các công cụ phần mềm để chạy chương trình Java (gồm JVM+ các thư viện và file mà JVM cần để hoạt động)
- JDK (Java Development Kit): tập hợp các công cụ phần mềm để chạy và phát triển ứng dụng Java (gồm JRE+ các công cụ phát triển)



## 3. Ngôn ngữ lập trình Java

### 3.3 Một số đặc trưng ngôn ngữ Java

#### 3.3.1 Kiểu dữ liệu:

- Kiểu logic: **boolean**
- Kiểu ký tự: **char**
- Kiểu số nguyên: **byte, short, int, long**
- Kiểu số thực: **double, float**
- Kiểu dữ liệu phức hợp: **String**, mảng
- Kiểu dữ liệu khác: **ArrayList, List, Stack, ...**

Kiểu dữ liệu  
cơ bản

Kiểu lớp

#### 3.3.2 Biểu thức:

- Tương tự **C++**

### 3. Ngôn ngữ lập trình Java

#### Các toán tử

Separator	.    []    ()    ;    ,
R to L	++   --   +   -   ~   !   ( <i>data type</i> )
L to R	*   /   %
L to R	+   -
L to R	<<   >>   >>>
L to R	<   >   <=   >=   instanceof
L to R	==   !=
L to R	&
L to R	^
L to R	
L to R	&&
L to R	
R to L	?:
R to L	=   *=   /=   %=   +=   -=   <<=   >>=   >>>=   &=   ^=    =

## 3. Ngôn ngữ lập trình Java


### 3.3 Một số đặc trưng ngôn ngữ Java

#### 3.3.3 Luồng điều khiển chương trình (tương tự C++)

- ✓ Rẽ nhánh : **if-else**, **switch –case**
- ✓ Lặp: **for**, **while**, **do-while**
  - Sử dụng **break** và **continue** với **label**: cho phép điều khiển vòng lặp lồng nhau.

VD: sử dụng break kết hợp với label cho phép thoát ra khỏi vòng lặp ngoài cùng

**label**



```
loop:
do
{
    statement;
    do
    {
        statement;
        statement;
        if (boolean expression)
            break loop;
    } while (boolean expression)
    statement;
} while (boolean expression);
```

## 3. Ngôn ngữ lập trình Java

### 3.3 Một số đặc trưng ngôn ngữ Java

#### 3.3.4 Khai báo biến và hằng dữ liệu

- Khai báo biến với kiểu cơ bản:

Cú pháp tương tự C++:

**<kiểu> <tên\_biến>;**

*Tên biến: chỉ bao gồm chữ cái, số, gạch dưới và dấu \$*

Ví dụ:

```
int x,y;
```

```
boolean check=false;
```

```
float $var1=1.25f; //ký tự f cho biết 1.25 có kiểu float
```

## 3. Ngôn ngữ lập trình Java

### 3.3 Một số đặc trưng ngôn ngữ Java

#### 1.3.3 Khai báo biến và hằng dữ liệu

– Khai báo hằng:

- + Trong java không có khái niệm hằng

- + Sử dụng từ khóa final cho các biến không thay đổi giá trị

Ví dụ:

```
final int MAX=100;
```

## 3. Ngôn ngữ lập trình Java

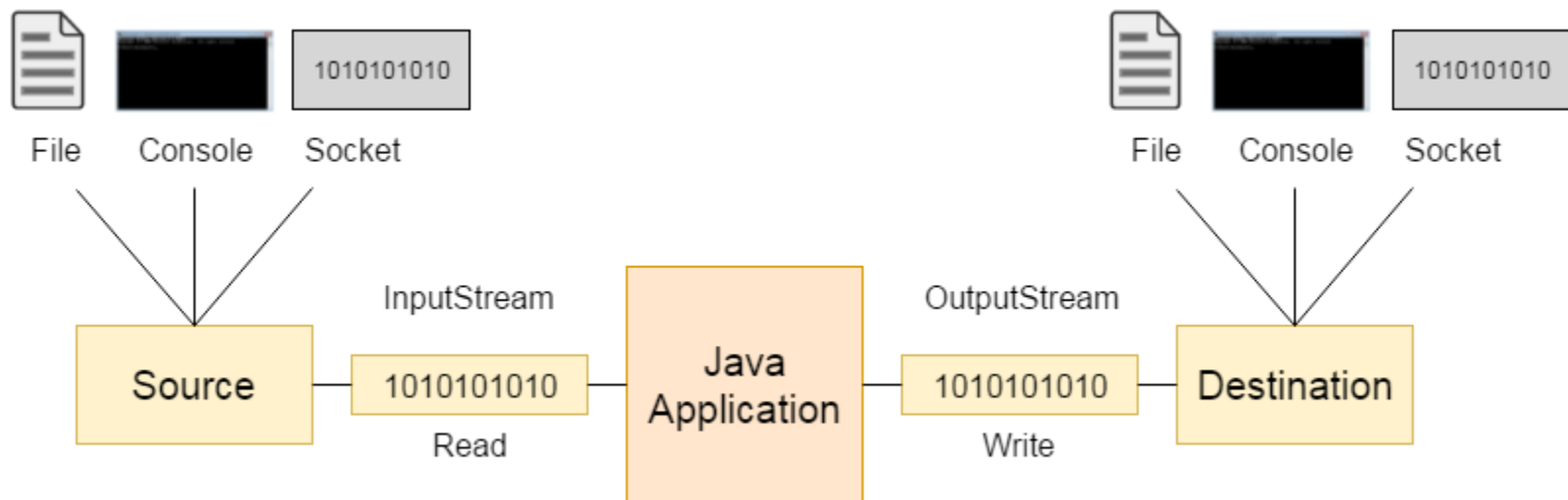
### 3.3 Một số đặc trưng ngôn ngữ Java

#### 1.3.4 Nhập xuất dữ liệu cơ bản trong Java:

- Java sử dụng khái niệm luồng dữ liệu để thực hiện vào ra dữ liệu giữa chương trình với bên ngoài (file, cửa sổ dòng lệnh, socket)

+ Luồng vào chuẩn: **System.in**

+ Luồng ra chuẩn: **System.out**



## 3. Ngôn ngữ lập trình Java

### 3.3 Một số đặc trưng ngôn ngữ Java

#### 1.3.4 Nhập xuất dữ liệu cơ bản trong Java:

- + Sử dụng `System.out` để ghi dữ liệu ra đầu ra chuẩn: dùng hàm **print** hoặc **println**.

Ví dụ:

```
int x=1;  
System.out.print("Giá trị của x=");  
System.out.println(x);
```

Hoặc:

```
System.out.println("Giá trị của x=" + x);  
//x tự động được convert về chuỗi ký tự
```



## 3. Ngôn ngữ lập trình Java

### 3.3 Một số đặc trưng ngôn ngữ Java

#### 1.3.4 Nhập xuất dữ liệu cơ bản trong Java:

- Sử dụng `System.in` để đọc dữ liệu ra đầu vào chuẩn
  - + Chỉ đọc được theo từng byte

Ví dụ:

```
int i = System.in.read();      System.out.println((char)i);
```

- Cách dễ hơn để đọc dữ liệu : sử dụng lớp `Scanner`

Ví dụ:

```
Scanner myObj = new Scanner(System.in);  
System.out.println("Enter username");  
String userName = myObj.nextLine();  
System.out.println("Username is: " + userName);  
int age = myObj.nextInt();  
double salary = myObj.nextDouble();  
System.out.println("Age: " + age);  
System.out.println("Salary: " + salary);
```

# Bài tập

Bài tập 1: Tìm số nguyên lẻ nhỏ nhất có 3 chữ số abc sao cho thỏa mãn điều kiện

$$abc = a^3 + b^3 + c^3$$

Bài tập 2: Nhập vào tháng, năm từ bàn phím. Viết chương trình hiển thị nên số ngày của tháng và năm vừa nhập.

## 3. Ngôn ngữ lập trình Java

### 3.3.5 Mảng

- Mảng trong Java có kiểu đối tượng (Object)
- Khai báo mảng:

Cú pháp: <kiểu> <tên biến mảng> [];

Hoặc <kiểu> [] <tên biến mảng>;

- Khởi tạo mảng:

Cú pháp: <tên biến mảng> = **new** <Kiểu> [số phần tử];

VD:

```
String names[];  
names = new String[3];  
names[0] = "Georgianna";  
names[1] = "Jen";  
names[2] = "Simon";
```

## 3. Ngôn ngữ lập trình Java

### 3.3.5 Mảng

Duyệt mảng: bắt đầu từ phần tử thứ 0

Ví dụ:

```
int list[] = new int [10];  
for (int i = 0; i < list.length; i++)  
{  
    System.out.println(list[i]);  
}
```

Khởi tạo lại biến mảng -> mảng mới

Ví dụ:

```
int elements[] = new int[6];  
elements = new int[10];
```

## 3. Ngôn ngữ lập trình Java

### 3.3.5 Mảng

#### + Mảng 2 chiều:

- Mảng mà mỗi phần tử là một mảng

Ví dụ:

```
int twoDim [][] = new int [4] [];  
twoDim[0] = new int[5];  
twoDim[1] = new int[5];
```

- Khởi tạo mảng với kích thước xác định

Ví dụ:

```
int twoDim[][] = new int[4][5];
```

- Chiều thứ nhất luôn có kích thước xác định

Ví dụ:

```
int twoDim [][] = new int [][4]; //sai
```

## 3. Ngôn ngữ lập trình Java

### 3.3.6 Xâu ký tự

- + Sử dụng kiểu lớp **String** để biểu diễn xâu ký tự.
- + Khai báo và gán dữ liệu cho xâu ký tự:

Ví dụ:

```
String name;//khai báo
```

```
name="Nguyễn Văn A";//gán dữ liệu cho xâu
```

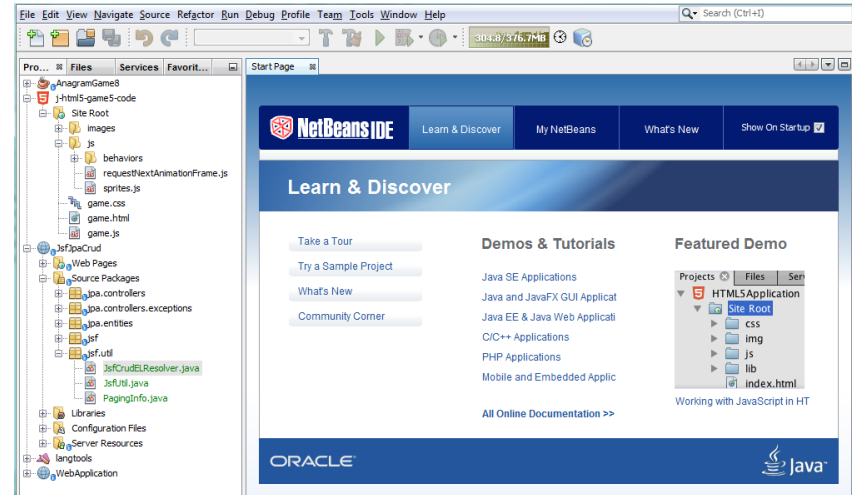
```
String fullname="Nguyễn Văn B";//vừa khai báo vừa  
gán dữ liệu
```

- + Nối hai xâu ký tự: sử dụng toán tử +
- + Một số phương thức với đối tượng kiểu String:
  - ✓ length: lấy độ dài xâu
  - ✓ compareTo: so sánh
  - ✓ indexOf : tìm kiếm xâu con

## 4. Công cụ lập trình hướng đối tượng với Java

### Apache Netbeans:

- + Công cụ miễn phí từ Apache Software Foundation
- + Dễ sử dụng, hỗ trợ nhiều ngôn ngữ và framework, đặc biệt với các nền tảng Java

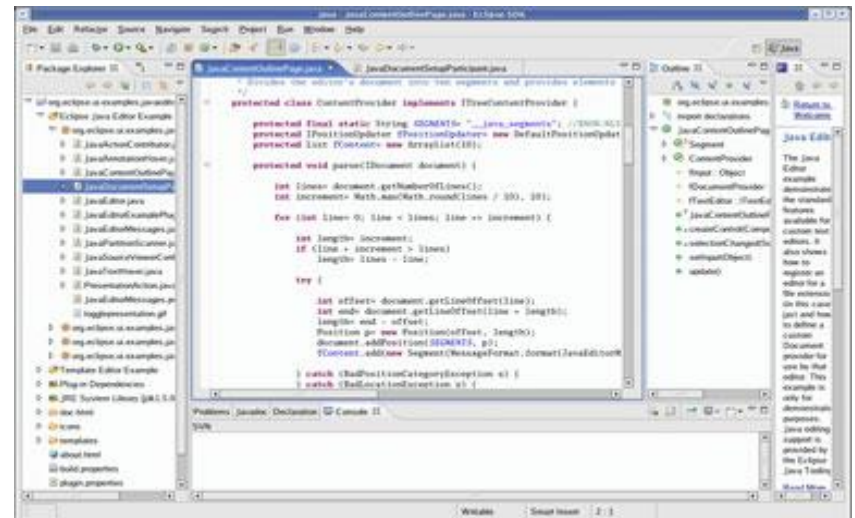


### Eclipse

- + Công cụ phát triển Java và C++ miễn phí mã nguồn mở phổ biến nhất.
- + Có hệ thống plugin mở rộng đa dạng và mạnh mẽ.

### Các công cụ khác

- IntelliJ IDEA
- Jdeveloper
- JCreator



# Bài tập

Bài tập 1: Cho ma trận  $A_{n \times m}$  gồm  $n$  hàng và  $m$  cột

- a) Nhập ma trận  $A$  từ bàn phím.
- b) Tính giá trị lớn nhất trên từng hàng của ma trận lưu vào véc tơ  $B$ .
- c) Hoán đổi hàng chứa phần tử có giá trị lớn nhất đầu tiên trong ma trận (ưu tiên đếm theo chỉ số hàng) cho hàng  $K$  ( $K$  nhập từ bàn phím với  $0 < K < n$ ).