

单元 5.3 我爱大明星（三）

单元教学设计

学习阶段		二、实用程序开发		学时	
项目 5		我爱大明星小游戏		学时	12
单元 5.3		我爱大明星（三）		学时	2
教 学 目 标	能力目标	知识目标		思政目标	
	能够灵活应用一维数组解决现实问题。	理解什么是数组？ 掌握数组的定义、初始化和使用		课上学习练习	引导学生对待偶像的态度：不要只关注偶像的外表，而忽视内在；盲目维护偶像，缺乏理智以及沉迷偶像崇拜，引导学生从偶像崇拜中挖掘其教育价值，让他们在选择偶像的时候，不要只看外表，而要更加注重偶像的内在品质，激励自己向更好的方向发展，从而树立崇高的人生目标，为实现伟大的中国梦而添砖加瓦。
		掌握如何使用 java 中的数组。 掌握如何使用对象数组。		课上学习练习	
	知 识 储 备	知识储备： 马士兵讲座： 《一维数组》	练习 1：数组求和 定义一个整型数组 score[]={22, 11, 33, 15, 23, 12, 14, 55}，求出数组元素的和。		
		练习 2：数组求最大最小值 定义一个整型数组 score[]={22, 11, 33, 15, 23, 12, 14, 55}，求出数组元素的最大值和最小值。			
十分钟测试		十分钟测试： 定义由 30 个 0~9 之间数字组成的数组，如 0, 0, 1, 2, 3, 5, 4, 5, 2, 8, 7, 6, 9, 5, 4, 8, 3, 1, 0, 2, 4, 8, 7, 9, 5, 2, 1, 2, 3, 9，请分别统计 0~9 这 10 个数字分别出现了多少次。			
教学实施					
能 力 训 练 任 务	任务一	定义 JButton 数组，代替 b0~b5 显示按钮			
	任务二	定义 Star 数组，用数组管理明星对象。 打印“杨辉三角”			
	任务三	改进 LoveStar 程序，进一步优化代码 课程思政 了解软件行业的最新发展，树立爱国情、报国志。引导学生： ② 了解计算机软件从业人员应当具备的职业道德守则，为进军软件行业做准备； ②了解 Java 语言强大的生态体系； ③了解国情与世界与 IT 相关的软件、硬件、企业的相关情况，芯片垄断、华为被挤压、抖音、微信被禁事件。 ③ 新时代与个人成才的关系； ④ ⑤通过软件行业发展前景，引发学生对未来的职业愿景，激发学生对社会主义核心价值观的认同感。			
课后学习资源		1. 微课《一维数组》课堂录像 2. 微课《用 JXL 读 Excel 表》 3. 参考文档： 《不会编写 JavaBean 就不是一个 Java 开发人员》 《Java 事件处理机制-事件监听器的四种实现方式》 《详谈类的封装的使用》			

单元教学进度设计

Step1：项目导入（25）分钟

新课引入

现在已经可以实现了显示多个明星信息的功能：



但是程序代码还可以进一步优化，比如我们显示了 6 个明星的信息，6 个明星和 6 个按钮相互独立，但是对这些对象的操作代码重复度很高，有什么方法可以把同类的对象管理起来，以简化代码。
我们可以用数组来解决。

课前测试与知识梳理

教学环节	教学内容
点 评 1	练习 1：数组求和 定义一个整型数组 <code>score[]={22, 11, 33, 15, 23, 12, 14, 55}</code>，求出数组元素的和。 <pre>public void test1() { int[] score={22, 11, 33, 15, 23, 12, 14, 55}; int sum=0; for(int i=0;i<score.length;i++){ sum+=score[i]; } System.out.println("整型数组的和="+sum); }</pre> 错误 1: <pre>Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 8 at unit3.practice.TestArray1.test2(TestArray1.java:19) at unit3.practice.TestArray1.main(TestArray1.java:26)</pre> <pre>public void test2() { int[] score={22, 11, 33, 15, 23, 12, 14, 55}; int sum=0; for(int i=1;i<=score.length;i++){ sum+=score[i]; } System.out.println("整型数组score[]的和="+sum); }</pre>
点 评 2	练习 1：数组求最大最小值 定义一个整型数组 <code>score[]={22, 11, 33, 15, 23, 12, 14, 55}</code>，求出数组元素的最大值和最小值。

```

public void test1() {
    int[] score={22, 11, 33, 15, 23, 12, 14, 55};
    int max=score[0];
    int min=score[0];
    for(int i=1;i<score.length;i++) {
        if(score[i]>max) {
            max=score[i];
        }
        if(score[i]<min) {
            min=score[i];
        }
    }
    System.out.println("最大值是: "+max+", 最小值是:"+min);
}

```

十分钟测试

1. 用if……else语句判断

```

public void test1() {
    int[] n={0, 0, 1, 2, 3, 5, 4, 5, 2, 8, 7, 6, 9, 5, 4, 8, 3, 1, 0, 2, 4, 8, 7, 9, 5, 2, 1, 2, 3, 9};
    int n0=0;
    int n1=0;
    int n2=0;
    int n3=0;
    int n4=0;
    int n5=0;
    int n6=0;
    int n7=0;
    int n8=0;
    int n9=0;
    for(int i=0;i<n.length;i++) {
        if(n[i]==0) { n0++;
        } else if(n[i]==1) { n1++;
        } else if(n[i]==2) { n2++;
        } else if(n[i]==3) { n3++;
        } else if(n[i]==4) { n4++;
        } else if(n[i]==5) { n5++;
        } else if(n[i]==6) { n6++;
        } else if(n[i]==7) { n7++;
        } else if(n[i]==8) { n8++;
        } else if(n[i]==9) { n9++;
        }
    }
    System.out.println("0出现的次数: "+n0);
    System.out.println("1出现的次数: "+n1);
    System.out.println("2出现的次数: "+n2);
    System.out.println("3出现的次数: "+n3);
    System.out.println("4出现的次数: "+n4);
    System.out.println("5出现的次数: "+n5);
    System.out.println("6出现的次数: "+n6);
    System.out.println("7出现的次数: "+n7);
    System.out.println("8出现的次数: "+n8);
    System.out.println("9出现的次数: "+n9);
}

```

```
}
```

2. 用switch语句判断

```
public void test2() {  
    int[] n={0, 0, 1, 2, 3, 5, 4, 5, 2, 8, 7, 6, 9, 5, 4, 8, 3, 1, 0, 2, 4, 8, 7, 9, 5, 2, 1, 2, 3, 9};  
    int n0=0;  
    int n1=0;  
    int n2=0;  
    int n3=0;  
    int n4=0;  
    int n5=0;  
    int n6=0;  
    int n7=0;  
    int n8=0;  
    int n9=0;  
  
    for(int i=0;i<n.length;i++){  
        switch(n[i]) {  
            case 0:n0++;break;  
            case 1:n1++;break;  
            case 3:n3++;break;  
            case 4:n4++;break;  
            case 5:n5++;break;  
            case 6:n6++;break;  
            case 7:n7++;break;  
            case 8:n8++;break;  
            case 9:n9++;break;  
        }  
    }  
    System.out.println("0出现的次数: "+n0);  
    System.out.println("1出现的次数: "+n1);  
    System.out.println("2出现的次数: "+n2);  
    System.out.println("3出现的次数: "+n3);  
    System.out.println("4出现的次数: "+n4);  
    System.out.println("5出现的次数: "+n5);  
    System.out.println("6出现的次数: "+n6);  
    System.out.println("7出现的次数: "+n7);  
    System.out.println("8出现的次数: "+n8);  
    System.out.println("9出现的次数: "+n9);  
}
```

十分钟测试教师的解决方案

n0	n1	n2	n3	n4	n5	n6	n7	n8	n9
0	1	2	3	4	5	6	7	8	9
n [0]	n [1]	n [2]	n [3]	n [4]	n [5]	n [6]	n [7]	n [8]	n [9]

```
public void test3() {  
    int[] n={0, 0, 1, 2, 3, 5, 4, 5, 2, 8, 7, 6, 9, 5, 4, 8, 3, 1, 0, 2, 4, 8, 7, 9, 5, 2, 1, 2, 3, 9};  
    int[] m=new int[10];  
    for(int i=0;i<n.length;i++){  
        int x=n[i];
```

```

        m[x]++;
    }
    for(int i=0;i<m.length;i++){
        System.out.println(i+"出现的次数："+m[i]);
    }
}

```

关于数组的总结

为什么要使用数组？

问：编写代码保存 4 名学生的考试成绩。

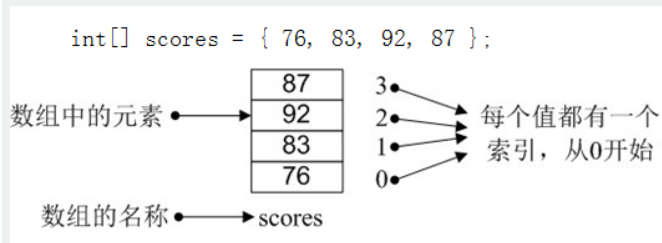
答：简单啊，定义 4 个变量呗

问：那“计算全年级 400 名学生的考试成绩”，肿么办

答：。。。。。。。

数组，就可以帮助你妥妥的解决问题啦！！

数组可以理解为一个巨大的“盒子”，里面可以按顺序存放多个类型相同的数据，比如可以定义 int 型的数组 scores 存储 4 名学生的成绩



数组中的元素都可以通过下标来访问，下标从 0 开始。例如，可以通过 scores[0] 获取数组中的第一个元素 76，scores[2] 就可以取到第三个元素 92 啦！

如何使用 Java 中的数组

Java 中操作数组只需要四个步骤：

1、声明数组

语法：数据类型[] 数组名；

或者 数据类型 数组名[]；

其中，数组名可以是任意合法的变量名，如：

```

int[] scores; // 存储学生成绩的数组，类型为整型
double height[]; // 存储学生身高的数组，类型为浮点型
String[] names; // 存储学生姓名的数组，类型为字符串

```

2、分配空间

简单地说，就是指定数组中最多可存储多少个元素

语法：数组名 = new 数据类型 [数组长度]；

其中，数组长度就是数组中能存放元素的个数，如：

```

scores = new int[5]; // 长度为5的整数数组
height = new double[5]; // 长度为5的浮点型数组
names = new String[5]; // 长度为5的字符串数组

```

话说，我们也可以将上面的两个步骤合并，在声明数组的同时为它分配空间，如：

```
int[] scores = new int[5];
```

3、 赋值

分配空间后就可以向数组中放数据了，数组中元素都是通过下标来访问的，例如向 scores 数组中存放学生成绩

```
scores[0] = 89;
scores[1] = 79;
```

4、 处理数组中数据

我们可以对赋值后的数组进行操作和处理，如获取并输出数组中元素的值

```
System.out.println("scores数组中第一个元素的值: " + scores[0]);
```

在 Java 中还提供了另外一种直接创建数组的方式，它将声明数组、分配空间和赋值合并完成，如

```
int[] scores={78,91,84,68};
```

它等价于：

```
int[] scores=new int[] {78,91,84,68};
```

使用循环操作 Java 中的数组

实际开发中我们经常使用循环控制数组成员的操作。如：

```
int[] scores={78,91,84,68};
for(int i=0;i<scores.length;i++){
    System.out.println("数组中第"+(i+1)+"个元素的值是: "+scores[i]);
}
```

运行结果：

```
数组中第1个元素的值是： 78
数组中第2个元素的值是： 91
数组中第3个元素的值是： 84
数组中第4个元素的值是： 68
```

其中， 数组名.length 用于获取数组的长度
需要注意的“小毛病”：

- 1、 数组下标从 0 开始。因此 scores[3] ，表示数组中的第 4 个元素，而并非第 3 个元素
- 2、 数组下标的范围是 0 至 数组长度-1 ，如果越界访问，就会报错。

Step2：任务实施

任务一【20 分钟】

教学环节	教学内容	教师活动	学生活动
任务引入	1. 在 LoveStar 程序中，我们对 Button 的操作，具有很多重复的部分，而且，当 button 数量增加时，须要修改多处代码。 通过分析，如果用数组来管理所有的 Button，一方面可以简化代码，另一方面，当 button 数量发生改变的时候，易于修改。	布置任务	参与讨论
任务	自定义对象数组，它们由同一个类的多个具体实例有序组成。 对象数组中，由于每个元素都是数据类型的引用，因此数组在创建后，		

部署	必须对每个对象分别进行实例化创建。也就是： 1. 声明对象类型数组。 2. 依次初始数组对象。		
任务实施	在测试窗口中，定义 10 个按钮，并依次添加到窗口上。 1. 声明数组对象 <code>JButton[] bs=new JButton[10];</code> 注意：new JButton[10];不带（），因为这个时候每个 button 只声明而未初始化。 2. 初始化数组对象 <pre>for(int i=0;i<bs.length;i++) { bs[i]=new JButton("JBUTTON"+(i+1)); }</pre> 3. 给数组加上事件监听器 <pre>for(int i=0;i<bs.length;i++) { bs[i].addActionListener(this); }</pre> 4. 给数组加上标示 <pre>for(int i=0;i<bs.length;i++) { bs[i].setActionCommand(Integer.toString(i)); }</pre> 5. 设置按钮位置和大小 <pre>for(int i=0;i<bs.length;i++) { bs[i].setBounds(20, 20+i*50, 100, 30); }</pre> 6. 将按钮添加到窗口上 <pre>for(int i=0;i<bs.length;i++) { c.add(bs[i]); }</pre>	演示编程	编程思考
	合并代码： <pre>for(int i=0;i<bs.length;i++) { bs[i]=new JButton("JBUTTON"+(i+1)); bs[i].addActionListener(this); bs[i].setActionCommand(Integer.toString(i)); bs[i].setBounds(20, 20+i*40, 150, 30); c.add(bs[i]); }</pre>	演示编程	编程思考
错误情境	对象数组必须要先初始化再使用，否则会出现如下错误： <pre>Exception in thread "main" java.lang.NullPointerException at unit3.job1.TestButtonArray_2.<init>(TestButtonArray_2.java:17) at unit3.job1.TestButtonArray_2.main(TestButtonArray_2.java:33)</pre>	讲解	听讲思考
任务实施	用数组来管理明星按钮。 1. 声明： <code>JButton[] b=new JButton[6];</code> 2. 初始化 <code>b[0]=new JButton("范冰冰");</code> <code>b[1]=new JButton("李易峰");</code>		

	<pre> b[2]=new JButton("杨幂"); b[3]=new JButton("林志颖"); b[4]=new JButton("姚笛"); b[5]=new JButton("高圆圆"); 3. 设置 for(int i=0;i<b.length;i++){ b[i].setBounds(20+i*120, 500, 100, 30); b[i].addActionListener(this); b[i].setFont(font2); c.add(b[i]); } </pre>		
--	--	--	--

任务二按钮事件监听【20 分钟】

教学环节	教学内容	教师活动	学生活动
任务引入	我们现在一共有 6 个明星，请用修改代码，用 Star 数组来管理明星信息。	演示	观看
任务实施	由学生自行对代码进行改造。 错误情境： <pre> Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 6 at unit3.job2.StarFrame10.initFrame(StarFrame10.java:54) at unit3.job2.StarFrame10.<init>(StarFrame10.java:27) at unit3.job2.StarFrame10.main(StarFrame10.java:129) </pre> 着重演示错误解决过程 1. 数组访问越界错误，非常常见的错误 2. 当错误出现时，依次查看错误可能原因，倒推法。	辅导	编程实现

任务三十分钟测试【15 分钟】

任务部署	虽然用 Star 对象数组来管理明星信息，但是程序依然么有太大的优化，如果不是 6 个明星，而是 16 个甚至 106 个明星呢？ 让我们来对代码进行优化。	布置任务	完成测试
任务实施	1. 在按钮上设置标示 2. 改进的 actionPerformed(ActionEvent e) 方法 <pre> public void actionPerformed(ActionEvent e) { this.showStar(star[Integer.parseInt(e.getActionCommand())]); } </pre>	引导编程	听讲思考
	对自己的程序进行修改和优化。	辅导	编程
课程思政	了解软件行业的最新发展，树立爱国情、报国志。引导学生： ⑤ 了解计算机软件从业人员应当具备的职业道德守则，为进军软件行业做准备； ②了解 Java 语言强大的生态体系；	引导主持	参与讨论

	③了解国情与世界与 IT 相关的软件、硬件、企业的相关情况，芯片垄断、华为被挤压、抖音、微信被禁事件。 ⑥ 新时代与个人成才的关系； ⑤通过软件行业发展前景，引发学生对未来的职业愿景，激发学生对社会主义核心价值观的认同感。		
--	---	--	--

Step3: 总结与课后安排【10 分钟】

教学小结	1. 为什么要使用数组 2. 数组有什么作用？ 3. 数组的声明、初始化和遍历 4. 对象数组的使用。
拓展实践	遵循同样的步骤，完善课外项目。
课后学习资源	1. 微课《一维数组》课堂录像 2. 微课《用 JXL 读 Excel 表》 3. 参考文档： 《不会编写 JavaBean 就不是一个 Java 开发人员》 《Java 事件处理机制—事件监听器的四种实现方式》 《详谈类的封装的使用》