【济南中心】JAVA 编程阶梯：基础篇之第二章

关键字

概述:被 Java 语言赋予特定含义的单词

特点:组成关键字的字母全部小写

注意事项:

goto 和 const 作为保留字存在,目前并不使用(保留字：在 JDK 的新版本中可能提升为关键字)

类似 Notepad++这样的高级记事本,针对关键字有特殊的颜色标记，非常直观

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				

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

标识符

概述：给类,接口,方法,变量等起名字时使用的字符序列

组成规则：英文大小写字母、数字字符、$和_命名规范:

包名：多单词组成时所有字母都小写,

类名接口名：多单词组成时，所有单词的首字母大写

变量名和函数名：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

常量名：所有字母都大写。多单词时每个单词用下划线连接

注意事项 ：不能以数字开头、不能是 Java 中的关键字、区分大小写要见名知意，驼峰命名

常量

概述：在程序执行的过程中其值不可以发生改变

常量分类：

1、字面值常量

2、自定义常量(面向对象部分讲)

字面值常量的分类

1. 整数常量：所有整数。
2. 小数常量：所有小数。
3. 布尔(boolean)型常量：只有两个数值，true、false。
4. 字符常量：将一个数字字母或者符号用单引号(' ')标识，如：'a'。
5. 字符串常量：将一个或者多个字符用双引号("")标识，如："hello world"、"a"、""（空字符串）。
6. null 常量：只有一个数值就是：null。

Java 针对整数常量提供了 4 种表现形式：二进制、八进制、十进制 、十六进制

进制：就是进位制，是人们规定的一种进位方法。对于任何一种进制--X 进制，就表示某一位置上的数运算时是逢 X 进一位。二进制就是逢二进一，八进制是逢八进一，十进制是逢十进一，十六进制是逢十六进一、

二进制的由来：任何数据在计算机中都是以二进制的形式存在的。二进制早期由电信号开关演变而来。

一个整数在内存中一样也是二进制的，但是使用一大串的 1 或者 0 组成的数值进行使用很麻烦。

八进制的由来：所以就想把一大串缩短点，将二进制中的三位用一位表示。

这三位可以取到的最大值就是 7.超过 7 就进位了，这就是八进制。

十六进制的由来：但是对于过长的二进制变成八进制还是较长，所以出现的用 4 个二进制位表示一位的情况，四个二进制位最大是 15，这就是十六进制。

不同进制的数据组成：

二进制 由 0,1 组成。以 0b 开头 例：0101

八进制 由 0,1,...7 组成。以 0 开头 例：032

十进制 由 0,1,...9 组成。整数默认是十进制的 例：12

十六进制 由 0,1,...9,a,b,c,d,e,f(大小写均可)。以 0x 开头 例：0x42b6

规律：进制越大，表现形式越短。

转换(可以通过计算机的程序员计算器)

8421 码（8421 码是 BCD 代码中最常用的一种。）

100 转成二进制 0110 0100

101101 转成十进制 45

二进制和八进制，十六进制进行转换是以十进制作为桥梁

二进制到八进制是 3 位二进制组成一位八进制

二进制到十六进制是 4 位二进制组成一位十六进制

在计算机内，有符号数有 3 种表示法：原码、反码和补码。所有数据的

运算都是采用补码进行的。

原码：就是二进制定点表示法，即最高位为符号位，“0”表示正，“1”表示负，其余位表示数值的大小。

反码：正数的反码与其原码相同；负数的反码是对其原码逐位取反，但符号位除外。

补码：正数的补码与其原码相同；负数的补码是在其反码的末位加 1。

变量

概述：在程序执行的过程中，在某个范围内其值可以发生改变的量

格式：数据类型 变量名 = 变量值;

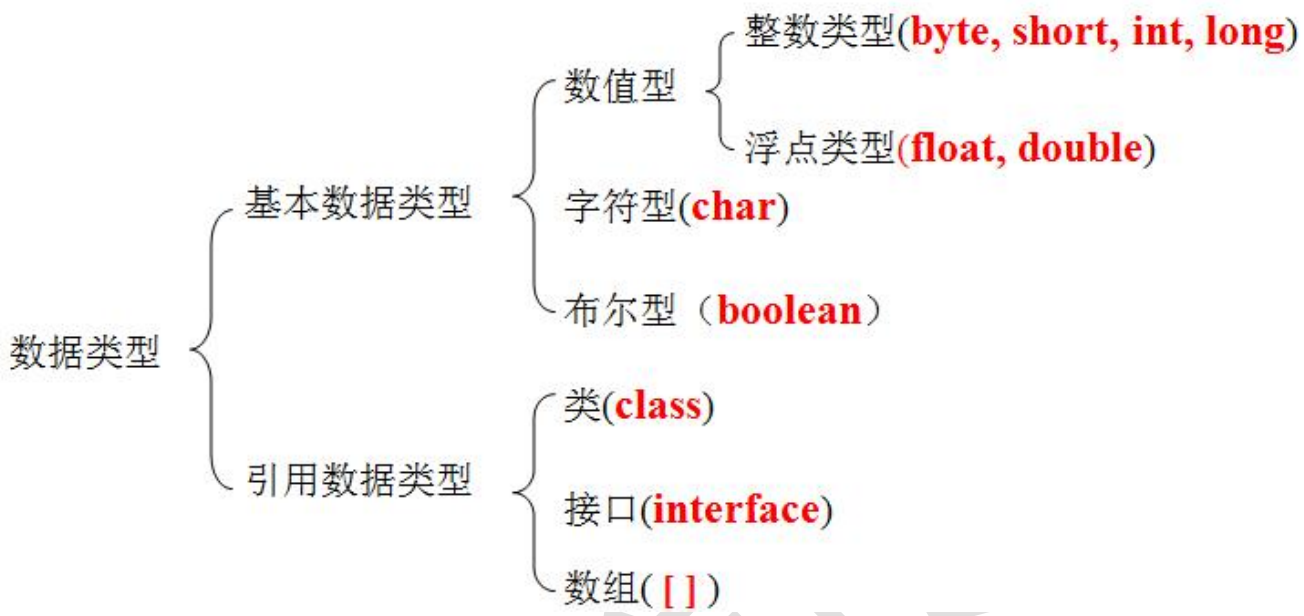
好处： 可以用来不断的存放同一类型的常量，并重复使用

数据类型

概述：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，在内存中分配了不同大小的内存空间

Java 中数据类型的分类

分类：基本数据类型(4 类 8 种) 和引用数据类型



整数型

byte 占一个字节 -128 到 127

short 占两个字 -2^15~2^15-1

int 占四个字节 -2^31~2^31-1

long 占八个字节 -2^63~2^63-1

浮点型

float 占四个个字节 -3.403E38~3.403E38

double 占八个字节-1.798E308~1.798E308

字符型

char 占两个字节 0~65535

布尔型

boolean

boolean 理论上是占八分之一个字节,因为一个开关就可以决定是 true 和 false 了,但是 java

中 boolean 类型没有明确指定他的大小

注意事项

作用域 ：变量定义在哪一级大括号中，哪个大括号的范围就是这个变量的作用域。相同的作用域中不能定义两个同名变量。

初始化值：没有初始化值不能直接使用

在一行上可以定义多个变量，但是不建议只定义一个

数据转换

默认转换

byte,short,char—int—long—float—double

byte,short,char 相互之间补转换，他们参与运算首先转换为 int 类型

强制转换

目标类型 变量名=(目标类型)(被转换的数据);

boolean 类型不能转换为其他的数据类型

运算符

算术运算符

运算符	运算	范例	结果
+	正号	+3	3
-	负号	b=4;-b;	-4
+	加	5+5	10
-	减	6-4	2
*	乘	3*4	12
/	除	5/5	1
%	取模	5%5	0
++	自增（前）	a=2;b=++a;	a=3;b=3
++	自增（后）	a=2;b=a++;	a=3;b=2
--	自减（前）	a=2;b=--a	a=1;b=1
--	自减（后）	a=2;b=a--	a=1;b=2
+	字符串相加	"He"+"llo"	"Hello"

加法（+）： 1、加法 System.out.println("a+b="+(a+b)) ,此处是求(a+b)的值

2、正数 int b = +3; 此处表示 b 是正数 3

3、字符串连接符 System.out.println("a=“+a+"b="+b)) ‘+’为字符串连接符号，不参与运算，显示的为 a=a, b=b

除法（/）：1、整数相除，只能得到整数 2、要想得到小数，可以*1.0

/和%的区别：

除法和平时用法一样，取模就是取余数，负数对正数取模结果为负数。正数对负数取模结果为正数。**举个栗子：**

-2%5= -2 -2 是被模数，5 是模数，负数的取模运算结果是不是负数看左边。

++和--的应用：

++运算单独存在时放左放右是没有区别的，

参与其他运算时 b = ++a 相等于 b = a+1，

b = a++ b 应为 a 的初始值，a 为 a+1

赋值运算符

符号：

= , +=, -=, *=, /=, %=

=为基本的赋值运算符，其他的为扩展的赋值运算符

=赋值号

+=加赋值

把左边和右边的结果赋值给左边。注意：左边不能是常量

关系运算符

比较运算符

运算符	运算	范例	结果
==	相等于	4==3	false
!=	不等于	4!=3	true
<	小于	4<3	false
>	大于	4>3	true
<=	小于等于	4<=3	false
>=	大于等于	4>=3	false
instanceof	检查是否是类的对象	"Hello" instanceof String	ture

比较运算符的结果都是 boolean 型，也就是要么是 true，要么是 false。

比较运算符“==”不能误写成“=” 。

识别二维码 关注黑马程序员视频库
免费获得更多 IT 资源