Java3:流程控制和数组

授课教师：邱元杰　电子邮箱：yuanjiq@126.com，微信电话: 13679081552

第3章 流程控制、数组

分支语句

循环语句

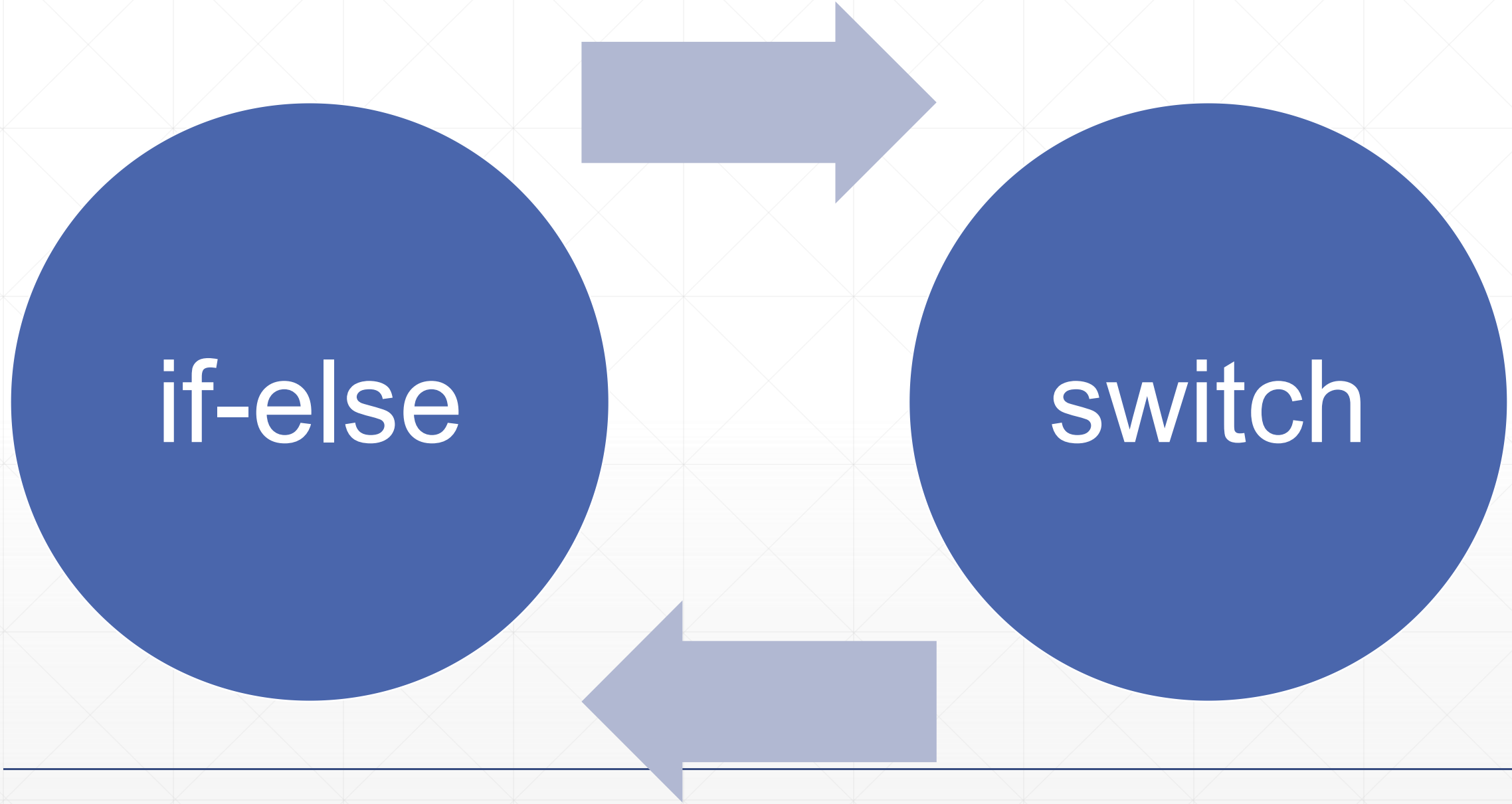
数组

流程控制

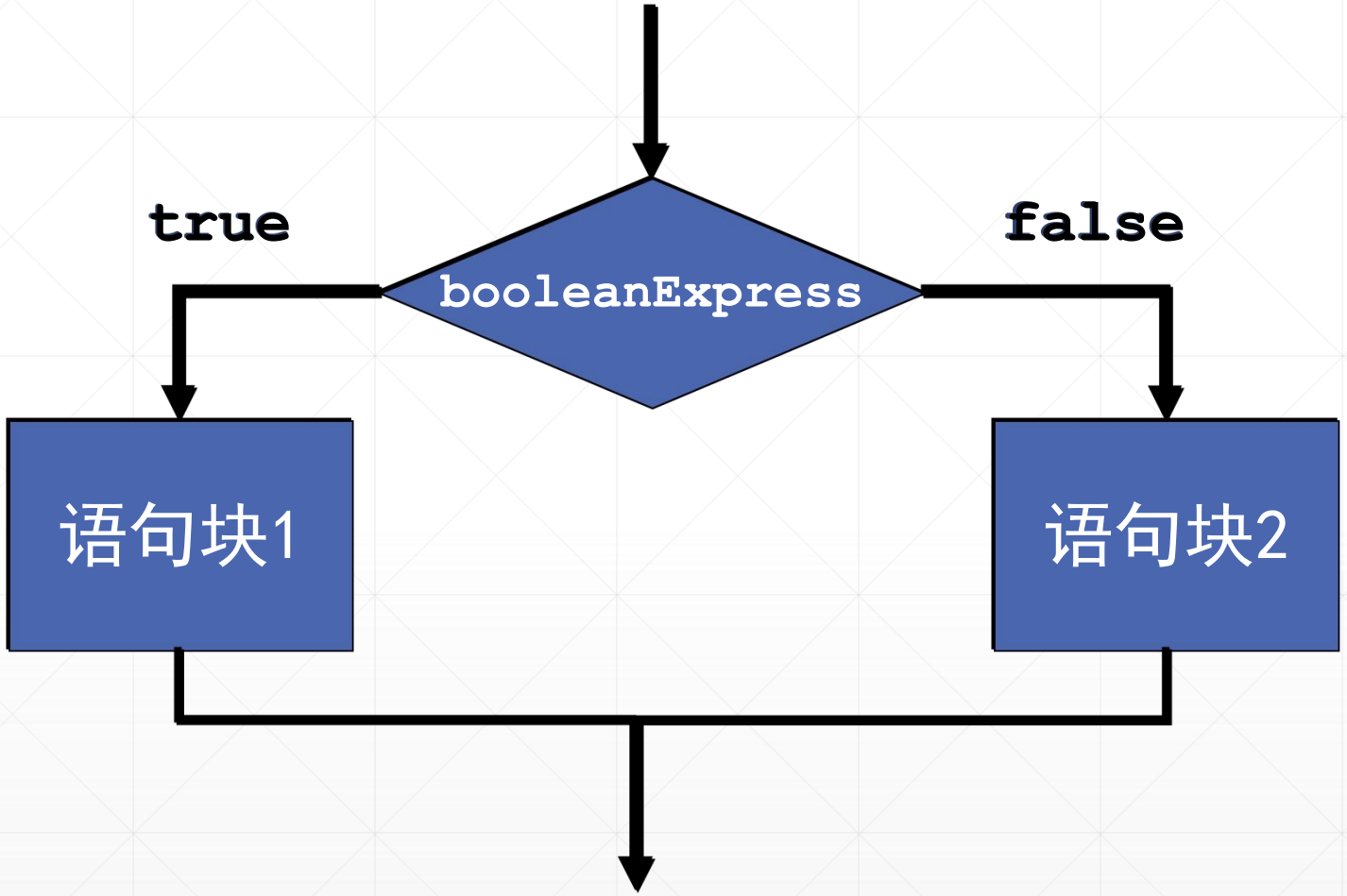
- Java语言中的流程控制语句提供了控制程序执行顺序的手段。流程控制是程序代码的重要部分。
- 流程控制语句分为：分支语句、循环语句、异常处理语句和跳转语句。

分支与跳转语句	if-else	switch	break	return
循环语句	while	do-while	for	continue
异常处理语句	try	catch	finally	throw

分支语句



分支语句：if-else



CompareTwoDat.java

ScoreLevel1.java

分支语句：switch

- 多分支语句switch与if-else语句一样，是根据相关表达式的值选择程序流程。它与if语句不同处在于多种情况可供程序流程选择。
- switch所用的表达式为int类型相容的数据表达式，它可以是byte、short、char或者int类型的值，特别要指出的是不能是布尔型的值。

分支语句：switch

- 格式：

```
switch (intexpression){
    case int1:
            statement or block (1)
            break;
    case int2:
            statement or block (2)
            break;
    ...
    default:
            statement or block(d)
}
```

ScoreLevel2.java

ScoreLevel3.java

分支语句：break

- break常用于switch语句的中，用break语句起跳出switch语句的作用。
- break语句不仅能用在switch语句，也可以用在循环语句，都同样起到结束它所在语句块流程。
- 处在break语句之后的语句将会被跳过而不被执行。

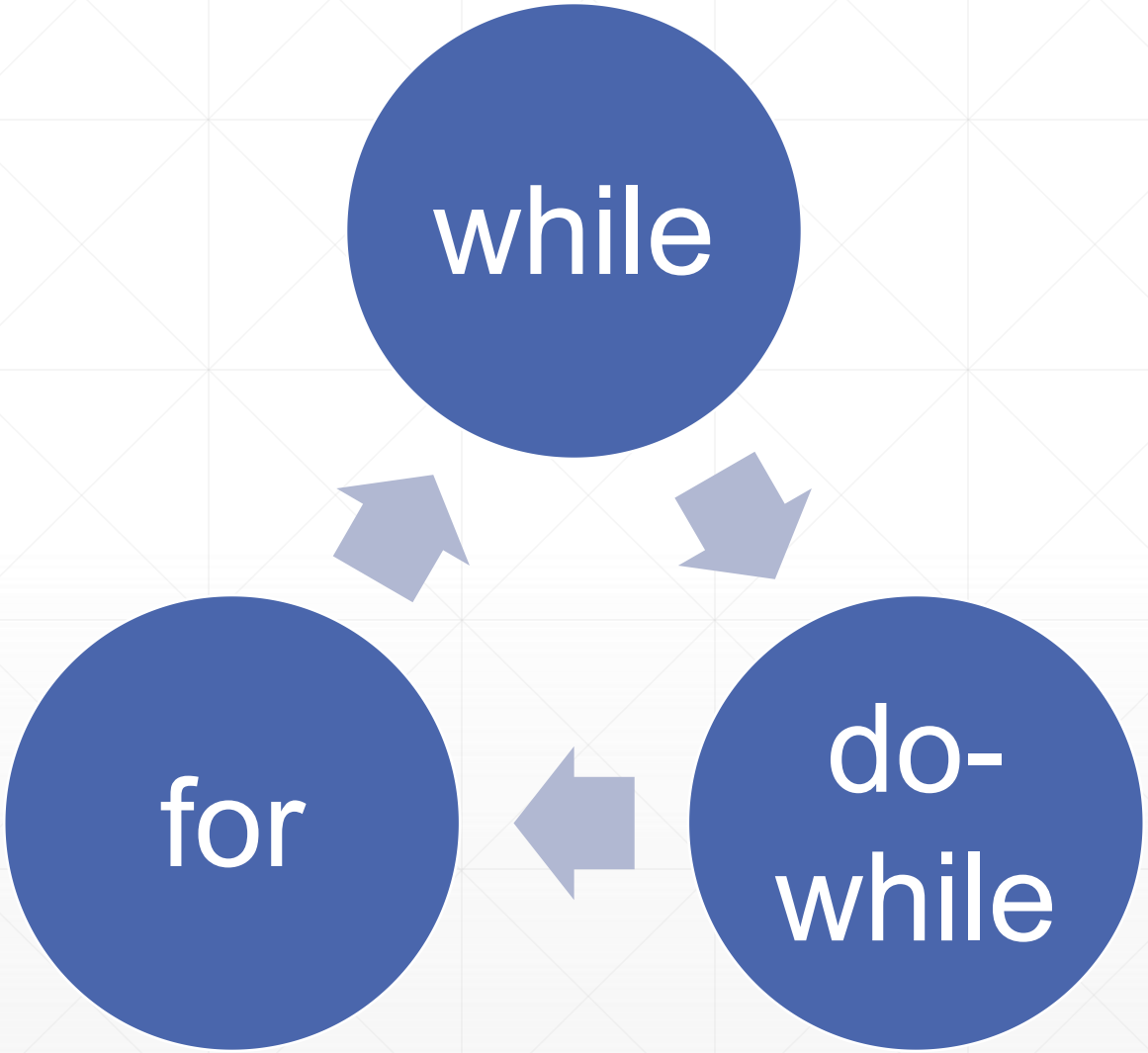
第3章 流程控制、数组

分支语句

循环语句

数组

循环语句



循环语句

- **循环是由四个部分组成，根据不同的循环语句，它们之间执行顺序有所不同，这四个组成部分是：**

初始化(initalization)	为循环设置初始量
判断(condition)	**布尔表达式的值决定循环是否继续**
循环体(body)	循环的代码段(语句)
迭代(iteration)	每循环一次，改变循环控制变量的值

循环语句：while

- while循环语句的格式是：

```
[initalization]
while (expressBool){
    statements;
    [iteration;]
}
```

循环语句：while

- while语句循环执行的顺序是：
  - 执行初始化initalization(如果有)；
  - 计算表达式expressBool的值；
  - 若expressBool值为true，则执行循环体statements；
  - 执行迭代部分iteration(如果有)；
  - 返回到2；
  - 若expressBool值为false，则终止while循环。

循环语句：do-while

- **do-while循环语句的格式是：**

```
[initalization]
do {
    statements;
    [iteration;]
}while (expressBool);
```

循环语句：do-while

- do-while语句循环执行的顺序是：
  - 执行初始化initalization(如果有)；
  - 执行循环体statements；
  - 执行迭代部分iteration(如果有)；
  - 计算表达式expressBool的值；
  - 若expressBool值为true，则返回到2；
  - 若expressBool值为false，则终止do-while循环。

TempConverDoWhile.java

循环语句：for

- for循环语句的格式是：

```
for(initalize; condit; iterat){
        statements;
}
```

循环语句：for

- for语句循环执行的顺序是：
  - 执行初始化initalize；
  - 计算表达式condit的值；
  - 若condit值为true，则执行循环体statements；
  - 执行迭代部分iterat；
  - 返回到2；
  - 若condit值为false，则终止for语句。

循环中的break语句

- break语句用于循环结构中，当程序执行break语句时，程序流程就结束循环
- break语句也可以带语句标记，它的作用是结束该语句标记的语句块。
- break语句使用格式如下：

```
break [outerLabel];
```

BreakLoop.java　BreakLoop2.java
BreakLabel.java　BreakLabel2.java

循环中的continue语句

- continue语句用于循环结构中，当程序执行contiune语句时，程序流程就结束本次循环，充当了循环体的最后一条语句作用。
- continue语句也可以带语句标记，它的作用是结束该语句标记的外层循环的本次循环。
- continue语句使用格式如下：

```
continue  [outerLabel];
```

第3章 流程控制、数组

分支语句

循环语句

数组

数组

- 在Java中，数组是引用类型。数组类型是一种有序数据的集合，数组中在每一维上的元素具有相同的数据类型。
- 数组通过数组名和它的下标对数组元素访问，数组元素的**下标不能越界**。
- 数组是一个对象，数组声明不能创建对象本身，而创建一个引用。数组元素由new语句或数组初始化软件动态分配。

数组：数组声明

- Java的数组声明采用与C语言类似的形式。数组可分为一维数组和多维数组。它们的声明的形式为：

```
type  arrayName[][]...];
```

- 或另一等价形式：

```
type[][]...]  arrayName;
```

```
int   count[];     //一维整型数组count
char  ch[][];     //二维字符型数组ch
float[] fNum;     //一维浮点型数组fNum
```

数组：数组实例化

- 在Java语言中，数组的声明是不能确定数组大小。数组的实例化即存储单元的分配是由new运算符实现。

```
arrayName = new type [arraySize1][[]...];
```

- 数组通过数组名和它的下标对数组元素访问，数组元素的下标不能越界。
- 数组实例化示例：

```
int[] a = new int[3];
```

- 数组a有元素：a[0]、a[1]、a[2]。

数组：数组实例化

- 数组在实例时，同时也有了初始化的值。

  例：int[] a = new int[3];

  数组a的三个元素有值都为0。

- 数组在创建时，也可显式初始化。

  例：int[] a = {1,2,3};

  数组a的三个元素的值分别为1，2，3。

- 数组实例化后就有了确定的元素，每个数组有一个属性length，其值就是这个数组的元素的数量。

  例：a.length的值为3。

ArrayRefer.java

数组：多维数组

- Java编程语言没有提供多维数组。它是通过创建数组的数组(和数组的数组的数组)。
- 数组通过数组名和它的下标对数组元素访问，数组元素的下标不能越界。
- 数组是一个对象，数组声明不能创建对象本身，而创建一个引用。数组元素由new语句或数组初始化软件动态分配。

数组：多维数组实例化

- 虽然在声明数组的格式中，允许方括号在数组名的左边或者右边，但这种方式不适合数组句法的其它部分。
- 必须首先将低位维初始化，再能对它后面的各位依次初始化。
- 利用对每维元素的分步初始化，可以创建非矩形数组的数组。

TwoDArray.java　Matrix.java　ThreeDMatrix.java

字符串

- 字符串是一串字符组成的数据，并用""包括起来。字符串常量是String类型的对象。
- 类String是Java语言的基础数据类型，它具有一定的特殊性。
- Java编译器在对字符串数据与其它类型数据使用“+”运算符连接操作编译时，总是首先将其它类型数据转换为字符串类型，然后再进行字符串连接。
- 例：　"Age: "+18 ==> "Age: 18"

StringConnect.java

字符串相关方法

- char charAt(int where)
- void getChars(int sourceStart, int sourceEnd, char target[ ], int targetStart)
- byte[ ] getBytes( )
- char[ ] toCharArray( )
- boolean equals(Object str)
- boolean equalsIgnoreCase(String str)
- boolean startsWith(String str)
- boolean endsWith(String str)
- int indexOf(int ch)
- int lastindexOf(int ch)
- String substring(int startIndex)
- String substring(int startIndex, int endIndex)

getCharsDemo.java
equalsDemo.java
EqualsNotEqualTo.java
indexOfDemo.java
StringReplace.java

字符串

- 字符串常量对方法的访问示例：

  "Hello".toUpperCase() ==> "HELLO"

  "Hello".length() ==> 5

  字符串常量不是char类型一维数组，不存在‘\0’结束符。 字符串和char数组可以通过相应方法转换：

  char[] data = "Car".toCharArray();

  则：data = {'C','a','r'}

  copyValueOf(data) ==> "Car"

字符串与基本数据的转化

- String2Integer
  - public static int parseInt(String s)
- Integer2String
  - public static String valueOf(int n)

StringBuffer

- StringBuffer是提供了大量的字符串功能的字符串类的对等类
- 字符串（String）表示了定长，不可变的字符序列。
- StringBuffer表示了可变长的和可写的字符序列。
- StringBuffer可有插入其中或追加其后的字符或子字符串。StringBuffer可以针对这些添加自动地增加空间，同时它通常还有比实际需要更多的预留字符，从而允许增加空间
- length( )和capacity( )
- append( )和insert( )

StringBufferDemo.java

appendDemo.java

思考

- 在Java语言中，流程控制分为哪些?
- 分支语句if-else和switch在断定条件上各为何种数据类型?
- 循环语句中由哪几个部分组成，各有何作用?
- 比较break和continue语句的区别。
- 数组的声明形式有哪些，其初始化是什么含义?
- 数组的length属性是指什么，在多维数组中如何来获取length的值?
- 创建一个2×3的二维int数组，并用1到6的整数作为元素的初始值。

作业

- 编写一个类TestArray，它只有一个main()方法，在该方法中，创建一个int类型的一维数组sim，实现结数组sim的元素从小到大排序，并输出排序后数组的值。

补充：如何输入数据

- **Scanner 类**

```
Scanner sc = new Scanner(System.in);
int i = sc.nextInt();
```

作业(2)

- toUpperCase()和toLowerCase()可以把一个字符串中的字符转变为大写或者小写。请编写一个程序，实现两个方法完成相同的功能，但是不能使用上述两个方法。