

# 数 组

## 为什么需要数组

问题

- Java考试结束后，老师给李四分配了一项任务，让他计算全班（30人）的平均分

30个变量太繁琐

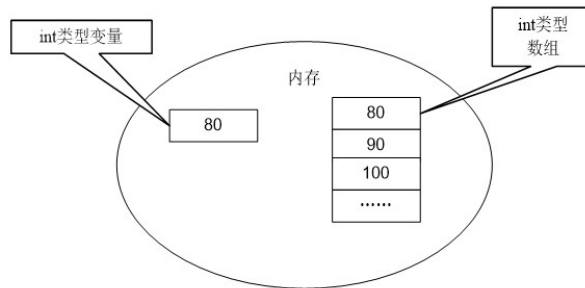
不利于数据处理

```
int stu1 = 95;  
int stu2 = 89;  
int stu3 = 79;  
int stu4 = 64;  
int stu5 = 76;  
int stu6 = 88;  
.....  
avg = (stu1+stu2+stu3+stu4+stu5...+stu30)/30;
```

30个变量

# 数组

## ■ 数组是一个变量，存储相同数据类型的一组数据



声明一个变量就是在内存空间划出一块合适的空间  
声明一个数组就是在内存空间划出一串连续的空间



## 一维数组

### ■ 一维数组的声明

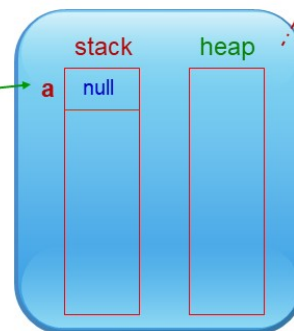
数组类型 数组名 []

数组类型 [] 数组名

或

### ■ 示例

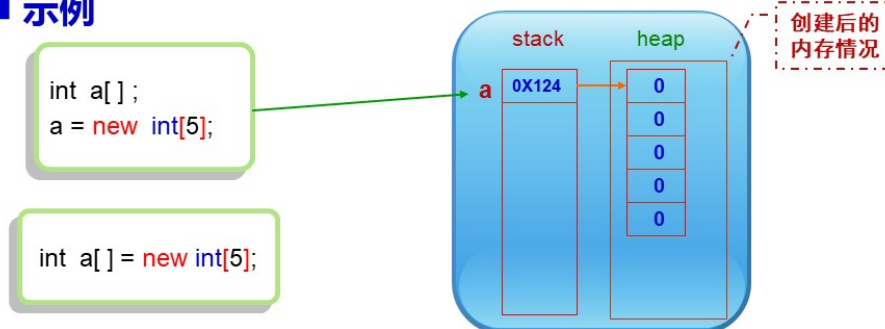
```
int a[];
int[] b;
double d[];
char[] ch;
Student stu[];
```



## ■ 一维数组的创建

数组名 = new 数据元素类型[元素个数]

### ■ 示例



## ■ 一维数组的初始化

- ✓ 动态初始化
- ✓ 静态初始化

### • 动态初始化

- ✓ 数组的声明创建与数组的赋值操作分开进行

- Java中，数组元素的下标从0开始
- 元素的下标可以是整型常量或整型表达式

数组名[元素下标] = 元素值;



```
int a[];  
a = new int[5];  
a[0] = 1;  
a[1] = 2;  
a[2] = 3;  
a[3] = 4;  
a[4] = 5;
```

## ■ 静态初始化

- ✓ 在创建数组的同时就为数组元素分配空间并赋值

数组类型 数组名[] = {元素1, 元素2, ...}

或

数组类型 数组名[] = new 数据类型[] {元素1, 元素2, ...}



int a[] = {1,2,3,4,5};

或

int a[] = new int[] {1,2,3,4,5};

## — 数组的常见错误 —

### ■ 数组中的常见错误

- ✓ 数组越界异常 (ArrayIndexOutOfBoundsException)

#### • 数组越界异常

```
class Array_exception2 {
    public static void main(String args[]){
        int a[]=new int[3];
        a[0]=1;
        a[1]=2;
        a[2]=3;
        a[3]=4;
        System.out.println(a[3]);
    }
}
```

→ Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 3  
at p05.Array\_exception2.main(Array\_exception2.java:10)



问题

## ■ 计算全班学员的平均分

```
public static void main(String[] args) {  
    int[] scores = new int[5]; //成绩数组  
    int sum = 0; //成绩总和  
    Scanner input = new Scanner(System.in);  
    System.out.println("请输入5位学员的成绩: ");  
    for(int i = 0; i < scores.length; i++){  
        scores[i] = input.nextInt();  
        sum = sum + scores[i]; //成绩累加  
    }  
    System.out.println("平均分是: " + (double)sum/scores.length);  
}
```

## 常见错误3-1

 代码阅读

```
public class ErrorDemo1 {  
    public static void main(String[] args){  
        int[] score = new int[ ];  
        score[0] = 89;  
        score[1] = 63;  
        System.out.println(score[0]);  
    }  
}
```

编译出错，没有写明数组的大小

```
public class ErrorDemo2 {
    public static void main(String[] args) {
        int[] scores = new int[2];
        scores[0] = 90;
        scores[1] = 85;
        scores[2] = 65;
        System.out.println(scores[2]);
    }
}
```

编译出错，数组越界

Console

```
<terminated> ErrorDemo2 [Java Application] D:\ProgramFile\MyEclipse10\Common\binary\com.sun.java.jdk.win32.x86_1.6.0
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 2
at ErrorDemo2.main(ErrorDemo2.java:9)
```

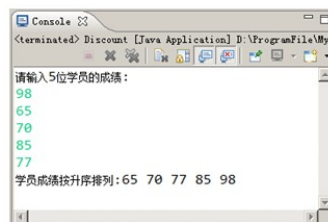
```
public static void main(String[] args){
    int[] score = new int[5];
    score = {60, 80, 90, 70, 85};

    int[] score2;
    score2 = {60, 80, 90, 70, 85};
}
```

编译出错，创建数组并赋值的方式必须在一条语句中完成

## 问题

- 循环录入5位学员成绩，进行升序排列后输出结果



## 分析

- 使用java.util.Arrays类
  - ◆ java.util包提供了许多工具类
  - ◆ Arrays类提供操作数组的方法
    - 如排序、查询
  - ◆ Arrays类的sort()方法：对数组进行升序排列

```
Arrays.sort(数组名);
```

## Arrays类

方法名称	说明
boolean equals(array1, array2)	比较array1和array2两个数组是否相等
sort(array)	对数组array的元素进行升序排列
String toString(array)	将一个数组array转换成一个字符串
void fill(array, val)	把数组array所有元素都赋值为val
copyOf(array, length)	把数组array复制成一个长度为length的新数组，返回类型与复制的数组一致
int binarySearch(array, val)	查询元素值val在数组array中的下标（要求数组中元素已经按升序排列）

## 示例

```
import java.util.*; //导入包
.....
int[] scores = new int[5]; //成绩数组
Scanner input = new Scanner(System.in);
System.out.println("请输入5位学员的成绩: ");
for(int i = 0; i < scores.length; i++){
    scores[i] = input.nextInt();
}
Arrays.sort(scores);
System.out.print("学员成绩按升序排列: ");
for(int i = 0; i < scores.length; i++){
    System.out.print(scores[i] + " ");
}
```

循环录入学生成绩并存储在数组中

数组中的元素被重新排列

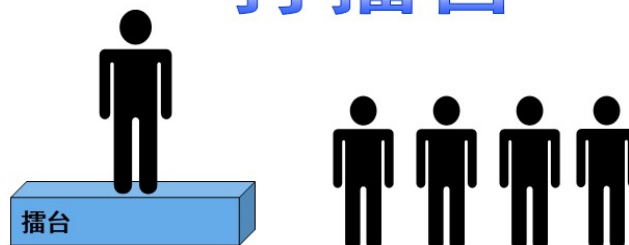
循环输出数组中的信息

## 求最大值

## 问题

- 从键盘输入本次Java考试五位学生的成绩，求考试成绩最高分

## 打擂台





### 示例

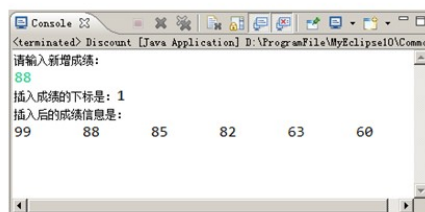
```
//计算成绩最大
max = scores[0];
for(int i = 1; i < scores.length; i++){
    if(scores[i] > max){
        max = scores[i];
    }
}
System.out.println("考试成绩最高分为: " + max);
```

使用max存储擂主初始值：第一个元素为擂主

循环打擂

### 问题

- 有一组学员的成绩{99, 85, 82, 63, 60}，将它们按升序排列。要增加一个学员的成绩，将它插入成绩序列，并保持升序



### 分析

- 将成绩序列保存在长度为6的数组中
- 通过比较找到插入位置
- 将该位置后的元素后移一个位置
- 将增加的学员成绩插入到该位置



```
int[] list = new int[6]; // 长度为6的数组
int index = list.length; // 保存新增成绩插入位置
//.....省略为数组和变量赋值
```

```
for(int i = 0; i < list.length; i++){
```

```
    if(num > list[i]){
```

```
        index = i;
```

```
        break;
```

```
    }
```

```
}
```

找到新元素的插入位置

```
for(int j = list.length-1; j > index; j--){
```

```
    list[j] = list[j-1];
```

```
}
```

元素后移

```
list[index] = num; // 插入数据
```

新元素放在index的位置

```
.....
```

## ■ 二维数组的声明

数组类型 数组名 [ ] [ ]

数组类型 [ ] [ ] 数组名

或

数组类型 [ ] 数组名 [ ]



```
int a[ ] [ ];
```


```
int [ ] [ ] b;
```

```
int [ ] c[ ];
```

## ■ 二维数组的创建

```
数组名 = new 数据类型[行数][列数];
```

```
数组名 = new 数据类型[行数][ ];
```



```
a = new int [3][4];  
a = new int [3][ ];  
a = new int [ ][4]; //非法
```

必须声明行的个数


## ■ 二维数组的初始化

### ➤ 静态初始化

```
数组类型 数组名[ ][ ] = { {元素11, 元素12, ...}, {元素21, 元素22, ...} }
```

或

```
数组类型 数组名[ ][ ] = new 数据类型[ ][ ] { {元素11, 元素12, ...}, {元素21, ...} }
```



```
int a[ ][ ] = { {1,2,3}, {4,5}, {7} };  
int a[ ][ ] = new int[ ][ ] { {1,2,3}, {4,5}, {7} };
```

## ■ 二维数组的初始化

### ➤ 动态初始化

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

```
a[0] = new int[2];
```

```
a[1] = new int[3];
```

```
a[2] = new int[4];
```

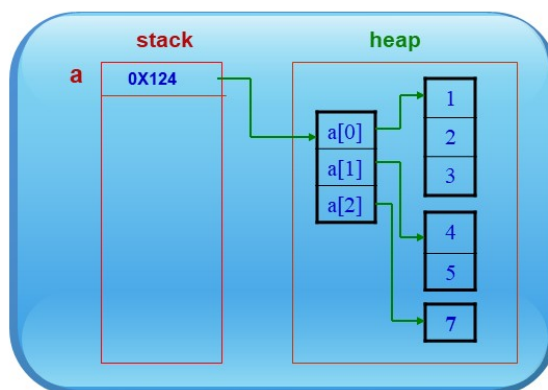
```
a[0][0] = 1;
```

```
a[0][1] = 2;
```

```
...
```

## 二维数组在内存中的表示

```
int a[ ][ ] = { {1,2,3}, {4,5}, {7} };
```





- 1、有5个班各5名学生某门课程的成绩，计算5个班各自的总成绩
- 2、现在有一个数组：11,31,23,54,0,0,77,90,0,5,42,71,63,79,60,53,  
要求将以上数组中值为0的项去掉，将不为0的值存入一个新的数组。