

Java数据类型划分

一，数据类型简介

程序是一套数字处理的游戏框架，也就说成功开发过程之中，所达到的最终数据定义。

1.1 Java数据类型分类

Java语言之中一共分为两类：

基本数据类型：描述的是一些数字单元；

数值型：

整型：byte\short\int\long: ——>默认值：0

浮点型：float\double; ——>默认值：0.0

布尔型：boolean; ——>默认值：false

字符型：char:

引用数据类型：牵扯到内存关系的使用；

数组、类、接口 ——>默认值：null

每一种数据类型都有数字的保存类型。

名称	类型	比特数	最小取值范围
整型	int	16	-32768~32767
短整型	short	16	-32768~32767
长整型	long	32	-2147483648~2147483647
无符号整型	unsigned int	16	0~65535
无符号短整型	unsigned short	16	0~65535
无符号长整型	unsigned long	32	0~4294967295
单精度型	float	32	10^{-37} ~ 10^{38}
双精度型	double	64	10^{-37} ~ 10^{308}
字符型	char	8	-128~127

参考原则：

如果要是描述数首选的一定是int（整数），double（小数）；

如果要进行数据传输或者是进行文字编码转化使用byte类型（二进制处理操作）

处理中文的时候最方便的操作使用的是字符char来完成（可选概念）

描述内存或文件大小，描述表主键列（自动增长）可以使用Long；

二，整型数据类型

整形数据一共有四种，能保存的范围由小到大：byte\short\int\long，那么在Java里面任何一个整型常量那么其默认的类型都收Int型（只要是整数就是Int类型的数据）

范例：定义int型变量；

```
public class JavalDt {  
    public static void main(String args []) {  
        // int变量名称 = 变量（10是一个常量，整数类型为int）；  
        int x = 10; //定义了一个整型变量x  
        //int型变量 * int型变量 = int型数据  
        System.out.println(x * x);  
    }  
}
```

```
}  
}
```

其输出结果是100；10永远不会改变，但是x是一个变量，x内容是可以改变的。

范例，修改x变量；

```
public class Javaltd {  
    public static void main(String args []) {  
        // int变量名称 = 变量 ( 10是一个常量，整数类型为int ) ;  
        int x = 10 ; //定义了一个整型变量x  
        x = 20 ; //改变了x已有的内容  
        //int型变量 * int型变量 = int型数据  
        System.out.println(x * x);  
    }  
}
```

其输出结果是400，改变了x的已有数据结果

任何的数据类型都是有其可以保存的数据范围的（正常使用下你很少会出现超过此范围的数据），如果真的超过了，会如何？

范例，如果超过Int范围会咋样

```
public class JavaStudy {  
    public static void main(String args[]) {  
        int max = Integer.MAX_VALUE ; //获取int最大值  
        int min = Integer.MIN_VALUE ; //获取int最小值  
        System.out.println(max) ; //2147483647  
        System.out.println(min) ; //-2147483648  
        System.out.println("-----");  
        //int型变量 + int型常量 = int型计算结果  
        System.out.println(max + 1) ; //-2147483648,最大值 + 1 = 最小值  
        System.out.println(max + 2) ; //-2147483647,最小值 + 2 = 次最小值  
        //int型变量 - int型常量 = int型计算结果。  
        System.out.println(max - 1) ; //2147483646,最小值 - 1 = 最大值，产生的循环过程  
    }  
}
```

通过此时的执行结果可以发现这些数字在进行处理的hi后如果超过了最大的保存范围，则会出现循环的问题，而这样的问题在Java里面称为数据溢出，那么如果想要解决这种溢出，那么可以继续扩大使用范围，比如Int

范例：解决数据溢出

在错做的时候预估数据范围，如果发现范围不够就使用更大的范围
（第一种解决方式）

```
public class JavaStudy {  
    public static void main(String args[]) {  
        //long long变量 = int的数值  
        long max = Integer.MAX_VALUE ; //获取int最大值  
        long min = Integer.MIN_VALUE ; //获取int最小值  
        System.out.println(max) ; //2147483647  
        System.out.println(min) ; //-2147483648  
        System.out.println("-----");  
        //long型变量 + long型常量 = long型计算结果  
        System.out.println(max + 1) ; //2147483648  
        System.out.println(max + 2) ; //2147483649  
        //long型变量 - long型常量 = long型计算结果。  
        System.out.println(min - 1) ; //-2147483649  
    }  
}
```

```

    }
}

```

除了可以定义long型变量之外，也可以直接在常量上进行处理，默认的正好素常两都是Int型，那么可以为他追加字母" L"或直接使用" (long)" "

```

public class JavaStudy {
    public static void main(String args[]) {
        //long long变量 = int的数值
        int max = Integer.MAX_VALUE ; //获取int最大值
        int min = Integer.MIN_VALUE ; //获取int最小值
        System.out.println(max) ; //2147483647
        System.out.println(min) ; //-2147483648
        System.out.println("-----");
        //int型变量 + long型常量 = long型计算结果
        System.out.println(max + 1L) ; //2147483648
        System.out.println(max + 2l) ; //2147483649
        //long型变量 - Int型常量 = long型计算结果。
        System.out.println((long)min - 1) ; //-2147483649
    }
}

```

现在发现数据之间是可以互相转换的，即：范围小的数据类型可以转为数据大的数据类型，但是反过来，凡伟大的数据类型要转换成小的数据类型，那就需要采用强制性的处理模式，同时还需要考虑可能带来的数据溢出。

范例：强制类型转换

```

public class JavaStudy {
    public static void main(String args[]) {
        long num = 2147483649L ; //此数据已经超过了int的范围
        int temp = (int)num ; //long范围比int范围大，不能够直接转换
        System.out.println(temp) ;
    }
}

```

程序只持有数据转换处理，但是如果不是必须情况下建议不建议这种转换。在进行整型处理的时候，还有一个byte类型特需要主要，这个类型的范围是：-128~127之间，保存的数据量很小。

范例：定义byte变量

```

public class JavaStudy {
    public static void main(String args[]) {
        byte num = 20 ;
        System.out.println(num) ;
    }
}

```

正常的情况将在Java程序里面20这个数字应该是Int型，但是在为byte赋值的时候并没有因为是Int发生强制类型转换，因为Java对byte进行了特殊处理，如果没有超过byte范围的常量，可以自动有Int变为byte，如果超过则会出现强制转换。

```

public class JavaStudy {
    public static void main(String args[]) {
        byte num = (byte) 200 ;
        System.out.println(num) ;
    }
}

```

其输出结果超过了200

```

public class JavaStudy {
    public static void main(String args[]) {
        int x = 200;
        byte num = x;
        System.out.println(num);
    }
}

```

由于200已经超过了byte的范围，所以产生了数据溢出的问题，但是需要提醒的是，对于变量还是必须使用强制转换处理。

三.浮点型数据

浮点型数据描述的是小数，而在Java里面任意的一个小数常量其对应的类型为double，所以double在以后描述小数的时候使用double进行定义。

```

public class JavaStudy {
    public static void main(String args[]) {
        //10.2一个小数其对应的类型为double
        double x = 10.2;
        int y = 10;
        // double类型 * int类型 = double类型
        double result = x * y;
        System.out.println(result);
    }
}

```

```

所有的数据类型进行自动转型的时候由public class JavaStudy {
    public static void main(String args[]) {
        float x = (float) 10.2;
        float y = 10.1F;
        System.out.println(x * y); //float型
    }
}

```

小类型转向大类型进行自动转换处理。

默认的类型double，但是也可以定义位数相对较少的float变量，此时赋值的时候就必须采用强制类型转换。

范例：定义一个float变量

```

public class JavaStudy {
    public static void main(String args[]) {
        float x = (float) 10.2;
        float y = 10.1F;
        System.out.println(x * y); //float型
    }
}

```

通过以上这些代码进行分析，整型是不包含有小数点的，二浮点型是包含小数点的。

范例：观察一个程序

```

public class JavaStudy {
    public static void main(String args[]) {
        int x = 10;
        int y = 4;
        System.out.println(x / y);
    }
}

```

输出结果为2，整型得到2的主要原因是整型是不保存整数点的，所以计算结果为 2.5 那么删除小数点来看，输出结果为2，如果想得到正确计算，那么则需要进行转型处理。

范例：数据转型

```
public class JavaStudy {  
    public static void main(String args[]) {  
        int x = 10 ;  
        int y = 4 ;  
        System.out.println((double)x / y) ;  
    }  
}
```

其输出结果为2.5，选择的数据类型将直接决定小数点的问题。

四，字符型数据

字符型数据是使用char进行定义的，在Java之中使用""定义的内容就是一个字符

范例：定义一个字符型变量

```
public class JavaStudy {  
    public static void main(String args[]) ;  
    char c = 'B' ; //一个字符变量  
    System.out.println(c) ;  
}  
}
```

在任何编程语言中，字符都可以于Int进行互相转换，就说字符所描述的内容，可以通过Int获取其对应的系统编码、

范例：观察char于Int的关系

```
public class JavaStudy {  
    public static void main(String args[]) {  
        char c = 'A' ; //一个字符变量  
        int num = c ; //可以获得字符的编码  
        System.out.println(num) ;  
    }  
}
```

对于以上程序获得了编码，需要注意一下几个问题；

大写字母范围：'A' (65) ~ 'Z' (90)

小写字母范围：'a' (97) ~ 'z' (122)

数字字母范围：'0' (48) ~ '9' (57)

通过以上编码范围可以发现大小写字母之间差了32个三个一组的长度，在这样的情况下就可以实现一个大小写的转换处理。

范例：将小写字母变为大写字母

```
public class JavaStudy {  
    public static void main(String args[]) {  
        char c = 'x' ; //一个字符变量  
        int num = c ; //可以获得字符编码  
        num = num - 32 ;//让自己内容减少32  
        System.out.println((char)num) ;  
    }  
}
```

到此为止的所有操作，都与C语言的从阿U走都一样的，但是注意的是，在Java里面char主要是进行中文的处理，所以一定要记住，Java里面的char可以保存中文数据。

范例：保存中文数据

```
public class JavaStudy {  
    public static void main(String args[]) {  
        char c = '樱'; //一个字符变量  
        int num = c; //可以获得字符编码  
        num = num - 32; //让自己内容减少32  
        System.out.println(num);  
    }  
}
```

char最大的优势就是可以储存中文。可以使用char之所以可以进行中文数据的保存，是因为Java语言所使用的unicode这种十六进制的编码，整个编码特点就是保存任意文字的编码

五.布尔数据

布尔是一位数学家的名字，主要描述的是一种逻辑的处理结果，在Java中使用boolean来进行布尔类型的变量定义，布尔类型的取值范围只有两个数据：true\false

范例：定义一个布尔型数据

```
public class JavaStudy {  
    public static void main(String args[]) {  
        boolean flag = true;  
        if (flag) { //判断flag内容，如果是true就执行  
            System.out.println("樱");  
        }  
    }  
}
```

像一些编程语言由于没有提供布尔类型，所以会使用0表示false，或者使用非0 表示true，而这个结果不会在Java里面实现。

六，string字符串

在任何语言之中，都没有所谓提供字符串这种基本数据类型，但是从实际上来讲各个语言为了方便程序的开发，也都会提供有字符串的相应描述。在Java里面使用的是string作为字符串的定义。

由于string类的存在比较特殊，说以其可以像普通那样采用直接赋值的方式进行字符串的定义。并且要求使用""进行字符串描述

范例：定义一个字符串

```
public class JavaStudy {  
    public static void main(String args[]) {  
        String str = "Hello,world!"; // 使用""进行描述  
        System.out.println(str);  
    }  
}
```

在进行字符串变量使用的时候也可以使用“+”来进行字符串的连接处理。

范例：实现字符串链接

```
public class JavaStudy {  
    public static void main(String args[]) {  
        String str = "Hello"; //使用""进行描述  
        str = str + " World"; //字符串连接  
        str += " !!!";  
        System.out.println(str);  
    }  
}
```

但是要考一点，这时候对“+”就有了两种描述：字符串的连接、数字的加法计算

范例：观察“+”的问题

```
public class JavaStudy {  
    public static void main(String args[]) {  
        double x = 10.1 ;  
        int y = 20 ;  
        String str = "计算结果： " + x + y ;  
        System.out.println(str) ;  
    }  
}
```

计算结果为 10.120

在Java语言之中，数据范围大的数据类型于数据小的数据类型进行计算的时候，所有范围小的数据类型自动转为范围大的数据类型。

但是此时有string字符串，则所有类型无条件先变为String,如果有“+”表示的是字符串的连接处理操作。

```
public class JavaStudy {  
    public static void main(String args[]) {  
        double x = 10.1 ;  
        int y = 20 ;  
        String str = "计算结果： " + (x + y) ;  
        System.out.println(str) ;  
    }  
}
```

其计算结果为30.1

在描述字符串的时候可以使用转义字符进行处理，例如TAB（\t），"（\"），（'），换行（\n），\（\）

范例：观察转义字符

```
public class JavaStudy {  
    public static void main(String args[]) {  
        System.out.println("\tHello,world!!!! \nHELLO! \"JIANGXUE\"");  
    }  
}
```

这些字符可以在学习之中进行简单的格式化显示，以后双引号定义的都是字符串