我们在先讨论在string和stringbuffer的相互转化

//String -> StringBuffer

13 //创建一个String对象

14 String str = "Hi Java!";

15 System.out.println(str);

16

17 //方式一：构造方法

18 StringBuffer buffer = new StringBuffer(str);

19 System.out.println(buffer);

20

21 //方式二：通过append方法

22 StringBuffer buffer2 = new StringBuffer();

23 buffer2.append(str);

24 System.out.println(buffer2);

//StringBuffer -> String

27 //创建一个StringBuffer对象

28 StringBuffer buffer3 = new StringBuffer();

29 buffer3.append("Happy birthday Java!");

30 System.out.println(buffer3);

31

32 //方式一：通过构造方法

33 String str2 = new String(buffer3);

34 System.out.println(str2);

35

36 //方式二：通过toString方法

37 String str3 = buffer3.toString();

38 System.out.println(str3);

注意构造方法。

题目：

给一个整数，然后判断这个整数是否为回文。

例如 12321，为回文。1221也是回文。

方法一：

我们拿到题目的时候最容易想到的方法就是将整数转化成为String类型，然后倒置，然后跟之前的整数相比，如果相等，则为回文。不相等则不是回文。

方法牵制出来的问题是：首先String没有reverse方法。需要把String类型再次进行转化成为Stringbuffer类型然后调用reverse方法进行转化。那么如何将int类型的数字转化成整数然后再转化成为stringbuffer类型呢？

String s = Integer.toString(int x);

StringBuffer sb = new StringBuffer(s);

然后需要做的是把倒置完的stringbuffer转化成为int

String ss = Stringbuffer.toString();//将stringbuffer转化成为string

Int res= Integer.valueOf(ss);//将string转化成为int

最后的问题是正负号的问题，那么如果负数怎么办呢？首先咱们都按正数处理，也就是说给数字做绝对值，然后处理完之后再管正负号的问题，数字倒置正负号肯定是不变的，最后再加上正负号就可以了。

方法代码如下：

public static int reverseInt\_2(int x){   
 int n = Math.*abs*(x);  
 String s = Integer.*toString*(n);  
 StringBuffer stringBuffer=new StringBuffer(s);  
 int res=Integer.*valueOf*

(stringBuffer.reverse().toString());  
 if(x>0)  
 return res;  
 else if (x==0)  
 return 0;  
 else  
 return 0-res;  
}

方法二：

方法一会占用出额外空间来处理字符串倒置，额外空间占用还很大。那么怎么样占用空间特别小就将整数倒置呢？

我们假设传入的数字为 123 ， 我们要把每一位的数字都“取”出来，123 %10 =3，

123 /10 =12；我们用10的求余就可以取出当前数字的最后一位，剩下的数字就是10的取整。用循环来进行各个位的数字取出工作。最后一位1/10 = 0；用 n / 10 = 0 的时候就是结尾了。取出各个位置的数字然后怎么变成一个整体的整数呢？3\*100+2\*10+1\*1=321.我们需要用数字乘以各个位的进制然后汇总就可以了。如何知道进位为多少呢？需要知道数字的长度。数字长度需要将数字转化成为string，然后用length（）方法知道数字的长度。

Int length = Integer.tostring(n).length();

各个位置的进位就等于，（int）Math.pow（10，length-1）。这里需要注意的是Math.pow方法的参数以及返回值都是double类型，要么就把参数改成double，要么就把结果前转化为int，如上我的代码。3 🡪 10^(3-1) = 3\*100, 2🡪10^(2-1)=2\*10=20 … 注意的是每次循环计算好数字然后需要更改进位也就是每次循环length都要减去1。正负数字处理如方法一，先都当整数处理，然后分正负。

代码如下：

public static int reverseInt\_1(int x){  
 int n=Math.*abs*(x);  
 int carry=(""+n).length();  
 int sum=0;  
 while(n!=0){  
 sum=sum+(int)Math.*pow*(10,carry-1)\*(n%10);  
 n=n/10;  
 carry-=1;  
 }  
 if(x>0){  
 return sum;  
 }  
 else if(x==0){  
 return 0;  
 }  
 else {  
 return 0-sum;  
 }  
}