|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**

МАГИСТЕРСКАЯ ПРОГРАММА **09.04.01/12 Интеллектуальный анализ больших**

**данных в системах поддержки принятия решений.**

**Отчет**

**по лабораторной работе № 4**

**Вариант № 13**

**Название:** внутренние классы и интерфейсы

**Дисциплина:** языки программирования для работы с большими данными

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-23М |  |  | С.П.Пантелеев |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | П.В. Степанов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2023

**Цель:** изучить работу с исключениями и файлами в java.

**Задание 1:** выполнить задания на основе варианта 1 лабораторной работы 3, контролируя состояние потоков ввода/вывода. При возникновении ошибок, связанных с корректностью выполнения математических операций, генерировать и обрабатывать исключительные ситуации. Предусмотреть обработку исключений, возникающих при нехватке памяти, отсутствии требуемой записи (объекта) в файле, недопустимом значении поля и т.д.

Код класса Main:

package Zadanie\_1;

import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.util.Scanner;

public class Main{

public static void main(String[] args) {

System.out.println("Сколько векторов будет вводится?");

Scanner console = new Scanner(System.in);

int col = console.nextInt();

BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));

Vector[] vectors = new Vector[col];

String x1;

String x2;

String y1;

String y2;

String z1;

String z2;

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

try {

System.out.println("Введите Х1: ");

x1 = console.next();

try {

Double.parseDouble(x1);

} catch (NumberFormatException e) {

System.out.println("Ошибка ввода. Введите повторно");

x1 = console.next();

}

System.out.println("Введите Х2: ");

x2 = console.next();

try {

Double.parseDouble(x2);

} catch (NumberFormatException e) {

System.out.println("Ошибка ввода. Введите повторно");

x2 = console.next();

}

System.out.println("Введите Y1: ");

y1 = console.next();

try {

Double.parseDouble(y1);

} catch (NumberFormatException e) {

System.out.println("Ошибка ввода. Введите повторно");

y1 = console.next();

}

System.out.println("Введите Y2: ");

y2 = console.next();

try {

Double.parseDouble(y2);

} catch (NumberFormatException e) {

System.out.println("Ошибка ввода. Введите повторно");

y2 = console.next();

}

System.out.println("Введите Z1: ");

z1 = console.next();

try {

Double.parseDouble(z1);

} catch (NumberFormatException e) {

System.out.println("Ошибка ввода. Введите повторно");

z1 = console.next();

}

System.out.println("Введите Z2: ");

z2 = console.next();

try {

Double.parseDouble(z2);

} catch (NumberFormatException e) {

System.out.println("Ошибка ввода. Введите повторно");

z2 = console.next();

}

vectors[i] = new Vector(Integer.parseInt(x1), Integer.parseInt(y1), Integer.parseInt(z1),

Integer.parseInt(x2), Integer.parseInt(y2), Integer.parseInt(z2));

} catch (OutOfMemoryError E) {

System.out.println("Нехватка памяти");

}

}

String number\_one;

String number\_two;

System.out.println("номер вектора 1: ");

number\_one = console.next();

try {

Double.parseDouble(number\_one);

} catch (NumberFormatException e) {

System.out.println("Ошибка ввода. Введите повторно");

number\_one = console.next();

}

System.out.println("номер вектора 2: ");

number\_two = console.next();

int x11 = vectors[Integer.parseInt(number\_one)].getX1();

int y11 = vectors[Integer.parseInt(number\_one)].getY1();

int z11 = vectors[Integer.parseInt(number\_one)].getZ1();

int x21 = vectors[Integer.parseInt(number\_one)].getX2();

int y21 = vectors[Integer.parseInt(number\_one)].getY2();

int z21 = vectors[Integer.parseInt(number\_one)].getZ2();

int x12 = vectors[Integer.parseInt(number\_two)].getX1();

int y12 = vectors[Integer.parseInt(number\_two)].getY1();

int z12 = vectors[Integer.parseInt(number\_two)].getZ1();

int x22 = vectors[Integer.parseInt(number\_two)].getX2();

int y22 = vectors[Integer.parseInt(number\_two)].getY2();

int z22 = vectors[Integer.parseInt(number\_two)].getZ2();

System.out.print("Комплонарны ли вектора? - ");

System.out.println(vectors[Integer.parseInt(number\_one)].ort(x11, x12, y11, y12, z11,

z12, x21, x22, y21, y22, z21, z22));

if(!vectors[Integer.parseInt(number\_one)].ort(x11, x12, y11, y12, z11, z12,

x21, x22, y21, y22, z21, z22)){

System.out.print("пересекаются ли вектора? - ");

System.out.println(vectors[Integer.parseInt(number\_one)].per(x11, x12, y11, y12, z11, z12,

x21, x22, y21, y22, z21, z22));

}

System.out.println("вектора одинаковой длинны? - ");

System.out.println(vectors[Integer.parseInt(number\_one)].sravn(x11, x12, y11, y12, z11, z12,

x21, x22, y21, y22, z21, z22));

}

}

Код класса Vector:

package Zadanie\_1;

import static java.lang.Math.pow;

import static java.lang.Math.sqrt;

public class Vector {

private int x1;

private int x2;

private int y1;

private int y2;

private int z1;

private int z2;

public Vector(int x1, int y1, int z1, int x2, int y2, int z2){

this.x1 = x1;

this.y1 = y1;

this.z1 = z1;

this.x2 = x2;

this.y2 = y2;

this.z2 = z2;

}

public int getX1() {

return x1;

}

public int getY1() {

return y1;

}

public int getZ1() {

return z1;

}

public int getX2() {

return x1;

}

public int getY2() {

return y1;

}

public int getZ2() {

return z1;

}

public boolean ort(int x11, int x12,

int y11, int y12,

int z11, int z12,

int x21, int x22,

int y21, int y22,

int z21, int z22){

int sum = x11 \* x12 + y11 \* y12 + z11 \* z12

\* x21 \* x22 + y21 \* y22 + z21 \* z22;

boolean check = false;

try {

Double.parseDouble(Integer.toString(sum));

check = true;

} catch (NumberFormatException e) {

System.out.println("Ошибка вычисления");

}

if (sum == 0 && check){

return true;

}

else {

return false;

}

}

public boolean per(int x11, int x12,

int y11, int y12,

int z11, int z12,

int x21, int x22,

int y21, int y22,

int z21, int z22){

int v1; int v2; int v3; int v4;

v1 = vektorMulti(x22 - x21, y22 - y21, x11 - x21, y11 - y21, z11 - z21, z11 - z21);

v2 = vektorMulti(x22 - x21, y22 - y21, x12 - x21, y12 - y21, z12 - z21, z12 - z21);

v3 = vektorMulti(x12 - x11, y21 - y11, x21 - x11, y21 - y21, z21 - z11, z21 - z11);

v4 = vektorMulti(x12 - x11, y21 - y11, x22 - x11, y22 - y21, z22 - z11, z22 - z11);

boolean check = false;

try {

Double.parseDouble(Integer.toString(v1 \* v2 \* v3 \* v4));

check = true;

} catch (NumberFormatException e) {

System.out.println("Ошибка вычисления");

}

if (v1 \* v2 < 0 && v3 \* v4 < 0 && check){

return true;

}

else {

return false;

}

}

public int vektorMulti(int ax, int ay,

int az, int bx,

int by, int bz){

boolean check = false;

try {

Double.parseDouble(Integer.toString(ax\*by - bx\*ay));

check = true;

} catch (NumberFormatException e) {

System.out.println("Ошибка вычисления");

}

if(check){

return ax\*by - bx\*ay;

}

else {

return 0;

}

}

public boolean sravn(int x11, int x12,

int y11, int y12,

int z11, int z12,

int x21, int x22,

int y21, int y22,

int z21, int z22){

double mod\_one = sqrt(pow(x11, 2) + pow(x12, 2) + pow(y11, 2)

+ pow(y12, 2) + pow(z12, 2) + pow(z12, 2));

double mod\_two = sqrt(pow(x21, 2) + pow(x22, 2) + pow(y21, 2)

+ pow(y22, 2) + pow(z22, 2) + pow(z22, 2));

boolean check = false;

try {

Double.parseDouble(Double.toString(mod\_one \* mod\_two));

check = true;

} catch (NumberFormatException e) {

System.out.println("Ошибка вычисления");

}

if(mod\_one == mod\_two && check){

return true;

}

else{

return false;

}

}

}

Работа программы показана на рисунке 1.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1 – Работа программы

**Задание 2:** выполнить задания на основе варианта 1 лабораторной работы 3, контролируя состояние потоков ввода/вывода. При возникновении ошибок, связанных с корректностью выполнения математических операций, генерировать и обрабатывать исключительные ситуации. Предусмотреть обработку исключений, возникающих при нехватке памяти, отсутствии требуемой записи (объекта) в файле, недопустимом значении поля и т.д.

Код класса Main:

package Zadanie\_2;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

System.out.println("Сколько матриц будет вводится?");

Scanner console = new Scanner(System.in);

int col = console.nextInt();

try {

Double.parseDouble(Integer.toString(col));

} catch (NumberFormatException e) {

System.out.println("Ошибка ввода. Введите повторно");

col = console.nextInt();

}

Matrix[] matrices = new Matrix[col];

System.out.println("Размер матриц: ");

int n1 = console.nextInt();

try {

try {

Double.parseDouble(Integer.toString(n1));

} catch (NumberFormatException e) {

System.out.println("Ошибка ввода. Введите повторно");

n1 = console.nextInt();

}

} catch (OutOfMemoryError E) {

System.out.println("Нехватка памяти");

}

System.out.println("Диапазон случайных чисел: ");

int n2 = console.nextInt();

try {

Double.parseDouble(Integer.toString(n2));

} catch (NumberFormatException e) {

System.out.println("Ошибка ввода. Введите повторно");

n2 = console.nextInt();

}

int n3 = console.nextInt();

try {

Double.parseDouble(Integer.toString(n3));

} catch (NumberFormatException e) {

System.out.println("Ошибка ввода. Введите повторно");

n3= console.nextInt();

}

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

matrices[i] = new Matrix(n1, n2, n3);

}

int number\_one;

int number\_two;

System.out.println("С какими матрицами будут производится вычисления");

System.out.println("Матрица 1: ");

number\_one = console.nextInt();

try {

Double.parseDouble(Integer.toString(number\_one));

} catch (NumberFormatException e) {

System.out.println("Ошибка ввода. Введите повторно");

number\_one= console.nextInt();

}

System.out.println("Матрица 2: ");

number\_two = console.nextInt();

try {

Double.parseDouble(Integer.toString(number\_two));

} catch (NumberFormatException e) {

System.out.println("Ошибка ввода. Введите повторно");

number\_two= console.nextInt();

}

operation operations = new operation();

matrices[number\_one].zapolnenie();

matrices[number\_two].zapolnenie();

double[] Matrix\_one = matrices[number\_one].oneArray;

double[] Matrix\_two = matrices[number\_two].oneArray;

double[][] Matrix\_one\_one = matrices[number\_one].twoArray;

double[][] Matrix\_two\_two = matrices[number\_two].twoArray;

System.out.println("Сложение: ");

operations.sum(Matrix\_one, Matrix\_two, n1);

System.out.println("Вычитания: ");

operations.vuch(Matrix\_one, Matrix\_two, n1);

System.out.println("умножение: ");

operations.umnoj(Matrix\_one\_one, Matrix\_two\_two, n1);

System.out.println("У какой матрицы считать норму: ");

int norma = console.nextInt();

try {

Double.parseDouble(Integer.toString(norma));

} catch (NumberFormatException e) {

System.out.println("Ошибка ввода. Введите повторно");

norma= console.nextInt();

}

double[][] norma\_matrix = matrices[norma].twoArray;

System.out.println("Первая норма матрицы: ");

System.out.println(operations.one\_norma(n1, norma\_matrix));

System.out.println("Вторая норма матрицы: ");

System.out.println(operations.two\_norma(n1, norma\_matrix));

double max\_norma\_one = -1000;

double max\_norma\_two = -1000;

for (int i = 0; i < n1; i++) {

if (max\_norma\_one < operations.one\_norma(n1, matrices[i].twoArray)){

max\_norma\_one = operations.one\_norma(n1, matrices[i].twoArray);

}

}

for (int i = 0; i < n1; i++) {

if (max\_norma\_two < operations.two\_norma(n1, matrices[i].twoArray)){

max\_norma\_two = operations.two\_norma(n1, matrices[i].twoArray);

}

}

}

}

Код класса Matrix:

package Zadanie\_2;

public class Matrix {

private int n1;

private int n2;

private int n3;

public Matrix(int n1, int n2, int n3) {

this.n1 = n1;

this.n2 = n2;

this.n3 = n3;

}

public int getN1() {

return n1;

}

public void setN1(int n1) {

this.n1 = n1;

}

public int getN2() {

return n2;

}

public void setN2(int n2) {

this.n2 = n2;

}

public int getN3() {

return n3;

}

public void setN3(int n3) {

this.n3 = n3;

}

public double[][] twoArray;

public double[] oneArray;

public void zapolnenie(){

twoArray = new double[n1][n1];

for (int i = 0; i < n1; i++) {

for (int j = 0; j < n1; j++) {

twoArray[i][j] = (Math.random() \* ((n3 - n2))) + n2;

System.out.print(twoArray[i][j] + " ");

}

System.out.println();

}

int one\_i = 0;

oneArray = new double[n1 \* n1];

for (int i = 0; i < n1; i++) {

for (int j = 0; j < n1; j++) {

oneArray[one\_i] = twoArray[i][j];

one\_i++;

}

}

}

}

Код класса Operation:

package Zadanie\_2;

public class operation {

public void sum(double[] martix\_one, double[] matrix\_two, int n1){

double[] rez = new double[martix\_one.length];

for (int i = 0; i < martix\_one.length; i++){

try {

rez[i] = martix\_one[i] + matrix\_two[i];

try {

Double.parseDouble(Integer.toString((int) rez[i]));

} catch (NumberFormatException e) {

System.out.println("Ошибка вычисления");

}

} catch (OutOfMemoryError E) {

System.out.println("Нехватка памяти");

}

}

for (int i = 0; i < n1; i++) {

for (int j = 0; j < n1; j++) {

try {

System.out.print(rez[i] + " ");

try {

Double.parseDouble(Integer.toString((int) rez[i]));

} catch (NumberFormatException e) {

System.out.println("Ошибка вычисления");

}

} catch (OutOfMemoryError E) {

System.out.println("Нехватка памяти");

}

}

System.out.println();

}

}

public void vuch(double[] martix\_one, double[] matrix\_two, int n1){

double[] rez = new double[matrix\_two.length];

for (int i = 0; i < martix\_one.length; i++){

try {

rez[i] = martix\_one[i] - matrix\_two[i];

try {

Double.parseDouble(Integer.toString((int) rez[i]));

} catch (NumberFormatException e) {

System.out.println("Ошибка вычисления");

}

} catch (OutOfMemoryError E) {

System.out.println("Нехватка памяти");

}

}

for (int i = 0; i < n1; i++) {

for (int j = 0; j < n1; j++) {

try {

try {

Double.parseDouble(Integer.toString((int) rez[i]));

} catch (NumberFormatException e) {

System.out.println("Ошибка вычисления");

}

} catch (OutOfMemoryError E) {

System.out.println("Нехватка памяти");

}

System.out.print(rez[i] + " ");

}

System.out.println();

}

}

public void umnoj(double[][] martix\_one, double[][] matrix\_two, int n1){

double[][] rez = new double[n1][n1];

for (int i = 0; i < martix\_one.length; i++) {

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

for (int k = 0; k < matrix\_two.length; k++) {

try {

rez[i][j] += martix\_one[i][k] \* matrix\_two[k][j];

try {

Double.parseDouble(Integer.toString((int) rez[i][j]));

} catch (NumberFormatException e) {

System.out.println("Ошибка вычисления");

}

} catch (OutOfMemoryError E) {

System.out.println("Нехватка памяти");

}

}

}

}

for (int i = 0; i < n1; i++) {

try {

for (int j = 0; j < n1; j++) {

System.out.print(rez[i][j] + " ");

try {

Double.parseDouble(Integer.toString((int) rez[i][j]));

} catch (NumberFormatException e) {

System.out.println("Ошибка вычисления");

}

}

System.out.println();

} catch (OutOfMemoryError E) {

System.out.println("Нехватка памяти");

}

}

}

public double one\_norma(int n1, double[][] twoArray){

double[] str = new double[n1];

for (int i = 0; i < n1; i++){

for (int j = 0; j < n1; j++) {

try {

str[i] += twoArray[i][j];

try {

Double.parseDouble(Integer.toString((int) str[i]));

} catch (NumberFormatException e) {

System.out.println("Ошибка вычисления");

}

} catch (OutOfMemoryError E) {

System.out.println("Нехватка памяти");

}

}

}

double max\_i = -10000;

for (int i = 0; i < n1; i++) {

try {

if(str[i] > max\_i){

max\_i = str[i];

try {

Double.parseDouble(Integer.toString((int) max\_i));

} catch (NumberFormatException e) {

System.out.println("Ошибка вычисления");

}

}

} catch (OutOfMemoryError E) {

System.out.println("Нехватка памяти");

}

}

return max\_i;

}

public double two\_norma(int n1, double[][] twoArray){

double[] str = new double[n1];

for (int i = 0; i < n1; i++){

for (int j = 0; j < n1; j++) {

try {

str[i] += twoArray[j][i];

try {

Double.parseDouble(Integer.toString((int) str[i]));

} catch (NumberFormatException e) {

System.out.println("Ошибка вычисления");

}

} catch (OutOfMemoryError E) {

System.out.println("Нехватка памяти");

}

}

}

double max\_j = -10000;

for (int i = 0; i < n1; i++) {

try {

if(str[i] > max\_j){

max\_j = str[i];

try {

Double.parseDouble(Integer.toString((int) max\_j));

} catch (NumberFormatException e) {

System.out.println("Ошибка вычисления");

}

}

} catch (OutOfMemoryError E) {

System.out.println("Нехватка памяти");

}

}

return max\_j;

}

}

Работа программы показана на рисунке 2.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2 – Работа программы

**Задание 3:** выполнить задания из варианта 2 лабораторной работы 3, реализуя собственные обработчики исключений и исключения ввода/вывода.

Код класса Main:

package Zadanie\_3;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Patient[] patients = new Patient[100];

Scanner console = new Scanner(System.in);

String first\_name;

String second\_name;

String third\_name;

String address;

String diagnoz\_in;

System.out.println("Сколько будет пациентов: ");

int col = console.nextInt();

if (col < 0){

System.out.println("Введите число больше 0");

col = console.nextInt();

} else if (col > 2147483647) {

System.out.println("Слишком много пациентов");

col = console.nextInt();

}

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

System.out.println("Введите имя: ");

first\_name = console.next();

if(!console.hasNextInt()){

if (first\_name.length() > 50){

System.out.println("Слишком длинное имя, повторите ввод");

first\_name = console.next();

}

} else {

System.out.println("Введено число, введите заново");

first\_name = console.next();

}

System.out.println("Введите фамилию: ");

second\_name = console.next();

if(!console.hasNextInt()){

if (second\_name.length() > 50){

System.out.println("Слишком длинная фамилия, повторите ввод");

second\_name = console.next();

}

} else {

System.out.println("Введено число, введите заново");

second\_name = console.next();

}

System.out.println("Введите отчество: ");

third\_name = console.next();

if(!console.hasNextInt()){

if (third\_name.length() > 50){

System.out.println("Слишком длинное отчество, повторите ввод");

third\_name = console.next();

}

} else {

System.out.println("Введено число, введите заново");

third\_name = console.next();

}

System.out.println("Введите адрес: ");

address = console.next();

if(!console.hasNextInt()){

if (address.length() > 50){

System.out.println("Слишком длинный адрес, повторите ввод");

address = console.next();

}

} else {

System.out.println("Введено число, введите заново");

address = console.next();

}

System.out.println("Введите диагноз: ");

diagnoz\_in = console.next();

if(!console.hasNextInt()){

} else {

System.out.println("Введено число, введите заново");

diagnoz\_in = console.next();

}

patients[i] = new Patient(i, first\_name, second\_name, third\_name,

address, i, i, diagnoz\_in);

}

System.out.println("Введите диагноз, чтобы поциентов с ним: ");

String diagnoz = console.next();

if(!console.hasNextInt()){

} else {

System.out.println("Введено число, введите заново");

diagnoz = console.next();

}

System.out.println("Пациенты с таким диагнозом: ");

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

if(patients[i].getDiagnoz().equals(diagnoz)){

System.out.println(patients[i].getFirst\_name() + " " + patients[i].getSecond\_name());

}

}

System.out.println("Введите диапазон значений: ");

int n1 = console.nextInt();

if (n1 > patients.length){

System.out.println("Введен слишком большой диапазон, введите заново: ");

n1 = console.nextInt();

}

int n2 = console.nextInt();

if (n2 > patients.length){

System.out.println("Введен слишком большой диапазон, введите заново: ");

n2 = console.nextInt();

}

System.out.println("Пациенты с номером медицинской карты, попадающими в этот диапазон: ");

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

if(patients[i].getNumber\_medic() > n1 && patients[i].getNumber\_medic() < n2){

System.out.println(patients[i].getFirst\_name() + " " + patients[i].getSecond\_name());

}

}

}

}

Код класса Patient:

package Zadanie\_3;

public class Patient {

private int id;

private String first\_name;

private String second\_name;

private String third\_name;

private String address;

private int number\_phone;

private int number\_medic;

private String diagnoz;

@Override

public String toString() {

return "Patient{" +

"number\_phone=" + number\_phone +

", number\_medic=" + number\_medic +

'}';

}

public Patient(int id, String first\_name, String second\_name, String third\_name, String address, int number\_phone, int number\_medic, String diagnoz) {

this.id = id;

this.first\_name = first\_name;

this.second\_name = second\_name;

this.third\_name = third\_name;

this.address = address;

this.number\_phone = number\_phone;

this.number\_medic = number\_medic;

this.diagnoz = diagnoz;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getFirst\_name() {

return first\_name;

}

public void setFirst\_name(String first\_name) {

this.first\_name = first\_name;

}

public String getSecond\_name() {

return second\_name;

}

public void setSecond\_name(String second\_name) {

this.second\_name = second\_name;

}

public String getThird\_name() {

return third\_name;

}

public void setThird\_name(String third\_name) {

this.third\_name = third\_name;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

public int getNumber\_phone() {

return number\_phone;

}

public void setNumber\_phone(int number\_phone) {

this.number\_phone = number\_phone;

}

public int getNumber\_medic() {

return number\_medic;

}

public void setNumber\_medic(int number\_medic) {

this.number\_medic = number\_medic;

}

public String getDiagnoz() {

return diagnoz;

}

public void setDiagnoz(String diagnoz) {

this.diagnoz = diagnoz;

}

}

Работа программы показана на рисунке 3.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 3 – Работа программы

**Задание 4:** выполнить задания из варианта 2 лабораторной работы 3, реализуя собственные обработчики исключений и исключения ввода/вывода.

Код класса Main:

package Zadanie\_4;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Abiturient[] abiturients = new Abiturient[100];

Scanner console = new Scanner(System.in);

String first\_name;

String second\_name;

String third\_name;

String address;

int[] ocenki = new int[100];

System.out.println("Сколько будет аббитуриентов: ");

int col = console.nextInt();

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

System.out.println("Введите имя: ");

first\_name = console.next();

if(!console.hasNextInt()){

if (first\_name.length() > 50){

System.out.println("Слишком длинное имя, повторите ввод");

first\_name = console.next();

}

} else {

System.out.println("Введено число, введите заново");

first\_name = console.next();

}

System.out.println("Введите фамилию: ");

second\_name = console.next();

if(!console.hasNextInt()){

if (second\_name.length() > 50){

System.out.println("Слишком длинное имя, повторите ввод");

second\_name = console.next();

}

} else {

System.out.println("Введено число, введите заново");

second\_name = console.next();

}

System.out.println("Введите отчество: ");

third\_name = console.next();

if(!console.hasNextInt()){

if (third\_name.length() > 50){

System.out.println("Слишком длинное имя, повторите ввод");

third\_name = console.next();

}

} else {

System.out.println("Введено число, введите заново");

third\_name = console.next();

}

System.out.println("Введите адрес: ");

address = console.next();

if(!console.hasNextInt()){

if (third\_name.length() > 50){

System.out.println("Слишком длинное имя, повторите ввод");

third\_name = console.next();

}

} else {

System.out.println("Введено число, введите заново");

third\_name = console.next();

}

System.out.println("Сколько оценок: ");

int kol\_oc = console.nextInt();

for (int j = 0; j < kol\_oc; j++) {

ocenki[i] = console.nextInt();

}

abiturients[i] = new Abiturient(i, first\_name, second\_name,

third\_name, address, i, ocenki);

}

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

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

if (abiturients[i].getOcenki().length == 2){

System.out.println(abiturients[i].getFirst\_name() + " " + abiturients[i].getSecond\_name());

break;

}

}

}

System.out.println("Средний балл: ");

int z\_sr\_ball = console.nextInt();

if (z\_sr\_ball == (int)z\_sr\_ball){

}else {

System.out.println("Введено число, повторный ввод");

z\_sr\_ball = console.nextInt();

}

double sr\_ball = 0;

int[] mass\_ocenki;

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

mass\_ocenki = abiturients[i].getOcenki();

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

sr\_ball += mass\_ocenki[i];

}

sr\_ball /= abiturients[i].getOcenki().length;

if (sr\_ball < z\_sr\_ball){

System.out.println(abiturients[i].getFirst\_name() + " " + abiturients[i].getSecond\_name());

}

sr\_ball = 0;

}

System.out.println("Высокий балл: ");

int viss\_ball = console.nextInt();

if (viss\_ball == (int)viss\_ball){

}else {

System.out.println("Введено число, повторный ввод");

viss\_ball = console.nextInt();

}

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

mass\_ocenki = abiturients[i].getOcenki();

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

sr\_ball += mass\_ocenki[i];

}

sr\_ball /= abiturients[i].getOcenki().length;

if (sr\_ball > viss\_ball){

System.out.println(abiturients[i].getFirst\_name() + " " + abiturients[i].getSecond\_name());

}

sr\_ball = 0;

}

System.out.println("Проходной балл: ");

double pr\_ball = console.nextInt();

if (pr\_ball == (double)pr\_ball){

}else {

System.out.println("Введено число, повторный ввод");

pr\_ball = console.nextInt();

}

System.out.println("Имеют полупроходной балл: ");

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

mass\_ocenki = abiturients[i].getOcenki();

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

sr\_ball += mass\_ocenki[i];

}

sr\_ball /= abiturients[i].getOcenki().length;

if (sr\_ball > pr\_ball/2){

System.out.println(abiturients[i].getFirst\_name() + " " + abiturients[i].getSecond\_name());

}

sr\_ball = 0;

}

}

}

Код класса Abiturient:

package Zadanie\_4;

import java.util.Arrays;

public class Abiturient {

public int id;

public String first\_name;

public String second\_name;

public String third\_name;

public String address;

public int number\_phone;

public int[] ocenki;

@Override

public String toString() {

return "Abiturient{" +

"id=" + id +

", first\_name='" + first\_name + '\'' +

", second\_name='" + second\_name + '\'' +

", third\_name='" + third\_name + '\'' +

", address='" + address + '\'' +

", number\_phone=" + number\_phone +

", ocenki=" + Arrays.toString(ocenki) +

'}';

}

public Abiturient(int id, String first\_name, String second\_name, String third\_name, String address, int number\_phone, int[] ocenki) {

this.id = id;

this.first\_name = first\_name;

this.second\_name = second\_name;

this.third\_name = third\_name;

this.address = address;

this.number\_phone = number\_phone;

this.ocenki = ocenki;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getFirst\_name() {

return first\_name;

}

public void setFirst\_name(String first\_name) {

this.first\_name = first\_name;

}

public String getSecond\_name() {

return second\_name;

}

public void setSecond\_name(String second\_name) {

this.second\_name = second\_name;

}

public String getThird\_name() {

return third\_name;

}

public void setThird\_name(String third\_name) {

this.third\_name = third\_name;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

public int getNumber\_phone() {

return number\_phone;

}

public void setNumber\_phone(int number\_phone) {

this.number\_phone = number\_phone;

}

public int[] getOcenki() {

return ocenki;

}

public void setOcenki(int[] ocenki) {

this.ocenki = ocenki;

}

}

Работа программы показана на рисунке 4.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 4 – Работа программы

**Задание 5:** ввести последовательность строк из текстового потока и выполнить указанные действия. При этом могут рассматриваться два варианта:

• каждая строка состоит из одного слова;

• каждая строка состоит из нескольких слов.

Имена входного и выходного файлов, а также абсолютный путь к ним могут быть введены как параметры командной строки или храниться в файле. В каждой строке найти слова, начинающиеся с гласной буквы.

Код класса Main:

package Zadanie\_5;

import java.io.\*;

public class Main {

public static void main(String[] args) {

char[] text = new char[100];

char[] text\_gl = new char[100];

int kol = 0;

try(FileReader reader = new FileReader("C:/University/Java\_labs/laba\_5/old\_file.txt"))

{

int c;

while((c=reader.read())!=-1){

text[kol] = (char)c;

kol++;

//System.out.print((char)c);

}

}

catch(IOException ex){

System.out.println(ex.getMessage());

}

char mas[] = new char[9];

mas[0] = 'у';

mas[1] = 'е';

mas[2] = 'ы';

mas[3] = 'а';

mas[4] = 'о';

mas[5] = 'э';

mas[6] = 'я';

mas[7] = 'и';

mas[8] = 'ю';

int g = 0;

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

if((i == 0) || (text[i-1] == '\n') || (text[i-1] == ' ')){

for (int j = 0; j < 9; j++){

if(text[i] == mas[j]){

int k = i;

while (text[k] != '\r' && text[k] != ' ' && k < kol){

//System.out.print(text[k]);

text\_gl[g] = text[k];

k++; g++;

}

text\_gl[g] = '\r';

g++;

text\_gl[g] = '\n';

g++;

}

}

}

}

try(FileWriter writer = new FileWriter("C:/University/Java\_labs/laba\_5/new\_file.txt", false))

{

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

writer.append(text\_gl[i]);

}

writer.flush();

}

catch(IOException ex){

System.out.println(ex.getMessage());

}

}

}

Работа программы показана на рисунках 5 – 6.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 5 – Работа программы

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 6 – Работа программы

**Задание 6:** ввести последовательность строк из текстового потока и выполнить указанные действия. При этом могут рассматриваться два варианта:

• каждая строка состоит из одного слова;

• каждая строка состоит из нескольких слов.

Имена входного и выходного файлов, а также абсолютный путь к ним могут быть введены как параметры командной строки или храниться в файле. Найти и вывести слова текста, для которых последняя буква одного слова совпадает с первой буквой следующего слова.

Код кл package Zadanie\_6;

import java.io.\*;

public class Main {

public static void main(String[] args) {

char[] text = new char[100];

char[] text\_gl = new char[100];

int kol = 0;

try (FileReader reader = new FileReader("C:/University/Java\_labs/laba\_5/old\_file.txt")) {

int c;

while ((c = reader.read()) != -1) {

text[kol] = (char) c;

kol++;

}

} catch (IOException ex) {

System.out.println(ex.getMessage());

}

int i\_nach\_one = 0;

int g = 0;

for (int i = 3; i < kol; i++) {

if (text[i] == '\n') {

i\_nach\_one = i + 1;

} else if (text[i + 2] == '\n' && text[i + 3] == text[i]) {

int k = i\_nach\_one;

while (text[k] != '\r' && text[k] != ' ' && k < kol) {

text\_gl[g] = text[k];

k++;

g++;

}

text\_gl[g] = '\r';

g++;

text\_gl[g] = '\n';

g++;

}

if (text[i - 3] == text[i] && text[i - 1] == '\n') {

int k = i;

while (text[k] != '\r' && text[k] != ' ' && k < kol) {

//System.out.print(text[k]);

text\_gl[g] = text[k];

k++;

g++;

}

text\_gl[g] = '\r';

g++;

text\_gl[g] = '\n';

g++;

}

if (text[i] == ' ') {

i\_nach\_one = i + 1;

} else if (text[i] == ' ' && text[i + 1] == text[i]) {

int k = i\_nach\_one;

while (text[k] != '\r' && text[k] != ' ' && k < kol) {

//System.out.print(text[k]);

text\_gl[g] = text[k];

k++;

g++;

}

text\_gl[g] = '\r';

g++;

text\_gl[g] = '\n';

g++;

}

if (text[i - 2] == text[i] && text[i - 1] == ' ') {

int k = i;

while (text[k] != '\r' && text[k] != ' ' && k < kol) {

text\_gl[g] = text[k];

k++;

g++;

}

text\_gl[g] = '\r';

g++;

text\_gl[g] = '\n';

g++;

}

}

try(FileWriter writer = new FileWriter("C:/University/Java\_labs/laba\_5/new\_file.txt", false))

{

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

writer.append(text\_gl[i]);

}

writer.flush();

}

catch(IOException ex){

System.out.println(ex.getMessage());

}

}

}асса Main:

Работа программы показана на рисунке 7.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 7 – Работа программы

**Задание 7:** при выполнении следующих заданий для вывода результатов создавать новую директорию и файл средствами класса File. Прочитать текст Java-программы и в каждом слове длиннее двух символов все строчные символы заменить прописными.

Код класса Main:

package Zadanie\_7;

import java.io.File;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

public class Main {

public static final String PATH = "C:/University/Java\_labs/laba\_5";

public static void main(String[] args) {

char[] text\_file = new char[1000];

String[] text\_file\_UP = new String[1000];

int kol = 0;

try(FileReader reader = new FileReader("C:/University/Java\_labs/laba\_5/text.txt"))

{

int c;

while((c=reader.read())!=-1){

text\_file[kol] = (char)c;

kol++;

}

}

catch(IOException ex){

System.out.println(ex.getMessage());

}

String string\_text\_file = new String(text\_file);

String[] word = string\_text\_file.split(" ");

word = string\_text\_file.split("\r\n");

for (int i = 2; i < word.length - 2; i++) {

if(!word[i].equals(" ") && !word[i + 1].equals(" ") && !word[i + 2].equals(" ")

&& !word[i + 1].equals("\r") && !word[i + 2].equals("\r")

|| !word[i].equals(" ") && !word[i - 1].equals(" ") && !word[i - 2].equals(" ")

&& !word[i - 1].equals("\n") && !word[i - 2].equals("\n")){

text\_file\_UP[i] = word[i].toUpperCase();

}

}

/\*for (int i = 0; i < word.length; i++) {

System.out.println(text\_file\_UP[i]);

}\*/

File dir = new File(PATH);

File newFile = new File(dir, "rez\_file.txt");

if(!newFile.exists()){

try {

newFile.createNewFile();

} catch (IOException e) {

throw new RuntimeException(e);

}

}

try(FileWriter writer = new FileWriter("C:/University/Java\_labs/laba\_5/rez\_file.txt", false))

{

for (int i = 0; i < text\_file\_UP.length; i++) {

writer.append(text\_file\_UP[i]);

}

writer.flush();

}

catch(IOException ex){

System.out.println(ex.getMessage());

}

}

}

Работа программы показана на рисунке 8.

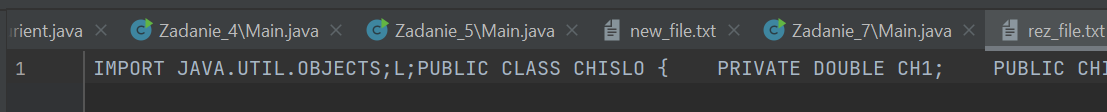


Рисунок 8 – Работа программы

**Задание 8:** при выполнении следующих заданий для вывода результатов создавать новую директорию и файл средствами класса File. В файле, содержащем фамилии студентов и их оценки, записать прописными буквами фамилии тех студентов, которые имеют средний балл более “7”.

Код класса Main:

package Zadanie\_8;

import java.io.File;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

public class Main {

public static final String PATH = "C:/University/Java\_labs/laba\_5";

public static void main(String[] args) {

//System.out.println("Hello world!");

char[] text\_file = new char[1000];

int kol = 0;

try(FileReader reader = new FileReader("C:/University/Java\_labs/laba\_5/students.txt"))

{

int c;

while((c=reader.read())!=-1){

text\_file[kol] = (char)c;

kol++;

}

}

catch(IOException ex){

System.out.println(ex.getMessage());

}

String string\_text\_file = new String(text\_file);

String[] word = string\_text\_file.split(" ");

String[] mass = new String[10];

mass[0] = "1";

mass[1] = "2";

mass[2] = "3";

mass[3] = "4";

mass[4] = "5";

mass[5] = "6";

mass[6] = "7";

mass[7] = "8";

mass[8] = "9";

mass[9] = "10";

String[] name = new String[3];

name[0] = word[0];

int na = 1;

int s = 0;

double n = 0;

String ch1 = null;

double[] mass\_sr = new double[3];

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

String ch = word[i];

String rn = "\r\n";

if (ch.contains(rn)){

String[] ch\_split = ch.split("\r\n");

ch1 = ch\_split[0];

name[na] = ch\_split[1];

na++;

}

if (!ch.contains(rn)){

for (int j = 0; j < mass.length; j++) {

if (ch.equals(mass[j])){

mass\_sr[s] += Integer.parseInt(mass[j]);

n++;

}

}

} else {

mass\_sr[s] += Integer.parseInt(ch1);

n++;

mass\_sr[s] /= n;

s++;

n = 0;

}

}

mass\_sr[s] += Integer.parseInt(ch1);

n++;

mass\_sr[s] /= n;

File dir = new File(PATH);

File newFile = new File(dir, "rez\_file2.txt");

if(!newFile.exists()){

try {

newFile.createNewFile();

} catch (IOException e) {

throw new RuntimeException(e);

}

}

try(FileWriter writer = new FileWriter("C:/University/Java\_labs/laba\_5/rez\_file2.txt", false))

{

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

if (mass\_sr[i] > 7){

writer.append(name[i]);

}

}

writer.flush();

}

catch(IOException ex){

System.out.println(ex.getMessage());

}

}

}

Работа программы показана на рисунках 9 – 10.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 9 – Работа программы

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 10 – Работа программы

**Вывод:** были изучена работа с исключениями и файлами в java.