

03.数据类型与运算符

讲师：李刚

本章要点

- 注释
- 文档注释与javadoc
- 标识符与关键字
- 数据类型、8个基本数据类型
- 表达式类型的自动提升
- 运算符

注释

- 单行注释 用双斜线 “//” 表示
- 多行注释 用 /*-----*/ 表示
- 文档注释 用 /**-----*/ 表示

文档注释与javadoc

- 文档注释还可以通过javadoc 命令来生成API文档
- 格式: javadoc -d apidoc 源文件名
 - 其中-d path指定存放API文档的路径。
- API 文档包括三个区域分别是:包区域,类区域,详细说明区域
- 注: 如果要出现包区域,要让不同类处于不同的包下面才会出现包区域。

标识符和关键字

- 分隔符：常见的分隔符有分号；花括号{} 方括号[] 圆括号() 空格 圆点。
 - 在 Java 语言中每一条语句的结束用分号表示；
- 标识符的规则：由字母,数字,下划线_,美元\$组成,但数字不能开头.
- 标识符不能是java的关键字和保留字,但中间可以包含标识符，不能包含空格。
- Java 9规定：不允许单独使用下划线作为分隔符。
- Java 的关键字加上goto const true false nulll 一共有53个关键字和保留字，都不能作为Java的标识符。

标识符举例

- | | |
|--------------|---|
| – abc_xyz | √ |
| – HelloWorld | √ |
| – abc\$ | √ |
| – \$中文 | √ |
| – _abc 对 | √ |
| – xyz#abc | × |
| – abc1 | √ |
| – 1abc | × |

数据类型的分类

- Java语言是一个强类型语言。
 - 所有变量，必须先声明类型，再使用。
 - 指定类型的变量，只能装对应的类型的数据。
什么样的瓶子装什么样的酒。
- Java 语言支持的类型分为两大类：基本数据类型和引用类型
 - 基本数据类型包括八大数据类型。
 - 引用类型包括数组、类、接口。

变量

- 变量就是用来记录数据的。
- 计算机内存里数据就需要通过变量来访问它。
- 通俗一点，你可以把变量理解成容器，它可以装数据。

使用Scanner获取键盘输入

- 使用Scanner类可以很方便地获取用户的键盘输入，Scanner是一个基于正则表达式的文本扫描器，它可以从文件、输入流、字符串中解析出基本类型值和字符串值。Scanner类提供了多个构造器，不同的构造器可接受文件、输入流、字符串作为数据源，用于从文件、输入流、字符串中解析数据。
- Scanner主要提供了两个方法来扫描输入：
 - hasNextXxx(): 是否还有下一个输入项，其中Xxx可以是Int、Long等代表基本数据类型的字符串。如果需要判断是否包含下一个字符串，则可以省略Xxx。
 - nextXxx(): 获取下一个输入项。Xxx的含义与前一个方法中Xxx相同。

基本数据类型

- Java 的基本数据类型可以分为两大类: boolean 类型 和 数值类型.
- 而数值类型又可以分为整型和浮点型
 - 整型包括: byte short int long (广义的整型包括 char 型)
 - 在内存中各占: 1, 2, 4, 8 个字节, 各有各的取值范围, 略记一下。
- 在定义 long 类型时 必须在数字后面加 L,
- 整数常量有三种表示方式: 十进制, 八进制(以 O 开头), 十六进制(以 OX 开头), 二进制 (以 OB 开头—jdk1.7)
- 在 java 语言中, 运算过程中, 整型类型默认为 int 类型
- 什么样的类型的变量 只能存放什么样类型的值

浮点型

- 浮点型 包括 : float 和 double 型 两种。他们分别占有4个字节和8个字节,但他们远远比int 和 long 的 4 和 8 个字节大的多
- Java 语言的浮点数有两种表示方式: 十进制数形式,浮点数必须包含小数,否则系统将认为是一个int 型 如: 8.0 不能省略为: 8
- 科学计数法形式:
 - 注: 只有浮点数才有科学计数法,,
- Java 语言的浮点数默认是double 型,,如果要表示float 型 必须在数字后面加 f(如: 1.3f),
- Java 的三个特殊的浮点数值:正无穷大,负无穷大和非数
- 所有的正无穷大都是相等,负无穷大都是相等,,而非数不与任何数值相等,与自己也不相等,只有浮点数除0 才可以得到这三个数,整数除0 将报错;

数值中使用下画线分隔

- Java 7引入了一个新功能：程序员可以在数值中使用下画线，不管是整型数值，还是浮点型数值，都可以自由地使用下画线。
- 通过使用下画线分隔，可以更直观地分辨数值中到底包含多少位。

字符型

- 字符也占2个字节，字符型也可当成整型用。
- 字符型常量以单引号 ‘ ’ 括起来
- 字符常量有三种表示形式：
 - 直接通过单个字符来指定字符常: 如: ‘a’, ‘2’ 等
 - 通过转义字符: 如: ‘\n’
 - 直接使用Unicode 值来表示字符常量
- 特别注意: 字符和字符串存在很大的区别,,字符串是以双引号括起来,而字符串则是引用类型。

boolean 类型

- boolean 类型的值只能是true 和false
- 通常用在判断条件中使用boolean型

使用var声明局部变量

- 使用var定义的局部变量的类型由编译器自动推断：定义变量时分配了什么类型的初始值，那该变量就是什么类型。
- 使用var定义的变量依然有明确的类型——为局部变量指定初始值时，该变量的类型就确定下来了。
- 使用var声明变量简化了变量的声明语法，但降低了程序的可读性。在以下两种情况下应该避免使用var声明变量。
 - 变量的类型不容易判断——比如变量的初始值是由复杂的方法调用所得到的。
 - 局部变量的使用范围很长——随着局部变量的使用范围的增大，后面的代码就更难判断该变量的类型。

基本数据类型的类型转换

- 自动类型的转换：表数范围小的自动转换为表数范围大的。
- 注意：byte和short 不能和char 相互转换,这是由于char 的取值范围是从0 到65535。 ,而byte 和short 中都包含负数,所以不能相互转换。
- 我们经常用到的是任何数据类型和字符串相连都会转换为字符串类型。使用 + 来连接。

强制类型转化

- 强制类型转换的运算符是一对圆括号()。
要将表数范围大的数据类型 转换为表数范围小的数据类型就要用()里面写小的数据类型
- 将变量转换为另一种类型。
 - 通常情况下,字符串不能直接转换为基本类型,如需转换使用.parseXxx(String s) 的静态方法用于将字符串转换成基本类型。

表达式的自动提升

- 如果运算中有多种类型的,表达式将结果自动提升为最大的类型。

直接量

- 直接在源代码中指定的值。
- 能指定直接量的只有3种类型：基本类型、字符串类型和null类型。

运算符

- 算术运算符: $+$, $-$, $*$, $/$, $\%$, $++$, $--$
 - 注意,自加和自减放在变量前后的区别: 放在前面先完成自加和自减再把变量拿出来用,在后面先拿出来用,再自加和自减
 - 自加和自减只能用于操作变量,不能用于操作常量
- 赋值运算符: $=$ 、 $+=$ 、 $*=$ 、 $/=$ 、 $\%=$;
- 位运算符: $\&$ 、 $|$ 、 \sim 、 \wedge 、 \ll 、 \gg 、 \ggg 。
- 扩展后的赋值运算符
- 比较运算符 $<$ 、 $>$ 、 \geq 、 \leq 、 $!=$ 、 $==$
- 逻辑运算符: $\&\&$ 、 $\|\|$ 、 $!$ 、 \wedge
- 三目运算符 条件? true的取值: false 的取值
- 优先级,,,尽量用小括号,,,引起,这样可读性更好。