**Java 基本数据类型**

变量就是申请内存来存储值。也就是说，当创建变量的时候，需要在内存中申请空间。

内存管理系统根据变量的类型为变量分配存储空间，分配的空间只能用来储存该类型数据。

因此，通过定义不同类型的变量，可以在内存中储存整数、小数或者字符。

Java的两大数据类型:

* 内置数据类型
* 引用数据类型

**内置数据类型**

Java语言提供了八种基本类型。六种数字类型（四个整数型，两个浮点型），一种字符类型，还有一种布尔型。

byte：

* byte数据类型是8位、有符号的，以二进制补码表示的整数；
* 最小值是-128（-2^7）；
* 最大值是127（2^7-1）；
* 默认值是0；
* byte类型用在大型数组中节约空间，主要代替整数，因为byte变量占用的空间只有int类型的四分之一；
* 例子：byte a = 100，byte b = -50。

short：

* short数据类型是16位、有符号的以二进制补码表示的整数
* 最小值是-32768（-2^15）；
* 最大值是32767（2^15 - 1）；
* Short数据类型也可以像byte那样节省空间。一个short变量是int型变量所占空间的二分之一；
* 默认值是0；
* 例子：short s = 1000，short r = -20000。

int：

* int数据类型是32位、有符号的以二进制补码表示的整数；
* 最小值是-2,147,483,648（-2^31）；
* 最大值是2,147,485,647（2^31 - 1）；
* 一般地整型变量默认为int类型；
* 默认值是0；
* 例子：int a = 100000, int b = -200000。

long：

* long数据类型是64位、有符号的以二进制补码表示的整数；
* 最小值是-9,223,372,036,854,775,808（-2^63）；
* 最大值是9,223,372,036,854,775,807（2^63 -1）；
* 这种类型主要使用在需要比较大整数的系统上；
* 默认值是0L；
* 例子： long a = 100000L，Long b = -200000L。

float：

* float数据类型是单精度、32位、符合IEEE 754标准的浮点数；
* float在储存大型浮点数组的时候可节省内存空间；
* 默认值是0.0f；
* 浮点数不能用来表示精确的值，如货币；
* 例子：float f1 = 234.5f。

double：

* double数据类型是双精度、64位、符合IEEE 754标准的浮点数；
* 浮点数的默认类型为double类型；
* double类型同样不能表示精确的值，如货币；
* 默认值是0.0d；
* 例子：double d1 = 123.4。

boolean：

* boolean数据类型表示一位的信息；
* 只有两个取值：true和false；
* 这种类型只作为一种标志来记录true/false情况；
* 默认值是false；
* 例子：boolean one = true。

char：

* char类型是一个单一的16位Unicode字符；
* 最小值是’\u0000’（即为0）；
* 最大值是’\uffff’（即为65,535）；
* char数据类型可以储存任何字符；
* 例子：char letter = ‘A’。

**实例**

对于数值类型的基本类型的取值范围，我们无需强制去记忆，因为它们的值都已经以常量的形式定义在对应的包装类中了。请看下面的例子：

public class PrimitiveTypeTest {

public static void main(String[] args) {

// byte

System.out.println("基本类型：byte 二进制位数：" + Byte.SIZE);

System.out.println("包装类：java.lang.Byte");

System.out.println("最小值：Byte.MIN\_VALUE=" + Byte.MIN\_VALUE);

System.out.println("最大值：Byte.MAX\_VALUE=" + Byte.MAX\_VALUE);

System.out.println();

// short

System.out.println("基本类型：short 二进制位数：" + Short.SIZE);

System.out.println("包装类：java.lang.Short");

System.out.println("最小值：Short.MIN\_VALUE=" + Short.MIN\_VALUE);

System.out.println("最大值：Short.MAX\_VALUE=" + Short.MAX\_VALUE);

System.out.println();

// int

System.out.println("基本类型：int 二进制位数：" + Integer.SIZE);

System.out.println("包装类：java.lang.Integer");

System.out.println("最小值：Integer.MIN\_VALUE=" + Integer.MIN\_VALUE);

System.out.println("最大值：Integer.MAX\_VALUE=" + Integer.MAX\_VALUE);

System.out.println();

// long

System.out.println("基本类型：long 二进制位数：" + Long.SIZE);

System.out.println("包装类：java.lang.Long");

System.out.println("最小值：Long.MIN\_VALUE=" + Long.MIN\_VALUE);

System.out.println("最大值：Long.MAX\_VALUE=" + Long.MAX\_VALUE);

System.out.println();

// float

System.out.println("基本类型：float 二进制位数：" + Float.SIZE);

System.out.println("包装类：java.lang.Float");

System.out.println("最小值：Float.MIN\_VALUE=" + Float.MIN\_VALUE);

System.out.println("最大值：Float.MAX\_VALUE=" + Float.MAX\_VALUE);

System.out.println();

// double

System.out.println("基本类型：double 二进制位数：" + Double.SIZE);

System.out.println("包装类：java.lang.Double");

System.out.println("最小值：Double.MIN\_VALUE=" + Double.MIN\_VALUE);

System.out.println("最大值：Double.MAX\_VALUE=" + Double.MAX\_VALUE);

System.out.println();

// char

System.out.println("基本类型：char 二进制位数：" + Character.SIZE);

System.out.println("包装类：java.lang.Character");

// 以数值形式而不是字符形式将Character.MIN\_VALUE输出到控制台

System.out.println("最小值：Character.MIN\_VALUE="

+ (int) Character.MIN\_VALUE);

// 以数值形式而不是字符形式将Character.MAX\_VALUE输出到控制台

System.out.println("最大值：Character.MAX\_VALUE="

+ (int) Character.MAX\_VALUE);

}

}

编译以上代码输出结果如下所示：

基本类型：byte 二进制位数：8

包装类：java.lang.Byte

最小值：Byte.MIN\_VALUE=-128

最大值：Byte.MAX\_VALUE=127

基本类型：short 二进制位数：16

包装类：java.lang.Short

最小值：Short.MIN\_VALUE=-32768

最大值：Short.MAX\_VALUE=32767

基本类型：int 二进制位数：32

包装类：java.lang.Integer

最小值：Integer.MIN\_VALUE=-2147483648

最大值：Integer.MAX\_VALUE=2147483647

基本类型：long 二进制位数：64

包装类：java.lang.Long

最小值：Long.MIN\_VALUE=-9223372036854775808

最大值：Long.MAX\_VALUE=9223372036854775807

基本类型：float 二进制位数：32

包装类：java.lang.Float

最小值：Float.MIN\_VALUE=1.4E-45

最大值：Float.MAX\_VALUE=3.4028235E38

基本类型：double 二进制位数：64

包装类：java.lang.Double

最小值：Double.MIN\_VALUE=4.9E-324

最大值：Double.MAX\_VALUE=1.7976931348623157E308

基本类型：char 二进制位数：16

包装类：java.lang.Character

最小值：Character.MIN\_VALUE=0

最大值：Character.MAX\_VALUE=65535

Float和Double的最小值和最大值都是以科学记数法的形式输出的，结尾的"E+数字"表示E之前的数字要乘以10的多少倍。比如3.14E3就是3.14×1000=3140，3.14E-3就是3.14/1000=0.00314。

实际上，JAVA中还存在另外一种基本类型void，它也有对应的包装类 java.lang.Void，不过我们无法直接对它们进行操作。

**引用类型**

* 引用类型变量由类的构造函数创建，可以使用它们访问所引用的对象。这些变量在声明时被指定为一个特定的类型，比如Employee、Pubby等。变量一旦声明后，类型就不能被改变了。
* 对象、数组都是引用数据类型。
* 所有引用类型的默认值都是null。
* 一个引用变量可以用来引用与任何与之兼容的类型。
* 例子：Animal animal = new Animal(“giraffe”)。

**Java常量**

常量就是一个固定值。它们不需要计算，直接代表相应的值。

常量指不能改变的量。 在Java中用final标志，声明方式和变量类似：

final double PI = 3.1415927;

虽然常量名也可以用小写，但为了便于识别，通常使用大写字母表示常量。

字面量可以赋给任何内置类型的变量。例如：

byte a = 68;

char a = 'A'

byte、int、long、和short都可以用十进制、16进制以及8进制的方式来表示。

当使用常量的时候，前缀0表示8进制，而前缀0x代表16进制。例如：

int decimal = 100;

int octal = 0144;

int hexa = 0x64;

和其他语言一样，Java的字符串常量也是包含在两个引号之间的字符序列。下面是字符串型字面量的例子：

"Hello World"

"two\nlines"

"\"This is in quotes\""

字符串常量和字符常量都可以包含任何Unicode字符。例如：

char a = '\u0001';

String a = "\u0001";

Java语言支持一些特殊的转义字符序列。

|  |  |
| --- | --- |
| **符号** | **字符含义** |
| \n | 换行 (0x0a) |
| \r | 回车 (0x0d) |
| \f | 换页符(0x0c) |
| \b | 退格 (0x08) |
| \s | 空格 (0x20) |
| \t | 制表符 |
| \" | 双引号 |
| \' | 单引号 |
| \\ | 反斜杠 |
| \ddd | 八进制字符 (ddd) |
| \uxxxx | 16进制Unicode字符 (xxxx) |

这一节讲解了Java的基本数据类型。下一节将探讨不同的变量类型以及它们的用法。