



NGÔN NGỮ JAVA CƠ BẢN

Giảng viên: Đỗ Ngọc Như Loan

GIỚI THIỆU JAVA

- Java là ngôn ngữ lập trình hướng đối tượng kế thừa trực tiếp từ C/C++, được phát triển bởi **Sun Microsystems** vào năm 1995.
- Java là 1 trong những ngôn ngữ lập trình phổ biến nhất.
- **JDK** (Java Development Kit) là một công cụ phần mềm cung cấp môi trường lập trình để viết và thực thi một chương trình Java.
- Chương trình viết bằng Java sẽ được biên dịch ra mã java bytecode. Sau đó **JVM** (Java Virtual Machine) - máy ảo Java sẽ chuyển java bytecode thành mã máy tương ứng.



ĐẶC ĐIỂM CỦA JAVA

- **Đơn giản:** Cú pháp dựa trên C/C++ nhưng đã lược bỏ phép toán con trỏ và đa kế thừa.
- **Hướng đối tượng:** Java là một ngôn ngữ lập trình hướng đối tượng.
- **Độc lập nền tảng:** Chương trình Java có thể chạy trên nhiều thiết bị, nhiều hệ điều hành khác nhau chỉ cần môi trường đó có cài máy ảo java (JVM).



ĐẶC ĐIỂM CỦA JAVA

- **Mạnh mẽ:** Quá trình giải phóng bộ nhớ được thực hiện tự động, có cơ chế xử lý ngoại lệ và kiểm tra lỗi nghiêm ngặt.
- **Bảo mật:** Hạn chế tối đa các vấn đề bảo mật tiềm ẩn. Java bỏ sử dụng pointer, ép kiểu chặt chẽ và cơ chế xác minh code nghiêm ngặt. Chương trình Java chạy trong máy ảo JVM nên lỗi phát sinh dễ dàng kiểm soát, không ảnh hưởng tới hệ thống.
- **Đa luồng:** Java cho phép chạy nhiều tiến trình chạy song song trong một thời điểm và tương tác lẫn nhau.



CÁC BƯỚC TẠO 1 CHƯƠNG TRÌNH

- Xác định mục tiêu của chương trình.
- Xác định những phương pháp giải quyết vấn đề.
- Viết chương trình để giải quyết vấn đề.
- Thực thi chương trình để xem kết quả.



NGÔN NGỮ JAVA CƠ BẢN

- Dấu mở và đóng ngoặc nhọn “{” và “}” là bắt đầu và kết thúc một khối lệnh.
- Dấu chấm phẩy “;” để kết thúc một dòng lệnh.
- Java được tổ chức theo lớp (class). Các lệnh và các hàm (kể cả hàm main) phải thuộc một lớp nào đó, chúng không được đứng bên ngoài của lớp.
- Ngôn ngữ Java hỗ trợ ba kiểu chú thích sau:

`/* text */`

`// text`

`/** documentation */` công cụ javadoc trong bộ JDK sử dụng chú thích này để tự động phát sinh tài liệu.



NGÔN NGỮ JAVA CƠ BẢN

// File: Program.java

```
class Program
```

```
{
```

```
    public static void main(String[] args)
```

```
    {
```

```
        System.out.println( "Hello World" );
```

```
    }
```

```
}
```

Lưu ý: Tên lớp chứa hàm main phải giống tên file



Kiểu dữ liệu cơ sở

Kiểu	Kích thước	Khoảng giá trị
<i>Kiểu số nguyên</i>		
byte	8 bits	-256...255
short	16 bits	-32768...32767
int	32 bits	$-2^{32} \dots 2^{32} - 1$
long	64 bits	$-2^{64} \dots 2^{64} - 1$
<i>Kiểu số thực</i>		
float	32 bits	-3.4e38...3.4e38
double	64 bits	-1.7e308...1.7e308
<i>Kiểu khác</i>		
char	16 bits	Ký tự Unicode
boolean	1 bit	true/false



ĐỊNH DANH (IDENTIFIER)

- Định danh là dùng biểu diễn tên của biến, của phương thức, của lớp.
- Trong Java, định danh có thể sử dụng ký tự chữ, ký tự số và ký tự dấu.
- Ký tự đầu tiên phải là ký tự chữ, dấu gạch dưới (`_`), hoặc dấu dollar (`$`).
- Có sự phân biệt giữa ký tự chữ hoa và chữ thường.

Ví dụ: `Hello`, `_prime`, `var8`, `tvLang`



BIẾN

- Tên biến: là một định danh hợp lệ. Không phải từ khóa. Duy nhất trong phạm vi (scope) của nó.
- **<kiểu dữ liệu> <tên biến>;**
- **<kiểu dữ liệu> <tên biến>=<giá trị>;**
- Ví dụ:
 - `int x;`
 - `int y=0;`
 - `char c;`
 - `string s;`
 - `float r=5f;`



HẲNG

- **final <kiểu dữ liệu> <tên hằng>=<giá trị>;**

- Ví dụ:

```
final double PI=3.14;
```

```
final int MAX_SPEED=90;
```



TOÁN TỬ SỐ HỌC

Toán tử	Ý nghĩa
+	Cộng
-	Trừ
*	Nhân
/	Chia nguyên
%	Chia dư
++	Tăng 1
--	Giảm 1



TOÁN TỬ QUAN HỆ VÀ LOGIC

Toán tử	Ý nghĩa
==	So sánh bằng
!=	So sánh khác
>	So sánh lớn hơn
<	So sánh nhỏ hơn
>=	So sánh lớn hơn hay bằng
<=	So sánh nhỏ hơn hay bằng
	OR (biểu thức logic)
&&	AND (biểu thức logic)
!	NOT (biểu thức logic)



HÀM CHUYỂN ĐỔI KIỂU

- `Integer.parseInt(chuỗi số)`: đổi chuỗi số thành số nguyên
- `Float.parseFloat(chuỗi số)`: đổi chuỗi số thành số thực (kiểu float)
- `Double.parseDouble(chuỗi số)`: đổi chuỗi số thành số thực (kiểu double)
- `<tên biến>.toString()`: đổi thành chuỗi
- `String.valueOf()`: đổi thành chuỗi
- Ví dụ:

```
int i = Integer.parseInt("123");  
String s = i.toString();
```



NHẬP DỮ LIỆU TỪ BÀN PHÍM

○ Sử dụng lớp **BufferedReader**:

```
import java.io.BufferedReader; // cần import ở đầu file
BufferedReader inStream =
new BufferedReader(new InputStreamReader(System.in));
String s = inStream.readLine();
```

○ Sử dụng lớp **Scanner**:

```
import java.util.Scanner; // cần import ở đầu file
Scanner reader = new Scanner(System.in);
String s = reader.nextLine();
int i = reader.nextInt();
```



XUẤT DỮ LIỆU RA MÀN HÌNH

- `System.out.print(...);`
- `System.out.println(...);`
- `System.out.printf(...);`
- Ví dụ:

```
System.out.println("Xin chào!");
```

```
System.out.print("sum="+s);
```

```
System.out.println("a = " + a + ", b = " + b + ", c = " + c);
```

```
System.out.printf("a = %i, b = %i, c = %i", a, b, c);
```



VÍ DỤ

```
import java.io.*;

public class Example
{
    public static void main(String[] args) throws Exception
    {
        BufferedReader inStream =
            new BufferedReader(new InputStreamReader(System.in));
        System.out.print("Nhap mot so nguyen:");
        String siNumber = inStream.readLine();
        int iNumber = Integer.parseInt(siNumber);
        System.out.print("Nhap mot so thuc:");
        String sfNumber = inStream.readLine();
        float fNumber = Float.parseFloat(sfNumber);

        System.out.println("So nguyen:“ + iNumber);
        System.out.println("So thuc:“ + fNumber);
    }
}
```



MẢNG

- **<kiểu dữ liệu> <tên mảng>[];**
- **<kiểu dữ liệu>[] <tên mảng>;**
- Ví dụ:


```
int arrInt[] = new int[100];  
int[] arrInt = new int[100];  
int[][] arr = new int[4][2];
```
- Có thể **khởi tạo giá trị ban đầu** cho các phần tử của mảng khi nó được khai báo.
- Ví dụ:

```
int arrInt[] = {1, 2, 3};  
char arrChar[] = {'a', 'b', 'c'};  
String arrString[] = {"ABC", "EFG", 'GHI'};
```



VÍ DỤ

```
class ArrayDemo
{
    public static void main(String args[])
    {
        int arrInt[] = new int[10];
        int i;
        for(i = 0; i < 10; i = i+1)
            arrInt[i] = i;
        for(i = 0; i < 10; i = i+1)
            System.out.println("This is arrInt[" + i + "]: " + arrInt[i]);
    }
}
```



CÁC CẤU TRÚC ĐIỀU KHIỂN

- IF
- SWITCH
- FOR
- WHILE
- DO .. WHILE
- FOR-EACH



IF

```
if (condition) {  
    statements;  
}  
[else  
{  
    statements;  
}]
```

Ví dụ:

```
int x = 3;  
if(x > 2) {  
    System.out.println("X > 2");  
}  
else {  
    System.out.println("X <= 2");  
}
```



VÍ DỤ

```
int x = 3;  
if (x > 2) {  
    System.out.println("X > 2");  
}  
else if (x < 2) {  
    System.out.println("X < 2");  
} else {  
    System.out.println("X = 2");  
}
```



SWITCH

```
switch (expression)
{
    case <const value 1>:
        statements;
        break;
    case <const value 2>:
        statements;
        break;
    [default:
        statements;
}
```



VÍ DỤ

```
int day = 2;
switch (day)
{
case 0 : System.out.println("Sunday");
        break;
case 1 : System.out.println("Monday");
        break;
case 2 : System.out.println("Tuesday");
        break;
case 3 : System.out.println("Wednesday");
        break;
case 4 : System.out.println("Thursday");
        break;
case 5: System.out.println("Friday");
        break;
case 6 : System.out.println("Satuday");
        break;
default:
        System.out.println("Invalid day of week");
}
```



FOR

```
for (initialization; condition; increment/decrement) {  
    statements;  
}
```

Ví dụ:

```
for (int i=0;i<100;i++)  
{  
    if (i%10 == 0)  
    {  
        System.out.println(i);  
    }  
}
```



WHILE

```
while (condition) {  
    statements;  
}
```

Ví dụ

```
int i = 0;  
while (i < 10)  
{  
    System.out.println (i);  
    i++;  
}
```



DO ... WHILE

```
do {  
    statements;  
} while (condition);
```

Ví dụ

```
int i = 11;  
do  
{  
    System.out.println (i);  
    i++;  
} while (i < 10);
```



FOR-EACH

```
for (data-type item: array/collection) {  
    statements;  
}
```

Ví dụ:

```
int[] intArray = { 2, 4, 6, 8, 10 };  
for (int i : intArray)  
{  
    System.out.println(i);  
}
```



LỆNH BREAK, CONTINUE, RETURN

- **break** dùng để kết thúc vòng lặp for, while, do while: `break [label];`
- **continue** dùng để bỏ qua vòng lặp hiện tại: `continue [label];`
- **return** thoát khỏi phương thức và trả về giá trị (nếu có): `return [<expression>];`



VÍ DỤ 1 (BREAK)

```
class BreakExample1 {  
    public static void main(String args[]) {  
        for (int i = 0; i < 10; i++) {  
            System.out.println(i + " ");  
            if (i % 2 == 0)  
                break;  
        }  
    }  
}
```



VÍ DỤ 2 (BREAK)

```
class BreakExample2 {  
    public static void main(String args[]) {  
        loop1:  
        for(i = 0; i < 100; i++) {  
            for(j = 0; j < 1000; j++) {  
                if (j == i * 12 + 7)  
                    break loop1;  
            }  
        }  
    }  
}
```



VÍ DỤ (CONTINUE)

```
class ContinueExample {  
    public static void main(String args[]) {  
        for (int i = 0; i < 10; i++) {  
            System.out.println(i + " ");  
            if (i % 2 == 0)  
                continue;  
        }  
    }  
}
```



VÍ DỤ (RETURN)

```
class MyClass {  
    static int myMethod(int x) {  
        return 5 + x;  
    }  
  
    public static void main(String[] args) {  
        System.out.println(myMethod(3));  
    }  
}
```



CHUỖI (STRING)

- Ví dụ:

```
String str = "Hello";
```

```
String str = new String("Hello");
```

```
int myInteger = 5;
```

```
String integerString = myInteger.toString( );
```

```
String integerString = String.valueOf(myInteger);
```



MỘT SỐ PHƯƠNG THỨC CHUỖI

- length
- charAt
- indexOf
- contains
- trim
- substring
- toUpperCase
- toLowerCase
- replace
- split



VÍ DỤ (STRING)

```
class Sample_String{  
    public static void main(String[] args){  
        String str1 = "Good";  
        System.out.println(str1.length());  
        System.out.println(str1.substring(0,2));  
        String str2 = "Morning";  
        String str3 = str1+str2;  
        System.out.println(str3.indexOf("Morning"));  
        System.out.println(str3.charAt(8));  
    }  
}
```



LUYỆN TẬP

- 1) Viết chương trình nhập 2 số tự nhiên N và M, xuất ra số lớn hơn trong 2 số đó. Nếu 2 số bằng nhau thì xuất số nào cũng được.
- 2) Viết chương trình nhập số tự nhiên N, sau đó nhập N phần tử của mảng A (lưu vào mảng), tìm phần tử lớn nhất và nhỏ nhất trong mảng.
- 3) Viết chương trình nhập số tự nhiên N, sau đó nhập N chuỗi lưu vào mảng A. Nhập chuỗi s. Tìm các chuỗi trong mảng A có chứa chuỗi s và xuất ra vị trí bắt đầu của chuỗi s trong các chuỗi đó.

