

数组的异常:

- 1) 数组下标越界, `java.lang.ArrayIndexOutOfBoundsException`
- 2) 空指针访问异常, `NullPointerException`

数组中的常见算法:

- 1) 数组元素的最大值, 最小值, 平均值, 总和
- 2) 数组中元素的复制和反转
- 3) 数组元素的排序

数组的排序:

- 1) 插入排序: 直接插入排序, 折半插入排序, shell排序
- 2) 交换排序: 冒泡排序, 快速排序
- 3) 选择排序: 简单选择排序, 堆排序
- 4) 归并排序:
- 5) 基数排序

冒泡排序: 嵌套循环, 进行`a.length-1`次

以后每次都从头开始进行比较相邻的俩个元素

```
for(int i=0;i<arr2.length-1;i++){
    for(int j=0;j<arr2.length-i-1;j++){
        if (arr2[j]>arr2[j+1]) {
            int temp=arr2[j+1];
            arr2[j+1]=arr2[j];
            arr2[j]=temp;
        }
    }
}
```

直接选择排序

//直接选择排序

```
for(int i=0;i<a.length-1;i++){
    for(int j=i;j<a.length;j++){
        if (a[i]>a[j]) {
            int temp=a[i];
            a[i]=a[j];
            a[j]=temp;
        }
    }
}
```

```
        }  
    }  
}
```

优化:

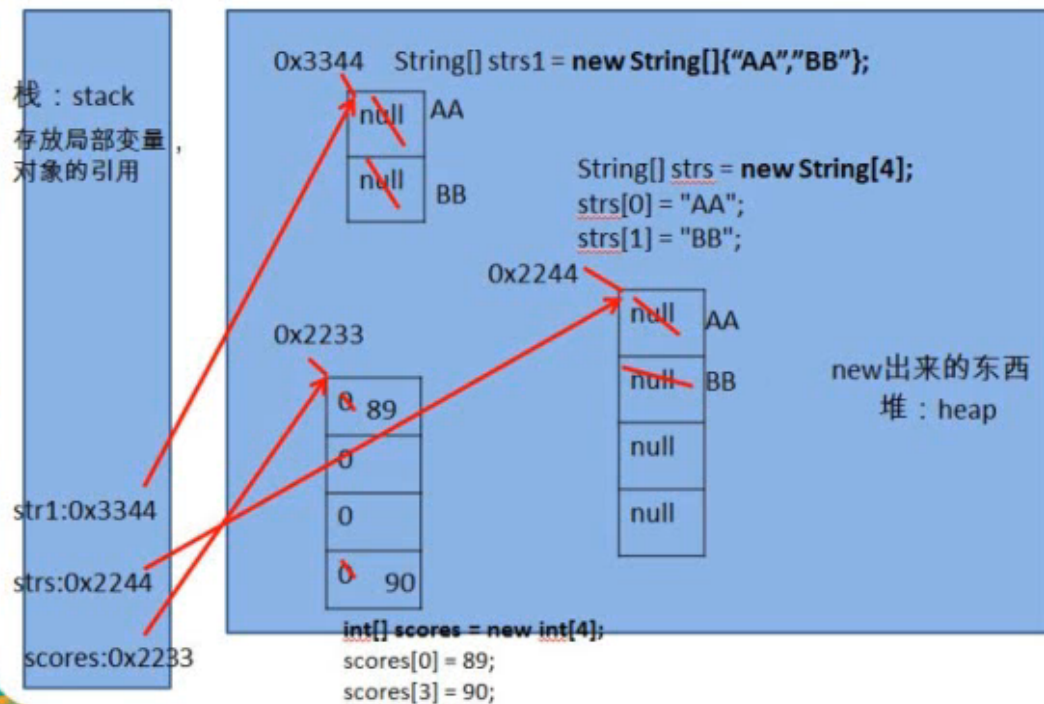
//直接选择排序

```
for(int i=0;i<a.length-1;i++) {  
    int t=i;  
    for(int j=i;j<a.length;j++) {  
        if (a[t]>a[j]) {  
            t=j;  
        }  
    }  
    if (t!=i) {  
        int temp=a[t];  
        a[t]=a[i];  
        a[i]=temp;  
    }  
}
```

java中的排序方法:

Arrays.sort(a);默认使用快速排序

对于数组来讲,



二维数组:

1) 声明并初始化

一维:

```
int [] i = new int[19];
```

```
i[0] = 32;
```

```
int [] j = new int[] {21, 23, 3};
```

二维:

```
string [][] str = new string[4][3]; // 长度为4, 4行3列
```

```
string [][] str = new string[4][]; // 指明长度为四, 列数不一定
```

```
str[0] = new string[3];
```

```
int a[][] = new int[] { {1, 2, 3}, {4, 3}, {3, 8} };
```

2) 如何引用二维数组的元素

```
a[2][2] = 34;
```

3) 二维数组的长度

```
a.length;
```

a[1].length;//每一行又是一个数组，依然有长度

#### 4) 遍历二维数组

```
for (int i=0; i<a.length; i++) //控制行数
{
    for(int j=0;j<a[i].length;j++)
    {
        sysoutem.out.println(a[i][j]);
    }
}
```

关于数组元素的默认初始化值：

- 1) byte, short, int, long默认初始化为0;
- 2) float, double默认初始化为0.0
- 3) char 默认空格
- 4) boolean默认false
- 5) 引用类型默认null