

第二章 Java基础语法1

贾生



务实、协助、服务、创新

内容提纲

- ❖ 关键字
- ❖ 注释
- ❖ 标识符
- ❖ 常量/变量
- ❖ 数据类型及转换
- ❖ 进制表现形式及转换



关键字

- ❖ 关键字的定义： 被**Java**语言赋予了特殊的含义，并且所有字母为小写字母。
- ❖ 在Editplus/Urltraedit编辑工具中被高亮表示，易于观察
- ❖ goto和const作为保留关键字，并不去使用



关键字

用于定义数据类型的关键字

class	interface	byte	short	int
long	float	double	char	boolean
void				

用于定义数据类型值的关键字

true	false	null		
------	-------	------	--	--

用于定义流程控制的关键字

if	else	switch	case	default
while	do	for	break	continue
return				

关键字

用于定义访问权限修饰符的关键字				
private	protected	public		
用于定义类，函数，变量修饰符的关键字				
abstract	final	static	synchronized	
用于定义类与类之间关系的关键字				
extends	implements			
用于定义建立实例及引用实例，判断实例的关键字				
new	this	super	instanceof	
用于异常处理的关键字				
try	catch	finally	throw	throws
用于包的关键字				
package	import			
其他修饰符关键字				
native	strictfp	transient	volatile	assert

- ❖ 用于解释说明源程序，**Java**中注释分为三种
 - 单行注释 格式：//注释内容
 - 多行注释 格式：/*注释内容*/
 - 文档注释 格式：/** 注释内容 */
- ❖ 按照惯例，应该先写注释，再写代码
- ❖ 提高代码的可阅读性,对于单行和多行注释,被注释的文字,不会被JVM(java虚拟机)解释执行。
- ❖ 对于文档注释，是java特有的注释，其中注释内容可以被JDK提供的工具 **javadoc** 所解析，生成一套以网页文件形式体现的该程序的说明文档。

标识符

- 标识符就是给Java程序中的接口、类、方法、变量等命名的字符序列。
- 命名规则
 - 26个英文大小写字母
 - 数字字符
 - \$和_
- 注意事项
 - 数字不可以开头
 - 不能使用关键字
 - 严格区分大小写

Java中的名称规范：

包名：多单词组成时所有字母都小写。

- **com.sina.user**

类名接口名：多单词组成时，所有单词的首字母大写

- **UserAgentInterface**

变量名和方法名：多单词组成时，第一个单词首字母小写，第二个单词开始每个单词首字母大写。

- **getMaxAge()**

常量名：所有字母都大写。多单词时每个单词用下划线连接。

- **MAX_INTEGER_VALUE**

常量和变量

❖ 常量表示其值不可改变

■ Java中常量的分类:

- 1, 整数常量。所有整数
- 2, 浮点型常量。所有小数, 包括单精度(float)和双精度(double), 默认双精度
- 3, 布尔型常量。较为特殊, 只有两个数值。true 或false
- 4, 字符常量。将一个数字字母或者符号用单引号(')标识。
- 5, 字符串常量。将一个或者多个字符用双引号标识。

❖ 转义字符 “\”, 讲其后的转义字符转变为其他含义。

- \n 换行, 讲当前位置移动到下一行
- \t 水平制表, 讲当前位置移动到下一行
- \" 双引号字符

❖ 变量的概念

- 变量是一个基本的存储单元，其存储的数据是可以改变的，程序通过一个变量名来代表该变量存储的数据。

❖ 变量的作用域/生存期

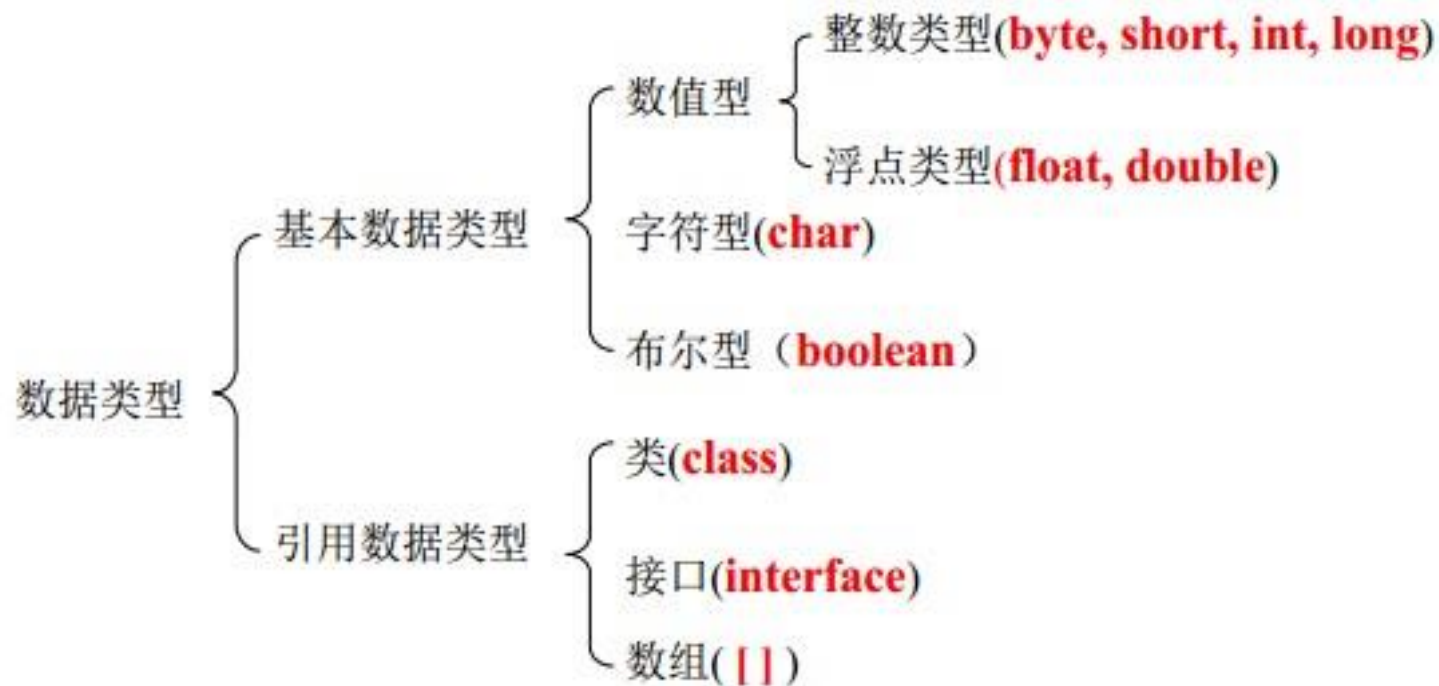
- 变量的作用域（一对{}之间有效），且不能使用相同变量名
- {}内变量可以使用{}外的变量，{}外的不可以使用内的，局部变量在使用之前必须赋值。一条语句可以同时定义多个变量。

❖ 变量声明格式

- 数据类型 标识符 [= 值];

基本数据类型

❖ **Java**语言是强类型语言，对于每一种数据都定义了明确的 具体数据类型，在内存总分配了不同大小的内存空间。



数据类型转换

❖ 隐式转换

- `int + int`
- `byte + int` (类型提升)

❖ 强制转换

- `byte = byte + int`
- 强转注意事项，向下强转会损失精度
- `byte b = (byte)(130)`

❖ `byte, char, short` 参与运算时不会自动转换，会自动提升为 `int` 类型

- `byte, char, short -> int -> long -> float -> double`。

❖ 字符(串)运算

- `System.out.println('a' + 1);`
- `System.out.println((char)('a' + 1));`
- `System.out.println("6+6=" + 6 + 6);`

❖ **ASCII**码表

❖ **char**数据类型

- `char c = 97; System.out.println(c2);`
- 可以存储中文，Java 是采用Unicode编码，其每个字符占用两个字节，中文同样占用两个字节。

进制表现形式

❖ 二进制

- 由0和1组成（以0b）开头，jdk1.7版本引进

❖ 八进制

- 由0,1,2,3,4,5,6,7组成。以0开头

❖ 十进制

- 由 0,1,2,3,4,5,6,7,8,9组成，默认十进制

❖ 十六进制

- 由0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f以0x开头，大小写都可以。

❖ 任意进制到十进制的转换

- 把系数 \times 基数的权次幂相加即可

❖ 十进制到任意进制的转换 除积倒取余

❖ 快速进制转换法(**8421**码)

- 在这种编码方式中每一位二进制代码代表一个 固定的数值，每一位1代表的数值加起来，得到的它的十进制数
- 二进制转换成八进制
- 二进制转换成十六进制

原码/反码/补码

- ❖ 对于一个数, 计算机要使用一定的编码方式进行存储. 原码, 反码, 补码 是机器存储一个具体数字的编码方式. **Java**的强制类型转换中有所涉及
- ❖ 原码
 - 原码就是符号位加上真值的绝对值, 即用第一位表示符号, 其余位表示值
- ❖ 反码
 - 正数的反码是其本身
 - 负数的反码是在其原码的基础上, 符号位不变, 其余各个位取反
- ❖ 补码
 - 正数的补码就是其本身
 - 负数的补码是在其原码的基础上, 符号位不变, 其余各位取反, 最后+1. (即在反码的基础上+1)

作业

- ❖ 请说出常用的关键字
- ❖ 注释分为哪几类，描述各自的功能
- ❖ 描述标识符的命名规则
- ❖ 常量和变量概念、作用域、命名格式
- ❖ 数据类型有哪几种，分别是什么
- ❖ 数据类型的隐式转换和强制转换
- ❖ 进制转换