# 按类型

## 字符串

### 5.替换空格

P51·实现一个函数，把字符串中的每个空格都替换成“%20”。例如，输入“We are happy.”，则输出“We%20are%20happy.”。

#### Java StringBuffer和StringBuilder类

当对字符串进行修改的时候，需要使用StringBuffer和StringBuilder类。

与String类不同的是，StringBuffer和StringBuilder类的对象能够被多次的修改，并且不产生新的未使用对象。StringBuilder类在Java 5中被提出，它和StringBuffer之间的最大不同在于StringBuilder的方法不是线程安全的(不能同步访问)。由于StringBuilder相较于StringBuffer有速度优势，所以多数情况下建议使用StringBuilder类。然而在应用程序要求线程安全的情况下，则必须使用StringBuffer类。

public class Test{

public static void main(String args[]){

StringBuffer sBuffer = new StringBuffer("sjtu");

sBuffer.append("www");

sBuffer.append(".sjtu");

sBuffer.append(".edu.cn");

System.out.println(sBuffer);

}

}



**public class** ReplaceSpaces\_5  
{  
 **public static** String replaceSpace(StringBuffer stringBuffer)  
 {  
 **int** P1= stringBuffer.length() - 1;  
 *//System.out.println(P1);* **for** (**int** i = 0; i <= P1; i++)  
 {  
 **if** (stringBuffer.charAt(i) == **' '**)  
 stringBuffer.append(**" "**); *// 两个空格* }  
  
 **int** P2 = stringBuffer.length() - 1;  
 *//System.out.println(P2);* **while** (P1 >= 0 && P2 > P1)  
 {  
 **char** c = stringBuffer.charAt(P1--);  
 **if** (c == **' '**)  
 {  
 stringBuffer.setCharAt(P2--, **'0'**);  
 stringBuffer.setCharAt(P2--, **'2'**);  
 stringBuffer.setCharAt(P2--, **'%'**);  
 }  
 **else** stringBuffer.setCharAt(P2--, c);  
 }  
 **return** stringBuffer.toString();  
 }  
  
 **public static void** main(String[] args)  
 {  
 StringBuffer preString = **new** StringBuffer(**"We are happy "**);  
 String newString = *replaceSpace*(preString);  
 System.***out***.println(preString);  
 System.***out***.println(newString);  
 }  
}  
*/\*  
We%20are%20happy%20  
We%20are%20happy%20  
 \*/*