

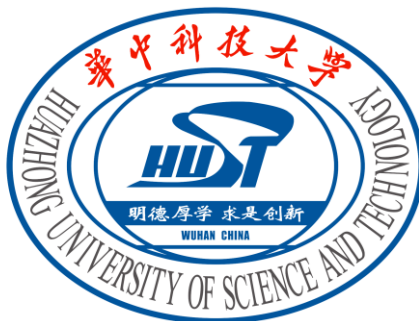
# 基于Java的面向对象程序设计

陈维亚

*weiya\_chen@hust.edu.cn*

华中科技大学软件学院

## 第2讲：“没有对象的” Java



1. Java简介
2. Java数据类型
3. Java运算符
4. Java控制结构
5. 总结

# 1. Java简介

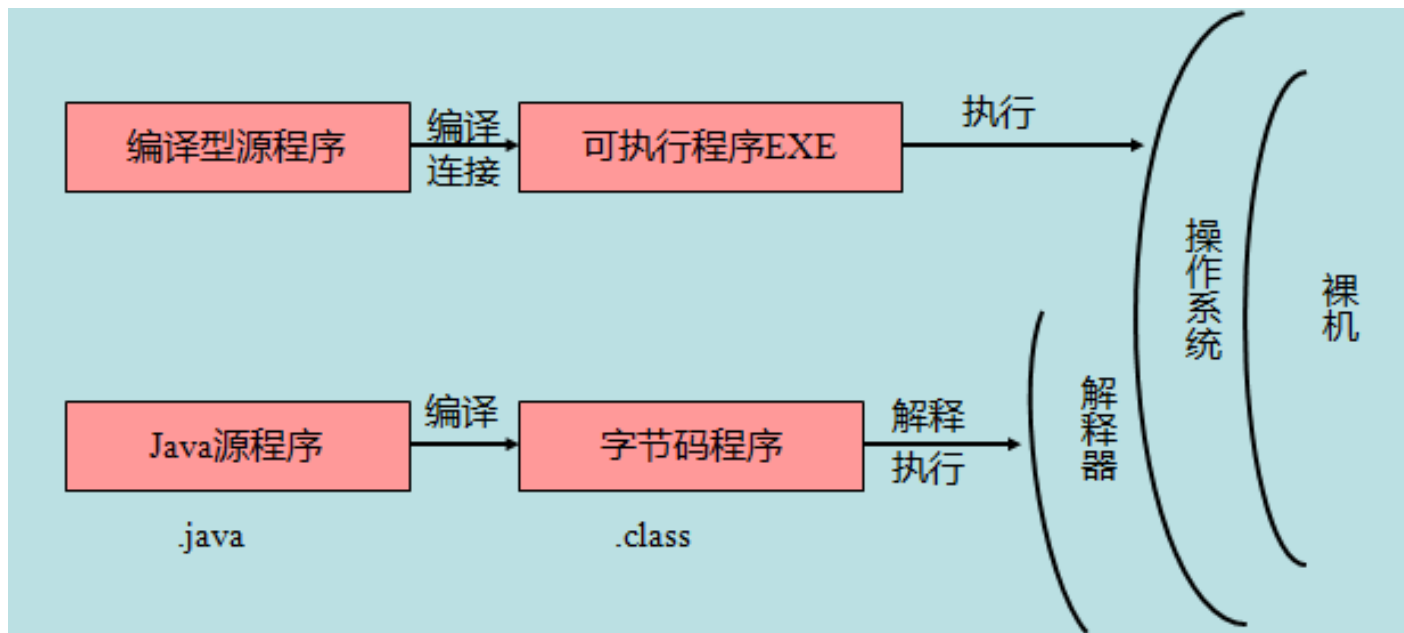
## Java起源

是由Sun Microsystems公司的James Gosling和同事们共同研发，于1995年5月推出的Java面向对象程序设计语言和Java平台的总称。



## Java核心特点

- 为了分布式系统而生：安全、重用、可移植
- 类似C语言的语法结构
- **纯**面向对象
- **非常丰富**的库函数
- 编译后由JVM解释执行（**一处编译，处处执行**）



## Java分为三个体系

- JavaSE (J2SE) : Java2 Platform Standard Edition
- JavaEE (J2EE) : Java 2 Platform Enterprise Edition
- JavaME (J2ME) : Java 2 Platform Micro Edition



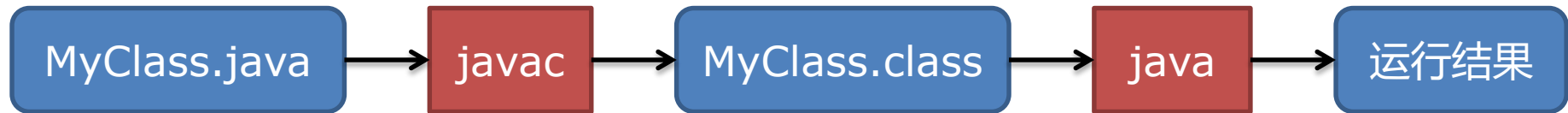
2005年6月，JavaOne大会召开，SUN公司公开Java SE 6。此时，Java的各种版本已经更名以取消其中的数字"2"：J2EE更名为Java EE，J2SE更名为Java SE，J2ME更名为Java ME。

## 配置Java开发环境

- 安装JDK ( 包含JRE )
- 选择合适的编辑器 , 用命令行编译执行
- 或者使用IDE开发



IntelliJ IDEA ?



- 编译源文件

`javac <source_file_name>`

e.g. `javac HelloWorld.java`

- 运行可执行文件

`java <class_name>`

e.g. `java HelloWorld`

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

## 2. Java数据类型



### 内置数据类型

#### 整型

byte 8 bits

short 16 bits

int 32 bits

Long 64 bits

#### 浮点型

float 32 bits

double 64 bits

#### 布尔型

boolean 1 bit {true, false}

#### 字符型

char 16 bits Unicode

```
int decimal = 100;  
int octal = 0144;  
int hexa = 0x64;
```



### 引用类型

- 在Java中，引用类型的变量非常类似于C/C++的指针。
- 引用类型指向一个对象，指向对象的变量是引用变量。
- 对象、数组都是引用数据类型。
- 所有引用类型的默认值都是null。

```
City wf = new City("Winterfell");  
Int[] fibonacci = {1, 1, 2, 3, 5, 8};
```

引用类型变量在声明后必须通过**实例化**开辟数据空间，才能对变量所指向的对象进行访问。

### 常量

在 Java 中使用 final 关键字来修饰常量，  
声明方式和变量类似：

```
final double PI = 3.1415926;  
final double SPEED_OF_LIGHT = 3e8;  
final double GRAVITY = 9.8;
```

### 类型转换

整型、浮点型、字符型数据可以混合运算。

运算中，不同类型的数据先转化为同一类型，然后进行运算。

#### A. 自动类型转换

低 -----> 高

byte —> short, char—> int —> long—> float —> double

```
char c1='a';  
int i1 = c1;           i1=?  
char c2 = 'A';  
int i2 = c2+1;         i2=?  
float f1 = i2 + 1.0;   f1=?
```

查ascii码表可知，'a'对应的int类型值为97，'A'对应值为65

### 类型转换

#### B. 强制类型转换

低 -----> 高

byte —> short, char—> int —> long—> float —> double

```
int i1 = 123;  
byte b = (byte)i1;
```

- 整数的默认类型是 int
- 定义 float 类型时必须在数字后面跟上 F 或者 f
- 使用 long 类型时必须在数字后面跟上 L

### 算术运算符

**+**   **-**   **\***   **/**   **%**   **++**   **--**

```
int a = 10;
int b = 20;
int c = 25;
int d = 25;
System.out.println("b / a = " + (b / a) );
System.out.println("b % a = " + (b % a) );
System.out.println("c % a = " + (c % a) );
System.out.println("a++ = " + (a++) );
System.out.println("a-- = " + (a--) );
System.out.println("d++ = " + (d++) );
System.out.println("++d = " + (++d) );
```

### 关系运算符

== != > < >= <=

```
int a = 10;  
int b = 20;  
System.out.println("a == b = " + (a == b));
```

### 位运算符

&   |   ^   ~   <<   >>   >>>

Java定义了位运算符，应用于整数类型(int)，长整型(long)，短整型(short)，字符型(char)，和字节型(byte)等类型。

```
int a = 60; /* 60 = 0011 1100 */
int b = 13; /* 13 = 0000 1101 */
int c = 0;
c = a & b;           12
c = a | b;           61
c = a ^ b;           49
c = ~a;              -61
c = a << 2;          240
c = a >> 2;           15
c = a >>> 2;          15
```

### 逻辑运算符

&&

||

!

```
int a = 10;  
int a = 5;  
boolean b = (a<4)&&(a++<10);  
System.out.println("使用短路逻辑运算符的结果为"+b);  
System.out.println("a的结果为"+a);
```

### 短路逻辑运算符

当使用“与”逻辑运算符时，当得到第一个操作为false时，其结果就必定是false，这时候就不会再判断第二个操作了。



### 赋值运算符

=      +=      -=      ...

### 条件运算符 (?:)

variable x = (expression) ? value if true : value if false

```
int a , b;  
a = 10;  
  
b = (a == 1) ? 20 : 30;  
System.out.println( "Value of b is : " + b );  
  
b = (a == 10) ? 20 : 30;  
System.out.println( "Value of b is : " + b );
```

### 3. Java运算符



#### 运算符优先级符

优先级递增

| 类别   | 操作符            | 关联性  |
|------|----------------|------|
| 后缀   | () [] . (点操作符) | 左到右  |
| 一元   | ++ -- ! ~      | 从右到左 |
| 乘性   | * / %          | 左到右  |
| 加性   | + -            | 左到右  |
| 移位   | >> >>> <<      | 左到右  |
| 关系   | >> = << =      | 左到右  |
| 相等   | == !=          | 左到右  |
| 按位与  | &              | 左到右  |
| 按位异或 | ^              | 左到右  |
| 按位或  |                | 左到右  |
| 逻辑与  | &&             | 左到右  |
| 逻辑或  |                | 左到右  |
| 条件   | ? :            | 从右到左 |
| 赋值   | = , +=等        | 从右到左 |
| 逗号   | ,              | 左到右  |

### 分支结构

#### if 语句

```
int x = 30;
if( x < 20 ){
    System.out.print("这是 if 语句");
} else {
    System.out.print("这是 else 语句");
}
```

#### 条件运算符 ( ? : )

```
int a , b;
a = 10;

b = (a == 1) ? 20 : 30;
```

### 分支结构

#### switch 语句

```
char weather = 'C';
switch(weather)
{
    case 'S' :
        System.out.println("阳光普照");
        break;
    case 'X' :
    case 'C' :
        System.out.println("阴云密布");
        break;
    case 'W' :
        System.out.println("大风起兮");
        break;
    default :
        System.out.println("别问我我什么也不知道");
}
System.out.println("今天的天气是 " + weather);
```

### 循环结构

#### **while** 循环

```
int x = 10;
while( x < 20 ){
    System.out.print("value of x : " + x);
    x++;
    System.out.print("\n");
}
```

#### **do...while** 循环

```
int x = 10;
do {
    System.out.print("value of x : " + x);
    x++;
    System.out.print("\n");
} while(x < 20);
```

### 循环结构

#### **for** 循环

```
for(int x=10; x<20; x=x+1){  
    System.out.print("value of x : " + x + "\n");  
}
```

#### **Java 增强 for** 循环

```
int[] numbers = {1, 2, 3, 4, 5};  
for(int x : numbers){  
    System.out.print("value of x : " + x + "\n");  
}
```

### 循环结构

#### **break** 关键字

```
int[] numbers = {1, 2, 3, 4, 5};
for(int x : numbers){
    if( x == 3 ) {
        break;
    }
    System.out.print("value of x : " + x + "\n");
}
```

#### **continue** 关键字

```
int[] numbers = {1, 2, 3, 4, 5};
for(int x : numbers){
    if( x == 3 ) {
        continue;
    }
    System.out.print("value of x : " + x + "\n");
}
```

【练习1】请用Java实现如下小程序：

- 1) 火车在计算托运行李费用时以kg为单位计算费用（12元/kg），忽略重量中的小数部分，即忽略不足1kg的部分。
- 2) 汽车在计算托运行李费用时以kg为单位计算费用（22元/kg），将重量中的小数部分进行四舍五入
- 3) 飞机在计算托运行李费用时以g为单位计算费用（0.062元/g），将重量中的小数部分，即不足1g的部分进行四舍五入。

请设计一个程序，让用户从键盘输入行李重量（以kg为单位），然后程序分别计算出用火车、汽车和飞机托运行李的费用。

火车，12元/kg，小数忽略  
汽车，22元/kg，小数四舍五入  
飞机，0.062元/g，小数四舍五入



## 【练习1】

火车，12元/kg，小数忽略  
汽车，22元/kg，小数四舍五入  
飞机，0.062元/g，小数四舍五入

```
public class Baggage{  
    public static void main(String[] args){  
        double weight = 65.7;  
        int p_train = 12, p_bus = 22;  
        double p_plane = 0.062;  
        double c_train, c_bus, c_plane;  
  
        // TODO  
  
    }
```

【练习2】编写一个java应用程序，在主类的main方法中实现下列功能：

- 程序随机分配给客户一个1~100之间的整数。
- 用户输入自己的猜测。
- 程序返回提示信息，提示信息分别是：“猜大了”、“猜小了”或“猜对了”。
- 用户可根据提示信息再次输入猜测，直到提示信息是“猜对了”。

```
public class GuessNumber{
    public static void main(String[] args){
        Scanner reader = new Scanner(System.in);
        Random random = new Random();
        System.out.println("给你一个1至100之间的整数, 请猜测这个数");
        int realNumber = random.nextInt(100)+1;

        int yourGuess = 0;
        System.out.print("输入您的猜测:");
        yourGuess = reader.nextInt();

        // TODO
    }
```

【课后练习】编写一个java应用程序，在主类的main方法中实现下列功能：

- 用户输入一个1~99999之间的整数；
- 程序返回提示信息，告诉用户此数是否为回文数。

\* 回文数左右对称，如121、12321等，正着反着都一样。

### Java简介

Java数据类型

Java运算符

Java控制结构

Java 其实有很多对象!



类与对象的概念

Java中类的实现