

Java 应用与开发

Java 语言基础与流程控制

王晓东

wangxiaodong@ouc.edu.cn

中国海洋大学

August 28, 2018



参考书目

1. 陈国君等编著, Java 程序设计基础 (第 5 版), 清华大学出版社
2. Bruce Eckel, Thinking in Java (3rd)



学习目标

► Java 语言基础

1. 数据类型
2. 常量和变量
3. 关键字与标识符
4. 运算符与表达式
5. 从键盘输入数据

► 流程控制

1. 语句和复合语句
2. 分支结构（选择结构）
3. 循环结构
4. 跳转语句



大纲

数据类型

常量和变量

关键字与标识符

运算符与表达式

从键盘获得输入

语句

分支结构

循环结构



接下来...

数据类型

常量和变量

关键字与标识符

运算符与表达式

从键盘获得输入

语句

分支结构

循环结构



数据类型

❖ 数据类型的基本要素

- ▶ 数据的性质（数据结构）
- ▶ 数据的取值范围（字节大小）
- ▶ 数据的存储方式
- ▶ 参与的运算



数据类型

❖ 基本数据类型

由程序设计语言系统所定义、不可再划分的数据类型。所占内存大小固定，与软硬件环境无关。在内存中存放的是数据值本身。

整型 byte short int long

浮点型 float double

逻辑型 boolean

字符型 char

❖ 引用数据类型（复合数据类型）

在内存中存放的是指向该数据的地址，不是数据值本身。包括：类、数组、接口等。



数据类型

❖ 整型

类型	数据位数	取值范围
byte (字节型)	8	$-128 \sim 127$, 即 $-2^7 \sim 2^7 - 1$
short (短整型)	16	$-32768 \sim 32767$, 即 $-2^{15} \sim 2^{15} - 1$
int (整型) (默认)	32	$-2147483648 \sim 2147483647$, 即 $-2^{31} \sim 2^{31} - 1$
long (长整型) (l 或 L)	64	$-2^{63} \sim 2^{63} - 1$



数据类型

❖ 浮点型

类型	数据位数	取值范围
float (单精度) (f 或 F)	32	$1.4E - 45 \sim 3.4E + 38$
double (双精度) (默认)	64	$4.9E - 324 \sim 1.8E + 308$



数据类型

❖ 逻辑型 boolean 布尔型

- ▶ 只有 true（“真”）和 false（“假”）两个取值。
- ▶ 占 1 个字节，默认取值为 false。
- ▶ true 和 false 不能转换成数字表示形式。



数据类型

❖ 字符型

- ▶ 字符型数据类型用来存储单个字符，采用的是 Unicode 字符集编码方案¹。
- ▶ 字符声明用单引号表示单个字符。
- ▶ 字符型数据可以转化为整型。

CODE ▶ 字符数据类型示例

```
1 public class CharDemo {  
2     public static void main(String[] args) {  
3         char a = 'J';  
4         char b='Java'; //会报错  
5     }
```

¹请自行学习掌握什么是字符集和字符编码规则。



数据类型转换

❖ 数值型不同类型数据的转换

自动类型转换

1. 转换前的数据类型与转换后的类型兼容。
2. 转换后的数据类型的表示范围比转换前的类型大。
3. 条件 2 说明不同类型的数据进行运算时，需先转换为同一类型，然后进行运算。转换从“短”到“长”的优先关系为：

byte→short→char→int→long→float→double

强制类型转换

如果要将较长的数据转换成较短的数据时（不安全）就要进行强制类型转换。格式如下：

- ▶ (预转换的数据类型) 变量名



数据类型转换

❖ 字符串型数据与数值型数据相互转换

```
1 String myNumber = "1234.56";  
2 float myFloat = Float.parseFloat(MyNumber);
```

❖ 数值型数据转换成字符串

字符串可用加号“+”来实现连接操作。若其中某个操作数不是字符串，该操作在连接之前会自动将其转换成字符串。所以可用加号来实现自动的转换。

```
1 int myInt = 1234;           //定义整形变量MyInt  
2 String myString = "" + MyInt; //将整型数据转换成了字符串
```



接下来...

数据类型

常量和变量

关键字与标识符

运算符与表达式

从键盘获得输入

语句

分支结构

循环结构



常量

❖ 变量的属性

- ▶ 变量名
- ▶ 类型
- ▶ 值
- ▶ 地址



常量

整型常量 八进制、十六进制、十进制长整型后需要加 l 或 L。

浮点型常量 单精度后加 f 或 F，双精度后加 d 或 D 可省略。

逻辑型常量

字符型常量 单引号。

字符串常量 双引号。

❖ 常量的声明

```
1 final int MAX = 10;  
2 final float PI = 3.14f;
```



变量

❖ 变量声明、初始化和赋值

```
1    int i, j = 0;  
2    i = 8;  
  
4    float k;  
5    k = 3.6f;
```

Java 语言程序中可以随时定义变量，不必集中在执行语句之前。



接下来...

数据类型

常量和变量

关键字与标识符

运算符与表达式

从键盘获得输入

语句

分支结构

循环结构



关键字与标识符

❖ 关键字 (Java 保留字)

abstract	assert	boolean	break	byte	case
catch	char	class	continue	default	do
double	else	enum	extends	false	final
finally	float	for	if	implements	import
instanceof	int	interface	long	native	new
null	package	private	protected	public	return
short	static	super	switch	synchronized	this
volatile	throws	transient	true	try	void



关键字与标识符

❖ 标识符

用来表示变量名、类名、方法名、数组名和文件名的有效字符序列。规定如下：

- ▶ 可以由字母、数字、下划线 (__)、美元符号 (\$) 组合而成。
- ▶ 必须以字母、下划线或美元符号开头，不能以数字开头。
- ▶ 关键字不能当标识符使用。
- ▶ 区分大小写。

编码习惯：驼峰命名，类名首字母大写，变量、方法及对象首字母小写。



接下来...

数据类型

常量和变量

关键字与标识符

运算符与表达式

从键盘获得输入

语句

分支结构

循环结构



运算符

按照运算符功能来分，基本的运算符有下面几类：

算术运算符	+, -, *, /, %, ++, --
关系运算符	>, <, >=, <=, ==, !=
逻辑运算符	!, &&, , &, , ^
位运算符	>>, <<, >>>, &, , ^, ~
赋值运算符	=, 扩展赋值运算符, 如 +=, /= 等
条件运算符	? :
其他运算符	包括分量运算符 .、下标运算符 []、实例运算符 instanceof、内存分配运算符 new、强制类型转换运算符(类型)、方法调用运算符()等



课后小作业

请自行总结 Java 运算符的基本用法，包括运算符的优先级。



接下来...

数据类型

常量和变量

关键字与标识符

运算符与表达式

从键盘获得输入

语句

分支结构

循环结构



从键盘获得输入

由键盘输入的数据，不管是文字还是数字，Java 皆视为**字符串**，若是要由键盘输入获得数字则必须再经过类型转换。

CODE 获得键盘输入字符串并转换为数字

```
1  import java.io.*;
2  public class MyClass {
3      public static void main(String[] args) throws IOException {
4          int num1, num2;
5          String str1, str2;
6          InputStreamReader in;
7          in = new InputStreamReader(System.in);
8          BufferedReader buf;
9          buf = new BufferedReader(in);
10         System.out.print("请输入第一个数: ");
11         str1 = buf.readLine();    //将输入的内容赋值给字符串变量 str1
12         num1 = Integer.parseInt(str1); //将 str1 转成 int 类型后赋给 num1
13         System.out.print("请输入第二个数: ");
14         str2 = buf.readLine();    //将输入的内容赋值给字符串变量 str2
15         num2 = Integer.parseInt(str2); //将 str2 转成 int 类型后赋给 num2
16         System.out.println(num1 + " * " + num2 + " = " + (num1 * num2));
17     }
18 }
```



从键盘获得输入

为了简化输入操作，从 JavaSE 5 版本开始在 java.util 类库中新增了一个类专门用于输入操作的类 **Scanner**，可以使用该类输入一个对象。

CODE ▶ 使用 Scanner 获得键盘输入并转换为特定数据类型

```
1  import java.util.*;
2  public class MyClass {
3      public static void main(String[] args)
4      {
5          Scanner reader = new Scanner(System.in);
6          double num;
7          num = reader.nextDouble(); //按照 double 类型读取键盘输入
8          ...
9      }
10 }
```

其他可用的读取方法

nextByte() nextDouble() nextFloat() nextInt() nextLong()
nextShort() next() nextLine()



接下来...

数据类型

常量和变量

关键字与标识符

运算符与表达式

从键盘获得输入

语句

分支结构

循环结构



语句与复合语句

- ▶ Java 语言中语句可以是以分号 “;” 结尾的简单语句，也可以是用一对花括号 “{}” 括起来的复合语句。
- ▶ Java 中的注释形式：
 - ▶ 单行注释：//
 - ▶ 多行注释：/* */
 - ▶ 文件注释：/** */



接下来...

数据类型

常量和变量

关键字与标识符

运算符与表达式

从键盘获得输入

语句

分支结构

循环结构



分支结构

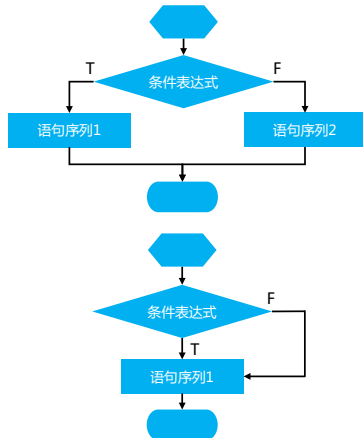
❖ if 分支结构 1

if 双路条件结构

```
if (条件表达式) {  
    语句序列 1  
} else {  
    语句序列 2  
}
```

if 单路条件结构

```
if (条件表达式) {  
    语句序列  
}
```



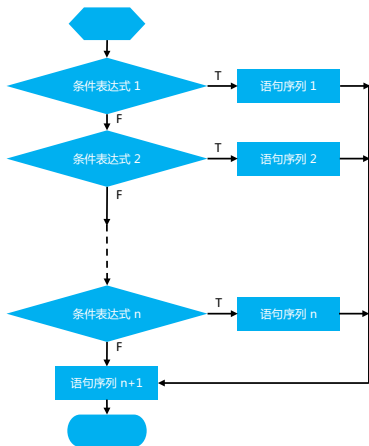
分支结构

❖ if 分支结构 2

if 多重条件选择结构

```
if (条件表达式1){  
    语句序列 1  
}  
else if (条件表达式2){  
    语句序列 2  
}  
...  
else if (条件表达式n){  
    语句序列 n  
}  
else {  
    语句序列 n+1  
}
```

条件表达式的结果必须
都是布尔型



分支结构

❖ switch 分支结构²

switch 多路选择结构

```
switch (表达式) {  
    case 常量表达式1:  
        语句序列1;  
        break;  
    case 常量表达式2:  
        语句序列2;  
        break;  
    ...  
    case 常量表达式n:  
        语句序列n;  
        break;  
    default:  
        语句序列n+1;  
}
```

■ switch表达式后面的数据类型只支持byte、short、char、int四种整形类型、枚举类型和String类型，且与各个case后面的常量表达式值的类型一致

■ 每个case一般都有break，若没有则多个case共享一个分支

■ 表达式的值与任何一个case后的常量表达式的值不相同同时执行

📖 课后小作业

自行搜索总结 switch 中的 case 语句如果没有 break 会出现什么情况，程序逻辑是如何执行的。

²在 Java 1.7 版本之后，switch 里表达式的类型可以为 String。



接下来...

数据类型

常量和变量

关键字与标识符

运算符与表达式

从键盘获得输入

语句

分支结构

循环结构



循环结构

❖ while 循环

```
1 while(conditional expression) {  
2     statements goes here ...  
3 }
```

❖ do-while 循环

```
1 do {  
2     statements goes here ...  
3 }  
4 while(conditional expression);
```



循环结构

❖ for 循环 1

```
1    int[] integers = {1, 2, 3, 4};  
3    for (int j = 0; j < integers.length; j++) {  
4        int i = integers[j];  
5        System.out.println(i);  
6    }
```

❖ for 循环 2

```
1    int[] integers = {1, 2, 3, 4};  
3    for (int i : integers) {  
4        System.out.println(i);  
5    }
```



循环中的跳转

break 语句 使程序的流程从一个语句块（switch 或循环结构）内跳出。

continue 语句 终止当前这一轮（次）的循环，进入下一轮（次）循环。

return 语句 用来使程序从方法（函数）中返回，可返回一个值。



本节习题

❖ 简答题

1. Java 语言定义类哪些基本数据类型？其存储结构分别是什么样的？
2. 自动类型转换的前提是什么？转换时的优先级顺序如何？
3. 数字字符串转换为数值类型数据时，可以使用的方法有哪些？

❖ 小编程

1. 编写程序，从键盘输入一个浮点数，然后将该浮点数的整数部分输出。
2. 编写程序，从键盘输入 2 个整数，然后计算它们相除后得到的结果并输出，注意排除 0 除问题。



THE END

wangxiaodong@ouc.edu.cn

