



情境1：命名规则



1.标识符和关键字

- (1) 标识符
- (2) 关键字
- (3) 注释

2.常量和变量

- (1) 常量
- (2) 变量

3.基础数据类型

- (1) 布尔类型
- (2) 字符类型
- (3) 整数类型
- (4) 浮点类型
- (5) 各类型数据间的相互转换

■ 标识符

- 在程序设计语言中的每个元素（如变量、常量、方法、类和包等）需要有一个名字以标识它的存在和唯一性，这个名子就是标识符（**identifier**）。
- 标识符的命名规则：
 - 必须以**字母、下划线（_）或数字**开头，后面可以是字母、下划线、美元符号、数字（**0~9**）。
 - 标识符**不能包含运算符**，如+，-等。 **要记忆！！**
 - 标识符**不能是关键字**。如main, public, class, import等。
 - 标识符**不能是true、false和null**。
 - 标识符可有**任意长度**。
 - 标识符**区分大小写**，如Mybook与mybook是完全不同的两个标识符。



- **Java**语言中的关键字均用小写字母表示。
- 分类：
 - 访问控制：
private, protected, public;
 - 类、方法和变量修饰符：
abstract, class, extends, final, implements, interface, native, new, static, strictfp, synchronized, transient, volatile;
 - 程序控制语句：
break, continue, return, do, while, if, else, for, instanceof, switch, case, default;

 关键字

- 错误处理: **catch, finally, throw, throws, try;**
 - 包相关: **import, package;**
 - 基本类型: **boolean, byte, char, double, float, int, long, short;**
 - 变量引用: **super, this, void;**
 - 语法保留字: **null, true, false;**
- 关键字 **goto** 和 **const** 是 C++ 保留的关键字, 在 Java 中不能使用。

- 为增加程序的可读性，便于日后维护及修改，需要为程序添加注释(**comment**)。
- **Java**的注释符有三种：
 - 行注释符。“//”。
 - 块注释符。“/*”和“*/”。
 - 文档注释符。“/**”和“*/”。



- 常量是在程序运行过程中其值始终保持不变的量。
- 常量定义
`final 数据类型 常量名=值;`
例如: `final int a=12;`
- Java常用的整型常量、浮点常量说明如下:
 - 整型常量: 十进制整数, 八进制整数, 十六进制整数。
 - 浮点型常量: 十进制数形式, 科学计数法形式。
 - 字符型常量: 用于表示单个字符, 要求用单引号把字符括起来。
 - 字符串常量: 用双引号 (“ ”) 括起来的由0个或多个字符序列。
 - 布尔型常量: 只有真 (true)、假 (false) 两种值。



常量

■ 转义字符

- 转义字符代表一些特殊字符。
- 转义字符主要通过在一个字符前加一个反斜杠“\”来实现。
- 常用的转义字符

转义字符	含 义	转义字符	含 义
'\b'	退格	'\a'	响铃
'\t'	水平制表符 tab	'\"'	双引号
'\v'	垂直制表符	'\''	单引号
'\n'	换行	'\\'	反斜线
'\f'	换页	'\ddd'	用3位8进制数表示字符
'\r'	回车	'\uxxxx'	用4位16进制数表示字符



变量

- 变量是在程序运行过程中其值可以被改变的量，通常用来记录运算中间结果或保存数据。
- 变量包括变量名、变量值两部分。
- 变量必须先声明后使用。
 - 声明变量的格式如下：

数据类型 变量名1[, 变量名2,];

或：数据类型 变量名1 [=初值][, 变量名2 [=初值], ...];

例如：`int k ; //声明一个存放整型且名是k的变量`

`float x, y; //声明浮点型变量x, y`

`String studentname = "WangXin";`

`//声明字符变量studentname, 其初值为WangXin`

`double a=1.0, b=2.0; //声明变量a、b并分别赋值1.0、2.0`



例1: 源程序名 “ComputeArea.java”, 计算半径为10的圆的面积, 并显示结果。

```
public class ComputeArea
```

```
{
```

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

```
{
```

```
final double PI=3.14159;
```

```
double area;
```

```
area=PI*10*10;
```

```
System.out.println(“The area for the circle of 10 is “+area);
```

```
}
```

```
}
```

```
命令提示符
D:\Java>java ComputeArea
The area for the circle of 10 is 314.159
```

简单数据类型

■ 数据类型概述

- 每个数据类型都有一个**值域**，或者叫做**范围**。编译器根据变量或常量的**数据类型**对其**分配存储空间**。Java为数值、字符值和布尔值数据提供了几种基本数据类型。

Java语言的数据类型划分如图2-1

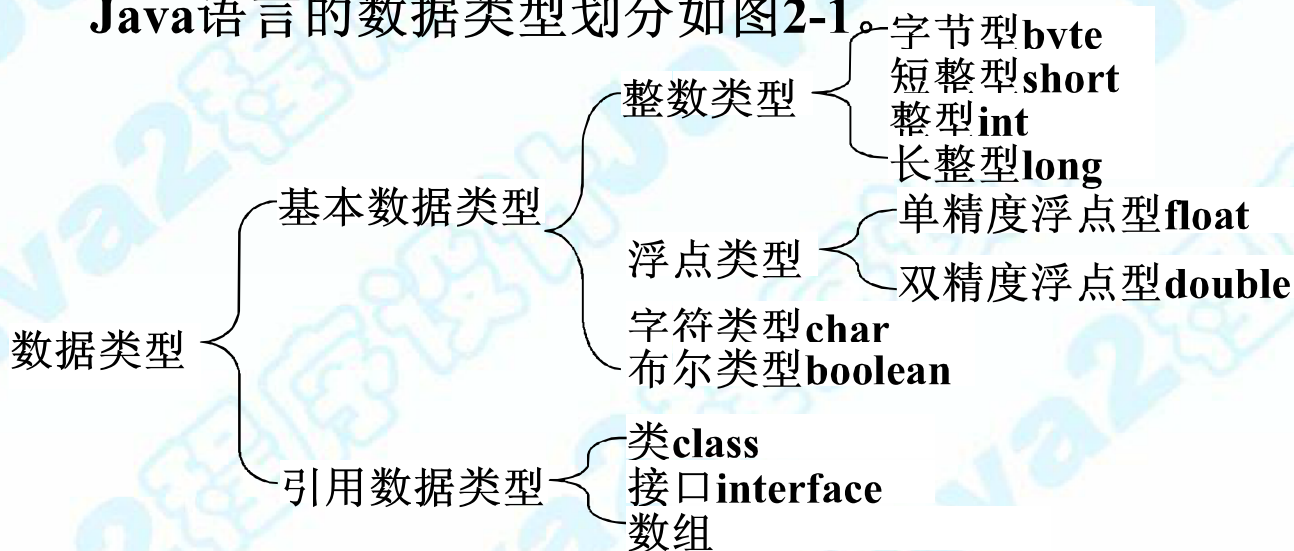


图2-1 Java语言的数据类型



简单数据类型

■ 1. 数值数据类型

Java有六种数值类型：四种整型和两种浮点型

数据类型	所占位数	数的取值范围
byte	8	$-2^7 \sim 2^7 - 1$
short	16	$-2^{15} \sim 2^{15} - 1$
int	32	$-2^{31} \sim 2^{31} - 1$
long	64	$-2^{63} \sim 2^{63} - 1$
float	32	$-3.4 \times 10^{38} \sim 3.4 \times 10^{38}$ (精度为6到7位有效数字)
double	64	$-1.7 \times 10^{308} \sim 1.7 \times 10^{308}$ (精度为14到15位有效数字)



简单数据类型

【例2】源程序名“**TypeMaxValue.java**”，数值数据类型的最大值示例。

```
1 public class TypeMaxValue
2 {
3     public static void main(String args[])
4     { // 整数类型
5         byte largestByte = Byte.MAX_VALUE;// 定义一个byte类型的
6         short largestShort = Short.MAX_VALUE;//定义一个short类型的变量
7         int largestInteger = Integer.MAX_VALUE;//定义一个int类型的变
8         long largestLong = Long.MAX_VALUE;//定义long类型的变量
9         float largestFloat = Float.MAX_VALUE;//定义一个float类型的变量
10        double largestDouble = Double.MAX_VALUE;//定义double类型变量
11                // 在屏幕上显示对应类型的最大值
12        System.out.println("最大的byte值是: " + largestByte);
13        System.out.println("最大的short值是: " + largestShort);
14        System.out.println("最大的integer值是: " + largestInteger);
15        System.out.println("最大的long值是: " + largestLong);
```




2.1 简单数据类型

```
16 System.out.println("最大的float值是: " + largestFloat);
17 System.out.println("最大的double值是: " + largestDouble);
18 }
19 }
```

【运行结果】

```
C:\WINDOWS\system32\cmd.exe
D:\Java>java TypeMaxUalue
最大的byte值是: 127
最大的short值是: 32767
最大的integer值是: 2147483647
最大的long值是: 9223372036854775807
最大的float值是: 3.4028235E38
最大的double值是: 1.7976931348623157E308
D:\Java>
```

【程序分析】

第4-11句：Java中每种数据类型都封装为一个类，通过类型类的MAX_VALUE方法找到各种数值数据类型的取值最大值。

第12-18句：输出各种数值数据类型的取值最大值。



简单数据类型

■ 2. 字符数据类型

- 字符数据类型**char**用于表示单个字符。一个字符的值由单引号括起来，如：

char letter = 'A'; //将字符A赋给了变量**letter**

char numChar = '4'; //将数字4赋给了变量**numChar**

- 但是，如果‘4’中的单引号去掉是非法的，因为“**char numChar=4;**”中，4是一个数值，不能赋给**char**型变量。



简单数据类型

- 另外，**char**型只表示一个字母，表示一串字符要用叫做**String**的数据结构。如：

`String message="Welcome to Java!";` //字符串必须用双引号括住。

■ 3. 布尔数据类型

- 布尔数据类型的值域包括两个值：真（**true**）和假（**false**）。例如，下面一行代码给变量**doorOpen**赋值为**true**，如：
 - **Boolean doorOpen=true;**



简单数据类型

4. 数值类型转换

- 当两个类型不同的运算对象进行二元运算时，Java自动把精度较低的类型转换成另一个精度较高的类型。运算中，不同类型的数据先转化为同一类型，然后进行运算。

低----->高

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

- 例：操作数1（为byte型）+操作数2（为int型）

转换类型如下：

操作数1（为int型）+操作数2（为int型）

- 常用数值类型转换的具体情况见表2-3所示。



简单数据类型

表 Java语言数据类型转换

操作数1类型	操作数2类型	转换后的类型
byte、short、char	int	int
byte、short、char、int	long	long
byte、short、char、int、long	float	float
byte、short、char、int、long、float	double	double

如从高级转换成低级，则需要强制类型转换，但会导致溢出或精度下降。

如：**int i = 8; byte b=(byte)i;**



简单数据类型

- 【例3】源程序名“TypeTest.java”，数据类型转换示例。

```
1 public class TypeTest
2 {
3     public static void main (String args[ ])
4     {
5         int c;
6         long d=6000;
7         float f;
8         double g=123456789.987654321;
9         c=(int)d;
10        f=(float)g; //导致精度的损失.
11        System.out.println("c= "+c);
12        System.out.println("d= "+d);
13        System.out.println("f= "+f);
14        System.out.println("g= "+g);
15    }
16 }
```




简单数据类型

【运行结果】

```
C:\WINDOWS\system32\cmd.exe
D:\Java>javac TypeTest.java
D:\Java>java TypeTest
c= 6000
d= 6000
f= 1.23456792E8
g= 1.2345678998765433E8
D:\Java>
```

【程序分析】

第9句：将long类型数据强制转换为int类型，有些情况可能导致数据溢出。

第10句：将double类型数据强制转换为float类型，将导致精度的损失，通过运行结果可以看出。

所以，进行数据类型的强制转换时一定需要慎重。



简单数据类型

■ 【例4】源程序名“**MathDemo.java**”，调用**Java API**函数完成数值运算示例。

```
1 public class MathDemo
2 {
3     public static void main(String[] args)
4     {
5         System.out.println("abs(-5)=" + Math.abs(-5));
6         System.out.println("max(6.75, 3.14)=" + Math.max(6.75, 3.14));
7         System.out.println("min(100, 200)=" + Math.min(100, 200));
8         System.out.println("round(3.5)=" + Math.round(3.5));
9         System.out.println("round(-6.5)=" + Math.round(-6.5));
10        System.out.println("sqrt(2)=" + Math.sqrt(2));
```



简单数据类型

```
11 System.out.println("pow(2, 5) = " + Math.pow(2, 5));
12 System.out.println("E = " + Math.E);
13 System.out.println("exp(2) = " + Math.exp(2));
14 System.out.println("log(2) = " + Math.log(2));
15 System.out.println("ceil(6.75)=" + (int)Math.ceil(6.75));
16 System.out.println("floor(6.75)=" + (int)Math.floor(6.75));
17 System.out.println("Pi = " + Math.PI);
18 System.out.println("sin(Pi/4)= " + Math.sin(Math.PI / 4));
19 System.out.println("cos(1) = " + Math.cos(1));
20     }
21 }
```



简单数据类型

【运行结果】

```
C:\WINDOWS\system32\cmd.exe
D:\Java>java MathDemo
abs(-5) = 5
max(6.75, 3.14) = 6.75
min(100, 200) = 100
round(3.5) = 4
round(-6.5) = -6
sqrt(2) = 1.4142135623730951
pow(2, 5) = 32.0
E = 2.718281828459045
exp(2) = 7.38905609893065
log(2) = 0.6931471805599453
ceil(6.75) = 7
floor(6.75) = 6
Pi = 3.141592653589793
sin(Pi / 4) = 0.7071067811865475
cos(1) = 0.5403023058681398
D:\Java>
```

【程序分析】

第5-7句：演示求绝对值和求最大、最小值函数的用法。

第8-9句：演示四舍五入函数的用法。

第10-11句：演示求平方根和求幂函数的用法。

第12-14句：演示指数与对数函数的用法。

第15-16句：演示向上取整与向下取整函数的用法。

第17-19句：演示三角函数的用法。



Thank you!

