Министерство образования и науки, молодежи и спорта Украины

ГВУЗ "Донецкий национальный технический университет"

кафедра Прикладной математики и информатики

Лабораторная работа №4

по курсу " Введение в программирование на Java"

по теме "Пакеты и интерфейсы"

Выполнил студент гр. ІПЗ-12а Егоров А. А.

Проверил: Середа А.А.

Донецк – 2014

Задание

1. Предыдущие 3 лабораторные работы (№1, 2, 3) разбить по файлам.

Каждый класс должен находиться в отдельном файле.

Каждая лабораторная работа должна находиться в отдельном пакете.

Общий метод main должен вызывать все тесты.

2. Придумать задачу и обосновать, почему ее нельзя решить без использования интерфейсов.

Исходный код

package lab.yegorov;

import lab1.yegorov.Lab1;

import lab2.yegorov.Lab2;

import lab3.yegorov.Lab3;

public class Main {

public static void main(String[] args)

{

System.out.println("Lab1:");

Lab1.StartTestLab1(null);

System.out.println("Lab2:");

Lab2.StartTestLab2(null);

System.out.println("Lab3:");

Lab3.StartTestLab3(null);

}

}

package lab1.yegorov;

/\*\*

\* Created by AdminPC on 13.02.14.

\*/

/\*

Задание на лабораторную работу:

Необходимо в основном классе Java - программы создать:

1. 9 методов (один оператор по варианту задания – один метод).

2. 5 методов для проверки приоритетов (методы, в которых используется одновременно два разных

оператора – например, «a+b\*c» - результат вычисления зависит от приоритета).

3. 5 методов для проверки ассоциативности (методы, в которых один и тот же оператор используется дважды,

и результат зависит от порядка выполнения операторов. Например: «a-b-c» - результат зависит от того,

вычисляется ли сначала a-b или b-c).

4. 10 методов тестирования. Для каждого метода тестирования провести минимум 3 теста.

\*/

public class Lab1 {

public static void StartTestLab1(String args[]) {

for(int a = 10; a < 16; ++a) {

priorityTest1(a, a + 2, a - 2);

priorityTest2(a, a + 2, a - 2);

priorityTest3(a, a + 2, a - 2);

}

priorityTest4(true, false, false);

priorityTest5(true, false, false);

priorityTest4(false, false, true);

priorityTest5(false, false, true);

priorityTest4(true, true, false);

priorityTest5(true, true, false);

priorityTest4(true, true, true);

priorityTest5(true, true, true);

priorityTest4(false, false, false);

priorityTest5(false, false, false);

for(int a = 10; a < 16; ++a) {

associativityTest1(a, a + 2, a - 5);

associativityTest2(a, a + 2, a - 5);

associativityTest3(a, a + 2, a - 5);

}

associativityTest4(true, false, false);

associativityTest5(true, false, false);

associativityTest4(false, false, true);

associativityTest5(false, false, true);

associativityTest4(true, true, false);

associativityTest5(true, true, false);

associativityTest4(true, true, true);

associativityTest5(true, true, true);

associativityTest4(false, false, false);

associativityTest5(false, false, false);

for(int a = 5; a < 9; ++a) {

addTest(a, a+3);

modTest(a, a-3);

assignmentWithSubtractionTest(a, a+3);

orTest(a, a+3);

xorWithAssignmentTest(a, a+3);

moreTest(a, a+3);

}

boolXorTest(true, true);

boolXorWithAssignmentTest(true, true);

boolXorWithAssignmentTest2(true, true);

boolXorTest(true, false);

boolXorWithAssignmentTest(true, false);

boolXorWithAssignmentTest2(true, false);

boolXorTest(false, true);

boolXorWithAssignmentTest(false, true);

boolXorWithAssignmentTest2(false, true);

boolXorTest(false, false);

boolXorWithAssignmentTest(false, false);

boolXorWithAssignmentTest2(false, false);

boolNotTest(true);

boolNotTest(false);

}

public static int add(int a, int b) {

return a + b;

}

public static int mod(int a, int b) {

return a % b;

}

public static int assignmentWithSubtraction(int a, int b) {

return a -= b;

}

public static int or(int a, int b) {

return a | b;

}

public static int xorWithAssignment(int a, int b) {

return a ^= b;

}

public static boolean more(int a, int b) {

return a > b;

}

public static boolean boolXor(boolean a, boolean b) {

return a ^ b;

}

public static boolean boolNot(boolean a) {

return !a;

}

public static boolean boolXorWithAssignment(boolean a, boolean b) {

return a ^= b;

}

//-------//

public static void priorityTest1(int a, int b, int c) {

int t;

if((t = add(a,mod(b,c))) == (a + b % c))

System.out.println("Priority operators \"a + b % c\" "+ t +" is work!");

else

System.out.println("Priority operators \"a + b % c\" not working correctly!");

}

public static void priorityTest2(int a, int b, int c) {

int t;

if((t = assignmentWithSubtraction(a, add(b, c))) == (a -= b + c))

System.out.println("Priority operators \"a -= b + c\" "+ t +" is work!");

else

System.out.println("Priority operators \"a -= b + c\" not working correctly!");

}

public static void priorityTest3(int a, int b, int c) {

int t;

if((t = xorWithAssignment(a, or(b, c))) == (a ^= b | c))

System.out.println("Priority operators \"a ^= b | c\" "+ t +" is work!");

else

System.out.println("Priority operators \"a ^= b | c\" not working correctly!");

}

public static void priorityTest4(boolean a, boolean b, boolean c) {

boolean t;

if((t = boolXorWithAssignment(a, boolXor(b, c))) == (a ^= b ^ c))

System.out.println("Priority operators \"a ^= b ^ c\" "+ t +" is work!");

else

System.out.println("Priority operators \"a ^= b ^ c\" not working correctly!");

}

public static void priorityTest5(boolean a, boolean b, boolean c) {

boolean t;

if((t = boolXorWithAssignment(a, boolXor(boolNot(b), boolNot(c)))) == (a ^= !b ^ !c))

System.out.println("Priority operators \"a ^= !b ^ !c\" "+ t +" is work!");

else

System.out.println("Priority operators \"a ^= !b ^ !c\" not working correctly!");

}

//-----//

public static void associativityTest1(int a, int b, int c) {

int t;

if((t = mod(mod(a, b), c)) == (a % b % c))

System.out.println("Associativity operators \"a % b % c\" "+ t +" is work!");

else

System.out.println("Associativity operators \"a % b % c\" not working correctly!");

}

public static void associativityTest2(int a, int b, int c) {

int t;

if((t = assignmentWithSubtraction(a, assignmentWithSubtraction(b, c))) == (a -= b -= c))

System.out.println("Associativity operators \"a -= b -= c\" "+ t +" is work!");

else

System.out.println("Associativity operators \"a -= b -= c\" not working correctly!");

}

public static void associativityTest3(int a, int b, int c) {

int t;

if((t = xorWithAssignment(a, xorWithAssignment(b, c))) == (a ^= b ^= c))

System.out.println("Associativity operators \"a ^= b ^= c\" "+ t +" is work!");

else

System.out.println("Associativity operators \"a ^= b ^= c\" not working correctly!");

}

public static void associativityTest4(boolean a, boolean b, boolean c) {

boolean t;

if((t = boolXor(boolXor(a, b), c)) == (a ^ b ^ c))

System.out.println("Associativity operators \"a ^= b ^ c\" "+ t +" is work!");

else

System.out.println("Associativity operators \"a ^= b ^ c\" not working correctly!");

}

public static void associativityTest5(boolean a, boolean b, boolean c) {

boolean t;

if((t = boolXorWithAssignment(a, boolXorWithAssignment(b, c))) == (a ^= b ^= c))

System.out.println("Associativity operators \"a ^= b ^= c\" "+ t +" is work!");

else

System.out.println("Associativity operators \"a ^= b ^= c\" not working correctly!");

}

//-----//

public static void addTest(int a, int b) {

int t;

if((t = add(a, b)) == (a + b))

System.out.println("Method \"add\" "+ t +" is work!");

else

System.out.println("Method \"add\" not working correctly!");

}

public static void modTest(int a, int b) {

int t;

if((t = mod(a, b)) == (a % b))

System.out.println("Method \"mod\" "+ t +" is work!");

else

System.out.println("Method \"mod\" not working correctly!");

}

public static void assignmentWithSubtractionTest(int a, int b) {

int t;

if((t = assignmentWithSubtraction(a,b)) == (a -= b))

System.out.println("Method \"assignmentWithSubtraction\" "+ t +" is work!");

else

System.out.println("Method \"assignmentWithSubtraction\" "+ t +" is work!");

}

public static void orTest(int a, int b) {

int t;

if((t = or(a, b)) == (a | b))

System.out.println("Method \"or\" "+ t +" is work!");

else

System.out.println("Method \"or\" not working correctly!");

}

public static void xorWithAssignmentTest(int a, int b) {

int t;

if((t = xorWithAssignment(a, b)) == (a ^= b))

System.out.println("Method \"xorWithAssignment\" "+ t +" is work!");

else

System.out.println("Method \"xorWithAssignment\" not working correctly!");

}

public static void moreTest(int a, int b) {

boolean t;

if((t = more(a, b)) == (a > b))

System.out.println("Method \"more\" "+ t +" is work!");

else

System.out.println("Method \"more\" not working correctly!");

}

public static void boolXorTest(boolean a, boolean b) {

boolean t;

if((t = boolXor(a, b)) == (a ^ b))

System.out.println("Method \"boolXor\" "+ t +" is work!");

else

System.out.println("Method \"boolXor\" not working correctly!");

}

public static void boolNotTest(boolean a) {

boolean t;

if((t = boolNot(a)) == (!a))

System.out.println("Method \"boolNot\" "+ t +" is work!");

else

System.out.println("Method \"boolNot\" not working correctly!");

}

public static void boolXorWithAssignmentTest(boolean a, boolean b) {

boolean t;

if((t = boolXorWithAssignment(a, b)) == (a ^= b))

System.out.println("Method \"boolXorWithAssignment\" "+ t +" is work!");

else

System.out.println("Method \"boolXorWithAssignment\" not working correctly!");

}

public static void boolXorWithAssignmentTest2(boolean a, boolean b) {

boolean t;

if((t = boolXorWithAssignment(b, a)) == (b ^= a))

System.out.println("Method \"boolXorWithAssignment\" "+ t +" is work!");

else

System.out.println("Method \"boolXorWithAssignment\" not working correctly!");

}

}

package lab2.yegorov;

/\*\*

\* Created by AdminPC on 13.02.14.

\*/

/\*

1. Написать на java класс, который реализует функционал работы с массивом, приведенный в варианте задания.

Обязательно создать методы: введение массива, вывод массива, сортировка массива (любым методом).

2. Создать класс для тестирования методов работы с массивом, создать минимум 10 тестов.

Variant 3. Создать матрицу M на N, для которой подсчитать сумму элементов каждой строки.

\*/

public class Lab2 {

public static void StartTestLab2(String args[]) {

Test.testing();

}

}

package lab2.yegorov;

import java.text.DecimalFormat;

import java.util.Random;

/\*\*

\* Created by AdminPC on 02.03.14.

\*/

class Matrix {

private double[][] matrix;

public Matrix(int m, int n) {

matrix = new double[m][n];

}

public void watchMatrix() {

DecimalFormat format = new DecimalFormat();

format.setDecimalSeparatorAlwaysShown(false);

for(int i = 0; i < matrix.length; ++i) {

for(int j = 0; j < matrix[i].length; ++j) {

System.out.print(format.format(matrix[i][j]) + "\t");

}

System.out.println();

}

System.out.println();

}

public void inputMatrix() {

//Scanner scan = new Scanner(System.in);

//double tempInput;

for(int i = 0; i < matrix.length; ++i) {

for(int j = 0; j < matrix[i].length; ++j) {

/\*

while (true)

try {

System.out.print("Введите элемент матрицы (" + (i + 1) + ", " +(j + 1) + ") :\n>>> ");

tempInput = scan.nextDouble();

break;

} catch (Exception e) {

System.out.println("Exception. " + e.getMessage() + " Try Again...");

scan.nextLine(); //Clear buffer

}

matrix[i][j] = tempInput;

\*/

Random rand = new Random();

matrix[i][j] = 20 + (100 - 20) \* rand.nextDouble();

}

}

//scan.close();

}

public void sortMatrix() {

for(int i = 0; i < matrix.length; ++i) {

/\* Сортировка методом вставки с прямым включением \*/

int k;

double temp;

for(int ii = 1; ii < matrix[i].length; ++ii ) {

if(matrix[i][ii] < matrix[i][ii-1]) {

temp = matrix[i][ii];

k = ii - 1;

while(k >= 0 && temp < matrix[i][k]) {

matrix[i][k + 1] = matrix[i][k];

k--;

}

matrix[i][k+1] = temp;

}

}

}

}

public void sumMatrixElementLine() {

System.out.println("Сумма элементов строк: ");

double tempSum = 0;

for(int i = 0; i < matrix.length; ++i) {

for(int j = 0; j < matrix[i].length; ++j) {

tempSum += matrix[i][j];

}

System.out.println("" + (i + 1) + "\t = " + tempSum);

tempSum = 0;

}

}

}

package lab2.yegorov;

import java.util.Random;

/\*\*

\* Created by AdminPC on 02.03.14.

\*/

class Test {

public final static int N = 9;

public static void testing() {

Matrix m = new Matrix(5,5);

m.inputMatrix();

System.out.println("Введенная матрица:");

m.watchMatrix();

m.sumMatrixElementLine();

m.sortMatrix();

System.out.println("\nОтсортированная по строкам матрица:");

m.watchMatrix();

System.out.println("\n\n");

m = null;

//System.gc();

Random rand = new Random();

int countTest = 0;

int i, j;

while(true) {

countTest++;

i = 1 + rand.nextInt(5);

j = 1 + rand.nextInt(5);

m = new Matrix(i,j);

m.inputMatrix();

System.out.println("Введенная матрица:");

m.watchMatrix();

m.sumMatrixElementLine();

m.sortMatrix();

System.out.println("\nОтсортированная по строкам матрица:");

m.watchMatrix();

System.out.println("\n\n");

m = null;

//System.gc();

if(countTest == N) break;

}

}

}

package lab3.yegorov;

/\*\*

\* Created by AdminPC on 14.02.14.

\*/

/\*

1. Разработать в программе следующие классы:

- класс, содержащий функцию main;

- класс для методов основной программы (методы должны содержать решение задачи по номеру своего варианта;

в основном методе должны быть использованы условия и циклы);

- класс для методов тестирования, производный от класса основной программы

(методы тестирования должны реализовывать тестирование методов основной программы;

в методах тестирования должны быть протестированы все крайние случаи работы программы).

2. Создать объекты классов программы и тестирования в функции main().

3. Выполнить и протестировать программу.

Винни-Пух на каждый свой день рождения съедает столько еды, сколько съел на двух предыдущих вместе.

На первых двух днях он съел по M граммов.

Нужно узнать, сколь граммов еды он съест на N-й день рождения...

\*/

public class Lab3 {

public static void StartTestLab3(String args[]) {

Task t1, t2, t3;

t1 = new Task(5.0, 7);

t2 = new Task(3.2, 7);

t3 = new Task(1.0, 7);

System.out.println("Task1 = " + t1.answer());

System.out.println("Task2 = " + t2.answer());

System.out.println("Task3 = " + t3.answer());

TestTask test1, test2, test3;

test1 = new TestTask(5.0, 7, 65);

test2 = new TestTask(3.2, 7, 40);

test3 = new TestTask(1.0, 7, 13);

test1.testMethod();

test2.testMethod();

test3.testMethod();

}

}

package lab3.yegorov;

/\*\*

\* Created by AdminPC on 02.03.14.

\*/

class Task {

private double m; // граммы

private int n; // n-й день рождения

private double firstDay;

private double secondDay;

private double newDay;

public Task(double m, int n) {

this.m = m;

this.n = n;

firstDay = secondDay = m;

}

private double fibonacci(int n) {

if(n <= 2) return m;

return fibonacci(n - 2) + fibonacci(n - 1);

}

private double cycleSolve() {

for(int i = 2; i < this.n; ++i) {

newDay = firstDay + secondDay;

firstDay = secondDay;

secondDay = newDay;

}

return newDay;

}

public double answer() {

return cycleSolve();

//return fibonacci(n);

}

}

package lab3.yegorov;

/\*\*

\* Created by AdminPC on 02.03.14.

\*/

class TestTask extends Task {

final private double EPS = 0.000001;

private double myAnswer;

public TestTask(double m, int n, double myAnswer) {

super(m, n);

this.myAnswer = myAnswer;

}

public void testMethod() {

double testAnswer = answer();

if(Math.abs(testAnswer - myAnswer) < EPS)

System.out.println("true! " + testAnswer + " (programm answer) = " + myAnswer + " (my answer)");

else

System.out.println("false! " + testAnswer + " (programm answer) != " + myAnswer + " (my answer)");

}

}

Экранные формы

