数组

回顾

1 方法：完成特定功能的一段代码。

作用：提高代码的可读性，可维护性，可重用性。

语法：

```
访问修饰符    其他修饰符    返回类型   方法名(参数列表){
        方法体
    }
```

分为：根据有没有参数：无参方法和有参方法

根据有没有返回值：无返回值和有返回值

```
public static  void method1(){

}

public static void  method2(int x){

}

public static  int method(){

    return 10;

}
```

方法的返回值：

(1) 没有返回值 void ，在方法体中可以写return（return; 结束方法） ,也可以不写。

(2) 有返回值， return 变量或常量、表达式;

(3) 有返回值，并且方法中存在多个分支,每个分支都要写return 变量或常量、表达式。

方法调用：

无参方法调用:方法名();

有参方法调用:方法名(参数);

参数要求：实参和形参 个数、类型（兼容，可以自动类型转换）、顺序都要一致。

如果有返回值，创建一个变量接收返回值。

2 方法重载：同一个类中，方法名相同，参数不同

（1）参数列表不同： 个数不同，类型不同，顺序不同

（2）和修饰符、返回值没有关系

3 递归算法： 在一个方法内部调用自己本身。

(1) 递归必须要有结束条件 (没有结束条件:StackOverFlowError)

(2) 递归有层层递推的规律

3.1 阶乘

```
public static int jieCheng(int n){
    if(n==1){
```

```
                return 1;
            }
            return jieCheng(n-1)*n;

        }
    3.2 求和
        public static int sum(int n){
            if(n==1){
                return 1;
            }
            return jieCheng(n-1)+n;

        }
    3.3 求第30个斐波那契数
        public static int feiBo(int n){
            if(n==1||n==2){
                return 1;
            }
            return feiBo(n-1)+feiBo(n-2);
        }
```

今天任务

```
1.开发工具eclipse的介绍
    第一阶段：eclipse
    第二阶段：eclipse 、idea
    第三、四阶段：idea
    其他的开发工具：JBuilder、MyEclipse、STS
2.数组的声明和初始化
3.数组的元素访问以及遍历
4.数组的应用(冒泡排序、选择排序、顺序查找和二分法查找)
5.工具类的使用
6.可变参数
```

教学目标

```
1.了解eclipse工具的基本使用
2.掌握数组的声明和初始化
3.掌握数组的使用
4.掌握数组的排序和查找
5.工具类的使用
6.可变参数
```

第一节 开发工具的介绍

1.1 工作空间的概念

工作空间（workspace）：是用户在同一个工程中（或者是一个事务）工作环境的集合,简单来说，就是项目存放的位置。
工作空间有明显的层次结构。项目在最顶级，项目里头可以有文件和文件夹。

File:文件

Edit:编辑

Source:源码

Refactor:重构

Navigate:导航

Search:搜索

Project:项目

Run:运行

Window:窗口

Help: 帮助

Debug:调试

Package Explorer: 包资源管理器

Preferences: 首选项（配置）

Eclipse的配置 切换为java透视图: window--->Perspective--->open Perspective--->other--->java

重置透视图: window--->Perspective--->reset Perspective

Package Explorer： 包资源管理器,用来管理项目的包

入门演示如何创建项目：

file--->New--->Java Project

项目结构：

jre :运行环境

src: 存放源码

1 创建包， 公司域名的倒置 www.baidu.com com.baidu.helloworld com.qf.helloworld 小写

2 创建类，类名HelloWorld

3 编写代码

4 运行 点击run按钮

配置Eclipse

1 修改字体大小 window-->preferences--->输入font---->colors and fonts--->basic--->text font

2 修改编码 utf-8 window-->preferences--->输入 encoding--> workspace--> other utf-8

3 修改代码提示 window-->preferences--->输入 assist --->java--editor> content assist

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ(@,

1.2 常用快捷键

```
常见模板代码（代码片段）
    main--->主函数
    syso--->输出语句
自定模板
    window-->preferences-->java--->editor--->templates
    添加模板
    psvm
    public static void main(String[] args){
        ${cursor}
    }
```

快捷键
ctrl+D：删除一行
ctrl+shift+f：格式化代码
ctrl+S：保存文件
ctrl+shift+o：导包
ctrl+alt+方向键上下键：向上或向下复制
alt+方向键的上下键：向上或向下移动代码
alt+shift+j：文档注释
ctrl+/：单行注释
ctrl+/：取消单行注释
ctrl+shift+/：多行注释
ctrl+shift+\:取消多行注释
shift+enter：强制向下换行
ctrl+shift+enter 强制向上换行
ctrl+f11:运行
alt+shift+a:矩形选择

第二节 数组

回顾数据类型：（1）基本类型 byte short int long float double char boolean

（2）引用类型 类 数组 接口 枚举 注解

回顾：变量：内存中的一块存储空间，存储的就是常量。

特点：一个变量只能存储一个数据，不能存储多个。

需求：统计把一个班级的所有学生30个人java成绩？

解决1:使用变量存储，需要声明30个变量：int java1=90; int java2=80;

解决2:使用数组

2.1 什么是数组

数据的组合,不止一个数据，包含多个数据。数组就是一个变量。

2.2 数组声明和初始化

声明:

方式一：数据类型[] 数组名
方式二：数据类型 数组名[]
推荐使用方式一，C#等越来越多的语言已经不支持方式二定义数组

初始化:

Java中的数组必须先初始化，然后才可以使用，所谓初始化，就是为数组中的数组元素分配内存空间，并为每个数组元素赋初始值。

2.2.1 静态初始化

初始化时由程序员指定每个数组元素的初始值，由系统计算数组长度。

语法：数组元素类型[] 数组名 = new 数组元素类型[]{元素0，元素1,....};

可简写为：数组元素类型[] 数组名 = {元素0，元素1,....};

说明：任何一个变量都得有自己的数据类型，这里的arr表示数组变量名称，int表示数组中元素的类型，int[]才是数组类型
注意：简写静态初始化只能一条语句完成，不能分割成两条语句。

代码实现：

```
/*
    静态初始化:由我们指定元素的初始值，由系统计算长度或者元素的个数
*/
int[] arr = new int[]{1,56,76,87};
int[] arr1 = {1,56,76,87};

String[] arr2 = new String[]{"434","gfg","gjf545"};
String[] arr3 = {"434","gfg","gjf545"};

//Scanner[] arr4 =

char[] arr5 = new char[]{'2','g','*'};
char[] arr5 ={'2','g','*'};
```

2.2.2 动态初始化

初始化时程序员只指定数组长度，由系统为数组元素分配初始值

语法：元素类型[] 数组名 = new 元素类型[元素个数或者数组长度]；

系统对初始值分配规则如下：
a.整数型为0
b.浮点型为0.0
c.字符型为‘\u0000’(不同的系统平台显示结果不同)
d.布尔类型为false
e.引用类型为null

代码实现：

```
        /*
        动态初始化：初始化时由程序员指定数组的长度，系统负责分配元素的初始值
        */
        int[] array1 = new int[5];//0

        String[] array2 = new String[3];//null

        char[] array3 = new char[10];//\u0000
```

注意：
    a.在初始化数组时，不要静态初始化和动态初始化同时使用，也就是说，不要在进行数组初始化时，既指定数组的长度，也为每个数组元素分配初始值。

2.3 数组的使用

2.3.1 通过下标访问指定元素

注意：1Java语言的数组索引(下标、角标)是从0开始的,数组的下标的最大值长度-1
2不要超出索引的范围，如果超出范围出现异常 java.lang.ArrayIndexOutOfBoundsException

代码实现：

```
//使用静态初始化的方式定义一个数组
//数组中可以存放重复数据
int[] arr1 = new int[]{2,65,76,83,32,5,5};

//1.访问数组中的元素
//格式：数组名称[下标]    表示获取指定下标所对应的值
//需求：获取下标3对应的元素
int num1 = arr1[3];
System.out.println(num1);//83
System.out.println(arr1[3]);//83
```

2.3.2 获取数组元素的个数

在Java中，所有数组都提供了一个length属性，通过这个属性可以访问到数组的长度或者数组中元素的个数

代码实现：

```
//2.获取数组中的元素个数或者数组的长度
//格式:数组名称.length;
int len = arr1.length;
System.out.println("数组arr1的长度为：" + len);
```

2.3.3 修改数组元素的值

代码实现：

```
//3.修改数组元素的值
int num2 = arr1[6];
System.out.println(num2);//
//格式：数组名称[下标] = 被修改之后的值
//注意：不管是静态初始化还是动态初始化，都可以采用这种方式修改元素的值
arr1[6] = 100;
System.out.println(arr1[6]);//100
```

2.3.4 遍历数组

依次访问数组中的每一个元素，获取每个下标对应的元素值
方式一：简单for循环
方式二：增强for循环（foreach）无法使用下标

代码实现：

```
//4.遍历数组
int n0 = arr1[0];
int n1 = arr1[1];
int n2 = arr1[2];
int n3 = arr1[3];
int n4 = arr1[4];
int n5 = arr1[5];
int n6 = arr1[6];

//1>简单for循环
```

```
        //i表示下标，0~arr1.length
        for(int i = 0;i < arr1.length;i++) {
            int n = arr1[i];
            System.out.println(n);
        }

        /*
        2>增强for循环【foreach】
        JDK1.5之后新增的
        优点：用于遍历数组和集合，无需通过数组下标，就可以直接访问数组或者集合中的元素
        语法：
        for(元素数据类型 变量名：数组名称) {
            System.out.println(变量名);
        }
        */
        //底层工作原理：根据下标获取数组元素
        for(int num : arr1) {
            System.out.println("增强for循环的结果：" + num);
        }
        /*
        两种遍历方式的选择：不需要知道下标，只需要获取元素值，则采用增强for循环
        */

        //需求：打印下标为偶数的元素值【只能采用简单for循环】
        for(int i = 0;i < arr1.length;i++) {
            if(i % 2 == 0) {
                int n = arr1[i];
                System.out.println(n);
            }
        }
```

上机练习：键盘录入学生java成绩，并把成绩保存在数组中，然后计算总分和平均分?

```
public static void main(String[] args) {
        Scanner input=new Scanner(System.in);
        System.out.println("请输入班级人数");
        int count=input.nextInt();
        int[] score=new int[count];
        for (int i = 0; i < score.length; i++) {
            System.out.println("请输入第"+(i+1)+"个人的java成绩");
            score[i]=input.nextInt();
        }
        System.out.println("--------------------");
        int sum=0;//总分
        double avg=0;//平均分
        for (int n : score) {
            sum=sum+n;
        }
        avg=(double)sum/count;
        System.out.println("总分是:"+sum);
        System.out.println("平均分是:"+avg);
    }
```

2.3.5 数组的内存分配

java内存：

栈：存储基本类型数据和引用类型的地址，特点：先进后出，一般空间比较小，存取速度较快

堆：存储引用类型数据，特点：空间比较大，存储速度相对较慢

方法区： 存储字符串常量池，静态数据，代码和类的元数据

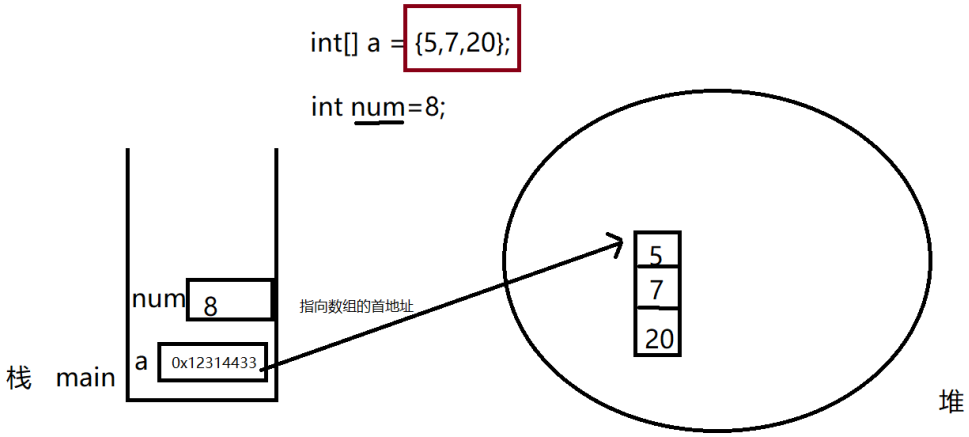
数组属于引用类型，数组引用变量只是一个引用，这个引用变量可以指向任何有效的内存空间，只有当这个引用指向有效的空间时，才可以通过引用去操作真正数组中的元素。

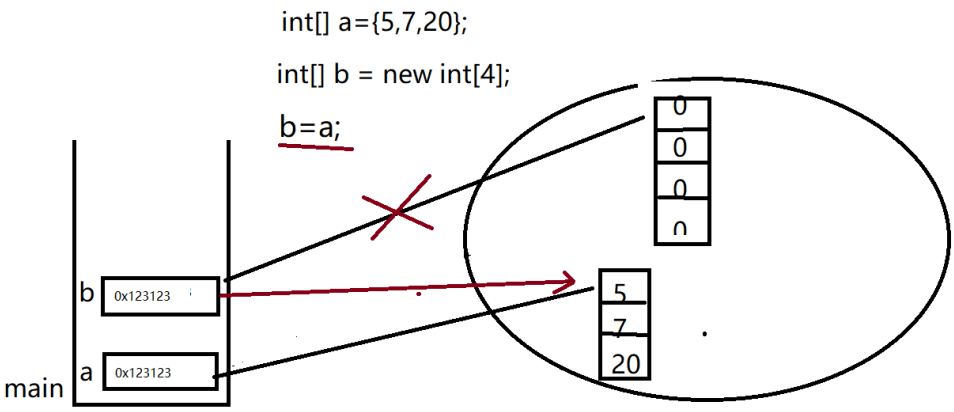
结论：数组的引用变量存储在栈空间中，而真正的数组数据存储在堆空间在中。

代码实现：

```
class ArrayUsageDemo04
{
    public static void main(String[] args)
    {
        //图1 使用静态初始化的方式初始化一个数组a
        int[] a = {5,7,20};
        System.out.println("a的长度为： " + a.length);//3
        int  num =8;
        System.out.println("num:"+num);


        //图2
        int[] b=new int[4];
        System.out.println("b的长度是:"+b.length);
        b=a;
        System.out.println("b的长度是:"+b.length);
    }
}
```

画图分析：





2.3.6 使用数组时常见的问题

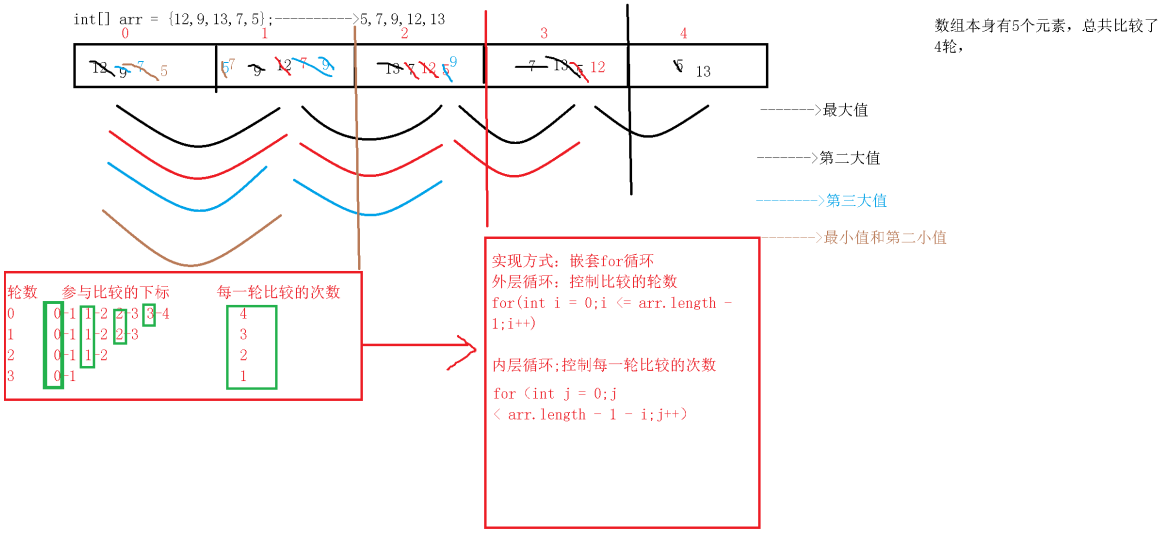
```
1>数组越界异常：ArrayIndexOutofBoundsException
        出现的时机：当使用了不存在的下标时，则会出现这个错误
        0~length - 1

2>空指针异常：NullPointerException
        出现的时机：当数组的引用变量赋值为null，还在后面的代码中使用这个引用
```

2.4 数组的排序

2.4.1 冒泡排序

```
排序思路：比较两个相邻的下标对应的元素，如果符合条件就交换位置（最值出现在最后位）
```

画图分析



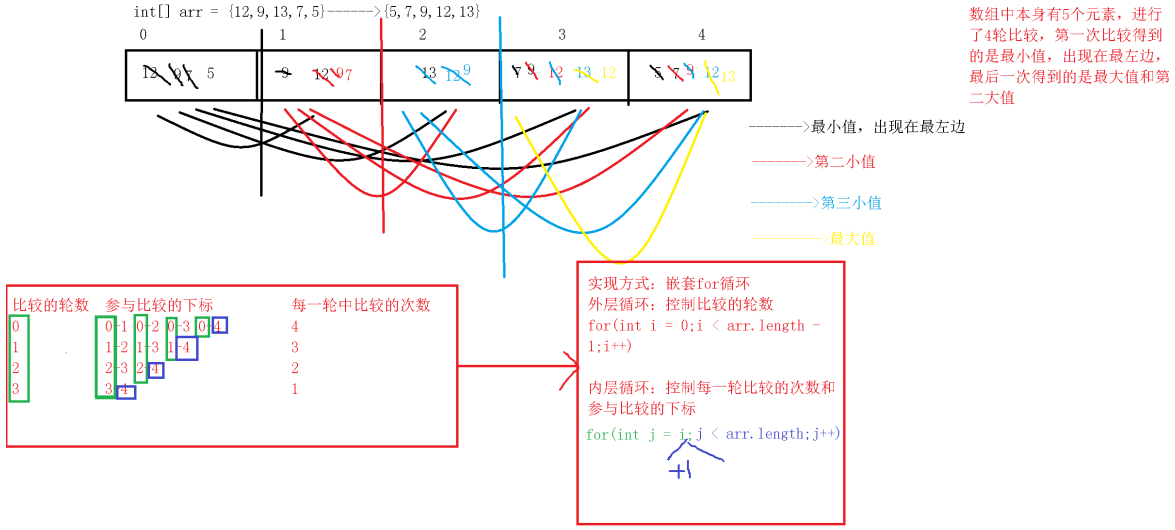
代码实现：

```
class ArraySortedDemo01
{
    public static void main(String[] args)
    {
        int[] arr = {23,54,65,3,5,2,87};
        //以升序为例
        //外层循环：控制比较的轮数
        for(int i = 0;i < arr.length - 1;i++) {
            //内层循环：控制每一轮比较的次数和参与比较的下标
            for(int j = 0;j < arr.length - 1 - i;j++) {
                if(arr[j] > arr[j + 1]) {
                    //交换位置
                    int temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                }
            }
        }

        for(int num:arr) {
            System.out.println(num);
        }
    }
}
```

2.4.2 选择排序

排序思路：固定一个下标，然后拿这个下标对应的值依次和后面的元素进行比较

画图分析：



代码实现：

```
class ArraySortedDemo02
{
    public static void main(String[] args)
    {
        int[] arr = {23,54,65,3,5,2,87};
        //以升序为例
        //外层循环：控制比较的轮数
        for(int i = 0;i < arr.length - 1;i++) {
            //内层循环:控制每一轮比较的次数，参与比较的下标
            for(int j = i + 1;j < arr.length;j++) {
                //交换位置
                if(arr[i] > arr[j]) {
                    int temp = arr[i];
                    arr[i] = arr[j];
                    arr[j] = temp;
                }
            }
        }

        for(int num:arr) {
            System.out.println(num);
        }
    }
}
```

2.5 数组的查找

2.5.1 顺序查找

查找思路：遍历这个数组，依次把每一位元素和要查找的数据进行比较。

代码实现：

```
class ArraySearchDemo01
{
    public static void main(String[] args)
    {
```

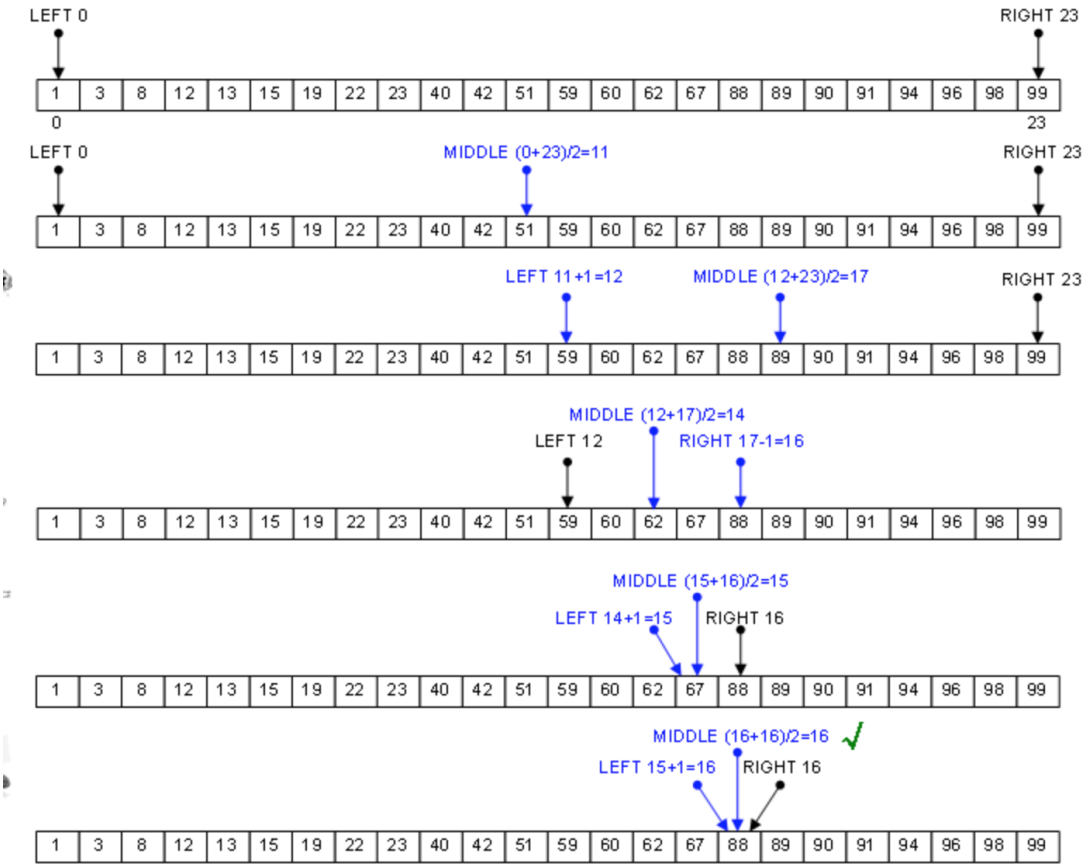
```
        int[] arr = {23,54,65,3,5,2,87};

        //需求：查找65在数组中的位置
        int key = 65;
        for(int i = 0;i < arr.length;i++) {
            if(key == arr[i]) {
                System.out.println(i);
            }
        }
    }
}
```

2.5.2 二分法查找

查找思路：前提是数组是有序（升序或者降序）的，通过折半来缩小查找范围，提高查找效率

将待查找的元素与中间下标对应的元素比较，如果大于中间下标对应的元素，则去右半部分查找

画图分析：



代码实现：

```
class ArraySearchDemo02
{
    public static void main(String[] args)
    {
        int[] arr = {12,43,54,65,87,88,90,343};

        //待查找的元素
        int key = 88;

        //相应的下标
        int left = 0;
```

```
        int right = arr.length - 1;

        while(left <= right) {
            //中间下标
            int middle = (left + right) / 2;//取整
            if(arr[middle] > key) {
                right = middle - 1;
            } else if(arr[middle] < key) {
                left = middle + 1;
            } else {
                System.out.println(middle);
                break;
            }
        }
    }
}
```

第三节 Arrays工具类

作用：主要用于对数组进行排序，查找，填充，比较等的操作
Arrays工具类存在于java.util包下，所以使用的第一步就是导包：import java.util.Arrays;

演示代码

```
//演示Arrays工具类的使用
package com.qf.day07;

import java.util.Arrays;

/*
 * Arrays工具类的使用
 * 1 二分查找
 * 2 排序
 * 3 复制
 * 4 填充
 * 5 把数组转成字符串
 *
 *
 */

public class Demo1 {
    public static void main(String[] args) {
    //  binarySearch();
        //sort();
        //copy();
        //fill();
        toStr();
    }
    //binarySearch 二分查找
    public static void binarySearch() {
        int[] arr=new int[] {5,8,10,20,65,100};
        int result=Arrays.binarySearch(arr,22);
        if(result>=0) {
            System.out.println("找到了");
        }else {
            System.out.println("没找到 ");
        }
    }
```

```
    //排序
    public static void sort() {
        int[] arr=new int[] {12,8,100,2,9};
        Arrays.sort(arr);
        System.out.println("排序之后:");
        for(int i=0;i<arr.length;i++) {
            System.out.print(arr[i]+" ");
        }
    }
    //复制
    public static void copy() {
        int[] arr=new int[] {12,8,100,2,9};
        int[] arr2=Arrays.copyOf(arr, arr.length);
        for(int i=0;i<arr2.length;i++) {
            System.out.println(arr2[i]);
        }
    }
    //填充
    public static void fill() {
        int[] arr=new int[] {12,8,100,2,9};
        Arrays.fill(arr, 10);
        for(int i=0;i<arr.length;i++) {
            System.out.println(arr[i]);
        }
    }
    //把数组转成字符串
    public static void toStr() {
        int[] arr=new int[] {12,8,100,2,9};
        String s=Arrays.toString(arr);
        System.out.println(s);
    }

}
```

第四节 可变参数

不定长参数

```
在设计方法时，方法的形参的个数是不确定的
语法
类型... 变量名称
例如：int... num

好处：不用创建数组，直接写数组元素
```

代码实现：

```
package com.qf.day07;
/*
 * 可变参数
 * int...
 * 好处：不用创建数组，直接写数组元素
 * 注意事项：
 * 1 一个方法只能有一个可变参数
 * 2 可变参数只能方法参数列表最后
 */
public class Demo4 {
    public static void main(String[] args) {
```

```
        //调用
        //int[] nums= {10,4,8,20};
        //sort(nums);
        sort(10,4,8,20,1,2); //实际运行是，把这些数据变成数组

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

注意：

a.不定长的参数在进行使用的时候被当做数组来进行处理

b.一个方法只能有一个可变参数

c. 可变参数只能方法参数列表最后

总结

```
1.eclipse的基本使用
2.数组的基本使用
    （1）先声明
    （2）初始化
        静态初始化 ,不用指定长度，直接赋值元素。
        动态初始化， 需要指定长度，数据默认值
    (3) 声明同时也可以初始化。
3 数组的使用
    通过下标获取元素，下标的范围0-长度-1
    获取数组的长度  length
    修改数组元素
    nums[0]=10;
    遍历：
        （1） for
        (2) 增强for  不能使用下标

4 排序：  冒泡排序   选择排序
5 查找：   顺序查找   二分法查找
6 Arrays 工具类的使用
7 可变参数
```

默写

```
1.使用if语句实现判断一个数是否是5的倍数
2.根据键盘输入的数字，打印对应的星期（注：可以不全写）
3.求50~200之间所有整数的和
```

作业

初级

1.定义一个函数，获取某个数组中的最小值
2.定义一个数组,数组成员10个,找出数组中最大数连同下标一起输出
3.给定一个整型数组，数组成员10个,求该数组中第二大的数的下标
4.B哥去参加青年歌手大奖赛,有10个评委打分,(去掉一个最高一个最低)求平均分?
5.利用选择排序对数据进行降序排序
6.定义数组，存放5个学生的成绩【成绩值自己设定】，将成绩从大到小排序，获得成绩之和，平均成绩，最小成绩，最大成绩。
7.定义一个长度为10的int数组,统计数组中的最大值、最小值、以及奇数和偶数的个数
8.提取一个方法，将指定数组中的数组元素进行反转
例如：{10,23,2,45,6}--->{6,45,2,23,10}

中级
1.将一个数组逆序输出
2.输入数组，最大的与第一个元素交换，最小的与最后一个元素交换，输出数组
3.有一个已经排好序的数组。现输入一个数，要求按原来的规律将它插入数组中

面试题

1.基本数据类型和引用数据类型之间的区别
2.在java中，声明一个数组过程中，是如何分配内存的
3.数组的静态和动态初始化有什么不同
4.分别使用冒泡和选择对已知数组进行排序