Министерство образования и науки, молодежи и спорта Украины

ГВУЗ "Донецкий национальный технический университет"

кафедра Прикладной математики и информатики

Лабораторная работа №6

по курсу " Введение в программирование на Java"

по теме " Исключения в Java "

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

Проверил: Середа А.А.

Донецк – 2014

Задание

1. В предыдущих лабораторных работах (№1, 2, 3, 5) добавить контроль правильности исходных данных (как на этапе получения входных данных так и на этапе обработки).

2. При получении неправильных данных генерировать требуемое исключение.

3. Для выполнения лабораторной работы создать свой класс исключений.

Исходный код

package lab.yegorov;

/\*\*

\* Created by AdminPC on 13.02.14.

\*/

public class Main {

public static void main(String args[]) {

try

{

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);

}catch (MyException e)

{

System.out.println(e.message);

}

}

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) throws MyException {

int t;

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

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

else

throw new MyException("Priority operators \"a + b % c\" not working correctly!");

}

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

int t;

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

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

else

throw new MyException("Priority operators \"a -= b + c\" not working correctly!");

}

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

int t;

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

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

else

throw new MyException("Priority operators \"a ^= b | c\" not working correctly!");

}

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

boolean t;

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

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

else

throw new MyException("Priority operators \"a ^= b ^ c\" not working correctly!");

}

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

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

throw new MyException("Priority operators \"a ^= !b ^ !c\" not working correctly!");

}

//-----//

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

int t;

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

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

else

throw new MyException("Associativity operators \"a % b % c\" not working correctly!");

}

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

int t;

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

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

else

throw new MyException("Associativity operators \"a -= b -= c\" not working correctly!");

}

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

int t;

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

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

else

throw new MyException("Associativity operators \"a ^= b ^= c\" not working correctly!");

}

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

boolean t;

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

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

else

throw new MyException("Associativity operators \"a ^= b ^ c\" not working correctly!");

}

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

boolean t;

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

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

else

throw new MyException("Associativity operators \"a ^= b ^= c\" not working correctly!");

}

//-----//

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

int t;

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

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

else

throw new MyException("Method \"add\" not working correctly!");

}

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

int t;

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

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

else

throw new MyException("Method \"mod\" not working correctly!");

}

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

int t;

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

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

else

throw new MyException("Method \"assignmentWithSubtraction\" "+ t +" is work!");

}

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

int t;

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

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

else

throw new MyException("Method \"or\" not working correctly!");

}

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

int t;

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

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

else

throw new MyException("Method \"xorWithAssignment\" not working correctly!");

}

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

boolean t;

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

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

else

throw new MyException("Method \"more\" not working correctly!");

}

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

boolean t;

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

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

else

throw new MyException("Method \"boolXor\" not working correctly!");

}

public static void boolNotTest(boolean a) throws MyException {

boolean t;

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

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

else

throw new MyException("Method \"boolNot\" not working correctly!");

}

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

boolean t;

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

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

else

throw new MyException("Method \"boolXorWithAssignment\" not working correctly!");

}

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

boolean t;

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

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

else

throw new MyException("Method \"boolXorWithAssignment\" not working correctly!");

}

}

class MyException extends Exception {

public String message;

public MyException(String message)

{

this.message = message;

}

}

package lab.yegorov;

import java.text.DecimalFormat;

import java.util.NoSuchElementException;

import java.util.Random;

import java.util.Scanner;

/\*\*

\* 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 Main {

public static void main(String args[]) {

Test.testing();

}

}

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();

}

}

public void inputMatrix() {

Scanner scan = new Scanner(System.in);

double tempInput = 0;

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>>> ");

if(scan.hasNextDouble())

tempInput = scan.nextDouble();

else

{

scan.nextLine();

continue;

}

if (tempInput < 0)

throw new NegativDoubleNumber("Отрицательное число");

else

break;

} catch (NegativDoubleNumber ex) {

System.out.println("Exception. " + ex.about() + " Try Again...");

//scan.nextLine(); //Clear buffer

}

}

matrix[i][j] = tempInput;

//Random rand = new Random();

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

}

}

scan.close();

}

private double input(int i, int j, Scanner scan) throws NegativDoubleNumber {

double tempInput;

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

tempInput = Double.parseDouble(scan.nextLine());

if (tempInput < 0) throw new NegativDoubleNumber("Отрицательное число");

return tempInput;

}

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;

}

}

}

class Test {

public static void testing() {

Matrix m = new Matrix(3,3);

m.inputMatrix();

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

m.watchMatrix();

m.sumMatrixElementLine();

m.sortMatrix();

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

m.watchMatrix();

m = null;

}

}

class NegativDoubleNumber extends Exception {

public NegativDoubleNumber(String m)

{

super(m);

}

public String about() {

return super.getMessage();

}

}

package lab.yegorov;

/\*\*

\* Created by AdminPC on 14.02.14.

\*/

/\*

public class Main {

public static void main(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, 42.6);

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

try {

test1.testMethod();

test2.testMethod();

test3.testMethod();

} catch (IncorrectAnswerException e) {

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

}

}

}

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);

}

}

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() throws IncorrectAnswerException {

double testAnswer = answer();

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

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

else

throw new IncorrectAnswerException("false! " + testAnswer + " (programm answer) != " + myAnswer + " (my answer)");

}

}

class IncorrectAnswerException extends Exception {

public IncorrectAnswerException(String m)

{

super(m);

}

}

package lab.yegorov;

import java.util.Arrays;

import java.util.Random;

import java.util.Scanner;

import java.util.Vector;

/\*\*

\* Created by AdminPC on 14.02.14.

\*/

/\*

Задание на лабораторную работу:

1. Разработать в программе следующие классы:

- класс, содержащий функцию main;

- класс для методов работы со строками;

- класс для методов тестирования, производный от класса основной программы.

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

Все классы описать внутри отдельного пакета.

Тесты должны запускаться вместе c тестами остальных лабораторных работ.

3. Выполнить и протестировать программу.

Variant 3. Задана строка a. Преобразовать каждое слово в строке так,

чтобы все предыдущие вхождения его последней буквы были заменены на заданный символ b.

Пример

a=”минимум”,b=”.” => rez = “.ини.ум”.

\*/

public class Main {

public static void main(String args[]) {

ReplaceCharTest t = new ReplaceCharTest();

try {

t.Test();

t.writeInputStringAndReplaceChar();

t.Test();

} catch (RandomException e) {

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

}

}

}

class ReplaceChar {

private String inputString;

private char b;

public ReplaceChar(String inputString, char b) {

this.inputString = inputString;

this.b = b;

}

public String toConvert() {

Vector<String> word = new Vector<String>();

word.addAll(Arrays.asList(inputString.split("[ ,.?!:;—<>()\\\\\\-0-9]")));

word.removeAll(Arrays.asList("")); //удаляем пустые строки

word.trimToSize();

Vector<String> punct = new Vector<String>();

punct.addAll(Arrays.asList(inputString.split("\\p{L}")));

punct.removeAll(Arrays.asList("")); //удаляем пустые строки

punct.trimToSize();

char endChar;

int endNum;

StringBuilder strBuild;

String rezult = "";

for(int it = 0; it < word.size(); ++it) {

endNum = (word.elementAt(it)).length() - 1;

endChar = (word.elementAt(it)).charAt(endNum);

strBuild = new StringBuilder(word.elementAt(it));

for(int i = endNum-1; i >= 0; --i) {

if((strBuild.charAt(i)) == endChar)

strBuild.setCharAt(i,b);

}

word.remove(it);

word.add(it,strBuild.toString());

}

boolean t = isfirstPunct();

for(int i = 0, j = 0; i+j < word.size() + punct.size();) {

if(t) {

if(j < punct.size())

rezult += punct.elementAt(j++);

if(i < word.size())

rezult += word.elementAt(i++);

}

else {

if(i < word.size())

rezult += word.elementAt(i++);

if(j < punct.size())

rezult += punct.elementAt(j++);

}

}

return rezult;

}

private boolean isfirstPunct() {

char[] m = " ,.?!:;—()<>0123456789".toCharArray();

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

if(m[i]==inputString.charAt(0))

return true;

}

return false;

}

public String getInputString() {

return inputString;

}

public void setInputString(String inputString) {

this.inputString = inputString;

}

public char getReplaceChar() {

return b;

}

public void setReplaceChar(char b) {

this.b = b;

}

}

class ReplaceCharTest extends ReplaceChar {

public ReplaceCharTest() {

super("Java — объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems " +

"(в последующем приобретённой компанией Oracle)",'.');

}

public void writeInputStringAndReplaceChar() {

Scanner cin = new Scanner(System.in);

try {

System.out.print("Введите строку для тестирования:\n>>> ");

setInputString(cin.nextLine());

System.out.print("\nВведите символ для замены: (при вводе строки будет использован первый символ)\n>>> ");

setReplaceChar(cin.nextLine().charAt(0));

}catch (Exception e) {

System.out.println("exception...");

System.exit(0);

}

}

public void Test() throws RandomException {

System.out.println("Input string (replace symbol: "+getReplaceChar()+"): \n"+getInputString());

System.out.println("Output string: \n"+toConvert());

Random rand = new Random();

int r;

if(((r = rand.nextInt(20)) % 20) < 10 ) throw new RandomException("random number: "+r);

}

}

class RandomException extends Exception {

public RandomException(String s) {

super(s);

}

}

Экранные формы

