

**桂林航天工业学院**

**实验报告**

课程名称 **面向对象程序设计 （D）**

开课学期

　 实 验 室

班 级

姓 名 **张旭君**

学 号

成绩： （五级）

桂林航天工业学院学生实验报告

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 课程名称 | **面向对象程序设计（D）** | | | 实验项目名称 | | **实验一 Java运行与开发环境** | | |
| 开课教学单位及实验室 | | 计算机科学与工程学院启天楼B123 | | | | 实验日期 | |  |
| 学生姓名 |  | | 学号 | |  | 专业班级 | |  |
| 指导教师 |  | | | | 实验成绩 | |  | |
| 一、实验目的  1.熟练搭建Java运行的环境  2.熟悉Eclipse的开发环境  3.掌握编写一个Java应用程序并运行  二、实验操作方法和步骤  **1.在所在机器上安装JDK环境**  **2.进入系统中配置环境变量**  **①打开系统高级系统设置；②点击环境变量；**    **③在系统变量一栏中“新建”；**  **④打开的窗口中，变量名一栏输入：JAVA\_HOME，变量名选择你JDK安装的路径；**    **⑤设置完成后，在找到path，选中并在内新建：%JAVA\_HOME%\bin，随后全部确定，并关闭环境配置。**    **⑥设置完成后，键盘组合键WIN+R ，在输入cmd，打开后输入，javac，响应则配置成功。**    **3.编写、调试并运行一个简单的HelloWorld程序**  **①在一个喜欢的地方创建一个名为demo.java 的文本，输入以下内容**    ②使用组合键Win+R打开终端，然后通过终端命令进入到自己创建的文件路径；注意路径不能有中文    **4.通过Eclipse创建第一个Java工程，创建Java类，通过main方法运行此程序。** | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 课程名称 | **面向对象程序设计（D）** | | | 实验项目名称 | | **实验二 简单数据类型和输出处理** | | |
| 开课教学单位及实验室 | | 计算机科学与工程学院启天楼B117 | | | | 实验日期 | |  |
| 学生姓名 |  | | 学号 | |  | 专业班级 | |  |
| 指导教师 |  | | | | 实验成绩 | |  | |
| 一、实验目的  1.熟悉和理解Java中的数据类型、运算符、表达式、程序结构等内容。  2.掌握常用的输入/输出方法，编写、调试简单的程序。  3.学会使用Java提供的类及方法。  二、实验操作方法和步骤  **1.仔细阅读下面的Java语言源程序，给出程序运行的结果。**  **public** **class** Experiment2 {  **public** **static** **void** main(String[] args) {  **char** x, y, z;  x = 'K';  y = '\\';  z = '\117';  System.*out*.print(x);  System.*out*.print(y);  System.*out*.println(z);  **int** a = 2, b = 3;  **int** c = b / 2;  **double** d = b / 2.0;  **boolean** e, f, h;  e = (a > b);  f = (a != b);  h = (a + b == 36);  System.*out*.println("c=" + c);  System.*out*.println("d=" + d);  System.*out*.println("e=" + e);  System.*out*.println("f=" + f);  System.*out*.println("h=" + h);  }  }  **运行结果如下：**  K\0  c=1  d=1.5  e=false  f=true  h=false  **2.编写程序，从键盘上输入两个整数a和b（假设a=2，b=20），然后将a和b 之间所有的奇数输出显示出来（每行显示5个数）。**  代码如下：  import java.util.Scanner;  public class demo2 {  public static void main(String[] args) {  Scanner scanner = new Scanner(System.*in*);  System.*out*.println("请输入两个整数a和b：");  int a = scanner.nextInt();  int b = scanner.nextInt();  *//for 循环 a-b之间*  for (int i = a; i <= b; i++) {  *//判断如果 当没有余数时，表示偶数，有余数表示是基数*  if (i % 2 != 0) {  *//满足输出*  System.*out*.print(i + " ");  *//判断每隔五行输出*  if ((i - a + 1) % 5 == 0) {  *//满足换行*  System.*out*.println();  }  }  }  }  }  运行图如下： | | | | | | | | |
| **3.编程输出九九乘法表**  public class demo3 {  public static void main(String[] args) {  for (int i = 1; i <= 9 ; i++) {  for (int j = 1 ; j<= i ; j++){  System.*out*.print(j+" \* " + i + " = " + (i\*j)+ " ");  }  System.*out*.println();  }  }  }  运行图如下： | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 课程名称 | **面向对象程序设计（D）** | | | 实验项目名称 | | **实验三 流程控制语句** | | |
| 开课教学单位及实验室 | | 计算机科学与工程学院  启天楼B123 | | | | 实验日期 | |  |
| 学生姓名 |  | | 学号 | |  | 专业班级 | |  |
| 指导教师 |  | | | | 实验成绩 | |  | |
| 一、实验目的  1.熟练使用各类型的运算符操作  2.熟练使用选择和循环结构进行程序设计  二、实验操作方法和步骤  **1.编程求1+2+3+…+1000的结果。**  **代码如下：**  public class demo1 {  public static void main(String[] args) {  *//定义一个sum变量来计算总和*  int sum = 0 ;  *//通过for循环来实现1- 1000 的和*  for (int i = 1 ; i<=1000;i++){  sum += i ; *//累加*  }  System.*out*.println("1-1000的和是："+sum);  }  }  **运行结果如下：**    **2.编程找到所有的水仙花数（三位数，各位置上数字的三次方之和等于该数字）。**  public class demo2 {  public static int fun(int param) {  int a,b,c;  int d = param;  a = param % 10;  b= param % 100 / 10;  c = param /100;  if ((a\*a\*a + b\*b\*b + c\*c\*c) == d) {  return 1;  }  return 0;  }  public static void main(String[] args) {  for (int i =100 ; i<=999;i++) {  if (*fun*(i) == 1) {  System.*out*.println(i + "是一个水仙花数");  }  }  }  }  **运行结果如下：**    **3.编写程序，显示输出2~100中的素数，每行显示5个数。**  public class demo3 {  public static void main(String[] args) {  int count = 0;  *//遍历2-100*  for (int i = 2; i <= 100; i++) {  *//判断如果i满足的前提下（也就是素数）。加多一个空格字符*  if (*isPrime*(i)) {  System.*out*.print(i + " ");  *//每隔五个换行*  count++;  if (count % 5 == 0) {  System.*out*.println();  }  }  }  }  public static boolean isPrime(int num) {  *//这段代码是为了判断传过来的num是不是质数*  *//如果是质数，则返回否，跳出循环，如果它是，进行循环*  *//那它就是素数*  for (int i = 2; i <= Math.*sqrt*(num); i++) {  if (num % i == 0) {  return false;  }  }  return true;  }  }  代码运行图如下： | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 课程名称 | **面向对象程序设计（D）** | | | 实验项目名称 | | **实验四 数组处理** | | |
| 开课教学单位及实验室 | | 计算机科学与工程学院启天楼B123 | | | | 实验日期 | |  |
| 学生姓名 |  | | 学号 | |  | 专业班级 | |  |
| 指导教师 |  | | | | 实验成绩 | |  | |
| 一、实验目的  1.理解数组的定义、初始化、赋值。  2.掌握数组的运用。  二、实验操作方法和步骤  **1.定义一个包含10个整数元素的数组，初始值由用户给定。找出数组中的最大数并连同下标一起输出。**  import java.util.Scanner;  public class demo1 {  public static void main(String[] args) {  Scanner scanner = new Scanner(System.*in*);  int[] arr = new int[10];  System.*out*.println("请输入10个整数：");  *//遍历输入10个*  for (int i = 0; i < 10; i++) {  arr[i] = scanner.nextInt();  }  int maxNum = arr[0];  int maxIndex = 0;  *//遍历最大下标*  for (int i = 1; i < 10; i++) {  if (arr[i] > maxNum) {  maxNum = arr[i];  maxIndex = i;  }  }  System.*out*.println("最大数是：" + maxNum + "，下标是：" + maxIndex);  }  }    **2.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，计算并显示输出成绩总和，平均成绩，最小成绩，最大成绩。**  public class demo2 {  public static void main(String[] args) {  int[] grades = {60, 85, 98, 99, 100};  int sum = 0;  int min = grades[0];  int max = grades[0];  for (int grade : grades) {  sum += grade;  if (grade < min) {  min = grade;  }  if (grade > max) {  max = grade;  }  }  *//平均成绩*  double average = (double) sum / grades.length;  System.*out*.println("成绩总和： " + sum);  System.*out*.println("平均成绩： " + average);  System.*out*.println("最小成绩： " + min);  System.*out*.println("最大成绩： " + max);  }  }  运行图如下：    **3.定义一个长度为10的整型数组，循环输入10个整数。 然后将输入一个整数，查找此整数，找到后输出下标，没找到给出提示。**  import java.util.Scanner;  public class demo3 {  public static void main(String[] args) {  int[] arr = new int[10];  Scanner scanner = new Scanner(System.*in*);  System.*out*.println("请输入10个整数：");  for (int i = 0; i < 10; i++) {  arr[i] = scanner.nextInt();  }  System.*out*.println("请输入要查找的整数：");  int target = scanner.nextInt();  boolean found = false;  for (int i = 0; i < 10; i++) {  if (arr[i] == target) {  System.*out*.println("找到整数，下标为：" + i);  found = true;  break;  }  }  if (!found) {  System.*out*.println("未找到该整数");  }  }  }  运行图如下： | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 课程名称 | **面向对象程序设计（D）** | | | 实验项目名称 | | **实验五 Java继承与多态** | | |
| 开课教学单位及实验室 | | 计算机科学与工程学院启天楼B118 | | | | 实验日期 | |  |
| 学生姓名 |  | | 学号 | |  | 专业班级 | |  |
| 指导教师 |  | | | | 实验成绩 | |  | |
| 一、实验目的  1.理解 Java 语言是如何体现面向对象编程基本思想的。  2.掌握类的继承与多态。  二、实验操作方法和步骤  **1.按要求编写一个Java应用程序：**  **（1）定义一个类Rectangle，描述一个矩形，包含有长、宽两种属性（用length和width表示）和计算面积的方法（方面名定义为area）。**  **代码如下：**  class Rectangle{ *//定义一个Rectangle 矩形类*  double length; *// 长*  double width; *// 宽*  public Rectangle(double length, double width) {  this.length = length;  this.width = width;  }  public double area() { *//计算矩形面积*  return length \* width;  }  }   1. **编写一个类Cuboid，继承自矩形类，同时该类描述长方体，具有长、宽、高（height）属性和计算体积的方法（方法名为volume）。**   class Cuboid extends Rectangle{ *//定义一个长方体继承矩形*  double height; *//定义一个高*  public Cuboid(double length, double width , double height) {  super(length, width); *//父类调用*  this.height = height;  }  public double volume() { *// 计算长方体体积的方法*  return area() \* height; *// 面积乘高等于体积*  }  }  **（3）编写一个测试类（类名为Test），对以上两个类进行测试，创建一个长方体c1，定义其长、宽、高分别为3.1、4.2,和5.3，输出其底面积和体积。**  public class Test { *//测试类*  public static void main(String[] args) {  Cuboid c1 = new Cuboid(3.1, 4.2, 5.3);  *// 输出底面积和体积*  System.*out*.println("底面积： " + c1.area());  System.*out*.println("体积： " + c1.volume());  }  }  运行结果如图所示：    **2.先创建一个Point类，然后定义Triangle类。在Triangle类中定义三个Point的实体来表示一个三角形的三个顶点，再定义一个方法setPoints对这三个点进行初始化，然后定义两个方法perimeter和area求三角形的周长、面积。在main()中创建一个Triangle类的对象，求给定三个点的三角形的周长、面积。**  package text5;  class Point { *//定义Point类，用于表示坐标*  double x, y;  public Point(double x, double y) {  this.x = x;  this.y = y;  }  }  class Triangle { *//定义Triangle类*  Point p1, p2, p3;  public void setPoints(Point p1, Point p2, Point p3) {*//定义setPoints方法，用于初始化三个顶点*  this.p1 = p1;  this.p2 = p2;  this.p3 = p3;  }  public double perimeter() { *//定义perimeter方法，用于计算三角形的周长*  double side1 = Math.*sqrt*(Math.*pow*(p2.x - p1.x, 2) + Math.*pow*(p2.y - p1.y, 2));  double side2 = Math.*sqrt*(Math.*pow*(p3.x - p2.x, 2) + Math.*pow*(p3.y - p2.y, 2));  double side3 = Math.*sqrt*(Math.*pow*(p1.x - p3.x, 2) + Math.*pow*(p1.y - p3.y, 2));  return side1 + side2 + side3; *//计算周长*  }  public double area() { *//定义area方法，用于计算三角形的周长*  double side1 = Math.*sqrt*(Math.*pow*(p2.x - p1.x, 2) + Math.*pow*(p2.y - p1.y, 2));  double side2 = Math.*sqrt*(Math.*pow*(p3.x - p2.x, 2) + Math.*pow*(p3.y - p2.y, 2));  double side3 = Math.*sqrt*(Math.*pow*(p1.x - p3.x, 2) + Math.*pow*(p1.y - p3.y, 2));  double s = (side1 + side2 + side3) / 2; //计算半周长  return Math.*sqrt*(s \* (s - side1) \* (s - side2) \* (s - side3));*//计算面积 海伦公式*  }  }  public class Main {  public static void main(String[] args) {  *//创建对象*  Triangle triangle = new Triangle();  *//初始化三个点*  Point p1 = new Point(0, 0);  Point p2 = new Point(0, 3);  Point p3 = new Point(4, 0);  triangle.setPoints(p1, p2, p3);  *//调用方法*  System.*out*.println("Perimeter: " + triangle.perimeter());  System.*out*.println("Area: " + triangle.area());  }  }  运行结果： | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 课程名称 | **面向对象程序设计（D）** | | | 实验项目名称 | | **实验六 Java抽象类与接口** | | |
| 开课教学单位及实验室 | | 计算机科学与工程学院启天楼B118 | | | | 实验日期 | |  |
| 学生姓名 |  | | 学号 | |  | 专业班级 | |  |
| 指导教师 |  | | | | 实验成绩 | |  | |
| 一、实验目的  1.熟悉抽象类的定义与使用。  2.熟悉接口的定义与使用。  二、实验操作方法和步骤  **1.定义一个抽象类，并设计两个子类继承该抽象类，演示它们的使用。**  **具体，定义一个类Vehicle并将它声明为抽象类。在类Vehicle中声明一个InfoOfWheels方法，返回值为字符串类型。创建两个类Car和Motorbike从Vehicle类继承而来，并在这两个类中实现InfoOfWheels方法。在Car类中，应当返回“四轮车”信息；而在Motorbike类中，应当返回“双轮车”信息。创建另一个带main方法的测试类Test，在该类中创建Car和Motorbike的实例，并在控制台分别显示它们返回的信息。**  **代码如下：**  *// 定义抽象类Vehicle 使用 abstract 声明为抽象类*  abstract class Vehicle {  *// 声明InfoOfWheels方法*  public abstract String InfoOfWheels(); *//返回值带上String*  }  *// 创建子类Car，继承自Vehicle类*  class Car extends Vehicle {  *// 实现InfoOfWheels方法*  @Override  public String InfoOfWheels() {  return "四轮车";  }  }  *// 创建子类Motorbike，继承自Vehicle类*  class Motorbike extends Vehicle {  *// 实现InfoOfWheels方法*  @Override  public String InfoOfWheels() {  return "双轮车";  }  }  *// 创建测试类Test*  public class Test {  public static void main(String[] args) {  *// 创建Car实例*  Car car = new Car();  *// 在控制台显示Car返回的信息*  System.*out*.println(car.InfoOfWheels());  *// 创建Motorbike实例*  Motorbike motorbike = new Motorbike();  *// 在控制台显示Motorbike返回的信息*  System.*out*.println(motorbike.InfoOfWheels());  }  }  **运行图如下：**    **2.设计一个接口, 并设计一个实现类实现该接口，演示它们的使用。**  **具体，创建一个名称为Person的接口，在接口中定义两个方法sayHello()和sayBye()。然后，创建两个实现了Person接口的类Chinese和American。在Chinese类中，两个方法分别输出“你好，见到你很高兴！”和“再见！”；在American类中，两个方法分别输出“Hi，nice to meet you!”和“Goodbye！”。另外再创建一个带有main()方法的InterfaceDemo类，在main()中声明Person类型的两个引用c和a，并分别为它们创建Chinese和American类的实例，然后依次访问它们的sayHello()和sayBye()方法，查看最后的输出结果。**  **代码如下：**  *//第一步：创建一个名为Person的接口，并在接口中定义两个方法sayHello()和sayBye()*  public interface Person {  void sayHello();  void sayBye();  }  *//第二步：①创建两个实现了Person接口的类Chinese和American*  *//在Chinese类中，两个方法分别输出“你好，见到你很高兴！”和“再见！”；*  public class Chinese implements Person { *//继承Person接口 引入两个方法*  @Override  public void sayHello() { *//继承于Person sayHello 方法*  System.*out*.println("你好，很高心认识你！");  }  @Override  public void sayBye() { *//继承于Person sayBye 方法*  System.*out*.println("再见！");  }  }  *//②在American类中，两个方法分别输出“Hi，nice to meet you!”和“Goodbye！”*  public class American implements Person { *//继承Person接口 引入两个方法*  @Override  public void sayHello() { *//继承于Person sayHello 方法*  System.*out*.println("Hi，nice to meet you!");  }  @Override  public void sayBye() { *//继承于Person sayBye 方法*  System.*out*.println("Goodbye！");  }  }  *//第三步：另外再创建一个带有main()方法的InterfaceDemo类，在main()中声明Person类型的两个引用c和a*  public class InterfaceDemo {  public static void main(String[] args) {  *//创建Chinese和American类的实例*  Person c = new Chinese();  Person a = new American();  *//调用 Chinese 中的两个方法*  c.sayHello();  c.sayBye();  *//调用American 中的两个方法*  a.sayHello();  a.sayBye();  }  }  运行图如下： | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 课程名称 | **面向对象程序设计（D）** | | | 实验项目名称 | | **实验七 字符串处理** | | |
| 开课教学单位及实验室 | | 计算机科学与工程学院启天楼B118 | | | | 实验日期 | |  |
| 学生姓名 |  | | 学号 | |  | 专业班级 | |  |
| 指导教师 |  | | | | 实验成绩 | |  | |
| 一、实验目的  理解和掌握字符串的初始化以及处理。  二、实验操作方法和步骤  **1.编写程序将“jdk”全部变为大写并输出到屏幕，截取子串“DK”并输出到屏幕。**  **代码如下：**  public class demo1 {  public static void main(String[] args) {  String str = "JDK";*//定义字符串*  String upStr = str.toUpperCase(); *//调用toUpperCase 小转大 ， toLowerCase 大转小*  String subStr = upStr.substring(1,3); *// 字符串截取，从第一个元素开始。到第三个元素*  System.*out*.println(upStr); *//输出结果*  System.*out*.println(subStr);  }  }  运行图如下    **2.从键盘录入一个字符串，统计该串中有大写字母、小写字母、数字各有多少个。比如：Hello12345World大写：2个 小写：8个数字：5个。**  **代码如下：**  import java.util.Scanner; *// 导入Scanner类，用于获取用户输入*  public class demo2 {  public static void main(String[] args) {  Scanner scanner = new Scanner(System.*in*); *//Scanner对象，用于读取用户输入*  System.*out*.println("请输入一个字符串："); *// 提示用户输入字符串*  String input = scanner.nextLine(); *// 读取用户输入的字符串*  int big = 0; *// 初始化大写字母计数器*  int small = 0; *// 初始化小写字母计数器*  int mumber = 0; *// 初始化数字计数器*  for (int i = 0; i < input.length(); i++) { *// 遍历输入字符串的每一个字符*  char ch = input.charAt(i); *// 获取当前字符*  if (Character.*isUpperCase*(ch)) { *// 判断当前字符是否为大写字母*  big++; *// 如果是大写字母，增加大写字母计数器的值*  } else if (Character.*isLowerCase*(ch)) { *// 判断当前字符是否为小写字母*  small++; *// 如果是小写字母，增加小写字母计数器的值*  } else if (Character.*isDigit*(ch)) { *// 判断当前字符是否为数字*  mumber++; *// 如果是数字，增加数字计数器的值*  }  }  System.*out*.println("大写字母个数：" + big); *// 输出大写字母的数量*  System.*out*.println("小写字母个数：" + small); *// 输出小写字母的数量*  System.*out*.println("数字个数：" + mumber); *// 输出数字的数量*  }  }  运行图如下：    **3.编写一个程序，将下面的一段文本中的各个单词的字母顺序翻转**  **“To be or not to be"，将变成"oT eb ro ton ot eb"。**  **代码如下：**  *//首先，我们需要将给定的字符串按照空格分割成单词。*  *//然后，对每个单词进行反转。*  *//最后，将反转后的单词用空格连接起来。*  public class demo3 {  public static void main(String[] args) {  String input = "To be or not to be";  System.*out*.println("input："+input);  String result = *modify*(input); *//把字符串参数传给 下面定义的modify方法中*  System.*out*.println("改变后："+result);  }  public static String modify(String input) {  String[] words = input.split(" ");*// 将输入字符串按照空格分割成单词*  StringBuilder result = new StringBuilder(); *// 定义一个StringBuilder对象*  *// 遍历每个单词*  for (int i = 0; i < words.length; i++) { *//分割了多少个 就进行多少此循环*  String word = words[i]; *//每次循环都用一个新的变量word 来接受*  *// reverse对单词进行反转 toString将对象转成字符串 并用一个新的字符串来接收*  String reversedWord = new StringBuilder(word).reverse().toString();  result.append(reversedWord);*// 将反转后的单词添加到结果中*  *//反转后因为前面通过空格截取了字符串，现在反转过来的，没有用空格隔开*  if (i < words.length - 1) { *//判断不是最后一个单词，添加一个空格*  result.append(" ");  }  }  return result.toString(); *//然后返回一个字符串到测试类中*  }  }  运行图如下：    总结：通过学习字符串初始化和处理，我掌握了如何更高效地编写代码。例如，使用字面量创建字符串、使用连接操作符拼接字符串等方法，都大大提高了我的编程效率；并且，丰富的字符串操作方法：如字符串的长度计算、子串的提取、字符串的替换、大小写转换等。这些方法使我能够更方便地对字符串进行处理，提高了我的编程能力；拓宽知识面：学习字符串初始化和处理，使我对计算机科学领域的知识有了更深入的了解。这不仅有助于我在编程方面的成长，还能为我日后的学习和发展奠定坚实的基础。 | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 课程名称 | **面向对象程序设计（D）** | | | 实验项目名称 | | **实验八 数据流和文件读写** | | |
| 开课教学单位及实验室 | | 计算机科学与工程学院启天楼B123 | | | | 实验日期 | |  |
| 学生姓名 |  | | 学号 | |  | 专业班级 | |  |
| 指导教师 |  | | | | 实验成绩 | |  | |
| 一、实验目的  1. 熟悉Java的字节输入输出流。  2. 熟悉Java的字符输入输出流。  二、实验操作方法和步骤  **1.从键盘上输入10个整数，写入到一个文件number.txt中，然后从此文件读出数据，显示在屏幕上。**  import java.io.\*;  import java.util.Scanner;  public class demo1 {  public static void main(String[] args) {  *// 创建Scanner对象用于接收键盘输入*  Scanner scanner = new Scanner(System.*in*);  *// 创建FileWriter对象用于写入文件*  FileWriter fileWriter = null;  *// 创建FileReader对象用于读取文件*  FileReader fileReader = null;  *// 创建BufferedReader对象用于缓冲读取文件*  BufferedReader bufferedReader = null;  try {  *// 创建FileWriter对象，将数据写入到number.txt文件中*  fileWriter = new FileWriter("number.txt");  *// 循环10次，接收用户输入的整数并写入文件*  for (int i = 0; i < 10; i++) {  System.*out*.print("请输入第" + (i + 1) + "个整数：");  int num = scanner.nextInt();  fileWriter.write(num + " "); *//写入number。tet*  }  fileWriter.flush(); *// 刷新缓冲区，确保数据写入文件*  *// 创建FileReader对象，从number.txt文件中读取数据*  fileReader = new FileReader("number.txt");  *// 创建BufferedReader对象，缓冲读取文件*  bufferedReader = new BufferedReader(fileReader);  *// 读取文件内容并显示在屏幕上*  System.*out*.println("从文件中读取的数据：");  String line;  while ((line = bufferedReader.readLine()) != null) {  System.*out*.println(line);  }  } catch (IOException e) {  e.printStackTrace();  } finally {  *// 关闭资源*  try {  if (fileWriter != null) {  fileWriter.close();  }  if (bufferedReader != null) {  bufferedReader.close();  }  if (fileReader != null) {  fileReader.close();  }  } catch (IOException e) {  e.printStackTrace();  }  }  }  }  运行图如下：      **2.从文件1中读取数据，修改数据的内容，然后存入文件2中，再从文件2中把数据读取出来显示在屏幕上。**  第一步：首先先创建一个file1.txt 在里面输入全是小写的aaaa；和创建一个file2.txt的空文本，用来接收file1.txt 修改过后的数据    import text8.\*;  import java.io.\*;  public class demo2 {  public static void main(String[] args) {  String inputFile = "file1.txt";  String outputFile = "file2.txt";  try {  *// 读取文件1中的数据*  BufferedReader reader = new BufferedReader(new FileReader(inputFile));  String line;  StringBuilder content = new StringBuilder();  while ((line = reader.readLine()) != null) {  *// 修改数据内容，这里只是简单地将每行文本转换为大写*  content.append(line.toUpperCase()).append(System.*lineSeparator*());  }  reader.close();  *// 将修改后的数据存入文件2中*  BufferedWriter writer = new BufferedWriter(new FileWriter(outputFile));  writer.write(content.toString());  writer.close();  *// 从文件2中读取数据并显示在屏幕上*  BufferedReader reader2 = new BufferedReader(new FileReader(outputFile));  System.*out*.println("文件2中的数据：");  while ((line = reader2.readLine()) != null) {  System.*out*.println(line);  }  reader2.close();  } catch (IOException e) {  e.printStackTrace();  }  }  }  运行结果如下：    file2.txt内容如下：（ 原本为空文本，执行后变成大写） | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 课程名称 | **面向对象程序设计（D）** | | | 实验项目名称 | | **实验九 用户界面设计** | | |
| 开课教学单位及实验室 | | 计算机科学与工程学院启天楼B118 | | | | 实验日期 | |  |
| 学生姓名 |  | | 学号 | |  | 专业班级 | |  |
| 指导教师 |  | | | | 实验成绩 | |  | |
| 一、实验目的  1.学会创建窗体  2.利用窗体面板设置不同的布局  3.给控件添加事件并对事件进行相应处理  二、实验操作方法和步骤  **1.创建窗体JFrame的一个对象并添加多个按钮**  **如：制作一个数字界面如下所示**    代码如下：  import javax.swing.\*;  import java.awt.\*;  public class demo1 extends JFrame {  public demo1() {  *// 设置窗体标题*  setTitle("网格布局DEMO");  *// 设置窗体大小*  setSize(400, 300);  *// 设置窗体关闭时的操作 关闭后，后台不再运行（释放资源）*  *//默认的是DISPOSE\_ON\_CLOSE 关闭后，程序还再后台运行*  setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  *// 因为效果图是 5行4列，使用表单最为合适*  setLayout(new GridLayout(5, 4));  *// 添加0-19的按钮 循环遍历*  for (int i = 0; i <= 19; i++) {  *//实例化btn 并将i带入button中*  JButton button = new JButton(String.*valueOf*(i));  *//遍历后，添加到窗口中*  add(button);  }  *//显示黄口*  setVisible(true);  }  public static void main(String[] args) {  *// 创建并实例化对象demo1*  new demo1();  }  }  效果图如下图：    **2.给窗体添加不同的控件，如按钮、文本框、标签等**  **如：设计一个录入职工信息的程序界面，信息量包括ID(身份证)、name(姓名)、sex(性别)、birthday(生日)、home(籍贯)、address(居住地)和number(职员号)。**  **为“退出”按钮添加事件，当点击“退出”按钮时，关闭界面。**    布局分析：    代码如下（详细步骤已写到注解中）：  import javax.swing.\*;  import java.awt.\*;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  public class demo2 extends JFrame {  public demo2() {  setTitle("公司职员信息");*//标题*  setSize(500,300);*//设置demo窗口大小*  *// 第一步：实例化并创建两个画板*  Panel panel1 = new Panel();  Panel panel2 = new Panel();  *// 给两个画板添加布局，为2行1列布局*  setLayout(new GridLayout(2,1));  *// 先给定一个背景方便观看。后面可以注释掉*  *//panel1.setBackground(Color.orange);*  *// 然后将两个画板添加到demo2窗口中 并显示出来*  add(panel1);  add(panel2);  *// 第二步：上层布局 画板用表单布局 4行4 列， 并且一共是使用7个文本，7个输入本文*  panel1.setLayout(new GridLayout(4,4));  JLabel id = new JLabel("身份证");  panel1.add(id);  TextField input1 = new TextField(20);  panel1.add(input1);  JLabel name = new JLabel("姓名");  panel1.add(name);  TextField input2 = new TextField(20);  panel1.add(input2);  JLabel sex = new JLabel("性别");  panel1.add(sex);  TextField input3 = new TextField(20);  panel1.add(input3);  JLabel birth = new JLabel("出生日期");  panel1.add(birth);  TextField input4 = new TextField(20);  panel1.add(input4);  JLabel address = new JLabel("籍贯");  panel1.add(address);  TextField input5 = new TextField(20);  panel1.add(input5);  JLabel adrs = new JLabel("居住地");  panel1.add(adrs);  TextField input6 = new TextField(20);  panel1.add(input6);  JLabel number = new JLabel("职工号");  panel1.add(number);  TextField input7 = new TextField(20);  panel1.add(input7);  *//第三步：下层布局 表单，1行1列*  panel2.setLayout(new GridLayout(1,1));  *//创建并实例化两个button ，btn2 是需要做事件监听的*  JButton btn1 = new JButton("操作");  JButton btn2 = new JButton("退出");  *//给两个button添加到画板中，并显示出来*  panel2.add(btn1);  panel2.add(btn2);  *//第四步：new 一个监听事件*  ActionListener actionListener = new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  *//关闭，并释放资源*  System.*exit*(0);  }  };  *//第五步：给btn2 加入监听事件*  btn2.addActionListener(actionListener);  *//可以忽视 我这边是为了点击关闭按钮的时候 释放资源才做的 可以忽视*  addWindowListener(new WindowAdapter() {  @Override  public void windowClosing(WindowEvent e) {  System.*exit*(0);  }  });  *//显示窗口*  setVisible(true);  }  public static void main(String[] args) {  *//创建并实例化demo2*  new demo2();  }  }  效果图： | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 课程名称 | **面向对象程序设计（D）** | | | 实验项目名称 | | **实验十 Swing组件使用** | | |
| 开课教学单位及实验室 | | 计算机科学与工程学院启天楼B129 | | | | 实验日期 | |  |
| 学生姓名 |  | | 学号 | |  | 专业班级 | |  |
| 指导教师 |  | | | | 实验成绩 | |  | |
| 一、实验目的  1.学会使用不同的控件  2综合的使用控件进行页面的布局  二、实验操作方法和步骤  **1.使用高级的控件进行布局和创建如下界面。**  **如：制作一个注册界面**    **或者一个包含菜单组件的界面**    **2.综合使用空间容器进行界面的布局，并编程实现在多个不同的界面之间相互跳转。**  **例如，在上面图一中：**  **(1)单击“登录”按钮，能够根据输入的账号密码进行登录；如果登录失败，能够提示；如果登录成功，提示登录成功之后，能到达操作界面。**  **(2)单击“注册”按钮，登录界面消失，出现注册界面。**  **(3)单击“退出”按钮，程序退出。**  import javax.swing.\*;  import java.awt.\*;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  public class demo1 {  public static void main(String[] args) {  *//实例化窗口 注册、登录、首页*  new RegisterFrame();  LoginFrame loginFrame = new LoginFrame();  new HomeFrame();  }  }  class RegisterFrame extends JFrame{  public RegisterFrame(){  *//窗口名称*  setTitle("注册窗口");  *//设置关闭事件*  setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  setSize(225,240);  *//创建画板*  JPanel jPanel = new JPanel();  add(jPanel);  JLabel jb1 = new JLabel("请您输入账号");  TextField username = new TextField(10);  jPanel.add(jb1);  jPanel.add(username);  JLabel jb2 = new JLabel("请您输入密码");  TextField password = new TextField(10);  jPanel.add(jb2);  jPanel.add(password);  JLabel jb3 = new JLabel("输入确认密码");  TextField rpassword = new TextField(10);  jPanel.add(jb3);  jPanel.add(rpassword);  JLabel jb4 = new JLabel("请您输入姓名");  TextField name = new TextField(10);  jPanel.add(jb4);  jPanel.add(name);  JLabel jb5 = new JLabel("请您选择部门");  *//创建下拉框*  JComboBox<String> select = new JComboBox<>();  *// 添加选项到下拉框*  select.addItem("财务部");  select.addItem("会计部");  select.addItem("技术部");  jPanel.add(jb5);  jPanel.add(select);  JButton btn1 = new JButton("注册");  JButton btn2 = new JButton("登录");  JButton btn3 = new JButton("退出");  jPanel.add(btn1);  jPanel.add(btn2);  jPanel.add(btn3);  *//登录事件处理 点击登录的时候进入登录页面*  btn2.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  LoginFrame loginFrame = new LoginFrame();  loginFrame.setVisible(true);  setVisible(false);  }  });  *//注册按钮监听事件*  MyRegisterListener myRegisterListener = new MyRegisterListener(username,password );  btn1.addActionListener(myRegisterListener);  *//退出按钮监听事件*  Mycossing mycossing = new Mycossing();  btn3.addActionListener(mycossing);  *//因为不是第一个要打开的窗口，所以不用设置setVisible();*  }  }  class LoginFrame extends JFrame{  public LoginFrame(){  setTitle("登录窗口");  *//设置关闭事件*  setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  setSize(230,200);  *//创建画板*  JPanel jPanel = new JPanel();  add(jPanel);  JLabel jb1 = new JLabel("请您输入账号");  TextField username = new TextField(10);  jPanel.add(jb1);  jPanel.add(username);  JLabel jb2 = new JLabel("请您输入密码");  TextField password = new TextField(10);  jPanel.add(jb2);  jPanel.add(password);  *//三个按钮*  JButton zhuce = new JButton("注册");  jPanel.add(zhuce);  *//注册按钮的事件跳转 点击注册 进入注册页面*  zhuce.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  RegisterFrame registerFrame = new RegisterFrame();  registerFrame.setVisible(true);  setVisible(false);  }  });  JButton loginbtn = new JButton("登录");  *//创建登录事件*  MyLoginListener myLoginListener = new MyLoginListener(username,password,this);  loginbtn.addActionListener(myLoginListener);  JButton tuichu = new JButton("退出");  *//实例化推出按钮事件*  Mycossing mycossing = new Mycossing();  tuichu.addActionListener(mycossing);  jPanel.add(loginbtn);  jPanel.add(tuichu);  *//实例化窗口*  setVisible(true);  }  }  class HomeFrame extends JFrame{  public HomeFrame(){  setTitle("首页");  setSize(300,300);  setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  JLabel jLabel = new JLabel("我是首页");  add(jLabel);  }  }  *//登录按钮的监听事件*  class MyLoginListener implements ActionListener {  private TextField username,password;  private LoginFrame loginFrame;  MyLoginListener(TextField username, TextField password,LoginFrame loginFrame){  this.username= username;  this.password= password;  *//把窗口本身赋值进来， 用于关闭窗口使用*  this.loginFrame = loginFrame;  }  @Override  public void actionPerformed(ActionEvent e) {  if ((username.getText()).equals("admin")&&(password.getText()).equals("123456")){  JOptionPane.*showMessageDialog*(null, "登陆成功！");  *//成功后，进入首页 登录销毁，首页出现*  loginFrame.setVisible(false);  HomeFrame homeFrame = new HomeFrame();  homeFrame.setVisible(true);  }else {  JOptionPane.*showMessageDialog*(null, "账号或密码错误！");  }  }  }  *//注册按钮的监听事件*  class MyRegisterListener implements ActionListener {  private TextField username,password;  public MyRegisterListener(TextField username ,TextField password){  this.username= username;  this.password= password;  }  @Override  public void actionPerformed(ActionEvent e) {  if ((username.getText()).equals("admin")&&(password.getText()).equals("123456")){  JOptionPane.*showMessageDialog*(null, "注册成功！");  }else {  JOptionPane.*showMessageDialog*(null, "注册失败！");  }  }  }  *//退出按钮的监听事件*  class Mycossing implements ActionListener{  @Override  public void actionPerformed(ActionEvent e) {  System.*exit*(0);  }  }  **效果图如下：**  **首先运行代码、会出现登录界面、用户可以通过已知道的密码进行登录**    **也可以通过注册按钮，进入注册界面**    **登录成功提示登录成功，可以进入首页、登陆失败，则会弹出账号或密码错误**    **在上面图二中：**  **(1)当单击“添加个人信息”菜单后进入“个人信息添加”界面，添加完成后跳转到个人信息显示界面，如果不想添加可返回到图二。**  **(2)当单击“修改个人信息”菜单进入到“个人信息修改”界面，修改成功后跳转到个人信息显示界面，如果不想修改可返回到图二。**  **(3)当单击“查看个人信息”菜单进入到“个人信息显示”界面，查看完毕可返回到图二。**  **(4)当单击“删除个人信息”菜单进入到“个人信息删除”界面，删除成功后跳转到个人信息显示界面，如果不想修改可返回到图二。** | | | | | | | | |
| **布局代码如下：**  import javax.swing.\*;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  public class demo2 {  public static void main(String[] args) {  new MyMenu();  new AddMyFrame();  }  }  class MyMenu extends JFrame{  public MyMenu(){  setTitle("公式员工信息表");  setSize(350,250);  setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  *// 创建一个 JMenuBar 实例*  JMenuBar menuBar = new JMenuBar();  *// 创建一个 JMenu 实例*  JMenu fileMenu = new JMenu("员工信息处理");  JMenuItem addData = new JMenuItem("添加个人信息");  JMenuItem reData = new JMenuItem("修改个人信息");  JMenuItem ViewData = new JMenuItem("查看个人信息");  JMenuItem DeData = new JMenuItem("删除个人信息");  fileMenu.add(addData);  fileMenu.add(reData);  fileMenu.addSeparator(); *// 添加分隔符*  fileMenu.add(ViewData);  fileMenu.add(DeData);  JMenu copy = new JMenu("打印处理");  JMenu help = new JMenu("帮助");  JMenu over = new JMenu("退出");  *//添加事件*  addData.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  System.*out*.println("我是添加事件");  }  });  *//修改事件*  reData.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  System.*out*.println("我是修改事件");  }  });  *//查看事件*  ViewData.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  System.*out*.println("我是查看事件");  }  });  *//删除事件*  DeData.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  System.*out*.println("我是删除事件");  }  });  *// 将 JMenu 添加到 JMenuBar 中*  menuBar.add(fileMenu);  menuBar.add(copy);  menuBar.add(help);  menuBar.add(over);  *// 将 JMenuBar 设置为 JFrame 的菜单栏*  setJMenuBar(menuBar);  setVisible(true);  }  }  class AddMyFrame extends JFrame{  public AddMyFrame(){  setTitle("我是添加事件");  setSize(300,300);  }  }  效果图如下： | | | | | | | | |