

实验内容	第三周实验 Array + Class 初步			成绩	
姓 名	王秋锋	学号	2015111948	班 级	计科三班
专 业	计算机科学与技术			日 期	2017 年 9 月 21 日

### 【实验目的 01】--数组

- ◆ 理解二维数组的概念，能够声明、创建和初始化各种类型的二维数组
- ◆ 掌握对一维或二维数组元素的访问方法
- ◆ 掌握不规则二维数组的声明、创建和初始化以及访问的方法了解 Arrays 类及 System 类中关于数组操作的方法的使用

### 【实验内容】

#### 1、编辑、编译、运行下面 java 程序

```
public class Demo {
    public static void main(String[] args){
        int i, j;
        int a[] = {2, 1, 4, 8, 9, 5, 3};
        for(i = 0; i < a.length-1; i++){
            int k = i;
            for(j = i; j < a.length; j++){
                if(a[j] < a[k])k = j;
            }
            int temp = a[i];
            a[i] = a[k];
            a[k] = temp;
        }
        for(int x : a)System.out.print(x + " ");
        System.out.println();
    }
}
```

**要求：**分析程序给你，给出运行结果

### 【实验结果与分析】

运行结果：

1 2 3 4 5 8 9

结果分析：

本程序是冒泡排序算法代码，实现了数组的从小到大排序

#### 2、有如下数组

要求: (1) 分别给出上述三问的程序代码和程序的运行结果

(2) 上述问题, 在输出数组元素时, for 语句和 for each 语句至少各用一次

### 【实验结果与分析】

(1) 程序代码:

```
public class Demo {  
    public static void main(String[] args){  
        int myArray[] = { 1, 2, 3, 4, 5, 6 };  
        int yourArray[] = { 10, 9, 8, 7, 6, 5, 4, 3, 2, 1};  
        int hisArray[] = {12,42,34,5,-34,36,323,-233,43,12};  
        System.arraycopy(myArray, 0, yourArray, 0, myArray.length);  
        for(int x:yourArray)System.out.print(x + " ");  
        System.out.println();  
    }  
}
```

运行结果:

1 2 3 4 5 6 4 3 2 1

(2) 程序代码:

```
import java.util.Arrays;  
public class Demo {  
    public static void main(String[] args){  
        int myArray[] = { 1, 2, 3, 4, 5, 6 };  
        int yourArray[] = { 10, 9, 8, 7, 6, 5, 4, 3, 2, 1};  
        int hisArray[] = {12,42,34,5,-34,36,323,-233,43,12};  
        Arrays.sort(hisArray);  
        for(int x:hisArray)System.out.print(x + " ");  
        System.out.println();  
    }  
}
```

运行结果:

-233 -34 5 12 12 34 36 42 43 323

(3) 程序代码:

```
import java.util.Arrays;  
public class a01_2 {  
    public static void main(String[] args){  
        int myArray[] = { 1, 2, 3, 4, 5, 6 };  
        int yourArray[] = { 10, 9, 8, 7, 6, 5, 4, 3, 2, 1};  
        int hisArray[] = {12,42,34,5,-34,36,323,-233,43,12};  
        Arrays.sort(hisArray);
```

```
int pos = Arrays.binarySearch(hisArray, 12);
if(pos >= 0)System.out.println("指定元素的位置为: "+pos);
else System.out.println("no found");
}
}
```

运行结果:

指定元素的位置为: 4

3、产生 20 个范围为 0~50 之间的不重复随机整数存入到一个数组中，将数组中元素按由小到大输出，并求数组所有元素的平均值。

【提示】每产生一个整数要跟前面已产生的数进行比较，如果已存在，则不算，要重新产生，可以用一个循环来控制产生一个新数，只有产生的数未出现过，才能出循环

要求：给出源码与程序运行结果

### 【实验结果与分析】

程序代码:

```
import java.util.Arrays;
import java.util.Random;
public class arc1_3 {
    public static void main(String[] args){
        int a[] = new int[20];
        Random rand = new Random();
        for(int i=0; i<20; i++){
            while(true){
                int x = rand.nextInt(50);
                boolean fg = true;
                for(int j=0; j<i; j++){if(a[j] == x){
                    fg = false;
                    break;
                }
            }
            if(fg){
                a[i] = x;
                break;
            }
        }
        Arrays.sort(a);
        for(int i=0; i<20; i++)System.out.print(a[i] + " ");
        System.out.println();
        double ave = 0.0;
        for(int i=0; i<20; i++)ave += a[i];
    }
}
```

```
    ave /= 20;
    System.out.println("平均值为: "+ave);
}
}
```

运行结果

```
1 2 8 9 10 11 12 16 21 22 23 25 26 27 31 39 40 44 47 49
平均值为: 23.15
```

#### 4、餐饮服务质量调查打分

在商业和科学研究中，经常需要对数据进行分析并将结果以直方图的形式显示出来。例如，一个公司的主管需要了解一年来公司的营业状况，比较一下各月份的销售收入状况，如果仅给出一大堆数据，这显然太不直观了，如果能将这些数据以条形图（直方图）的形式表示，那么将会大大增加这些数据的直观性，也便于数据的分析与对比。下面以顾客对餐饮服务打分为例，练习一下这方面的程序。假设现在有 100 个消费者被邀请来给自助餐厅的食品和服务质量打分，分数为从 1~5 的 5 个等级(1 意味着最低分,5 表示最高分)，试统计调查结果，并用 ‘\*’ 打印出如下形式的统计结果直方图。

Grade	Count	Histogram
1	5	*****
2	10	*****
3	7	*****

- 要求：**1、定义一个数组，随机产生 100 个 1~5 之间的随机数，作为 100 个消费者的打分
- 2、统计上述打分数据，并输出上面形式的图
- 3、给出源码与程序运行结果。

#### 【实验结果与分析】

程序代码：

```
import java.util.Arrays;
import java.util.Random;
public class arc1_4 {
    public static void main(String[] args){
        int a[] = new int[100];
        Random rand = new Random();
        for(int i=0; i<100; i++){
            int x = rand.nextInt(5)+1;
            a[i] = x;
        }
    }
}
```

```

    }
    int cnt[] = new int[6];
    for(int i=0; i<100; i++)cnt[a[i]]++;
    System.out.println("Grade      Count  Histogram");
    for(int i=1; i<=5; i++){
        System.out.print(i+"      "+cnt[i]+"  ");
        for(int j=1; j<=cnt[i]; j++)System.out.print("*");
        System.out.println("");
    }
}
}

```

运行结果

Grade	Count	Histogram
1	26	*****
2	18	*****
3	17	*****
4	27	*****
5	12	*****

## 【实验目的 02】---类与对象

- ◆ 掌握类的定义，理解成员变量、构造方法、成员方法的作用；
- ◆ 理解类和对象的区别，掌握构造函数的使用，熟悉通过对象名引用实例的方法和属性；

## 【实验内容】

### 1. 阅读并编辑下面的程序，给出程序运行结果

## 【实验结果与分析】

程序代码：

```

class Rect{ //定义矩阵类
    private double w,h;
    Rect(double w,double h){ //构造方法
        this.w = w;
        this.h = h;
    }
    Rect(Rect s){ //构造方法
        w = s.w;
        h = s.h;
    }
    double getW(){ //返回宽
        return w;
    }
    double getH(){ //返回高

```

```
        return h;
    }
    double area(){ //求面积
        return getW()*getH();
    }
}

public class Demo{ //测试类
    public static void main(String args[]){
        Rect s[] = new Rect[3];
        s[0] = new Rect(2,3);
        s[1] = new Rect(5,6);
        s[2] = new Rect(new Rect(7,8));
        for(int i= 0; i<s.length;i++){
            System.out.print("s["+i+"] area="+s[i].area());
            System.out.println(" wide =" +s[i].getW()+" high =" +s[i].getH());
        }
    }
}
```

运行结果

```
s[0] area=6.0 wide =2.0 high =3.0
s[1] area=30.0 wide =5.0 high =6.0
s[2] area=56.0 wide =7.0 high =8.0
```

## 2 阅读下面的程序（程序填空及运行结果）

### 【实验结果与分析】

程序代码：

```
public class Circle
{
    private float r; //定义私有变量圆半径
    Circle (){} //定义空构造函数
    Circle (float x) //定义空构造函数并初始化变量 r。
    {r=x;}
    public void set_value(float x) //初始化圆半径的值
    { r=x; }
    public double mj() //求圆面积
    { return(3.14*r*r); }
    public double zc() //求圆周长
    {
        double y;
        y=2*3.14*r;
        return(y);
    }
    public static void main(String args[])
```

```
{ Circle ex1 = new Circle(), ex2 = new Circle(); //定义类 Circle 对象（或  
类变量）并初始化  
    ex1.set_value(3.2f); //初始化 Circle 类对象圆半径的值  
    System.out.println("面积="+ex1.mj()+"周长="+ex1.zc());  
    ex2.set_value(5.5f); //定义类 Circle 对象并设置半径为 5.5.  
    System.out.println("面积="+ex2.mj()+"周长="+ex2.zc());  
}  
}
```

运行结果

```
面积=32.15360095825196周长=20.096000299453735  
面积=94.985周长=34.54
```

### 3. 创建学生类 Student 来完成下面问题。

- (1) 类 Student，属性：学号 no、姓名 name 和年龄 age，score（数组，多科成绩）等且都是 private；
- (2) 声明一个或多个构造方法，调用不同的构造方法，灵活初始化对象的所有的域；
- (3) 声明获得各属性（学号、姓名和年龄）的各个方法；
- (4) 声明设置各属性（学号、姓名和年龄）的各个方法；
- (5) 求平均成绩的方法；
- (6) 声明一个 public 型的 toString 方法，将该类所有域信息组合成一个字符串；
- (7) 声明统计创建 Student 对象的个数的成员变量 count 和得到 Student 对象的个数的方法。

**要求：**在主类（含 main 方法的类）中使用 Student 类创建两个 Student 对象，输出对象所有域信息；修改其姓名和年龄，并显示。

#### 【实验结果与分析】

程序代码：

```
import java.util.Arrays;  
import java.util.Scanner;  
public class Student {  
    private int no, age;  
    private double score[];  
    private String name;  
    static int count = 0;  
    Student(){}  
    Student(int no, int age, String name) {  
        count++;  
        this.no = no;  
        this.age = age;  
        this.name = name;  
        set_score();  
    }  
}
```

```
}
//声明获得各属性（学号、姓名和年龄）
int getno(){
    return no;
}
int getage(){
    return age;
}
String getname(){
    return name;
}
//声明设置各属性（学号、姓名、年龄和成绩）
void set_no(int _no){
    no = _no;
}
void set_age(int _age){
    age = _age;
}
void set_name(String _name){
    name = _name;
}
void set_score(){
    System.out.println("第"+count+"个学生: ");
    Scanner sc = new Scanner(System.in);
    System.out.println("请输入科目数: ");
    int n = sc.nextInt();
    score = new double[n];
    for(int i=1; i<=n; i++){
        System.out.print("请输入第"+i+"科成绩: ");
        score[i-1] = sc.nextDouble();
    }
}
String get_score(){
    return Arrays.toString(this.score);
}
//求平均成绩
double get_ave(){
    double ans = 0.0;
    for(double x : score)ans += x;
    ans /= score.length;
    return ans;
}
//将该类所有域信息组合成一个字符串
public String change(){
    return "学号:"+getno()+" 姓名: "+getname()+" 年龄: "+getage()+" 各科成绩: "+get_score()+" 平均成绩: "+get_ave();
}
```



```
}

public static void main(String[] args){
    Student s[] = new Student[2];
    s[0] = new Student(2015111948, 20, "王秋锋");
    s[1] = new Student(2015111666, 25, "Alice");
    System.out.println("输出学生信息: ");
    for(int i=1; i<=2; i++){
        System.out.println("第"+i+"个学生, 其信息为: "+s[i-1].change());
    }
    s[0].set_name("Bob");
    s[0].set_age(16);
    System.out.println("输出修改后第一个学生信息: "+s[0].change());
}
}
```

运行结果:

```
第1个学生:
请输入科目数:
2
请输入第1科成绩: 88
请输入第2科成绩: 99
第2个学生:
请输入科目数:
3
请输入第1科成绩: 56
请输入第2科成绩: 45
请输入第3科成绩: 98
输出学生信息:
第1个学生, 其信息为: 学号:2015111948 姓名: 王秋锋 年龄: 20 各科成绩: [88.0, 99.0] 平均成绩: 93.5
第2个学生, 其信息为: 学号:2015111666 姓名: Alice 年龄: 25 各科成绩: [56.0, 45.0, 98.0] 平均成绩: 66.33333333333333
输出修改后第一个学生信息: 学号:2015111948 姓名: Bob 年龄: 16 各科成绩: [88.0, 99.0] 平均成绩: 93.5
```

#### 4、编辑、编译、运行下面 java 程序，

```
import java.util.InputMismatchException;
import java.util.Scanner;
public class MyDate {
    int year;
    int month;
    int day;
    int num_order; // 接受来自命令行的输入, int类型, 只要负责showTips()的命令代号和
日期的修改
    boolean flag = true; // 控制程序的退出, 当flag = false时, 退出
    public MyDate(){
        year = 2012;
        month = 3;
        day = 14;
```

```
}  
// 构造函数  
public MyDate(int year, int month, int day) {  
    this.year = year;  
    this.month = month;  
    this.day = day;  
}  
// showTips()用于提示用户相关操作代号  
public void showTips() {  
    System.out.println("1修改年份");  
    System.out.println("2修改月份");  
    System.out.println("3修改日期");  
    System.out.println("4修改全部");  
    System.out.println("5退出修改");  
}  
public void changeDate() { //修改日期的函数  
    Scanner _order = new Scanner(System.in);  
    System.out.print("请输入您的命令: ");  
    //下面的数字异常没有捕获  
    num_order = _order.nextInt();  
    if (num_order == 1) {  
        System.out.println("请输入新的年份: ");  
        year = _order.nextInt();  
        System.out.println();  
    }  
    else if (num_order == 2) {  
        System.out.println("请输入新的月份: ");  
        month = _order.nextInt();  
        System.out.println();  
    }  
    else if (num_order == 3) {  
        System.out.println("请输入新的日份: ");  
        day = _order.nextInt();  
        System.out.println();  
    }  
    else if (num_order == 4) {  
        System.out.println("请输入新的年份: ");  
        year = _order.nextInt();  
        System.out.println("请输入新的月份: ");  
        month = _order.nextInt();  
        System.out.println("请输入新的日份: ");  
        day = _order.nextInt();  
        System.out.println();  
    }  
    else if (num_order == 5) {  
        flag = false;  
    }  
}
```

```
        System.out.println("程序结束");
    }
}
public void displayDate() { //输出现在的日期函数
    if (month >= 0 && month <= 12 && day >= 1 && day <= 31) {
        System.out.println("目前的日期为: " + year + "年" + month + "月" + day +
"日");
    }
    else {
        System.out.println("您的输入与设计情况不相符, 请确认后重新来过!");
    }
    showTips();
    changeDate();
}
public static void main(String[] args) {
    MyDate mydate = new MyDate();
    while (mydate.flag) {
        try {
            mydate.displayDate();
        }
        catch (InputMismatchException ime) {
            System.out.println("您的输入与要求不一致, 请确定后重新来过!");
            mydate.changeDate();
        }
    }
}
}
```

**要求:**

(1) 分析上面的程序, 写出运行结果 (该题要求: 运行结果用截图)

【实验结果与分析】 给出上述程序运行结果:

目前的日期为: 2012年3月14日

1修改年份

2修改月份

3修改日期

4修改全部

5退出修改

请输入您的命令: 1

请输入新的年份:

2017

目前的日期为: 2017年3月14日

1修改年份

2修改月份

3修改日期

4修改全部

5退出修改

请输入您的命令: 2

请输入新的月份:

9

目前的日期为: 2017年9月14日

1修改年份

2修改月份

3修改日期

4修改全部

5退出修改

请输入您的命令: 3

请输入新的日份:

21

目前的日期为: 2017年9月21日

1修改年份

2修改月份

3修改日期

4修改全部

5退出修改

请输入您的命令: 4

请输入新的年份:

2012

请输入新的月份:

12

请输入新的日份:

12

目前的日期为: 2012年12月12日

1修改年份

2修改月份

3修改日期

4修改全部

5退出修改

请输入您的命令: 5

程序结束

实验分析:

本程序定义了类 MyDate, 它有变量 year,month,day,num\_order,flag,分别代表年、月、日、接受来自命令行的输入、程序结束的标志,类中还定义了两个构造函数,函数 showTips()显示初始界面,函数 changeDate()可以修改日期,函数 displayDate()输出现在的日期函数,最后主函数构造类,并不断调用直到程序退出

