**面向对面程序设计JAVA第八次实验**

请完成IO和多线程的实验内容，实验的内容详见实验八的指导书。(100分)

一. 实验目的及实验环境

* 1 熟悉java的IO流的基本原理
* 2 掌握java的File的使用
* 3 掌握缓冲流，数据流，对象流等基本的流的使用
* 4 掌握多线程的生命周期
* 5 掌握多线程并行程序的分析和设计

二. 实验内容 1 完成如下程序：

* 1.1 Write a program that displays all the files (including the files in the sub directory.) after you input a file directory.
* 1.2 Write a program that copies the files with 3 approaches. 1.3 运行下面的程序，写出输出结果
* //The thread class for printing a specified character in specified times
* class PrintChar implements Runnable {
* private char charToPrint; //the character to print
* private int times; //the times to repeat
* //The thread class constructor
* public PrintChar(char c, int t) {
* charToPrint = c;
* times = t;
* }
* //override the run() method to tell the system what the thread will do
* public void run() {
* for (int i=1; i < times; i++)
* System.out.print(charToPrint+"");
* }
* }
* //The thread class for printing number from 1 to n for a given n.
* class PrintNum implements Runnable {
* private int lastNum;
* public PrintNum(int i) {
* lastNum = i;
* }
* public void run() {
* for (int i=1; i < lastNum; i++){
* System.out.print(i);
* }
* }
* }
* public class Exercise30\_01 {
* public static void main(String[] args) {
* Thread printA = new Thread(new PrintChar('a', 100));
* Thread printB = new Thread(new PrintChar('b', 100));
* Thread print100 = new Thread(new PrintNum(100));
* print100.start();
* printA.start();
* printB.start();
* }
* }
* 1.4 阅读下面的程序，并且给予代码注释。
* class ThreadsSum extends Thread{
* int id ;
* int start , end ;
* float sum;
* float[] a ;
* int arraylength ;
* public ThreadsSum(float[] a) {
* sum = 0;
* this.a = a;
* }
* public void run() {
* for(int i = start ; i <=end ; i++ ){
* sum = sum + a[i];
* }
* }
* public float getSum() {
* return sum;
* }
* }
* public class CalculateArraySum {
* /\*\*
  + @param args the command line arguments
* \*/
* public static void main(String[] args) throws Exception{
* // TODO code application logic here
* int numOfThreads = 10;
* int arrayLength = 10009;
* Object synObject = new Object();
* float sum=0 ;
* float [] a= new float[arrayLength];
* for(int i=0;i<a.length;i++) {
* a[i] = (float)Math.random() \* 10.0f;
* }
* ThreadsSum[] threads = new ThreadsSum[numOfThreads];
* int lengthPerThread = arrayLength/numOfThreads ;
* for(int i =0;i<numOfThreads ; i++) {
* threads[i] = new ThreadsSum(a);
* threads[i].id = i;
* threads[i].start = i\* lengthPerThread ;
* threads[i].end = (i +1)\* lengthPerThread -1 ;
* threads[i].start();
* }
* for(int i =0;i<numOfThreads ; i++) {
* threads[i].join();
* }
* for(int i =0;i<numOfThreads ; i++){
* synchronized(synObject){
* sum = sum + threads[i].getSum();
* }
* }
* int remainder = arrayLength - numOfThreads\* lengthPerThread ;
* for(int i=0;i<remainder ;i++) {
* sum = sum + a[arrayLength-i-1];
* }
* System.out.println(sum);
* }
* }
* 三．方案设计（核心代码和流程设计）
* 四．测试数据及运行结果
* 1．正常测试数据（3组）及运行结果；
* 2．非正常测试数据（2组）及运行结果。
* 五．总结
* 1． 实验过程中遇到的问题及解决办法。
* （例如：记录编译时错误信息，根据实验过程中出现的编译错误信息分析出错原因，并改正错误。）
* 2． 对设计及调试过程的心得体会。
* （例如：记录运行时错误结果或者信息，根据实验过程中出现的运行时错误信息进行分析，并利用设置断点、单步跟踪等方法定位错误，改正错误。）

1.初步体验了JAVA的编译环境，了解了package和class的关系：

一个project下可以有多个package,一个package下由可以有多个class,class在package范围内有效，同一package下的class之间可以直接用，不同package下要用class的话需要导入package，语法如下：import 包名.类名;需要注意的是只能用一些public方法和属性。

1. 了解了java的输入输出

输入: import java.until.Scanner;需要新建一个Scanner对象，然后利用这个对象来进行输入输出。

输出:import java.lang;但是一般情况下不用导包。

1. 数学计算的一些方法

import java.lang.Math;同上因为java.lang默认导入，所以不需要导包。

4.局部变量需要定义的时候需要赋初值,要不然编译不通过。

5.源码:

(1)

**package** zdsd;  
  
**import** java.util.Scanner;  
  
**public class** fcgcf {  
 **public static void** main(String[] args) {  
 Person p = **new** Person();  
 p.input();  
 p.getResult();  
 p.printAns();  
  
 }  
  
}  
  
**class** Person {  
 **public** String **name**;  
 **public double hours**,**hourly**, **fedral**, **state**;  
 **public double grosspay**,**dedralpay**,**statepay**,**totalpay**,**nextpay**;  
 **void** getResult(){  
 **grosspay** = **hours**\***hourly**;  
 **dedralpay** = **grosspay**\***fedral**;  
 **statepay** = **grosspay**\***state**;  
 **totalpay** = **dedralpay** + **statepay**;  
 **nextpay** = **grosspay** - **totalpay**;  
 **return** ;  
 }  
  
 **void** printAns(){  
 System.***out***.print(**"Employ Name: "**+ **name** + **"\n"**);  
 System.***out***.print( **"Hours Worked: "** + **hours** + **"\n"**);  
 System.***out***.print(**"Pay rate: $"** + **hourly** + **"\n"**);  
 System.***out***.print(**"Gross Pay: $"** + **grosspay** + **"\n"**);  
 System.***out***.println(**"Deduction"**);  
 System.***out***.println(**" edral Withholding(20.0%): $"** + **dedralpay**);  
 System.***out***.println(**" State Witholding (9.0%): $"** + **statepay**);  
 System.***out***.println(**" Total Deduction: $"** + **totalpay** + **"\n"**);  
 System.***out***.println(**"Net pay: $"** + **nextpay**);  
 **return** ;  
 }  
  
 **void** input() {  
 Scanner sc = **new** Scanner(System.***in***);  
 System.***out***.print(**"Enter employee's name:"**);  
 **name** = sc.nextLine();  
 System.***out***.print(**"Enter number of hours worked in aweek:"**);  
 **hours** = sc.nextDouble();  
 System.***out***.print(**"Enter hourly pay rate:"**);  
 **hourly** = sc.nextDouble();  
 System.***out***.print(**"Enter fedral tax withholding rate:"**);  
 **fedral** = sc.nextDouble();  
 System.***out***.print(**"Enter state tax withholding rate:"**);  
 **state** = sc.nextDouble();  
 }  
}  
(2)

**package** zdsd;  
  
**import** java.util.Scanner;  
**public class** Triangle {  
 **public double x1**,**y1**,**x2**,**y2**,**x3**,**y3**;  
 **public double l1**,**l2**,**l3**;  
 **public double s**;  
 **void** input() {  
 Scanner sc = **new** Scanner(System.***in***);  
 **x1** = sc.nextDouble();  
 **y1** = sc.nextDouble();  
 **x2** = sc.nextDouble();  
 **y2** = sc.nextDouble();  
 **x3** = sc.nextDouble();  
 **y3** = sc.nextDouble();  
 }  
  
 **void** getEdge(){  
 **l1** = Math.*sqrt*(Math.*pow*((**x1**-**x2**),2)+Math.*pow*((**y1**-**y2**),2));  
 **l2** = Math.*sqrt*(Math.*pow*((**x1**-**x3**),2)+Math.*pow*((**y1**-**y3**),2));  
 **l3** = Math.*sqrt*(Math.*pow*((**x3**-**x2**),2)+Math.*pow*((**y3**-**y2**),2));  
 }  
  
 **void** getArea(){  
 getEdge();  
 **double** p = (**l1**+**l2**+**l3**)/2;  
 **this**.**s** = Math.*sqrt*(p\*(p-**l1**)\*(p-**l2**)\*(p-**l3**));  
 *//System.out.println(this.s);*  
}  
  
  
 **public static void** main(String []args){  
 *//System.out.println("Hello world!");*  
Triangle t = **new** Triangle();  
 t.input();  
 t.getArea();  
 System.***out***.printf(**"%.1f"**,t.**s**);  
 *//System.out.println("The are of the triangle is " + t.s);*  
}  
}

(3)

**package** zdsd;  
  
**import** java.util.Scanner;  
  
**public class** equal {  
 **double a**,**b**,**c**;  
 **double q**;  
 **double res1**,**res2**;  
 **void** input(){  
 Scanner sc = **new** Scanner(System.***in***);  
 **a** = sc.nextDouble();  
 **b** = sc.nextDouble();  
 **c** = sc.nextDouble();  
 }  
  
 **int** getRootNum(){  
 **q** = **b**\***b**-4\***a**\***c**;  
 **if**(**q** == 0) **return** 0;  
 **else if**(**q** < 0) **return** -1;  
 **else return** 1;  
 }  
  
 **void** getResult(){  
 getRootNum();  
 **if**(**q** == 0){  
 **res1** = (-**b**+Math.*sqrt*(**q**))/(2\***a**);  
 }**else if**(**q** < 0){  
 **return** ;  
 }**else if**(**q** > 0) {  
 **res1** = (-**b**+Math.*sqrt*(**q**))/(2\***a**);  
 **res2** = (-**b**-Math.*sqrt*(**q**))/(2\***a**);  
 }  
 }  
  
 **void** printAns(){  
 **if**(**q** == 0) System.***out***.println(**"The root is "** + **res1**);  
 **else if**(**q** < 0) System.***out***.println(**"THe equartion has no real roots"**);  
 **else** System.***out***.println(**"THe root are "**+ **res1** + **"and "** + **res2**);  
 }  
  
 **public static void** main(String []args){  
 *//System.out.println("Hello java!");*  
Circle c = **new** Circle();  
 equal e = **new** equal();  
 e.input();  
 e.getResult();  
 e.printAns();  
 }  
}

(4)

**package** zdsd;  
  
**import** java.util.Scanner;  
**public class** Circle {  
 **double x1**,**y1**,**r1**;  
 **double x2**,**y2**,**r2**;  
 **double d**;  
 **double flag**;  
 **public void** input(){  
 Scanner sc = **new** Scanner(System.***in***);  
 **x1** = sc.nextDouble();  
 **y1** = sc.nextDouble();  
 **r1** = sc.nextDouble();  
 **x2** = sc.nextDouble();  
 **y2** = sc.nextDouble();  
 **r2** = sc.nextDouble();  
 }  
  
 **void** judgePos(){  
 **d** = Math.*sqrt*(Math.*pow*((**x1**-**x2**),2)+Math.*pow*((**y1**-**y2**),2));  
 **if**(**d** <= Math.*abs*(**r1**-**r2**) ) **flag** = 1;  
 **else flag** = 2;  
 }  
  
 **public void** printAns(){  
 judgePos();  
 **if**(**flag** == 1){  
 System.***out***.println(**"circle2 is inside circle1"**);  
 }**else if**(**flag** == 2){  
 System.***out***.println(**"Circle2 is overlaps with Circle1"**);  
 }  
 }  
  
 **public static void** main(String []args){  
 *//System.out.println("Hello Circle");*  
Circle c = **new** Circle();  
 c.input();  
 c.printAns();  
 }  
  
  
  
}