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 «17» ноября 2022 г. по «07» декабря 2022 г.

Место практики ГБУ КО ПОО «КИТиС»

|  |  |
| --- | --- |
| Выполнил: | студентка 4 курса,  группы ИСп 19-1  Полишките Анна Алексеевга  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |
| Проверила: | Большакова-Стрекалова Анна Викторовна  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (оценка)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) |

Калининград, 2022

Содержание

|  |  |
| --- | --- |
| Введение | 3 |
| 1 Решение задач | 4 |
| 1.1 Задача 1 | 4 |
| 1.2 Задача 2 | 6 |
| 1.3 Задача 3 | 8 |
| 1.4 Задача 4 | 9 |
| 1.5 Задача 5 | 15 |
| 1.6 Задача 6 | 16 |
| 1.7 Задача 7 | 17 |
| Заключение | 21 |
| Список использованных источников | 22 |

1. Введение

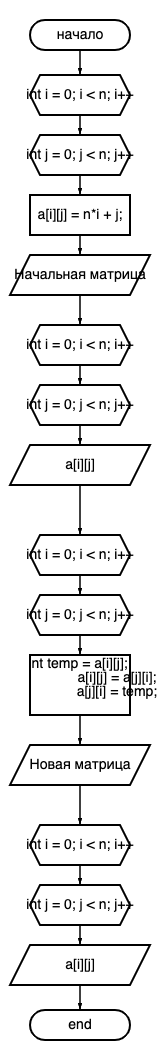
Учебная практика по модулю ПМ 01 «Разработка модулей программного обеспечения для компьютерных систем» проходит на базе колледжа.

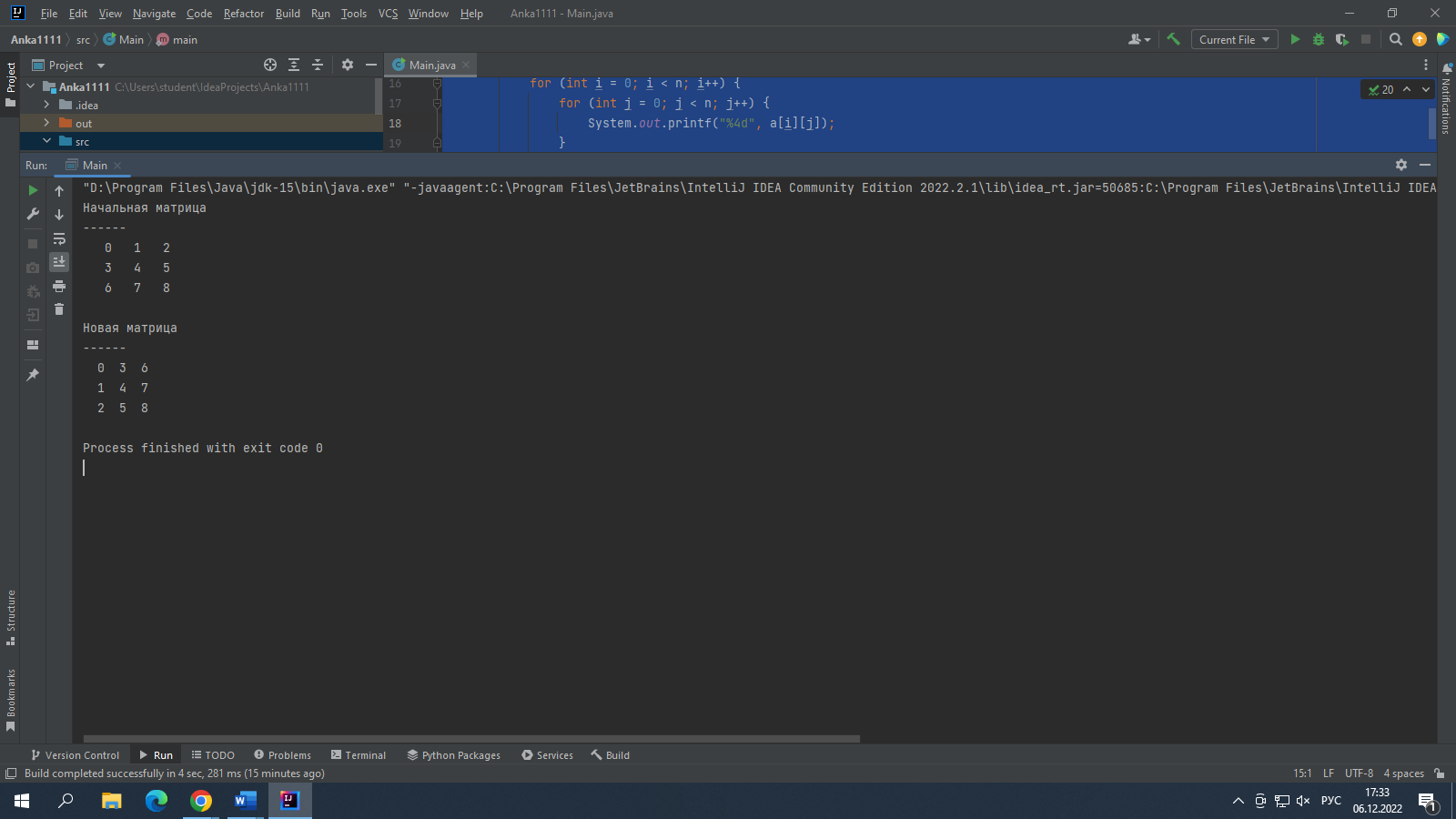
Данная практика являет способам показать и закрепить полученные знания в областях разработки автоматизированных информационных систем, разработки мобильных приложений, развитие профессиональных навыков в области проектирования баз данных и проектирования программных интерфейсов.

Задачами практики являются разработка приложений по работе с матрицами, классами и вложенными классами, работе с циклами, разработки визуального интерфейса, проектирование базы данных и мобильного приложения. Также в ходе работы требуется составить техническое задание и разработать диаграммы.

Задание 1

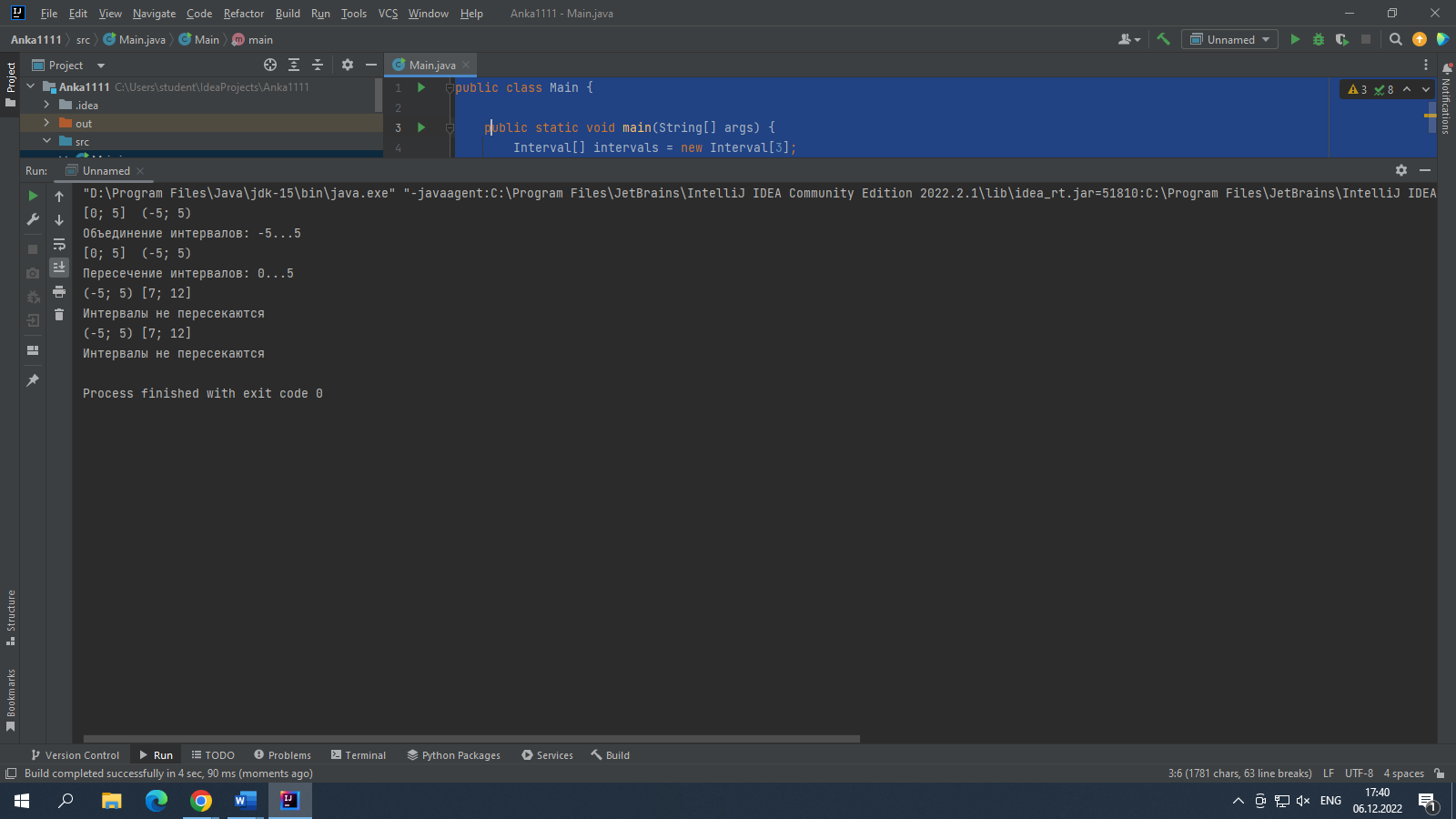
public class Main {  
 public static void main(String[] args) {  
  
 // часть 1 - создаем матрицу n на n  
 int n = 3;  
 int[][] a = new int[n][n];  
 for (int i = 0; i < n; i++) {  
 for (int j = 0; j < n; j++) {  
 a[i][j] = n\*i + j;  
 }  
 }  
  
 // часть 2 - выводит на экран начальную матрицу  
 System.*out*.println("Начальная матрица");  
 System.*out*.println("------");  
 for (int i = 0; i < n; i++) {  
 for (int j = 0; j < n; j++) {  
 System.*out*.printf("%4d", a[i][j]);  
 }  
 System.*out*.println();  
 }  
  
 // часть 3 - транспонирование матрицы  
 for (int i = 0; i < n; i++) {  
 for (int j = i+1; j < n; j++) {  
 int temp = a[i][j];  
 a[i][j] = a[j][i];  
 a[j][i] = temp;  
 }  
 }  
  
 // часть 4 - выводит на экран транспонированную матрицу  
 System.*out*.println();  
 System.*out*.println("Новая матрица");  
 System.*out*.println("------");  
 for (int i = 0; i < n; i++) {  
 for (int j = 0; j < n; j++) {  
 System.*out*.printf("%3d", a[i][j]);  
 }  
 System.*out*.println();  
 }  
  
 }  
 }





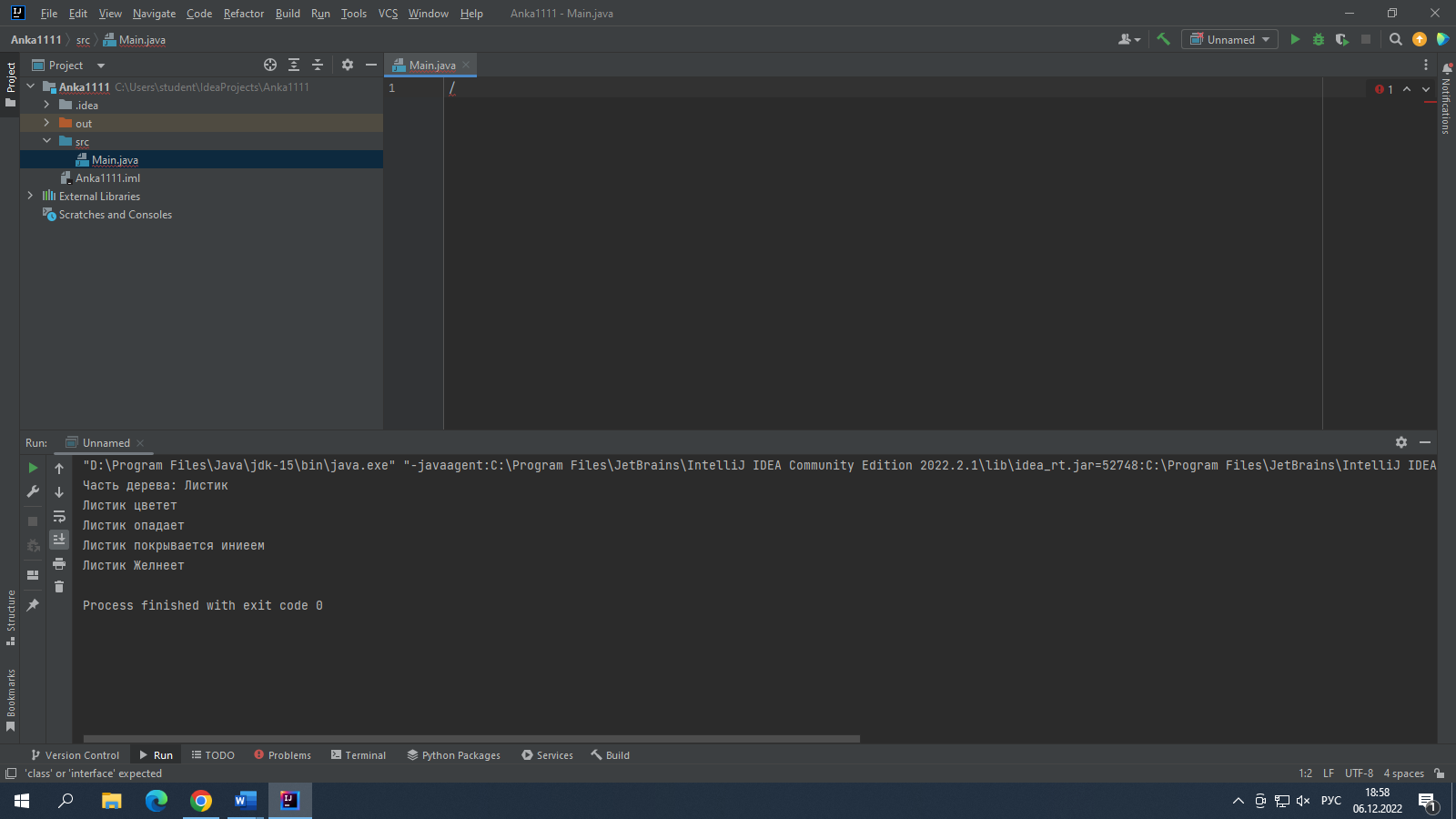
Задание 2

public class Main {  
  
 public static void main(String[] args) {  
 Interval[] intervals = new Interval[3];  
  
 intervals[0] = new Interval(0, 5, true);  
 intervals[1] = new Interval(-5, 5, false);  
 intervals[2] = new Interval(7, 12, true);  
  
 for (int i = 0; i < intervals.length - 1; i++) {  
 intervals[i].combine(intervals[i + 1]);  
 intervals[i].crossing(intervals[i + 1]);  
 }  
 }  
}  
class Interval {  
 private int start, end;  
 private boolean isInclude;  
  
 Interval(int start, int end, boolean isInclude) {  
 this.start = start;  
 this.end = end;  
 this.isInclude = isInclude;  
 }  
  
 private int getStart() {  
 return start;  
 }  
  
 private int getEnd() {  
 return end;  
 }  
  
 @Override  
 public String toString() {  
 String s1, s2;  
 if (isInclude) {  
 s1 = "[";  
 s2 = "]";  
 } else {  
 s1 = "(";  
 s2 = ")";  
 }  
 return s1 + start + "; " + end + s2;  
 }  
  
 void combine(Interval inter) {  
 System.*out*.println(this + "\t" + inter);  
 if (start < inter.getEnd() && inter.getStart() < end) {  
 int x = Math.*min*(start, inter.getStart());  
 int y = Math.*max*(end, inter.getEnd());  
 System.*out*.println("Объединение интервалов: " + x + "..." + y);  
 } else System.*out*.println("Интервалы не пересекаются");  
 }  
  
 void crossing(Interval inter) {  
 System.*out*.println(this + "\t" + inter);  
 int x = Math.*max*(start, inter.getStart());  
 int y = Math.*min*(end, inter.getEnd());  
 if (x < y) System.*out*.println("Пересечение интервалов: " + x + "..." + y);  
 else System.*out*.println("Интервалы не пересекаются");  
 }  
}



Задание 3

public class Main {  
 public static void main(String[] args) {  
 branch branch = new branch("Листик");  
  
 branch.showName();  
 branch.fly();  
 branch.sing();  
 branch.laysAnEgg();  
 branch.hatchingChicks();  
 }  
}  
class Listt{  
 String kind;  
 public Listt(String kind){  
 this.kind = kind;  
 }  
 public void showName (){  
 System.*out*.println("Часть дерева: " + kind );  
 }  
}  
class branch extends Listt {  
 public branch(String name) {  
 super(name);  
 }  
 public void showName(){  
 System.*out*.println("Часть дерева: " + kind);  
 }  
 public void fly(){  
 System.*out*.println(kind + " цветет");  
 }  
 public void sing(){  
 System.*out*.println(kind + " опадает");  
 }  
 public void laysAnEgg(){  
 System.*out*.println(kind + " покрывается иниеем ");  
 }  
 public void hatchingChicks(){  
 System.*out*.println(kind + " Желнеет");  
 }  
}



Задание 4

import java.util.ArrayList;

public class Main {

public static void main(String[] args) {

ArrayList<Cinema> Art = new ArrayList<>();

Art.add(new address("Кирова 65"));

Art.add(new address("Мира 8"));

Art.add(new records("16:30"));

Art.add(new records("18:50"));

Art.add(new film("Терминатор"));

Art.add(new film("Сумерки"));

print(Art);

}

public static void print(ArrayList<Cinema> city) {

System.out.println("Кино:");

for (Cinema object: city) {

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

}

}

}

class Cinema {

String name;

String info = "";

public void setName(String name) {

this.name = name;

}

public String getName() {

return name;

}

public String getInfo() {

return info;

}

public void setInfo(String info) {

this.info = info;

}

@Override

public String toString() {

return "CityObject{" +

"name='" + name + '\'' +

'}';

}

public void printInfo() {

System.out.println(info);

}

public Cinema(String name) {

this.name = name;

}

}

class address extends Cinema {

public address(String name) {

super(name);

}

public address(String name, String info) {

super(name);

this.info = info;

}

@Override

public String toString() {

if (info.isEmpty()) {

return "Адрес - " + name;

} else {

return "Адрес - " + name + " - " + info;

}

}

}

class film extends Cinema {

public film(String name) {

super(name);

}

public film(String name, String info) {

super(name);

this.info = info;

}

@Override

public String toString() {

if (info.isEmpty()) {

return "Фильм - " + name;

} else {

return "Фильм - " + name + " - " + info;

}

}

}

class records extends Cinema {

public records(String name) {

super(name);

}

public records(String name, String info) {

super(name);

this.info = info;

}

@Override

public String toString() {

if (info.isEmpty()) {

return "Время сеанса - " + name;

} else {

return "Время сеанса - " + name + " - " + info;

}

}

}

import java.util.ArrayList;

public class Main {

public static void main(String[] args) {

ArrayList<Cinema> Art = new ArrayList<>();

Art.add(new address("Кирова 65"));

Art.add(new address("Мира 8"));

Art.add(new records("16:30"));

Art.add(new records("18:50"));

Art.add(new film("Терминатор"));

Art.add(new film("Сумерки"));

print(Art);

}

public static void print(ArrayList<Cinema> city) {

System.out.println("Кино:");

for (Cinema object: city) {

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

}

}

}

class Cinema {

String name;

String info = "";

public void setName(String name) {

this.name = name;

}

public String getName() {

return name;

}

public String getInfo() {

return info;

}

public void setInfo(String info) {

this.info = info;

}

@Override

public String toString() {

return "CityObject{" +

"name='" + name + '\'' +

'}';

}

public void printInfo() {

System.out.println(info);

}

public Cinema(String name) {

this.name = name;

}

}

class address extends Cinema {

public address(String name) {

super(name);

}

public address(String name, String info) {

super(name);

this.info = info;

}

@Override

public String toString() {

if (info.isEmpty()) {

return "Адрес - " + name;

} else {

return "Адрес - " + name + " - " + info;

}

}

}

class film extends Cinema {

public film(String name) {

super(name);

}

public film(String name, String info) {

super(name);

this.info = info;

}

@Override

public String toString() {

if (info.isEmpty()) {

return "Фильм - " + name;

} else {

return "Фильм - " + name + " - " + info;

}

}

}

class records extends Cinema {

public records(String name) {

super(name);

}

public records(String name, String info) {

super(name);

this.info = info;

}

@Override

public String toString() {

if (info.isEmpty()) {

return "Время сеанса - " + name;

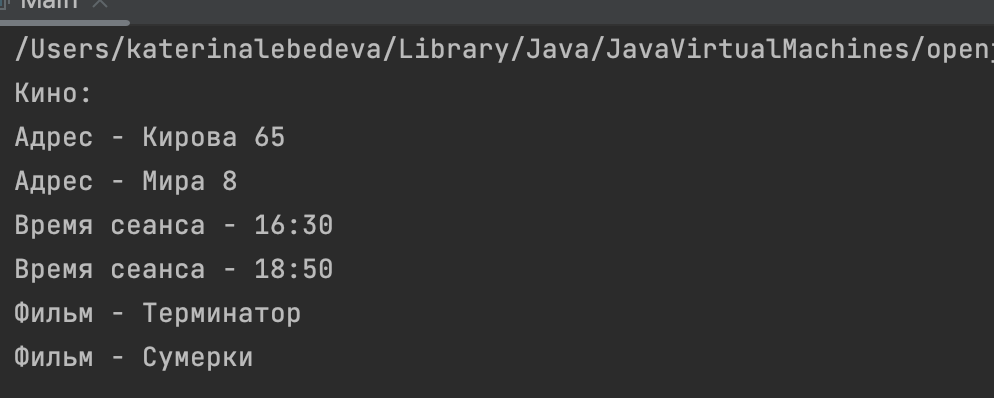
} else {

return "Время сеанса - " + name + " - " + info;

}

}

}



Задание 5

Main.java

package com.company;

public class Main

{

public static void main(String[] args)

{

StringBuffer text = new StringBuffer("Java - an object-oriented programming language \ndeveloped by Sun Microsystems, later acquired by Oracle.\nJava applications are usually compiled into a \nspecial byte-code, so they can run on any Java-virtual \nmachine, regardless of computer architecture.");

String subA="WordInsert ";

System.out.println(text);

int i=0, pos1=0;

while ((pos1 = text.indexOf("ed ", i)) != -1)

{

pos1 += 3;

text.insert(pos1, subA);

i = pos1;

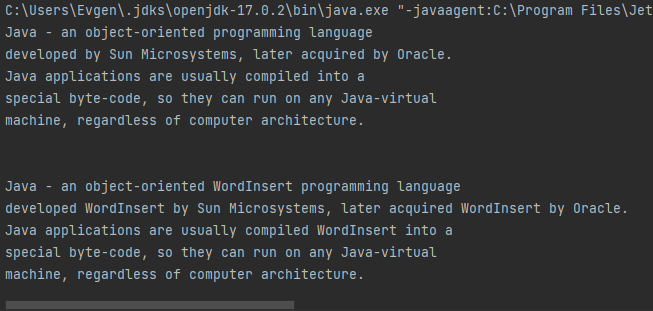
}

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

}

}

Результат:



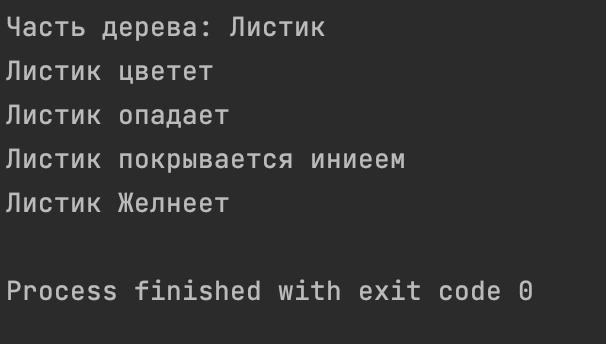
Задание 6

public class Main {  
 public static void main(String[] args) {  
 branch branch = new branch("Листик");  
  
 branch.showName();  
 branch.fly();  
 branch.sing();  
 branch.laysAnEgg();  
 branch.hatchingChicks();  
 }  
}  
class Listt{  
 String kind;  
 public Listt(String kind){  
 this.kind = kind;  
 }  
 public void showName (){  
 System.*out*.println("Часть дерева: " + kind );  
 }  
}  
class branch extends Listt {  
 public branch(String name) {  
 super(name);  
 }  
 public void showName(){  
 System.*out*.println("Часть дерева: " + kind);  
 }  
 public void fly(){  
 System.*out*.println(kind + " цветет");  
 }  
 public void sing(){  
 System.*out*.println(kind + " опадает");  
 }  
 public void laysAnEgg(){  
 System.*out*.println(kind + " покрывается иниеем ");  
 }  
 public void hatchingChicks(){  
 System.*out*.println(kind + " Желнеет");  
 try {} catch (Exception e) {}

}

}

Результат:



Задание 7

import java.awt.Color;

import java.awt.Dimension;

import java.awt.Graphics;

import java.awt.Graphics2D;

import java.awt.image.BufferedImage;

import javax.swing.ImageIcon;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JLayeredPane;

import javax.swing.JPanel;

public class Main

{

public JFrame win;

public JLayeredPane swap;

public Main()

{

makeWindow();

makeStar(new int[]{50,50,50,100,36});//положение на экране по Х, положение на экране по Y, малый радиус, большой радиус, количество лучей

}

private void makeWindow()

{

JPanel panel = new JPanel();

panel.setPreferredSize(new Dimension(300, 300));

panel.setOpaque(false);

win = new JFrame("Star");

swap = new JLayeredPane();

swap = win.getLayeredPane();

win.add(panel);

win.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

win.setResizable(false);

win.getContentPane().setBackground(new Color(0,191,255));

win.pack();

win.setLayout(null);

win.setLocationRelativeTo(null);

win.setVisible(true);

}

private void makeStar(int[] base)

{

int[] x = new int[base[4]\*2];

int[] y = new int[base[4]\*2];

for (int i = 0; i < base[4]\*2; i++)

{

x[i] = (int)(((i%2)==0 ? base[3] : base[2])\*Math.cos(((360/(base[4]\*2)\*i)-((360/(base[4]\*2)/2)))\*Math.PI/180))+0;

y[i] = (int)(((i%2)==0 ? base[3] : base[2])\*Math.sin(((360/(base[4]\*2)\*i)-((360/(base[4]\*2)/2)))\*Math.PI/180))+0;

}

BufferedImage bf = new BufferedImage(base[3]\*2, base[3]\*2, BufferedImage.TYPE\_INT\_ARGB);

Graphics g = bf.createGraphics();

Graphics2D g2 = (Graphics2D)g;

g2.setColor(new Color(255,255,28));

g2.translate(base[3], base[3]);

g2.fillPolygon(x, y, base[4]\*2);

g2.dispose();

JLabel jl = new JLabel();

jl.setBounds(base[0], base[1], base[3]\*2, base[3]\*2);

jl.setLayout(null);

jl.setOpaque(false);

jl.setIcon(new ImageIcon(bf));

jl.setVisible(true);

swap.add(jl, 0, 0);

bf.flush();

}

public static void main(String[] args)

{

new Main();

}

}

Результат:



Заключение

В ходе прохождения практики были освоены навыки по разработке автоматических информационных систем, закреплены знания в области проектирования программных интерфейсов и баз данных.

Цели и задачи практики достигнуты. Разработаны алгоритмы решения задач с матрицами и циклами.

Список использованных источников

1. Google Sites – <https://sites.google.com/>
2. Онлайн-редактор блок-схем – <https://programforyou.ru/>
3. учебник «Java – Методы  программирования»
4. Семакин И. Г. – «Основы алгоритмизации и программирования»
5. Руководство по использованию - <https://habr.com/ru/post/125799/>