Java数据类型划分

# 引言

1. Java数据类型划分方式；
2. 各个常用数据类型的使用。

# 具体内容

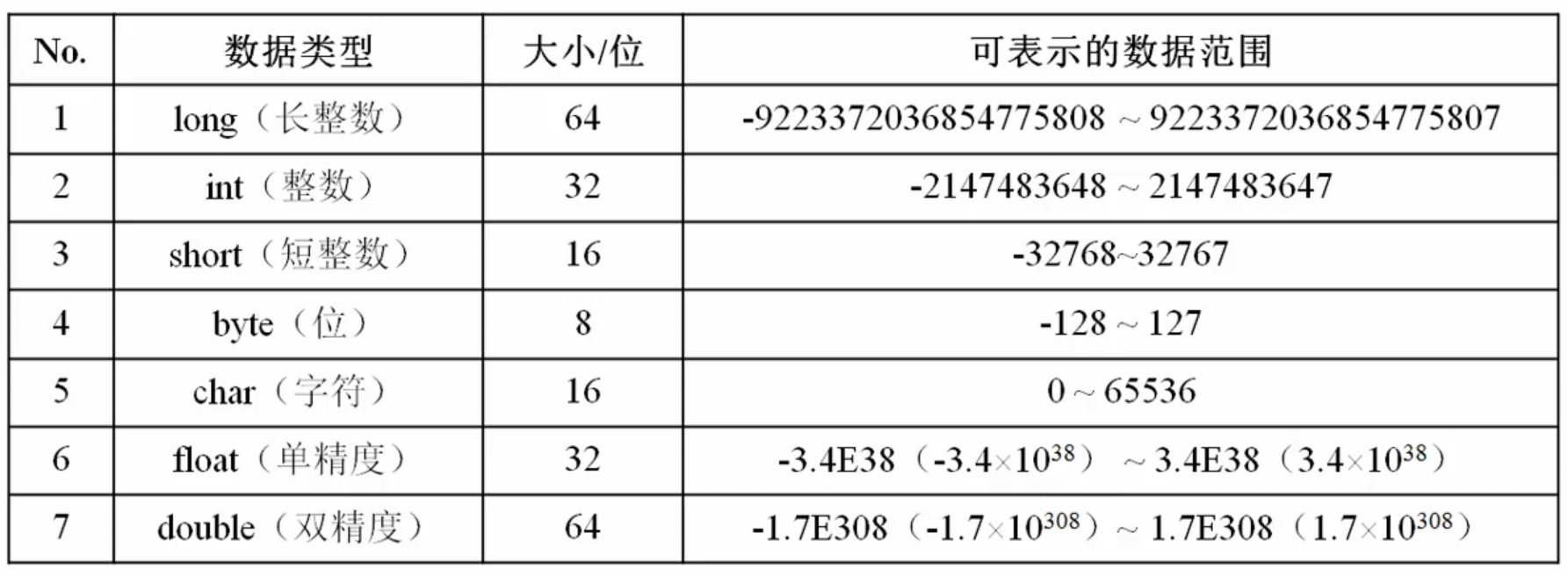
程序的本质实际上就是在于数据的处理上。所以如果要想进行程序的处理，那么必须要有相应的数据类型。在Java中针对于数据类型一共划分为两类：

* 基本数据类型：是进行内容的操作，而不是内存的操作；
  + 数值型：
    - 整型：byte、short、int、long； 默认值：0
    - 浮点型：float、double； 默认值：0.0
  + 字符型：char； 默认值：’\u0000’
  + 布尔型：boolean； 默认值：false
* 引用数据类型：在数据交互过程之中会占用内存空间指向
  + 分为：数组、类、接口； 默认值：null。

对于给出的八种基本数据类型，有其自己特定的使用环境：

* 表示整数首先考虑的就是int，例如：年龄；
* 表示小数一定使用double；
* 表示日期时间数组、文件、内存大小使用long（字节）；
* 进行编码转换、数据传输使用byte类型；
* 处理中文使用char类型；
* 处理逻辑使用boolean类型。

以后考虑代码的时候，float与short可以暂时不考虑。



通过以上的数据范围表就可以发现，double保存的数据范围是最大的。

## 整型

在程序开发之中如果要想表示整数就使用整型，同时如果要想定义整型可以使用byte、short、int、long。一定要记住，在Java之中有一个特点：默认的一个整数，其对应的类型永恒是int。

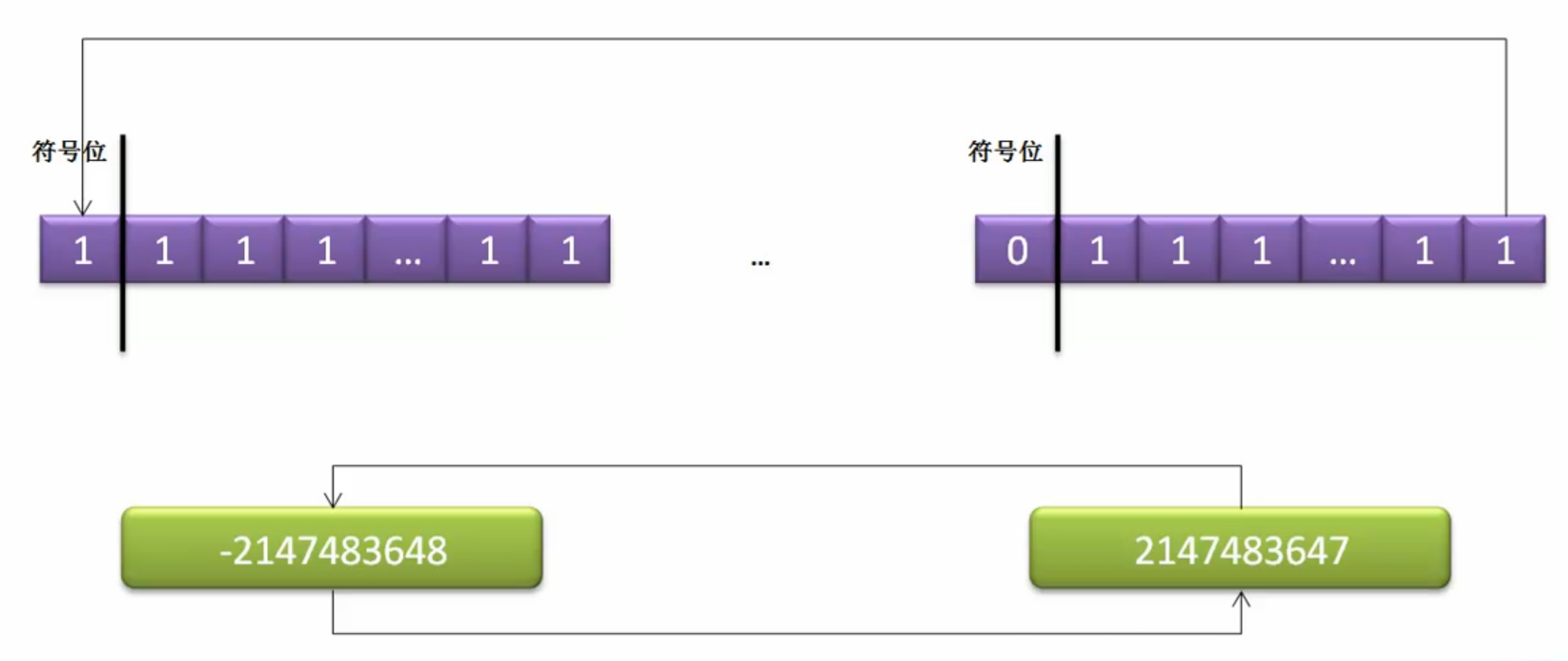
|  |
| --- |
| public class TestDemo {  public static void main(String[] args) {  //定义变量的语法：数据类型 变量名称 = 默认值；  int num = 10;// 定义一个num的int型变量，其内容为10  num = 20;//重复设置内容  System.out.println(num);// 输出num的内容  }  } |

变量的内容在开发之中是允许修改的，而常量的内容是不允许修改的。

|  |
| --- |
| public class TestDemo {  public static void main(String[] args) {  // 定义变量的语法：数据类型 变量名称 = 默认值；  // 10作为一个常量其类型是int型  int num = 10; // 定义一个num的int型变量，其内容为10  // 代码从等号右边开始执行  // int型变量 = int型变量 + int型常量  num = num + 20; //重复设置内容  System.out.println(num);// 输出num的内容  }  } |

**范例**：数据溢出的操作

|  |
| --- |
| public class TestDemo {  public static void main(String[] args) {  int max = Integer.MAX\_VALUE; //int的最大值  int min = Integer.MIN\_VALUE; //int的最小值  System.out.println(max); //2147483647  System.out.println(min); //-2147483648  //max是int型变量 + int型变量 = int型  System.out.println(max + 1);//-2147483648  System.out.println(max + 2);//-2147483647  System.out.println(min - 1);//2147483647  }  } |



之所以会出现数据的溢出操作，那么本质的问题就在于数据类型的操作位数太短了，那么在这种情况下（正常代码基本不会出现）、所以要想解决此时int的问题，那么就必须扩展操作类型，比int大的是long数据类型。

那么现在就会出现一个问题，给出了一个常量或者是变量怎么变为long呢？

* 将int变量转换为long：（long）int变量；
* 将int常量转换成long：常量L。
* 如果小范围的数据类型变量或常量与大范围的变量或常量计算时，先转换到大范围的数据类型。

范例：解决数据溢出

|  |
| --- |
| public class TestDemo {  public static void main(String[] args) {  int max = Integer.MAX\_VALUE; //int的最大值  int min = Integer.MIN\_VALUE; //int的最小值  System.out.println(max); //2147483647  System.out.println(min); //-2147483648  //max是int型变量 + long常量 = long类型  System.out.println(max + 1L);//214748364  }  } |

这样的转换操作确实很少出现的。但是通过以上的代码实际上发现了一种数据类型自动转换的功能。

不同种数据类型之间是可以进行转换的，而转换原则如下：

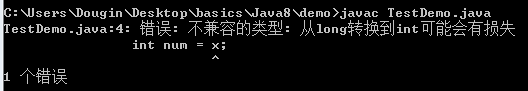
* 数据范围保存大的数据类型要转换为数据范围保存小的数据类型，使用强制转换；
* 数据范围保存小的数据类型可以自动转换为数据范围保存大的数据类型。

范例：数据的自动转换

|  |
| --- |
| public class TestDemo {  public static void main(String[] args) {  int x = 10;  long num = x;  System.out.println(num);  }  } |

范例：数据强制转换

|  |
| --- |
| public class TestDemo {  public static void main(String[] args) {  long x = 10L;  int num = (int)x;  System.out.println(num);  }  } |



在数据类型之中byte表示字节数据，字节数据的范围：-128~127。那么下面来看int与byte的转换。

范例：byte转换为int

|  |
| --- |
| public class TestDemo {  public static void main(String[] args) {  byte x = 130; // 定义byte变量  int num = x; // byte转换int  System.out.println(num);  }  } |

在Java中，由于byte类型较为特殊，那么进行赋值处理时，如果设置的整型（常量）数据在byte范围之内则自动转型，如果超过了byte的范围，那么则需要进行乡下的强制转换。



|  |
| --- |
| public class TestDemo {  public static void main(String[] args) {  byte x = (byte)130; // 定义byte变量  int num = x; // byte转换int  System.out.println(num);// -126  }  } |

现阶段表示数字就使用int。

## 浮点型

如果要表示小数(收入、成绩、身高)使用float、double。但是在Java之中，任何的一个小数常量其对应的类型都使用的是double。

范例：double操作

|  |
| --- |
| public class TestDemo {  public static void main(String[] args) {  double x = 10.0;  double y = 10.9;  System.out.println(x \* y);// 129.0  }  } |

但是千万不要忘记一点，double是整个Java数据类型之中保存范围最大的一个类型。所以任何的数据类型与double类型进行数学计算时，都会自动转换为double类型。

但是double并不是唯一的小数表示，对于小数还可以使用float表示。但是有一个问题：如果想要定义一个float变量，那么必须将double型的常量转换为float型。

|  |
| --- |
| public class TestDemo {  public static void main(String[] args) {  float x = 10.2F;  float y = (float) 10.2;  System.out.println(x);  System.out.println(y);  }  } |

但是这样的操作是有些啰嗦，之所以提出是因为很多时候，Java也许会用在一些内存操作严格的情况下。比如手机。

但是除了小数操作中要使用double类型之外，在一些计算中也需要使用到。

范例：除法计算

|  |
| --- |
| public class TestDemo {  public static void main(String[] args) {  int x = 10;  int y = 4;  System.out.println(x / y); // 2  }  } |

因为整型是不保留小数位的，所以如果利用整型进行数学计算，那么最终的结果小数位将全部被忽略掉。

范例：利用float、double进行小数位的保留。

|  |
| --- |
| public class TestDemo {  public static void main(String[] args) {  int x = 10;  int y = 4;  System.out.println(x / (double)y); // 2.5  }  } |

在大部分情况下，只要与小数点有关的一切操作都使用double完成。

**关于默认值的说明**

默认值现在不起作用。

范例：定义变量不设置内容

|  |
| --- |
| public class TestDemo {  public static void main(String[] args) {  int x ;  System.out.println(x);  }  } |
|  |

正常情况下，变量一定要设置内容之后才可以正常使用。而事实上，发现默认值没有用处。

|  |
| --- |
| public class TestDemo {  public static void main(String[] args) {  int x ;  x = 10;  System.out.println(x); // 10  }  } |

这个代码在JDK 1.5之后才可以使用，而JDK 1.5之前此代码是错误的。

范例：绝对正确的

|  |
| --- |
| public class TestDemo {  public static void main(String[] args) {  int x = 10;  System.out.println(x); // 10  }  } |

以后一定要养成习惯，方法中定义变量一定要设置默认值。

## 字符型

char主要是保存单个字符，并且单个字符要使用“’”声明。

范例：char类型

|  |
| --- |
| public class TestDemo {  public static void main(String[] args) {  char c = 'A'; // 字符变量  System.out.println(c); // A  }  } |

以后只要是使用了单引号声明的内容都表示单个字符。在Java里面，由于使用的是UNICODE这种十六进制编码，所以char可以保存中文。

范例：保存中文

|  |
| --- |
| public class TestDemo {  public static void main(String[] args) {  char c = '好'; // 字符变量  System.out.println(c); // 好  }  } |

首先明确的是，计算机的世界里面能保存的数据只是010101101，所以来讲如果要保存字符也需要有其对应的编码出现，而Java使用的是UNICODE编码，这个编码的特点：包含了ASC II码。那么在这样的情况下，char可以和int互相转换，而char转换为int之后就是具体的编码数据。

范例：观察转换

|  |
| --- |
| public class TestDemo {  public static void main(String[] args) {  char c = 'A'; // 字符变量  int num = c;  System.out.println(num); // 65  }  } |

常用字符范围：

* 大写字母：’A’（65）~ ‘Z’（90）；
* 小写字母：’a’（97）~ ‘z’（122）；
* 数字字母：’0’（48） ~ ‘9’（57）。

可以发现大写字母和小写字母的编码之间差了32个内容。

范例：大写与小写字母的转换

|  |
| --- |
| public class TestDemo {  public static void main(String[] args) {  char c = 'A'; // 字符变量  int num = c;  num = num + 32; // 大写字母和小写字母之间差了32  c = (char) num;  System.out.println(c); // a  }  } |

综合所述，char最大的优势只在一点上：可以保存中文。而byte是保存字节的。

## 布尔型

Boolean表示的是布尔型数据类型，布尔是一个数学家的名字，他发明了布尔类型。所谓布尔类型指的就是两个内容的概念：true、false。

一般情况下boolean类型往往都要结合判断语句完成处理。

范例：使用布尔型

|  |
| --- |
| public class TestDemo {  public static void main(String[] args) {  boolean flag = true;  if (flag) {  System.out.println("Hello World!");  }  }  } |

很多的编程语言实际上本身并没有直接提供布尔型数据，例如：C语言中，将0表示为false，而非0表示为true，但是Java并没有0、1描述，只有true、false。

但是可以借助于此概念在数据库设计上的时候使用。做一个超级管理员的标志位：1（是超级管理员）、0（普通管理员）。

## 字符串：String

利用char只能够保存单个字符，这在很大程度上是有限制的，所以如果要想表示多个字符，就可以利用String完成。但是String本身是一个类，但是这个类的使用较为特殊，可以像基本数据类型那样完成。在程序中使用双引号声明的数据就是字符串。

范例：定义字符串

|  |
| --- |
| public class TestDemo {  public static void main(String[] args) {  String str = "Hello World";  str = str + "Hello balabalabala";  System.out.println(str);  }  } |

现在发现在String进行连接的操作过程之中使用了“+”进行连接操作，可是“+”也属于数学操作。

范例：观察“+”操作

|  |
| --- |
| public class TestDemo {  public static void main(String[] args) {  int x = 10;  double y = 20.3;  String str1 = "计算结果：" + y + x;  String str2 = "计算结果：" + (y + x);  System.out.println(str1); // 计算结果：20.310  System.out.println(str2); // 计算结果：30.3  }  } |

另外一点，在程序中还存在有转义字符，包括：“\n”、“\\”、“\t”等

# 总结

1. 数据类型划分以及默认值；
2. 表示年龄使用int、表示成绩使用double、表示姓名使用String、表示逻辑使用boolean。