三、程序设计

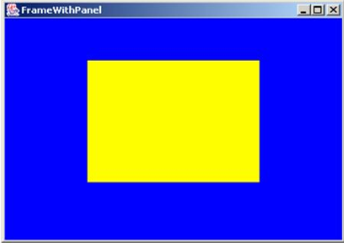
3.已有若干学生的数据，包括学号、姓名、成绩，要求输出这些学生的数据并计算出学生人数和平均成绩（学生人数和总成绩用静态数据成员表示）

public class StudentData {  
 private String studentId;  
 private String studentName;  
 private double score;  
  
 private static int studentCount ;  
 private static double totalScore ;  
  
 public StudentData(String studentId, String studentName, double score) {  
 this.studentId = studentId;  
 this.studentName = studentName;  
 this.score = score;  
  
 studentCount++;  
 totalScore += score;  
 }  
  
 public void displayStudentData() {  
 System.out.println("学号：" + studentId);  
 System.out.println("姓名：" + studentName);  
 System.out.println("成绩：" + score);  
 }  
  
 public static void displayStatistics() {  
 System.out.println("学生人数：" + studentCount);  
 System.out.println("平均成绩：" + (totalScore / studentCount));  
 }  
 public static void main(String[] args) {  
 // 创建学生对象并存储数据  
 StudentData student1 = new StudentData("001", "张三", 85.5);  
 StudentData student2 = new StudentData("002", "李四", 76.0);  
 StudentData student3 = new StudentData("003", "王五", 92.3);  
  
 student1.displayStudentData();  
 student2.displayStudentData();  
 student3.displayStudentData();  
  
 displayStatistics();  
 }  
}

4.实现一个金丝猴类，要求其继承自动物类，并实现包括金丝猴是受保护而且是有价值的动物的多重继承的功能。

// 动物类  
class Animal {  
 void speak() {  
 System.out.println("动物发出声音");  
 }  
}  
// 金丝猴类，继承自动物类  
class GoldenMonkey extends Animal {  
 void speak() {  
 System.out.println("金丝猴发出声音");  
 }  
 void isProtected() {  
 System.out.println("金丝猴是受保护的动物");  
 }  
 void isValuable() {  
 System.out.println("金丝猴是有价值的动物");  
 }  
}  
// 测试类  
public class Main {  
 public static void main(String[] args) {  
 GoldenMonkey monkey = new GoldenMonkey();  
 monkey.speak(); // 调用金丝猴类的 speak 方法  
 monkey.isProtected(); // 调用金丝猴类的 isProtected 方法  
 monkey.isValuable(); // 调用金丝猴类的 isValuable 方法  
 }  
}

实现如图所示的界面



import java.awt.\*;

import javax.swing.\*;   
public class a05 {  
 public static void main(String[] args) {  
 JFrame frame = new JFrame();  
 frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);   
 JPanel panel = new JPanel() {  
 @Override   
 protected void paintComponent(Graphics g) {  
 super.paintComponent(g);  
 g.setColor(Color.BLUE);  
 g.fillRect(0, 0, 300, 200)；  
 g.setColor(Color.YELLOW);  
 g.fillRect(75, 50, 150, 100);  
 }  
 };  
 panel.setPreferredSize(new Dimension(300, 200));   
 frame.getContentPane().add(panel);   
 frame.pack();   
 frame.setVisible(true);   
 }  
}

6.实现如图所示的界面，并实现将按钮“=”左侧两个文本框中的数相加的和显示在“=”右侧的文本框中的功能。



import javax.swing.\*;   
import java.awt.event.\*;   
public class a06 {  
 public static void main(String[] args) {  
 JFrame frame = new JFrame();  
 frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

JPanel panel = new JPanel();   
 JTextField textField1 = new JTextField(10);

JLabel plusLabel = new JLabel("+");

JTextField textField2 = new JTextField(10);  
 JButton addButton = new JButton("=");

JTextField resultField = new JTextField(10);  
 resultField.setEditable(false);   
 addButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 String text1 = textField1.getText();   
 String text2 = textField2.getText();  
 try {  
 int num1 = Integer.parseInt(text1);   
 int num2 = Integer.parseInt(text2);  
 int sum = num1 + num2;  
 resultField.setText(Integer.toString(sum));   
 } catch (NumberFormatException ex) {  
 resultField.setText("Invalid input");  
 }  
 }  
 });  
 panel.add(textField1);  
 panel.add(plusLabel);  
 panel.add(textField2);  
 panel.add(addButton);  
 panel.add(resultField);  
 frame.getContentPane().add(panel);  
 frame.pack();  
 frame.setVisible(true);  
 }  
}

7.实现通过键盘输入一个浮点数，然后将该浮点数的整数部分输出的程序。

import java.util.Scanner;  
public class a08 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.in);  
  
 System.out.print("请输入一个浮点数: ");  
 double number = scanner.nextDouble();  
  
 int integerPart = (int) number;   
  
 System.out.println("整数部分为: " + integerPart);  
  
 scanner.close();   
 }  
}

8.计算并输出一个整数各位数字之和的程序，如1234的各位数字之和为1+2+3+4。

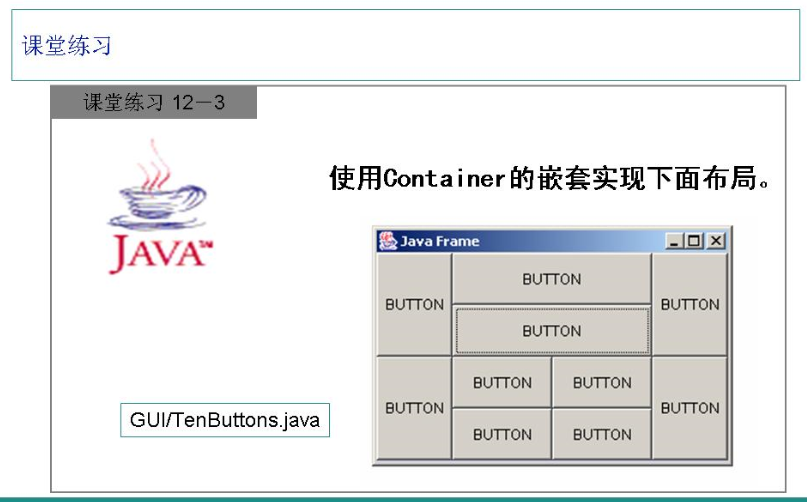
import java.util.Scanner;  
public class a09 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.in);  
 System.out.print("请输入一个整数: ");  
 int number = scanner.nextInt();  
 int sum = 0;  
 int digit;  
 while (number != 0) {  
 digit = number % 10; // 取最后一位数字  
 sum += digit; // 累加到总和中  
 number /= 10; // 去掉最后一位数字  
 }  
 System.out.println("各位数字之和为: " + sum);  
 scanner.close();  
 }  
}

9.实现字符串

String s= " SunJavaHPjavaOkjava兰理工java计通学院java计算机科学与技术java " 中统计“java”出现的次数的程序。

import java.util.Scanner;  
public class a09 {  
 public static void main(String[] args) {  
 String s = "SunJavaHPjavaOkjava 兰理工 java 计通学院 java 计算机科学与技术 java";  
 String target = "java";  
 int count = countSubstring(s, target);  
 System.out.println( "\""+target + "\" 出现的次数为: " + count);  
 }  
 public static int countSubstring(String s, String target) {  
 int count = 0;  
 int index = 0;  
 while (index != -1) {  
 index = s.indexOf(target, index); // 在字符串 s 中查找子串 target  
 if (index != -1) {  
 count++;  
 index += target.length(); // 继续从下一个位置开始查找  
 }  
 }  
 return count;  
 }  
}

10.实现如图所示的界面。



import java.awt.\*;  
  
public class Demo11 {  
 public static void main(String[] args) {  
 Frame f = new Frame("JavaFrame");  
  
 f.setLayout(new GridLayout(2,1));  
 f.setLocation(300,400);  
 f.setSize(300,200);  
  
 f.setBackground(new Color(204, 204, 255));  
 Panel p1 = new Panel(new BorderLayout());  
 Panel p2 = new Panel(new BorderLayout());  
 Panel p11 = new Panel(new GridLayout(2,1));  
 Panel p21 = new Panel(new GridLayout(2,2));  
  
 p1.add(new Button("Button"),BorderLayout.WEST);  
 p1.add(new Button("Button"),BorderLayout.EAST);  
  
 p11.add(new Button("Button"));  
 p11.add(new Button("Button"));  
 p1.add(p11,BorderLayout.CENTER);  
  
 p2.add(new Button("Button"),BorderLayout.WEST);  
 p2.add(new Button("Button"),BorderLayout.EAST);  
  
 for (int i = 0; i < 4; i++) {  
 p21.add(new Button("Button"));  
 }  
 p2.add(p21,BorderLayout.CENTER);  
 f.add(p1);  
 f.add(p2);  
 f.setVisible(true);  
 }  
}

11.15个红球和15个绿球排成一圈，从第1个球开始数，当数到第13个球时就拿出此球，然后再从下一个球开始数，当再数到第13个球时又取出此球，如此循环进行，直到仅剩15个球为止，问怎么样排法才能每次取出的球都是红球。

public class GetRedBalls5\_6 {

public static void main(String[] args){

int[] ball=new int[30];

int i, j, t;

for(i=0; i<30; i++)

ball[i]=i+1;

System.out.println("取出红球的编号为: ");

for(i=0, j=0, t=30; t!=15; i++) {

if(i==30)

i=0;

if(ball[i]!=0)

j++;

if(j==13&&ball[i]!=0) {

ball[i]=0;

System.out.print(" "+(i+1)+" ");

j=0;

t--;

}

}

}

12.请按照以下要求设计一个学术类Student和它的一个子类Undergraduate ，并进行测试。要求如下：

（1）Student类中包含姓名（字符串类型）、年龄（整型）两个属性。

（2）Student类中定义两个构造方法。一个是无参构造方法；一个接收两个参数的构造方法，两个参数分别为姓名和年龄属性赋值。

（3）Student包括以下成员方法：

void setName(String name); 设置姓名

void setAge(float grade)； 设置年龄

String getName(); 获取姓名

float getAge(); 获取年龄

Void show(): 输出姓名和年龄

（4）Undergraduate类继承Student类，并增加一个degree（学位）属性。有一个包括三个参数的构造方法，前两个参数用于给继承的name和age属性赋值，第三个参数给degree专业赋值，一个show（）方法用于打印Undergraduate的三个属性信息。

（5）在测试类中分别创建Student对象和Undergraduate对象，调用它们的show().方法。

class Student{  
 private String name;  
 private int age;  
 public Student(){  
  
 }  
 public Student(String name,int age){  
 this.name=name;  
 this.age=age;  
 }  
 public void setName(String name){  
 this.name=name;  
 }  
 public void setAge(int age){ //  
 this.age=age;  
 }  
 public String getName(){  
 return name;  
 }  
 public int getAge(){  
 return age;  
 }  
 public void show(){  
 System.out.println("学生姓名"+name);  
 System.out.println("学生年龄"+age);  
 }  
}  
class Undergraduate extends Student{  
 private String degree;  
 public Undergraduate(String name,int age,String degree){  
 super(name,age);  
 this.degree=degree;  
 }  
 public void show(){  
 super.show();  
 System.out.println("学生学位"+degree);  
 }  
}  
public class test12 {  
 public static void main(String[] args){  
 Student stu1=new Student("zhangsan",25);  
 stu1.show();  
 System.out.println();  
 Undergraduate stu2=new Undergraduate("lisi",26,"buoshi");  
 stu2.show();  
 }  
}

14.设有一个动物声音模拟器，希望模拟器可以模拟许多动物的叫声，要求如下：

（1）编写接口Animal。Animal接口有两个抽象方法cry()和getAnimalName()，要求实现该接口的各种具体动物类给出自己的叫声和种类名称。

（2）编写模拟器类Simulator。该类有一个playSound(Animal animal)方法，该方法的参数是Animal类型，即参数animal可以调用实现Animal接口类重写的cry()方法播放具体动物的声音，调用重写的getAnimalName()方法显示动物种类的名称。

（3）编写实现Animal接口的Dog类和Cat类。

（4）编写主类MainClass,在主类的main方法中至少包含如下代码：

Simulator simulator=new Simulator()

simulator.playSound(new dog())

simulator.playSound(new cat())

interface Animal {  
 void cry();  
 String getAnimalName();  
}  
// Dog 类实现 Animal 接口  
class Dog implements Animal {  
 public void cry() {  
 System.out.println("汪汪汪！");  
 }  
 public String getAnimalName() {  
 return "狗";  
 }  
}  
// Cat 类实现 Animal 接口  
class Cat implements Animal {  
 public void cry() {  
 System.out.println("喵喵喵！");  
 }  
 public String getAnimalName() {  
 return "猫";  
 }  
}  
// 模拟器类  
class Simulator {  
 public void playSound(Animal animal) {  
 animal.cry();  
 System.out.println("动物种类：" + animal.getAnimalName());  
 }  
}  
public class test13 {  
 public static void main(String[] args) {  
 Simulator simulator = new Simulator();  
 simulator.playSound(new Dog());  
 simulator.playSound(new Cat());  
 }  
}

15.使用Java的输入和输出流将一个文本文件的内容按行读出，每读出一行就顺序添加行号，并写入另一个文件中。（读取的文件路径：E:/Source.txt；写入的文件路径：E:/Object.txt）

import java.io.\*;  
public class test13 {  
 public static void main(String[] args) {  
 String inputFile = "E:/Source.txt";  
 String outFile = "E:/Object.txt";  
  
 try (BufferedReader reader = new BufferedReader(new FileReader(inputFile));  
 BufferedWriter writer = new BufferedWriter(new FileWriter(outFile))) {  
 String line;  
 int lineNumber = 1;  
 while ((line = reader.readLine()) != null) {  
 String lineWithLineNumber = lineNumber + ": " + line;  
 writer.write(lineWithLineNumber);  
 writer.newLine();  
 lineNumber++;  
 }  
 System.out.println("File line numbering completed.");  
 } catch (IOException e) {  
 System.out.println("An error occurred: " + e.getMessage());  
 }  
 }  
}

16.编写一个图形用户界面的应用程序，包括两个文本框和一个按钮，当单击按钮时，可以把一个文本框中的内容复制到另一个文本框中。

import java.awt.\*;  
import javax.swing.\*;  
import java.awt.event.\*;  
public class test16 {  
 public static void main(String[] args) {  
 JFrame frame = new JFrame("文本框复制");  
 frame.setLayout(new FlowLayout());  
 frame.setSize(300, 150);  
 frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  
 JTextField textField1 = new JTextField(20);  
 JTextField textField2 = new JTextField(20);  
 JButton button = new JButton("复制");  
 button.addActionListener(new ActionListener() {  
 public void actionPerformed(ActionEvent e) {  
 String text = textField1.getText();  
 textField2.setText(text);  
 }  
 });  
 frame.add(textField1);  
 frame.add(button);  
 frame.add(textField2);  
 frame.setVisible(true);  
 }  
}

17.请编写一个类，类名为MulThread ，该类实现Runnable接口。类中定义了含一个字符串参数的构造函数，设置线程名称；其中的run( )方法如下实现：方法中先在命令行显示“begin”+线程名，然后休眠1秒的时间，最后显示线程结束信息: "finished"+线程名。编写Application，在其中通过Runnable创建MulThread类的三个线程对象t1、t2、t3，并启动这三个线程。

class MulThread implements Runnable {  
 private String threadName;  
  
 public MulThread(String threadName) {  
 this.threadName = threadName;  
 }

public void run() {  
 System.out.println("begin " + threadName);  
 try {  
 Thread.sleep(1000); // 休眠1秒  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 System.out.println("finished " + threadName);  
 }  
}  
public class test17 {  
 public static void main(String[] args) {  
 MulThread t1 = new MulThread("Thread 1");  
 MulThread t2 = new MulThread("Thread 2");  
 MulThread t3 = new MulThread("Thread 3");  
  
 Thread thread1 = new Thread(t1);  
 Thread thread2 = new Thread(t2);  
 Thread thread3 = new Thread(t3);  
  
 thread1.start();  
 thread2.start();  
 thread3.start();  
 }  
}

18.编程实现：

（1）编写一个圆类Circle，该类拥有：

①一个成员变量Radius（私有，浮点型）； // 存放圆的半径；

②两个构造方法Circle( ) // 将半径设为0

Circle(double r ) //创建Circle对象时将半径初始化为r

③ 三个成员方法

double getArea( ) //获取圆的面积

double getPerimeter( ) //获取圆的周长

void show( ) //将圆的半径、周长、面积输出到屏幕

（2）编写一个圆柱体类Cylinder，它继承于上面的Circle类。还拥有：

①一个成员变量double hight（私有，浮点型）； // 圆柱体的高；

②构造方法 Cylinder (double r, double h ) //创建Circle对象时将半径初始化为r

③ 成员方法double getVolume( ) //获取圆柱体的体积

void showVolume( ) //将圆柱体的体积输出到屏幕

（3）编写应用程序，创建类的对象，分别设置圆的半径、圆柱体的高，计算并分别显示圆半径、圆面积、圆周长，圆柱体的体积。

class Circle{  
 private double Radius;  
 public Circle() {  
 Radius = 0;  
 }  
 public Circle(double r){  
 Radius=r;  
 }  
 public double getArea( ){  
 return Math.PI \* Radius \* Radius;  
 }  
 public double getPerimeter( ){  
 return 2 \* Math.PI \* Radius;  
 }  
 public void show( ){  
 System.out.println("半径：" + Radius);  
 System.out.println("周长：" + getPerimeter());  
 System.out.println("面积：" + getArea());  
 }  
}  
class Cylinder extends Circle {  
 private double height;  
 public Cylinder(double r, double h) {  
 super(r);  
 height = h;  
 }  
 public double getVolume() {  
 return getArea() \* height;  
 }  
 public void showVolume() {  
 System.out.println("体积：" + getVolume());  
 }  
}  
public class test18 {  
 public static void main(String[] args) {  
 Circle circle = new Circle(5);  
 circle.show();  
 System.out.println();  
 Cylinder cylinder = new Cylinder(3, 8);  
 cylinder.showVolume();  
 }  
}

19.按要求编写一个Java应用程序程序：

（1）定义一个接口CanFly，描述会飞的方法public void fly();

（2）分别定义类飞机和鸟，实现CanFly接口。

（3）定义一个测试类，测试飞机和鸟，在main方法中创建飞机对象和鸟对象，

再定义一个makeFly()方法，其中让会飞的事物飞。并在main方法中调用该方法，

让飞机和鸟起飞。

interface CanFly{  
 void fly();  
}  
class Bird implements CanFly{  
 @Override  
 public void fly() {  
 System.out.println("鸟儿飞翔！");  
 }  
}  
class Plan implements CanFly{  
 @Override  
 public void fly() {  
 System.out.println("飞机起飞！");  
 }  
}  
public class test19 {  
 public static void main(String[] args){  
 Plan p=new Plan();  
 Bird b=new Bird();  
 makeFly(p);  
 makeFly(b);  
 }  
 public static void makeFly(CanFly flyingObject){  
 flyingObject.fly();  
 }  
}

20.应用FileInputStream类，编写应用程序，从磁盘上读取一个Java程序，并将源程序代码显示在屏幕上。（被读取的文件路径为：E:/Hello.java）

import java.io.\*;  
public class test20 {  
 public static void main(String[] args) {  
 String filePath = "E:/Hello.java";   
 try (FileInputStream fis = new FileInputStream(filePath)) {  
 byte[] buffer = new byte[1024];  
 int bytesRead;  
 while ((bytesRead = fis.read(buffer)) != -1) {  
 String code = new String(buffer, 0, bytesRead);  
 System.out.print(code);  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
}

21.编写程序，在屏幕上显示带标题的窗口，并添加一个按钮。当用户单击按钮时，结束程序。

import javax.swing.\*;  
public class test21 {  
 public static void main(String[] args) {  
 JFrame frame = new JFrame("带标题的窗口");  
 frame.setSize(300, 200);  
 frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  
 JButton button = new JButton("结束程序")；  
 frame.getContentPane().add(button)  
 button.addActionListener(e -> {  
 JOptionPane.showMessageDialog(null, "程序已结束");  
 frame.dispose();  
 });  
 frame.setVisible(true);  
 }  
}