《Java语言程序设计》课程实验报告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 专业名称 | 计算机科学与技术 | 年级 | 2017 | 班级 | 计工本1704 |
| 学生姓名 | 付泽坤 | 指导老师 | 李焱 | 时间 | 2019.05.16 |

|  |  |
| --- | --- |
| 实验名称 | 继承与多态 |
| 实  验  目  的  及  要  求 | 目的：  了解熟悉顺序Java程序设计的形式，编写完整Java程序。  要求：   * 掌握类之间的关系。 * 掌握类的继承设计。 * 掌握用多个类解决问题。 * 会用Java编写完整的程序。 |
| 实  验  环  境 | Microsoft Windows 7 with SP1专业版（简体中文）32位  JDK 1.8  Eclipse 2017  或者Jcreator |
| 实  验  内  容 | 请按照要求编写出完整程序   * 第10章编程练习题: 10.7、10.10、10.12、10.23、10.28 * 第11章编程练习题: 11.1、11.2 |
| 实  验  步  骤  或  实  验  方  案 | **package cn.work.thirteen;**  **import sun.security.util.Password;**  **import java.util.Scanner;**  **public class \_13 {**  **public static void \_10\_7()**  **{**  **Account[] accounts = new Account[10];**  **for(int i = 0; i < 10; i++)**  **{**  **accounts[i] = new Account(i + 1);**  **}**  **while(true)**  **{**  **Scanner cin = new Scanner(System.in);**  **int id = cin.nextInt();**  **while(true)**  **{**  **System.out.println("Main menu:");**  **System.out.println("1. check balance");**  **System.out.println("2. withdraw");**  **System.out.println("3. deposit");**  **System.out.println("4. exit");**  **System.out.print("Enter a choice : ");**  **int tmp = cin.nextInt();**  **if(tmp == 1)**  **{**  **System.out.println(accounts[id].getMoney());**  **}**  **else if(tmp == 2)**  **{**  **int t = cin.nextInt();**  **accounts[id].withdraw(t);**  **}**  **else if(tmp == 3)**  **{**  **int t = cin.nextInt();**  **accounts[id].deposit(t);**  **}**  **else break;**  **}**  **}**  **}**  **public static void main(String[] args) {**  **//\_10\_7();**  **MyStringBuilder2 s1 =**  **new MyStringBuilder2(new char[]{'a', 'b', 'c', 'd', 'p'});**  **MyStringBuilder2 s2 = new MyStringBuilder2("xyz");**  **System.out.println(s1.length());**  **System.out.println(s1.charAt(3));**    **s2.append(123456789);**    **System.out.println(s2.toString());**  **System.out.println(s1.substring(1, 2));**    **s1.reverse();**  **System.out.println(s1.substring(2));**  **System.out.println(s1.toUpperCase().toString());**  **}**  **}**  **class MyStringBuilder2 {**  **private int size = 0;**  **private int capacity = 0;**  **private char[] buffer;**  **public MyStringBuilder2() {**  **}**  **public MyStringBuilder2(char[] chars) {**  **buffer = new char[chars.length];**  **size = capacity = chars.length;**  **System.arraycopy(chars, 0, buffer, 0, chars.length);**  **}**  **public MyStringBuilder2(String s) {**  **capacity = size = s.length();**  **buffer = new char[capacity];**  **for (int i = 0; i < s.length(); i++)**  **buffer[i] = s.charAt(i);**  **}**  **private void increaseCapacity(int newCapacity) {**  **char[] temp = new char[newCapacity];**  **System.arraycopy(buffer, 0, temp, 0, size);**  **capacity = newCapacity;**  **buffer = temp;**  **}**  **public int length() {**  **return size;**  **}**  **public MyStringBuilder2 substring(int begin, int end) {**  **char[] result = new char[end - begin];**  **for (int i = 0; i < result.length; i++)**  **result[i] = buffer[begin + i];**  **return new MyStringBuilder2(result);**  **}**  **// Exercise 8.30:**  **public MyStringBuilder2 insert(int offset, MyStringBuilder2 s) {**  **if (capacity < size + s.length()) {**  **increaseCapacity(2 \* (size + s.length()));**  **}**  **String temp = s.toString();**  **for (int i = 0; i < temp.length(); i++)**  **buffer[size - 1 + temp.length() - i] = buffer[size - 1 - i];**  **for (int i = 0; i < temp.length() ; i++)**  **buffer[i + offset] = temp.charAt(i);**  **size += temp.length();**  **return this;**  **}**  **public MyStringBuilder2 reverse() {**  **for (int i = 0; i < size / 2; i++) {**  **char temp = buffer[i];**  **buffer[i] = buffer[size - 1 - i];**  **buffer[size - 1 - i] = temp;**  **}**  **return this;**  **}**  **public MyStringBuilder2 substring(int begin) {**  **return substring(begin, size);**  **}**  **public MyStringBuilder2 toUpperCase() {**  **for (int i = 0; i < size; i++)**  **buffer[i] = Character.toUpperCase(buffer[i]);**  **return this;**  **}**  **public void append(int i) {**  **}**  **}**  **class MyString2**  **{**  **private char[] chars;**  **MyString2(char[] chars)**  **{**  **System.arraycopy(chars, 0, this.chars, 0, chars.length);**  **}**  **public MyString2 subString(int begin)**  **{**  **char[] tmp = new char[this.chars.length];**  **int len = 0;**  **for(int i = begin; i < this.chars.length; i++)**  **{**  **tmp[len++] = chars[i];**  **}**  **return new MyString2(tmp);**  **}**  **public MyString2 upletter()**  **{**  **char[] tmp = new char[chars.length];**  **for(int i = 0; i < tmp.length; i++)**  **{**  **tmp[i] = Character.toUpperCase(chars[i]);**  **}**  **return new MyString2(tmp);**  **}**  **}**  **class Triangle2D**  **{**  **private Mypoint p1;**  **private Mypoint p2;**  **private Mypoint p3;**  **public Triangle2D(Mypoint p1, Mypoint p2, Mypoint p3) {**  **this.p1 = p1;**  **this.p2 = p2;**  **this.p3 = p3;**  **}**  **Triangle2D()**  **{**  **this(new Mypoint(0, 0), new Mypoint(1, 1), new Mypoint(2, 5));**  **}**  **public double getDistance(Mypoint x, Mypoint y)**  **{**  **return Math.sqrt((x.getX() - y.getX()) \* (x.getX() - y.getX()) + (x.getY() - y.getY()) \* (x.getY() - y.getY()));**  **}**  **public double getArea()**  **{**  **double a = getDistance(p1, p2);**  **double b = getDistance(p1, p3);**  **double c = getDistance(p2, p3);**  **double p = (a + b + c) / 2.0;**  **return Math.sqrt(p\*(p - a)\*(p - b)\*(p - c));**  **}**  **public double getPerimeter()**  **{**  **}**  **}**  **class Mypoint**  **{**  **private int x;**  **private int y;**  **public Mypoint(int x, int y) {**  **this.x = x;**  **this.y = y;**  **}**  **public int getX() {**  **return x;**  **}**  **public void setX(int x) {**  **this.x = x;**  **}**  **public int getY() {**  **return y;**  **}**  **public void setY(int y) {**  **this.y = y;**  **}**  **}**  **class queue {**  **private int size;**  **private int[] que;**  **private int l;**  **private int r;**  **queue(int size) {**  **l = r = 0;**  **que = new int[size];**  **}**  **public void push(int val) {**  **que[r++] = val;**  **}**  **public int pop() {**  **return que[l++];**  **}**  **public int getSize() {**  **return size;**  **}**  **public boolean empty()**  **{**  **return l == r;**  **}**  **}**  **class Account**  **{**  **private int id;**  **private int money;**  **Account(int id)**  **{**  **this.id = id;**  **money = 0;**  **}**  **public void withdraw(int val)**  **{**  **money -= val;**  **}**  **public void deposit(int val)**  **{**  **money += val;**  **}**  **public int getId()**  **{**  **return id;**  **}**  **public void setId(int id) {**  **this.id = id;**  **}**  **public void setMoney(int money) {**  **this.money = money;**  **}**  **public int getMoney() {**  **return money;**  **}**  **}** |
| 调  试  过  程  及  实  验  结  果 |  |
| 总  结 |  |
| 附  录 |  |