

面向对象程序设计

实验要求

目录

实验报告模板	3
实验 07 类的多态实现与测试	6
1、实验目的	6
2、实验内容：	7
实验 08 数组功能与字符串编程	8
1、实验目的	8
2、实验内容	8
实验 09 线程程序设计	10
1. 实验目的	10
2. 实验内容	10
实验 10 IO 基础操作实现	13
1. 实验目的	13
2. 实验内容	13
实验 11 异常处理功能实现	14
1. 实验目的	14
2. 实验内容	14
实验 12 事件监听 GUI 编程实现	16
1. 实验目的	16
2. 实验内容	16

实验报告模板



山西工学院
SHANXI COLLEGE OF TECHNOLOGY

实验报告

学院名称：_____

专业名称：_____

课程名称：_____

班 级：_____ 学号：_____

学生姓名：_____

指导老师：_____

年 月 日

学生姓名		学号		实验成绩	
实验项目名称					
实验地点		实验日期			
<p>一、实验目的和要求</p> <p>二、实验内容和原理</p> <p>三、程序代码（要求有注释说明）</p> <p>四、实验结果截屏图</p>					

五、实验结果与分析

注：实验报告可根据专业实际情况进行调整，封面需使用教务部统一制定的模板

实验 07 类的多态实现与测试

1、实验目的

类的多态有多种体现方式，比较多的是在程序运行期间，根据类的引用关系，自动寻找匹配的方法，参考下面的案例内容。

分析下面代码的实现过程，尝试使用内存分布图分析代码的执行过程。

```
public class Test {  
    public static void main(String[] args) {  
        Lady missLiu= new Lady();  
        missLiu.pet = new Dog2();  
        missLiu.petEnjoy();  
        Lady missWang= new Lady();  
        missWang.pet = new Bird();  
        missWang.petEnjoy();  
    }  
}  
  
class Lady{  
    Pet pet;  
    void petEnjoy(){  
        pet.enjoy();  
    }  
}  
class Pet{  
    int name;  
    void enjoy(){  
        System.out.println("宠物高兴");  
    }  
}  
class Bird extends Pet{  
    void enjoy(){  
        System.out.println("喳喳.....");  
    }  
}  
class Dog2 extends Pet{  
    void enjoy(){  
        System.out.println("汪汪.....");  
    }  
}
```



2、实验内容：

(1) 编写应用程序，具体要求如下：

- ① 声明一个抽象类 Pet，封装属性 name 和 sex，声明一个带有两个参数的构造函数，声明抽象方法 void talk() 和 void eat()；
- ② 声明一个 Dog 类继承自 Pet，封装属性 color，声明带有三个参数的构造函数，重写 talk() 和 eat() 方法；
- ③ 声明一个 Cat 类继承自 Pet，封装属性 weight，声明带有三个参数的构造函数，重写 talk() 和 eat() 方法；
- ④ 编写测试类，通过有参构造函数实例化 Dog 类对象，调用 talk() 方法和 eat() 方法；通过有参构造函数实例化 Cat 类对象，调用 talk() 方法和 eat() 方法；

(2) 按照要求编写一个 Java 应用程序：

- ① 定义一个抽象类 Person，包含抽象方法 eat()，封装属性 name、sex、age，声明包含三个参数的构造方法；
- ② 定义一个 Chinese 类，继承自 Person 类，重写父类的 eat() 方法，并定义一个自己特有的方法 shadowBoxing()；
- ③ 定义一个 English 类，继承自 Person 类，重写父类的 eat() 方法，并定义一个自己特有的方法 horseRiding()；

编写测试类，定义一个 showEat()方法，使用父类作为方法的形参，实现多态，分别调用 showEat()方法，通过强制类型转换调用各自类特有的方法；

实验 08 数组功能与字符串编程

1、实验目的

Java 语言的数组与传统的面向过程中定义的不同，它是引用数据类型。基于类的定义模式。可以是不对称结构。实现数组处理使用的控制结构需要使用.length 方式。数组定义方式为：int a[] = new int[3];表示定义一组数组；int a[][] = new int[3][];表示定义二维数组，其中二维的内容继续使用一维数组的定义方式；

2、实验内容

(1) 编程使用 Scanner 类实现下面数组数组定义与数据录入。

请输入一维值:6

85	54	19	52	55	30		
94	10	68	79	31			
49	49	98	24	66	63	26	70
36	42	88	36				
37	37	31	90	52	29	43	92
33	16	28					

其中一维是通过键盘输入数据确定，各个二维则由系统随机生成 10 以内的数控制数据的数量。要求生成数组数据为两位整数，生成数据后能够按数组格式输出。

说明：上面的定义中只有一维是输入数据，其它数据由系统生成。

(2) 定义一个 Student 类（成员变量 name, sex, age, collage）。生成 10 个同学的数组并能够录入数据并最终显示整体数据。如下方式输入数据。

请输入姓名:张三

请输入年龄:20

请输入性别:男

请输入单位:太原理工大学

请输入姓名:李四

请输入年龄:25

请输入性别:女

请输入单位:山西大学

个人信息:姓名是:张三, 年龄是:20, 性别是:男, 工作单位是:太原理工大学

个人信息:姓名是:李四, 年龄是:25, 性别是:女, 工作单位是:山西大学

程序结构如下:

```
class Student{
    private String name;
    private int age;
    private char sex;
    private String collage;
    public String getName() {
    }
    public void setName(String name) {
    }
    public int getAge() {
    }
    public void setAge(int age) {
    }
    public char getSex() {
    }
    public void setSex(char sex) {
    }
    public String getCollage() {
    }
    public void setCollage(String collage) {
    }
    public Student(String name, int age, char sex, String collage) {
    }
    public String toString() {
    }
}
```

程序需要修改 toString()方法

实验 09 线程程序设计

1. 实验目的

- (1) 线程的概念、线程的生命周期。
- (2) 多线程的编程：扩展 Thread 类与使用 Runnable 接口。
- (3) 使用多线程机制实现动画

2. 实验内容

- (1) 运行下面的程序，理解用创建 Thread 子类的方法实现多线程。

```
import java.util.*;

class Test_Thread extends Thread {

    int pauseTime;

    String name;

    public Test_Thread (int x,String n){

        pauseTime=x;

        name=n;

    }

    public void run(){

        while(true){

            try{

                system.out.println(name+"."+new Date(System.currentTimeMillis()));

                Tread.sleep(pauseTime);
```

```

        }catch(Exception e){System.out.println(e);}

    }

}

static public void main(String srgs[]){

    Test_Thread tp1=new Test_Thread (1000,"Fast");

    tp1.start();

    Test_Thread tp2=new Test_Thread (3000,"Slow");

    tp2.start();

}

}

```

(2) 运行下面的程序，理解用实现 Runnable 接口的方法实现多线程。

```

Import java.util.*;

class Test_Thread implements Runnable {

    int pauseTime;

    String name;

    public Test_Thread (int x,String n){

        pauseTime=x;

        name=n;

    }

    public void run(){

        while(true){

            try{

```

```

        system.out.println(name+":"+new Date(System.currentTimeMillis()));

        Tread.sleep(pauseTime);

    }catch(Exception e){System.out.println(e);}

    }

}

static public void main(String srgs[]){

    Thread t1=new Thread(new Test_Thread (1000,"Fast"));

    t1.start();

    Thread t2=new Thread(new Test_Thread (3000,"Slow"));

    t2.start();

}

}

```

(3) 创建简单的程序 ThreeThread. java, 该程序将创建三个线程, 每个线程应当显示它所运行的时间 (可以考虑使用 Date 类或 Calendar 类)。

实验 10 IO 基础操作实现

1. 实验目的

- (1) 掌握 Java 两种基本输入/输出流类的使用。
- (2) 掌握 Java 文件类型的使用。
- (3) 掌握文件的输入、输出方法。

2. 实验内容

- (1) 编写程序，实现从键盘输入字符串并在屏幕上输出。
- (2) 编写一个程序，求 2~200 之间的素数，并将结果保存在文件 prime.dat 中。

再从该文件中读取内容并在屏幕上显示出来。注意异常处理。

实验 11 异常处理功能实现

1. 实验目的

掌握异常的概念，以及如何定义、抛出和捕获处理异常。

2. 实验内容

(1) 做实验之前要求思考以下问题，作为预习内容；

- ① 错误和异常有何区别？查看 Java 异常类的关系图。
- ② 异常是如何抛出、捕获和处理的？
- ③ Java 捕获和处理异常的结构式怎样的？
- ④ Try 语句如何嵌套？Throw 语句有何作用？Finally 程序块的作用是什么？

么？

(2) 运行下面的程序，理解异常的抛出、捕获与处理。

```
Import java.io.*;

public class Test{

    public static void main(String args[]){

        for(int i=0;i<4;i++){

            int k;

            try{

                switch(i){

                    case 0: //divided by zero

                        int zero=0;
```

```
        k=911 //zero

        break;

    case 1://null pointer

        int b[]=null;

        k=b[0];

        break;

    case 2; //array index out of bound

        int c[]=new int[2];

        k=c[9];

        break;

    case 3: //string index out of bound

        char ch="abc".charAt(99);

        break;

    }

} catch(Exception e){

    System.out.println("\nTestcase#" + i + "\n");

    System.out.println(e);

}

}

}
```

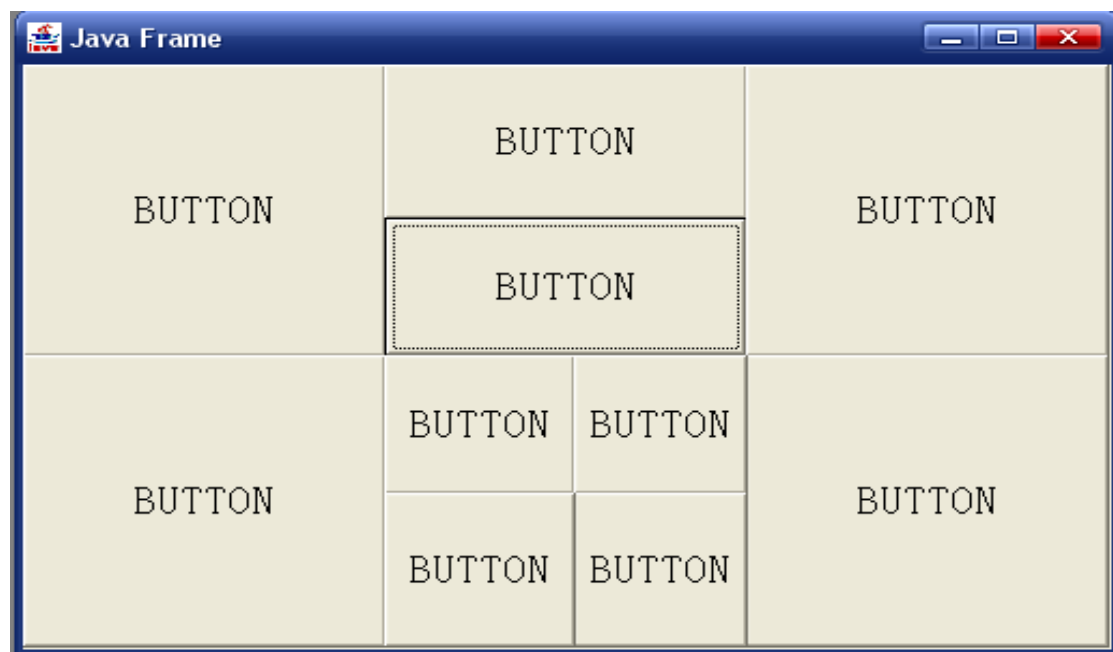
实验 12 事件监听 GUI 编程实现

1. 实验目的

掌握基本组件类的生成，掌握布局控制的方法。编程实现基础的事件监听功能

2. 实验内容

(1) 使用 Container 的嵌套实现下面的布局。



(2) 设计一窗口，内含一个按钮。开始运行时，按钮显示 “Click Me” 字样，当按钮按下时，按钮显示为 “Click Me Again” 字样，再按一次，则按钮显示 “Click Me” 字样，依次循环。