Министерство образования Калининградской области

государственное бюджетное учреждение Калининградской области

профессиональная образовательная организация

«Колледж информационных технологий и строительства»

(ГБУ КО ПОО «КИТиС»)

**Отчет по учебной практике**

УП.01 Разработка модулей программного обеспечения для компьютерных систем

по ПМ.01 Разработка модулей программного обеспечения для компьютерных систем

Специальность 09.02.07 «Информационные системы и программирование»

Сроки прохождения практики:

с «20» ноября 2023 г. по «09» декабря 2023 г.

Место практики ГБУ КО ПОО «КИТиС»

|  |  |
| --- | --- |
| Выполнил: | студент 4 курса,  группы ИСп20-1  Коржилов Михаил Владимирович  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |
| Проверила: | Большакова-Стрекалова Анна Викторовна  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (оценка)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) |

Калининград, 2023

Оглавление

[Введение 2](#_Toc153449586)

[Основная часть 3](#_Toc153449587)

[Решение задач 3](#_Toc153449588)

[Разработка АИС 26](#_Toc153449589)

[Разработка мобильного приложения 26](#_Toc153449590)

[Сайт и оценка эффективности 26](#_Toc153449591)

[Выводы 27](#_Toc153449592)

[Список использованных источников 28](#_Toc153449593)

[Приложения 29](#_Toc153449594)

# Введение

Цель: подготовится к демоэкзамену, проходя учебную практику.

Задачи:

* выявить свои слабые стороны;
* понять, как устроен демоэкзамен;
* понять то, что нужно доучить, для демоэкзамена;
* вспомнить, как работать с IDE IntelliJ IDEA.

Основная часть разделена на пункты такие как: «Решение задач», «Разработка АИС», «Разработка мобильного приложения», «Сайт и оценка эффективности».

В «Решение задач» будут предоставлены листинги кода программы, блок-схемы и ER-диаграммы программы и показом результатов работы программы.

В «Разработка АИС» будет показан процесс разработки и работы выбранного приложения для разработки, а также взаимосвязь между классами и их реализация.

«Разработка мобильного приложения» особым от «Разработка АИС» отличатся не будет, кроме того, что программа будет реализована на мобильном устройстве.

В «Сайт и оценка эффективности» будет показана работа сайта.

В выводах будет приведён вывод от проделанной работы.

В Списках использованных источников будут размещена использованная литература при прохождении учебной практики.

В приложениях будут размещены ссылки на GitHub и на сайт.

# Основная часть

### Решение задач

Задача 1.

Краткое описание варианта задания:

C5 – Транспонировать квадратную матрицу.

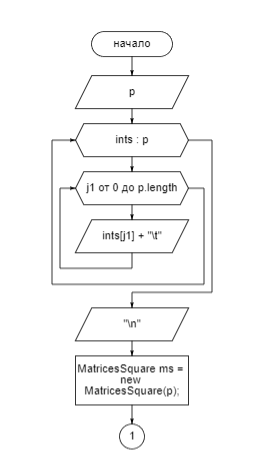
Ход выполнения задания

Num1.java:

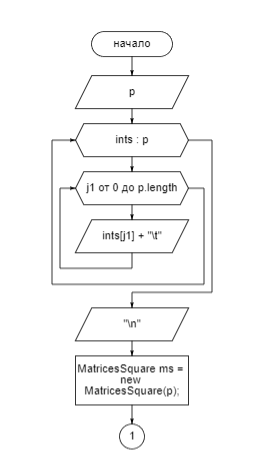
Таблица 1 – идентификаторы класса:

|  |  |  |
| --- | --- | --- |
| Наименование | Тип данных | Назначение |
| p | int[][] | Входящее |
| P1 | int[][] | Выходящее |
| ms | MatricesSquare | Входящее |
| ints | int[] | Выходящее |
| ints2 | int[] | Выходящее |
| j1 | int | Выходящее |
| j2 | int | Выходящее |

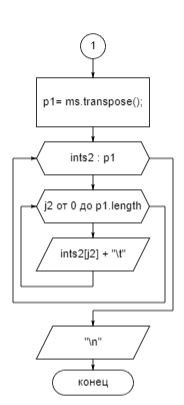
Блок схема:



1



1



Листинг программы:

public class Num1 {

public static void main(String[] args) {

int[][] p = {

{1,2,3},

{4,5,6},

{7,8,9}};

int[][] p1;

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

for (int[] ints : p) {

for (int j1 = 0; j1 < p.length; j1++) {

System.out.print(ints[j1] + "\t");

}

System.out.println();

}

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

System.out.println("Транспонированная Выходная матрица:");

MatricesSquare ms = new MatricesSquare(p);

p1= ms.transpose();

for (int[] ints2 : p1) {

for (int j2 = 0; j2 < p1.length; j2++) {

System.out.print(ints2[j2] + "\t");

}

System.out.println();

}

}

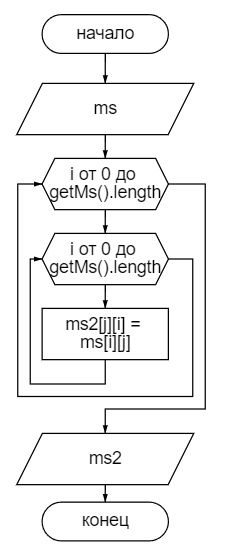
}

MatricesSquare.java:

Таблица 2 – идентификаторы класса:

|  |  |  |
| --- | --- | --- |
| Наименование | Тип данных | Назначение |
| ms | int[][] | Входящее |
| ms2 | int[][] | Выходящее |
| i, j | int | Выходящее |

Блок схема:



Листинг программы:

public class MatricesSquare {

private int [][] ms;

private int [][] ms2;

public MatricesSquare(int [][] ms){

this.ms = ms;

}

public int [][] transpose(){

ms2 = new int[getMs().length][getMs().length];

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

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

ms2[j][i] = ms[i][j];

}

}

return ms2;

}

public int[][] getMs() {

return ms;

}

public void setMs(int[][] ms) {

this.ms = ms;

}

public int[][] getMs2() {

return ms2;

}

public void setMs2(int[][] ms2) {

this.ms2 = ms2;

}

}

Результат выполнения листинга программы:

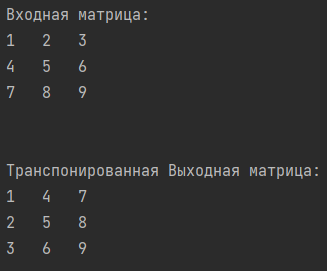


Рисунок 1 – Результат выполнения задания 1

На рисунке 1 показан результат работы задания 1.

Задача 2.

Краткое описание варианта задания:

А9 - Product: id, Наименование, UPC, Производитель, Цена, Срок хранения,

Количество.

Создать массив объектов. Вывести:

a) список товаров для заданного наименования;

b) список товаров для заданного наименования, цена которых не превосходит заданную;

c) список товаров, срок хранения которых больше заданного.

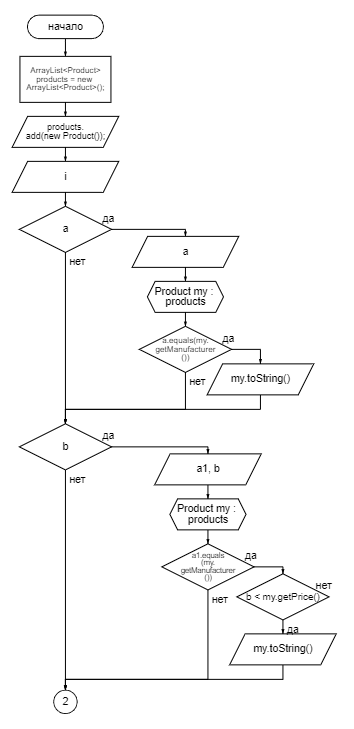
Ход выполнения задания:

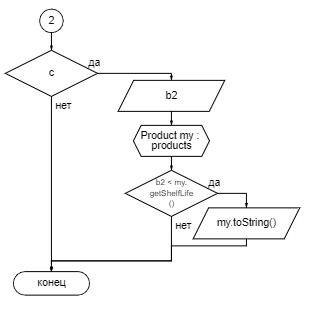
Main.java:

Таблица 3 – идентификаторы класса:

|  |  |  |
| --- | --- | --- |
| Наименование | Тип данных | Назначение |
| i, a, a1 | String | Входящее |
|  |  |  |
| products | Product | Входящее |
| my | Product | Выходящее |
| b | double | Входящее |
| b2 | short | Входящее |
| inputAction, in | Scanner | Входящее |

Блок схема:





Листинг программы:

import java.util.ArrayList;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner inputAction = new Scanner(System.in);

System.out.println("Введите запрос для вывода данных о продуктах:");

System.out.println("a) список товаров для заданного наименования;\n" +

"b) список товаров для заданного наименования, цена которых не превосходит заданную;\n" +

"c) список товаров, срок хранения которых больше заданного.\n\n");

System.out.print("Ввод: ");

String i = inputAction.nextLine();

ArrayList<Product> products = new ArrayList<Product>();

products.add(new Product(1, "Молоко", "123456789012", "Залейский\_фермер", 50.5, (short) 8, 25));

products.add(new Product(2, "Сыр", "123456789012", "Залейский\_фермер", 41.5, (short) 5, 25));

products.add(new Product(3, "Творог", "123456789012", "Залейский\_фермер", 55.5, (short) 9, 25));

products.add(new Product(4, "Колбаса", "123456789012", "Мясник", 50.0, (short) 6, 25));

products.add(new Product(5, "Капчёная рыба", "123456789012", "Мясник", 100.0, (short) 6, 25));

products.add(new Product(6, "Пельмени", "123456789012", "Мясник", 150.30, (short) 6, 25));

products.add(new Product(7, "Яблоко", "123456789012", "Фруктовый\_мир", 30.0, (short) 6, 25));

products.add(new Product(8, "Хлеб", "123456789012", "Пекарь", 45.0, (short) 6, 25));

products.add(new Product(9, "Шоколад", "123456789012", "Шоколадный\_заяц", 220.50, (short) 6, 25));

products.add(new Product(10, "Мороженной", "123456789012", "Залейский\_фермер", 120.0, (short) 7, 25));

switch (i) {

case "a" -> {

System.out.print("a) Ввдите наименование производителя: ");

String a = inputAction.next();

for (Product my : products) {

if (a.equals(my.getManufacturer())) {

System.out.println(my.toString());

}

}

}

case "b" -> {

System.out.print("b) Ввдите наименование производителя: ");

String a1 = inputAction.next();

System.out.print("Ввдите цену: ");

Scanner in = new Scanner(System.in);

double b = in.nextDouble();

for (Product my : products) {

if (a1.equals(my.getManufacturer()) && b >= my.getPrice()) {

System.out.println(my.toString());

}

}

}

case "c" -> {

System.out.print("c) Ввдите срок хранения: ");

short b2 = inputAction.nextShort();

for (Product my : products) {

if (b2 < my.getShelfLife()) {

System.out.println(my.toString());

}

}

}

}

}

}

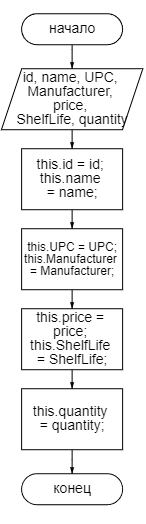
Product.java:

Таблица 4 – идентификаторы класса:

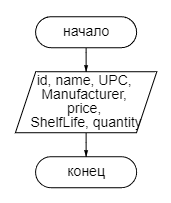
|  |  |  |
| --- | --- | --- |
| Наименование | Тип данных | Назначение |
| id, quantity | int | Входящее/ Выходящее |
| name, UPC, Manufacturer | String | Входящее/ Выходящее |
|  |  |  |
| price | double | Входящее/ Выходящее |
| ShelfLife | short | Входящее/ Выходящее |

Блок схема:

public Product:



toString:



Листинг программы:

public class Product {

private int id;

private String name;

private String UPC;

private String Manufacturer;

private double price;

private short ShelfLife;

private int quantity;

public Product(int id, String name, String UPC, String Manufacturer, double price, short ShelfLife, int quantity) {

this.id = id;

this.name = name;

this.UPC = UPC;

this.Manufacturer = Manufacturer;

this.price = price;

this.ShelfLife = ShelfLife;

this.quantity = quantity;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getUPC() {

return UPC;

}

public void setUPC(String UPC) {

this.UPC = UPC;

}

public String getManufacturer() {

return Manufacturer;

}

public void setManufacturer(String manufacturer) {

Manufacturer = manufacturer;

}

public double getPrice() {

return price;

}

public void setPrice(double price) {

this.price = price;

}

public short getShelfLife() {

return ShelfLife;

}

public void setShelfLife(short shelfLife) {

ShelfLife = shelfLife;

}

public int getQuantity() {

return quantity;

}

public void setQuantity(int quantity) {

this.quantity = quantity;

}

public String toString() {

return "ID: " + id + ", " +

"Наименование: " + name + ", " +

"UPC: " + UPC + ", " +

"Производитель: " + Manufacturer + ", " +

"Цена: " + price + ", " +

"Срок хранения: " + ShelfLife + ", " +

"Количество: " + quantity + ".";

}

}

Результат выполнения листинга программы:

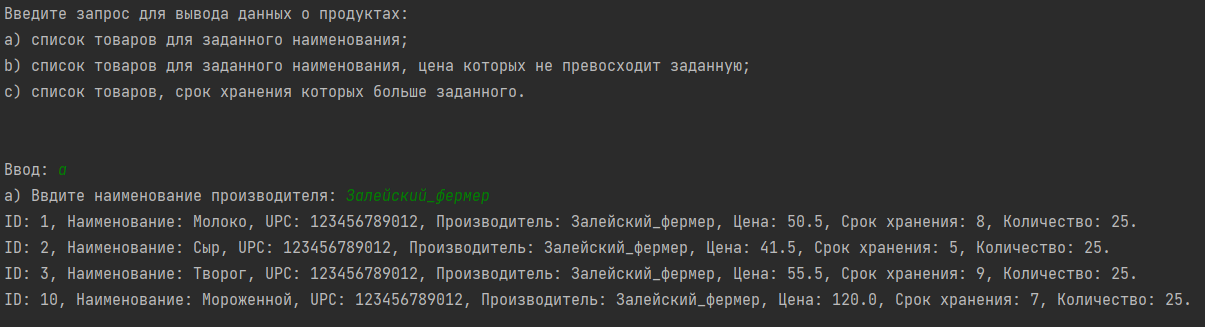


Рисунок 2 – Результат выполнения задания 2, ввод a

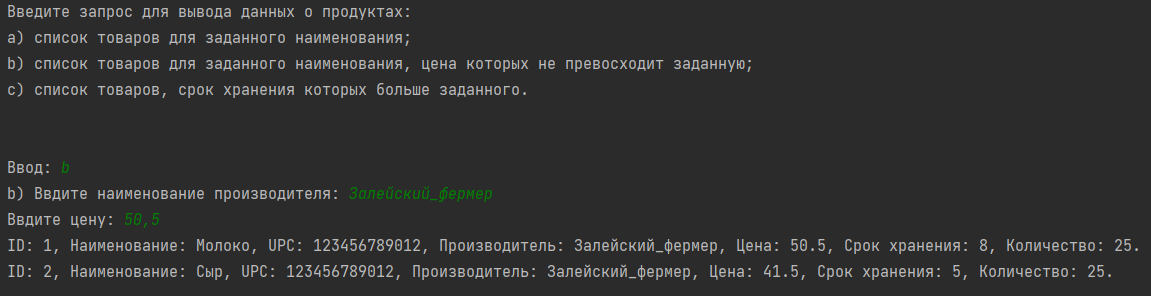


Рисунок 3 – Результат выполнения задания 2, ввод b

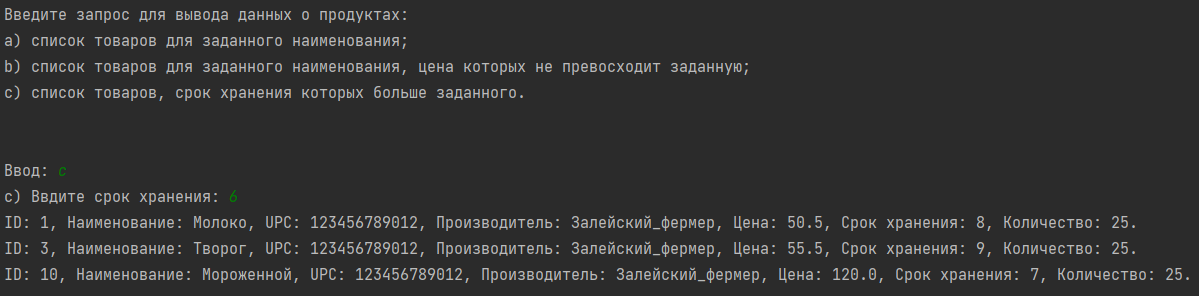


Рисунок 4 – Результат выполнения задания 2, ввод c

На рисунке 2, 3, 4 показан результат работы задания 2.

Задача 3.

Краткое описание варианта задания:

А9 - Создать объект класса Круг, используя классы Точка, Окружность.

Методы: задание размеров, изменение радиуса, определение принадлежности точки данному кругу.

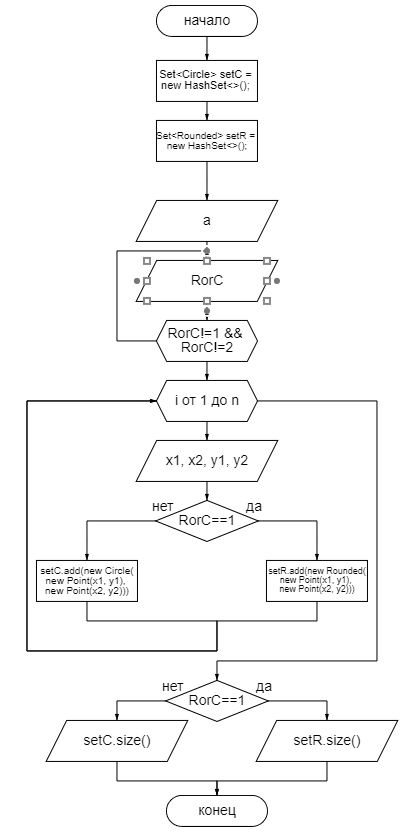
Ход выполнения задания:

Main.java:

Таблица 5 – идентификаторы класса:

|  |  |  |
| --- | --- | --- |
| Наименование | Тип данных | Назначение |
| scanner | Scanner | Входящее |
| a | int | Входящее |
| x1, x2, y1, y2 | double | Входящее |
|  |  |  |
| i | int | Выходящее |
| set | Set<Circle> | Входящее/ Выходящее |

Блок схема:



Листинг программы:

import java.util.HashSet;

import java.util.Scanner;

import java.util.Set;

public class Main {

public static void main(String[] args) {

//Тут произведён показ работы HashCod и Equals.

Set<Circle> setC = new HashSet<>();

Set<Rounded> setR = new HashSet<>();

Scanner scanner = new Scanner(System.in);

System.out.print("Введите количество элементов вводимого массива: ");

int a = scanner.nextInt();

byte RorC;

do{

System.out.println("Введите: \n\t1 - Rounded; \n\t2 - Circle");

System.out.print("Ввод: ");

RorC = scanner.nextByte();}while (RorC!=1 && RorC!=2);

System.out.println("Введите координаты точки.");

double x1, x2, y1, y2;

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

System.out.println("Фигура №"+i);

System.out.print("\tВведите x1: ");

x1=scanner.nextDouble();

System.out.print("\tВведите x2: ");

x2=scanner.nextDouble();

System.out.print("\tВведите y1: ");

y1=scanner.nextDouble();

System.out.print("\tВведите y2: ");

y2=scanner.nextDouble();

if (RorC==1) setR.add(new Rounded(new Point(x1, y1), new Point(x2, y2)));

else setC.add(new Circle(new Point(x1, y1), new Point(x2, y2)));

}

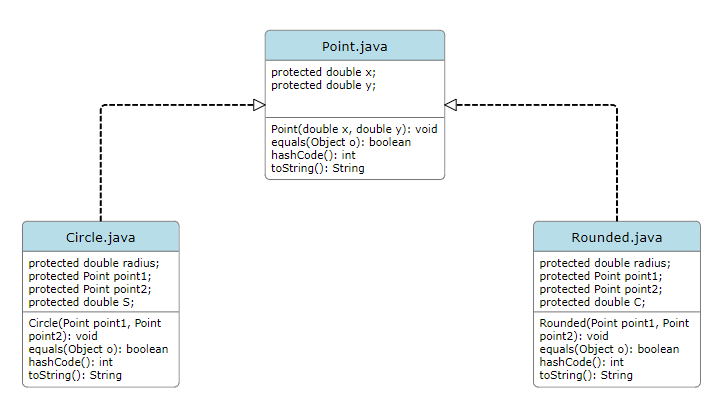
if (RorC==1) System.out.println("Размер массива:"+setR.size());

else System.out.println("Размер массива:"+setC.size());

}

}

UseCase-диаграмма



Листинг Point.java:

import java.util.Objects;

public class Point {

protected double x;

protected double y;

public Point(double x, double y) {

this.x = x;

this.y = y;

}

public double getX() {

return x;

}

public void setX(double x) {

this.x = x;

}

public double getY() {

return y;

}

public void setY(double y) {

this.y = y;

}

@Override

public boolean equals(Object o) {

if (this == o) return true;

if (o == null || getClass() != o.getClass()) return false;

Point point = (Point) o;

return Double.compare(point.x, x) == 0 &&

Double.compare(point.y, y) == 0;

}

@Override

public int hashCode() {

return Objects.hash(x, y);

}

@Override

public String toString() {

return "Point{" +

"x=" + x +

", y=" + y +

'}';

}

}

Листинг Rounder.java:

public class Rounded extends Point {

protected double radius;

protected Point point1;

protected Point point2;

protected double C;

public Rounded(Point point1, Point point2) {

super(point1.x, point1.y);

this.point1 = point1;

this.point2 = point2;

this.radius = Math.sqrt(Math.pow((point1.x - point2.x), 2) + Math.pow((point1.y - point2.y), 2));

C = 2 \* Math.PI \* radius;

}

public Point getPoint1() {

return point1;

}

public void setPoint1(Point point1) {

this.point1 = point1;

}

public Point getPoint2() {

return point2;

}

public void setPoint2(Point point2) {

this.point2 = point2;

}

public double getRadius() {

return radius;

}

public void setRadius(double radius) {

this.radius = radius;

}

public double getC() {

return C;

}

public void setC(double c) {

C = c;

}

@Override

public boolean equals(Object o) {

System.out.println("Equals -> "+ this + " : " + o);

if (this == o) return true;

if (o == null || getClass() != o.getClass()) return false;

if (!super.equals(o)) return false;

Rounded rounded = (Rounded) o;

return Double.compare(rounded.radius, radius) == 0 &&

Double.compare(rounded.C, C) == 0 &&

Objects.equals(point1, rounded.point1) &&

Objects.equals(point2, rounded.point2);

}

@Override

public int hashCode() {

System.out.println("HashCode -> "+ this);

return Objects.hash(super.hashCode(), radius, point1, point2, C);

}

@Override

public String toString() {

return "Rounded{" +

"radius=" + radius +

", point1=" + point1 +

", point2=" + point2 +

", C=" + C +

'}';

}

}

Листинг Circle.java:

import java.util.Objects;

public class Circle extends Point {

protected double radius;

protected Point point1;

protected Point point2;

protected double S;

public Circle(Point point1, Point point2) {

super(point1.x, point1.y);

this.point1 = point1;

this.point2 = point2;

this.radius = Math.sqrt(Math.pow((point1.x - point2.x), 2) + Math.pow((point1.y - point2.y), 2));

S = Math.PI\* Math.pow(radius, 2);

}

public double getRadius() {

return radius;

}

public void setRadius(double radius) {

this.radius = radius;

}

public Point getPoint1() {

return point1;

}

public void setPoint1(Point point1) {

this.point1 = point1;

}

public Point getPoint2() {

return point2;

}

public void setPoint2(Point point2) {

this.point2 = point2;

}

public double getS() {

return S;

}

public void setS(double s) {

S = s;

}

@Override

public boolean equals(Object o) {

System.out.println("Equals -> "+ this + " : " + o);

if (this == o) return true;

if (o == null || getClass() != o.getClass()) return false;

if (!super.equals(o)) return false;

Circle circle = (Circle) o;

return Double.compare(circle.radius, radius) == 0 &&

Double.compare(circle.S, S) == 0 &&

Objects.equals(point1, circle.point1) &&

Objects.equals(point2, circle.point2);

}

@Override

public int hashCode() {

System.out.println("HashCode -> "+ this);

return Objects.hash(super.hashCode(), radius, point1, point2, S);

}

@Override

public String toString() {

return "Circle{" +

"radius=" + radius +

", point1=" + point1 +

", point2=" + point2 +

", S=" + S +

'}';

}

Результат выполнения листинга программы:

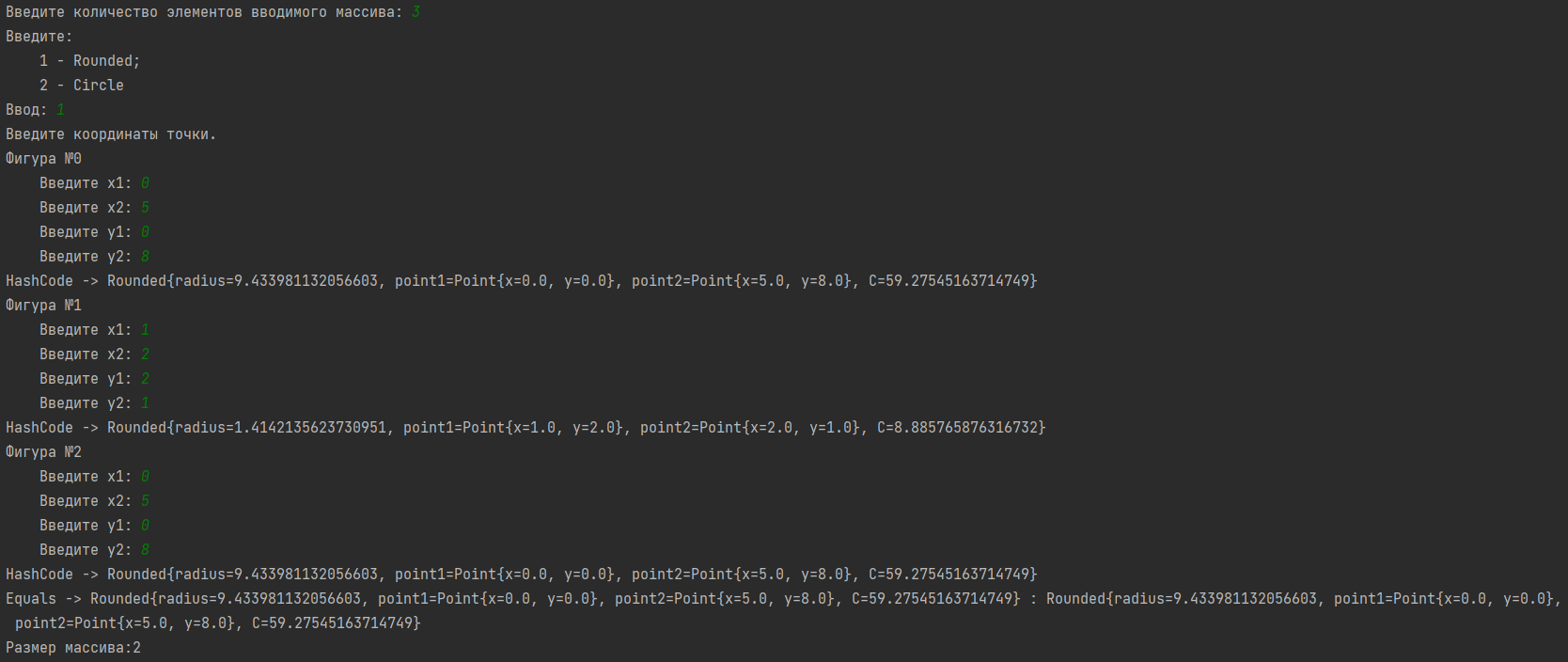


Рисунок 5 – Результат выполнения программы, выбран Rounder

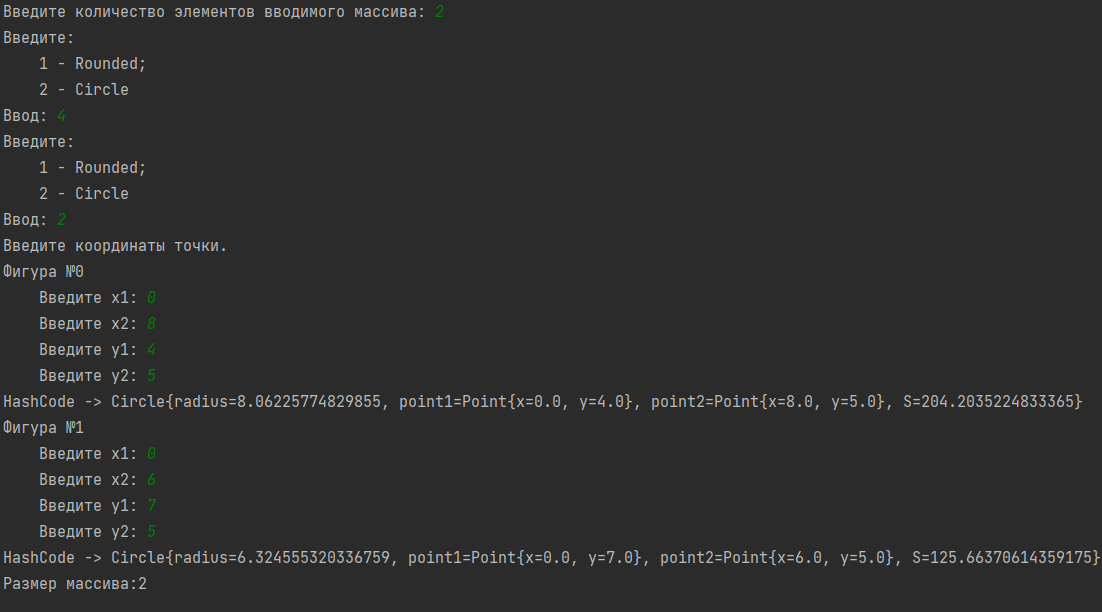


Рисунок 6 – Результат выполнения программы, выбран Circle

На рисунке 5, 6 показан результат работы задания 3. На рисунке 5 были введены одинаковых данных и тем самым было продемонстрирована работа hashcod-а и equals-а. На рисунке 6 была допущена ошибка и при выборе Rounder или Circle по вводу из id, был введён id который к ним не относится, и программа по вводу id запросила ещё раз ввести, но только уже правильный id.

Задача 4.

Краткое описание варианта задания:

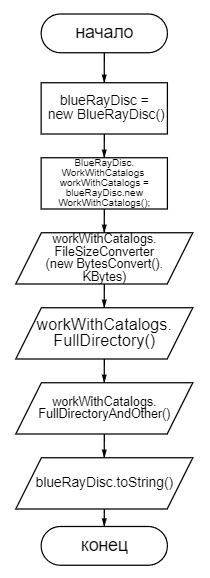
Ход выполнения задания

Main.java:

Таблица 6 – идентификаторы класса:

|  |  |  |
| --- | --- | --- |
| Наименование | Тип данных | Назначение |
| blueRayDisc | BlueRayDisc | Выходящее |
| workWithCatalogs | BlueRayDisc.WorkWithCatalogs | Выходящее |

Блок схема:



Листинг программы:

public class Main {

public static void main(String[] args) {

BlueRayDisc blueRayDisc = new BlueRayDisc("C:", "C:/Users/ko-ma/Desktop/Пока Не Нужное/ЛПЗ по Плешкову",

"1\_4\_Optimizatsia\_BD\_2","docx",33792, "02.03.2023","15:03");

BlueRayDisc.WorkWithCatalogs workWithCatalogs = blueRayDisc.new WorkWithCatalogs();

System.out.println("Кбайт программы: "+workWithCatalogs. FileSizeConverter (new BytesConvert().KBytes));

System.out.println(workWithCatalogs.FullDirectory());

System.out.println(workWithCatalogs.FullDirectoryAndOther());

System.out.println(blueRayDisc.toString());

}

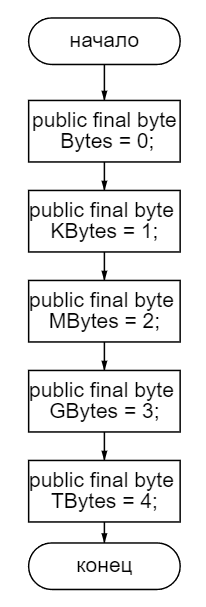
}

BytesConvert.java:

Таблица 7 – идентификаторы класса:

|  |  |  |
| --- | --- | --- |
| Наименование | Тип данных | Назначение |
| Bytes | byte | Выходящее |
| KBytes | byte | Выходящее |
| MBytes | byte | Выходящее |
| GBytes | byte | Выходящее |
| TBytes | byte | Выходящее |

Блок схема:



Листинг программы:

public class BytesConvert {

public final byte Bytes = 0;

public final byte KBytes = 1;

public final byte MBytes = 2;

public final byte GBytes = 3;

public final byte TBytes = 4;

}

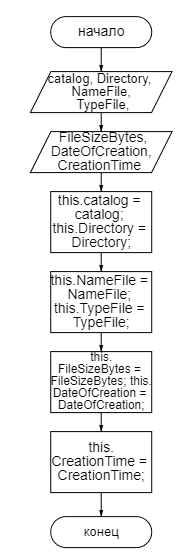
BlueRayDisc.java:

Таблица 8 – идентификаторы класса:

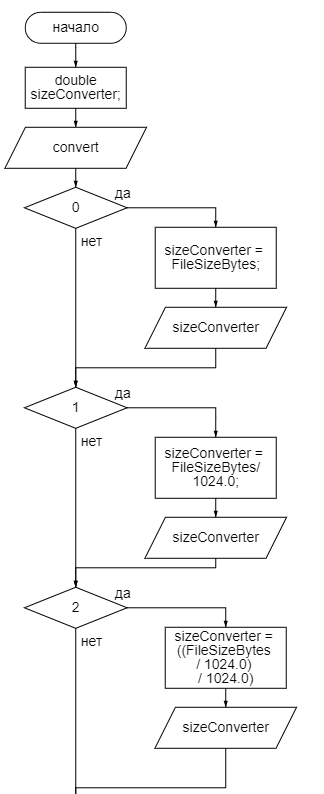
|  |  |  |
| --- | --- | --- |
| Наименование | Тип данных | Назначение |
| catalog | String | Входящее/ Выходящее |
| Directory | String | Входящее/ Выходящее |
| NameFile | String | Входящее/ Выходящее |
| TypeFile | String | Входящее/ Выходящее |
| FileSizeBytes | double | Входящее/ Выходящее |
| DateOfCreation | String | Входящее/ Выходящее |
| CreationTime | String | Входящее/ Выходящее |

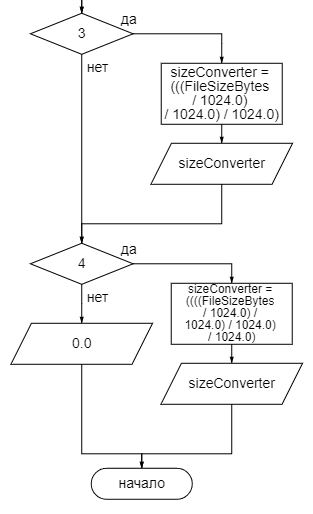
Блок схема:

public BlueRayDisc():



WorkWithCatalogs().FileSizeConverter():





Листинг программы:

public class BlueRayDisc {

private String catalog;

private String Directory;

private String NameFile;

private String TypeFile;

private double FileSizeBytes;

private String DateOfCreation;

private String CreationTime;

public BlueRayDisc(String catalog, String Directory, String NameFile,

String TypeFile, double FileSizeBytes, String DateOfCreation,

String CreationTime) {

this.catalog = catalog;

this.Directory = Directory;

this.NameFile = NameFile;

this.TypeFile = TypeFile;

this.FileSizeBytes = FileSizeBytes;

this.DateOfCreation = DateOfCreation;

this.CreationTime = CreationTime;

}

public String getCatalog() {

return catalog;

}

public void setCatalog(String catalog) {

this.catalog = catalog;

}

public String getDirectory() {

return Directory;

}

public void setDirectory(String directory) {

Directory = directory;

}

public String getNameFile() {

return NameFile;

}

public void setNameFile(String nameFile) {

NameFile = nameFile;

}

public String getTypeFile() {

return TypeFile;

}

public void setTypeFile(String typeFile) {

TypeFile = typeFile;

}

public double getFileSizeBytes() {

return FileSizeBytes;

}

public void setFileSizeBytes(double fileSizeBytes) {

FileSizeBytes = fileSizeBytes;

}

public String getDateOfCreation() {

return DateOfCreation;

}

public void setDateOfCreation(String dateOfCreation) {

DateOfCreation = dateOfCreation;

}

public String getCreationTime() {

return CreationTime;

}

public void setCreationTime(String creationTime) {

CreationTime = creationTime;

}

@Override

public String toString() {

return "CatalogFile{" +

"catalog='" + catalog + '\'' +

", Directory='" + Directory + '\'' +

", NameFile='" + NameFile + '\'' +

", TypeFile='" + TypeFile + '\'' +

", FileSizeBite=" + FileSizeBytes +

", DateOfCreation='" + DateOfCreation + '\'' +

", CreationTime='" + CreationTime + '\'' +

'}';

}

public class WorkWithCatalogs {

public double FileSizeConverter(byte convert) {

double sizeConverter;

switch (convert) {

case 0 -> {

sizeConverter = FileSizeBytes;

return sizeConverter;

}

case 1 -> {

sizeConverter = FileSizeBytes / 1024.0;

return sizeConverter;

}

case 2 -> {

sizeConverter = ((FileSizeBytes / 1024.0) / 1024.0);

return sizeConverter;

}

case 3 -> {

sizeConverter = (((FileSizeBytes / 1024.0) / 1024.0) / 1024.0);

return sizeConverter;

}

case 4 -> {

sizeConverter = ((((FileSizeBytes / 1024.0) / 1024.0) / 1024.0) / 1024.0);

return sizeConverter;

}

default -> {

return 0.0;

}

}

}

public String FullDirectoryAndOther() {

return Directory + "/" + NameFile + "." + TypeFile + "|\t|" +

FileSizeBytes + " bytes|\t|" + DateOfCreation + "|\t|" + CreationTime+"|";

}

public String FullDirectory() {

return Directory + "/" + NameFile + "." + TypeFile;

}

}

}

Результат выполнения листинга программы:

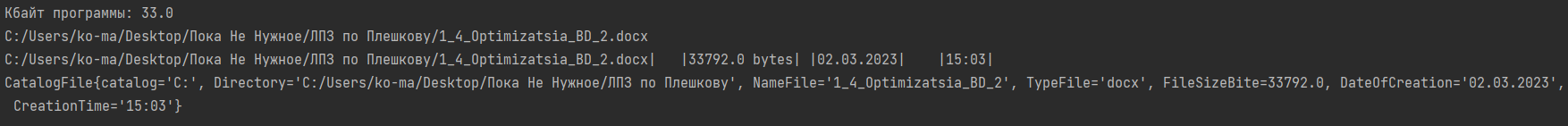


Рисунок 7 – Результат выполнения задания 1

На рисунке 7 показан результат работы задания 4, изображается вывод размера программы, его расположение.

### Разработка АИС

### Разработка мобильного приложения

### 

### Сайт и оценка эффективности

# Выводы

При выполнении отчёта я столкнутся со многими сложностями, и в отчёте было сделано не всё, но стало понятно, в чём мои проблемы и что мне нужно вспомнить и подтянуть, а что изучить.

# Список использованных источников

1. Java методы программирования. Учебно-методическое пособие. / И. Н. Блинов, В. С. Романчик / МИНСК ИЗДАТЕЛЬСТВО «ЧЕТЫРЕ ЧЕТВЕРТИ» / 2013 г.

# Приложения

1. Ссылка на GitHub - <https://github.com/1mishakor/UP>;