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»

|  |  |
| --- | --- |
| Выполнила: | студентка 4 курса,  группы ИСп 19-1  Осипова Арина Вячеславовна  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (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 4 | 6 |
| 1.4 Задача 5 | 8 |
| 1.5 Задача 9 | 9 |
| 1.6 Задача 10 | 16 |
| 2 Разработка АИС «Словарь» | 18 |
| 2.1 ER-диаграмма сущность-связь | 18 |
| 3 Создание сайта | 19 |
| Заключение | 22 |
| Список использованных источников | 23 |

Введение

Учебная практика по модулю ПМ 01 разработка программного обеспечения компьютерных систем проходит на базе колледжа.

Целью практики является закрепление знаний в разработке различного рода программ, разработка информационной базы, а также создание приложение и последующая разработка сайта.

Задачами практики является разработка приложений по работе с матрицами, классами и вложенными классами, работе с циклами, разработки визуального интерфейса, проектирование базы данных и мобильного приложения.

Задание №1

Найти максимальный элемент (ы) в матрице и удалить из матрицы все строки и столбцы, его содержащие.

Код программы:

import java.util.ArrayList;

import java.util.Random;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

int n;

try {

n = Integer.valueOf(s.nextLine());

} catch (NumberFormatException e) {

return;

}

int m;

try {

m = Integer.valueOf(s.nextLine());

} catch (NumberFormatException e) {

return;

}

int[][] arr = new int[n][n];

Random random = new Random();

int max = -m;

ArrayList<Integer> rows = new ArrayList<Integer>();

ArrayList<Integer> column = new ArrayList<Integer>();

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

arr[i] = random.ints(n, -m, m).toArray();

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

if (arr[i][j] > max) {

max = arr[i][j];

}

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

}

System.out.println();

}

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

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

if (max == arr[i][j]) {

if (rows.indexOf(j) == -1)

rows.add(j);

if (column.indexOf(i) == -1)

column.add(i);

}

}

}

System.out.println("\nMax: " + max);

int[][] newArr = new int[n - rows.size()][n - column.size()];

System.out.println("Size: " + (n - rows.size()) + "x" + (n - column.size()));

int newArrI = 0;

int newArrJ = 0;

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

if (rows.indexOf(i) == -1) {

newArrJ = 0;

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

if (column.indexOf(j) == -1) {

newArr[newArrI][newArrJ]

= arr[j][i];

newArrJ++;

}

}

newArrI++;

}

}

}

}

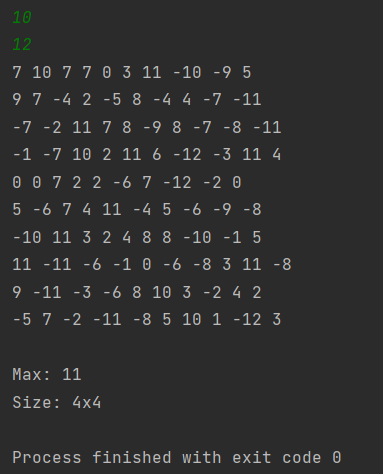


Рисунок 1 – результат задания №1

2 задание

Определить класс Комплекс. Создать массив/список/множество размерности n из комплексных координат. Передать его в метод, который выполнит сложение/умножение его элементов.

Код программы:

import java.util.ArrayList;

public class Complex {

private final ArrayList<Double> coordinates;

public Complex(ArrayList<Double> coordinates) {

this.coordinates = coordinates;

}

public double sumCoordinates() {

return coordinates.stream()

.mapToDouble(a -> a)

.sum();

}

public double productCoordinates(){

return coordinates.stream()

.mapToDouble(a -> a)

.reduce(1, (a, b) -> a \* b);

}

public static void main(String[] args) {

ArrayList<Double> list = new ArrayList<>();

int n = 5;

for (int i = 0; i < n-1; i++) {

list.add(1+Math.random()\*10);

}

Complex complex = new Complex(list);

System.out.println(complex.sumCoordinates());

System.out.println(complex.productCoordinates());

}

}

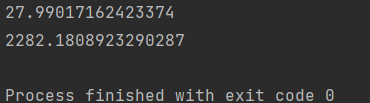


Рисунок 2 – результат задания №2

Задание №4

Создать класс Cправочная Cлужба Oбщественного Tранспорта с внутренним классом, с помощью объектов которого можно хранить информацию о времени, линиях маршрутов и стоимости проезда.

Код программы:

class Help\_Service\_Public\_Transport{ //класс "Cправочная Cлужба Oбщественного Tранспорта"

public class transport{ //внутренний клас "Транспорт"

double departure\_time;

double time\_of\_arrival;

double price;

public transport(double departure\_time, double time\_of\_arrival, double price){

this.departure\_time = departure\_time;

this.time\_of\_arrival = time\_of\_arrival;

this.price = price;

}

public void print(){

System.out.println("Время отправления: " + departure\_time);

System.out.println("Время прибытия: " + time\_of\_arrival);

System.out.println("Стоимость проезда: " + price);

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

}

}

private transport metro1;

transport metro = new transport(10.10, 15.14, 177);

private transport bus1;

transport bus = new transport(12.55, 17.35, 103);

private transport train1;

transport train = new transport(17.20, 22.50, 181);

public void print\_all()

{

bus.print();

metro.print();

train.print();

}

}

class practica{

public static void main(String args[]) {

Help\_Service\_Public\_Transport obj1 = new Help\_Service\_Public\_Transport();

obj1.print\_all();

}

}

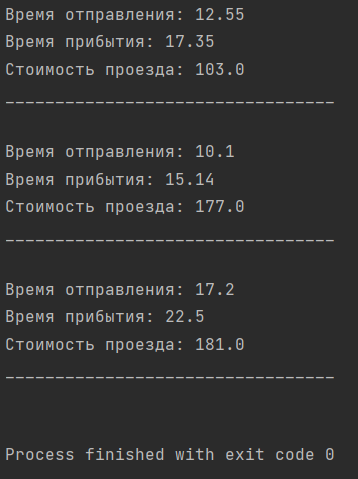


Рисунок 3 – результат задания №4

5 задание

В тексте найти и напечатать все слова максимальной и все слова минимальной длины.

Код программы:

import java.util.Scanner;  
public class Main {  
 public static String correctInputString(String in) {  
 in = in.replace(".", "");  
 in = in.replace(",", "");  
 in = in.replace(";", "");  
 in = in.replace(":", "");  
 in = in.replace("!", "");  
 in = in.replace("?", "");  
 in = in.replace(".", "");  
 in = in.replace("\"", "");  
 in = in.replace(" ", " "); //удаляем лишние пробелы  
 in = in.replace(" - ", " "); //удаляем тире  
 return in;  
 }  
 public static void findMinLengthAndOutput(String s) {  
 String[] words = s.split(" ");  
 int minLength = words[0].length(); //минимальная длина слова  
 for(String word : words) {  
 if(word.length() < minLength) {  
 minLength = word.length();  
 }  
 }  
 for(String word : words) {  
 if(word.length() == minLength) {  
 System.out.println(word);  
 }  
 }  
 }  
 public static void findMaxLengthAndOutput(String s) {  
 String[] words = s.split(" ");  
 int maxLength = words[0].length(); //максимальная длина слова  
 for (String word : words) {  
 if (word.length() > maxLength) {  
 maxLength = word.length();  
 }  
 }  
 for(String word : words) {  
 if(word.length() == maxLength) {  
 System.out.println(word);  
 }  
 }  
 }  
 public static void main(String[] args) {  
 Scanner scan = new Scanner(System.in);  
 String input = scan.nextLine();  
 System.out.println("Исходная строка: " + input);  
 input = correctInputString(input);  
 System.out.println("Обработанная строка: " + input);  
 System.out.println();  
 System.out.println("Слова минимальной длины:");  
 System.out.println();  
 findMinLengthAndOutput(input);  
 System.out.println("Слова максимальной длины:");  
 System.out.println();  
 findMaxLengthAndOutput(input);  
  
 }  
  
}

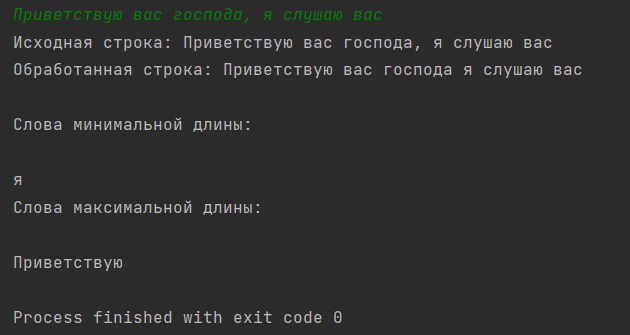
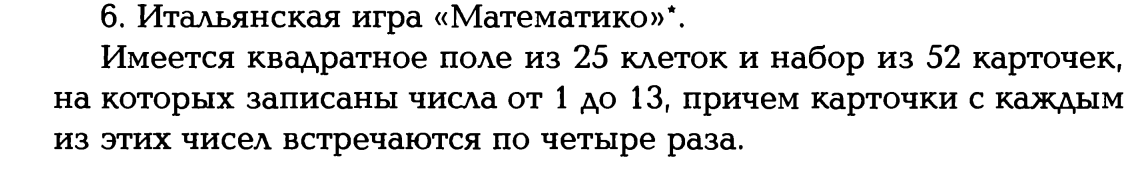
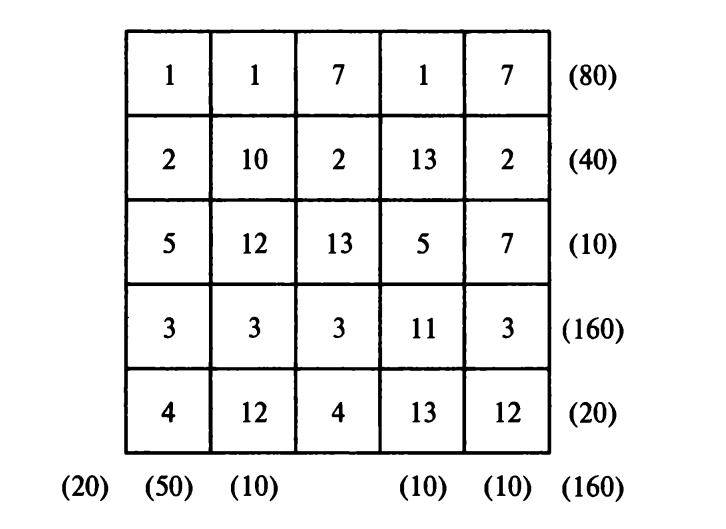


Рисунок 4 – результат задания №5

Задача №9





Код программы:

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.HashMap;

import java.util.LinkedList;

import java.util.List;

import java.util.Map;

import javax.swing.\*;

public class Main {

final static int btns\_count = 14;

final static int btn\_width = 60;

final static int btn\_height = 50;

static List<JButton>player\_buttons = new LinkedList<>();

static List<JButton> bot\_buttons = new LinkedList<>();

static List<Integer> bot\_buttons\_active = new LinkedList<>();

static Map<Integer, Integer> integers = new HashMap<>();

static Map<Integer, Integer> bot\_integers = new HashMap<>();

static int steps = 0;

static JLabel player\_score\_label = new JLabel("Очки: 0");

static JLabel bot\_score\_label = new JLabel("Очки: 0");

static int score\_sum = 0;

static int score\_sum\_bot = 0;

public static void main(String[] args) {

JFrame f = new JFrame("Игра Математико");

JPanel panel = new JPanel();

JLabel player = new JLabel("Игрок");

player.setBounds(0, 10, 200, 50);

JLabel bot = new JLabel("ИИ");

bot.setBounds(400, 10, 200, 50);

ActionListener actionListener = new TestActionListener();

panel.add(player);

panel.add(bot);

player\_score\_label.setBounds(0, 300, 200, 10);

bot\_score\_label.setBounds(400, 300, 200, 10);

panel.add(player\_score\_label);

panel.add(bot\_score\_label);

player\_score\_label.setVisible(false);

bot\_score\_label.setVisible(false);

for (int i = 0; i < 14; i++){

integers.put(i, 0);

bot\_integers.put(i, 0);

}

int value;

while (bot\_buttons\_active.size() < 25){

value = (int) (Math.random() \* 25);

if (!bot\_buttons\_active.contains(value)){

bot\_buttons\_active.add(value);

}

}

int x = 0;

int y = 0;

for (int i = 0; i < 25; i++){

if (i % 5 == 0){

x = 0;

y++;

}

JButton button1 = new JButton("0");

button1.setBackground(Color.BLUE);

button1.setForeground(Color.WHITE);

button1.setActionCommand(i + "");

button1.setBounds(btn\_width \* x,btn\_height \* y,btn\_width,btn\_height);

panel.add(button1, i);

button1.addActionListener(actionListener);

player\_buttons.add(button1);

x++;

}

x = 0;

y = 0;

for (int i = 0; i < 25; i++){

if (i % 5 == 0){

x = 0;

y++;

}

JButton button1 = new JButton("0");

button1.setActionCommand(i + "");

button1.setBounds(btn\_width \* x + 400,btn\_height \* y,btn\_width,btn\_height);

panel.add(button1, i);

button1.setEnabled(false);

bot\_buttons.add(button1);

x++;

}

panel.setLayout(null);

panel.setLocation(0,0);

f.setSize(720, 360);

f.add(panel);

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

f.setVisible(true);

}

public static class TestActionListener implements ActionListener {

public void actionPerformed(ActionEvent e) {

Object source = e.getSource();

((JButton) source).setText("" + rnd(btns\_count, false));

((JButton) source).setEnabled(false);

bot\_buttons.get(bot\_buttons\_active.get(0)).setText("" + rnd(btns\_count, true));

bot\_buttons\_active.remove(0);

if(steps == 24){

int[] result = result(player\_buttons,bot\_buttons);

player\_score\_label.setVisible(true);

player\_score\_label.setText("Счет: " + result[0]);

bot\_score\_label.setVisible(true);

bot\_score\_label.setText("Счет: " + result[1]);

}

else{

steps++;

}

}

}

public static int[] result( List<JButton>player\_buttons, List<JButton>bot\_buttons)

{

int[] result = new int[2];

result[0] = 0;

result[1] = 0;

int x1 = 0;

int y1 = 0;

int[][] values = new int[5][5];

for (int i = 0; i < 24; i++){

if (i % 5 == 0){

x1 = 0;

y1++;

}

values[y1-1][x1] = Integer.parseInt(player\_buttons.get(i).getText());

x1++;

}

int score\_row\_sum = 0;

int score\_row = 0;

for (int k = 0; k < 14; k++){

for (int i = 0; i < 5; i++){

for (int j = 0; j < 5; j++){

if (values[i][j] == k && values[i][j]!=-1){

score\_row++;

}

}

if (score\_row > 1){

score\_row\_sum = score\_row\_sum + score\_row;

}

score\_row = 0;

}

if (score\_row\_sum == 2){

score\_sum = score\_sum + 10;

}

if (score\_row\_sum >= 3){

score\_sum = score\_sum + 40;

}

score\_row\_sum = 0;

}

result[0] = score\_sum;

x1 = 0;

y1 = 0;

values = new int[5][5];

for (int i = 0; i < 24; i++){

if (i % 5 == 0){

x1 = 0;

y1++;

}

values[y1-1][x1] = Integer.parseInt(bot\_buttons.get(i).getText());

x1++;

}

score\_row\_sum = 0;

score\_row = 0;

for (int k = 0; k < 14; k++){

for (int i = 0; i < 5; i++){

for (int j = 0; j < 5; j++){

if (values[i][j] == k && values[i][j]!=-1){

score\_row++;

}

}

if (score\_row > 1){

score\_row\_sum = score\_row\_sum + score\_row;

}

score\_row = 0;

}

if (score\_row\_sum == 2){

score\_sum\_bot = score\_sum\_bot + 10;

}

if (score\_row\_sum >= 3){

score\_sum\_bot = score\_sum\_bot + 40;

}

score\_row\_sum = 0;

}

result[1] = score\_sum\_bot;

return result;

}

public static int rnd(final double max, boolean isBot)

{

int value = (int) (Math.random() \* max);

if (!isBot){

while (integers.get(value) > 3){

value = (int) (Math.random() \* max);

}

integers.put(value, integers.get(value) + 1);

}

else {

while (bot\_integers.get(value) > 3){

value = (int) (Math.random() \* max);

}

bot\_integers.put(value, bot\_integers.get(value) + 1);

}

return value;}}

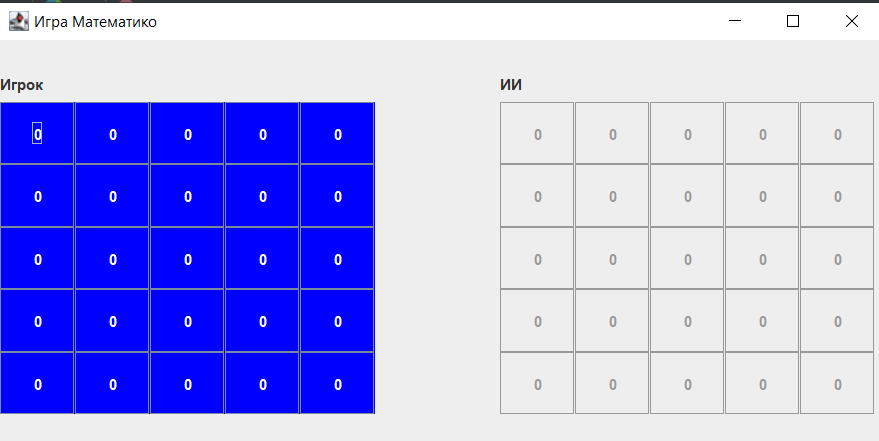


Рисунок 5.1 – начало игры «Математико»

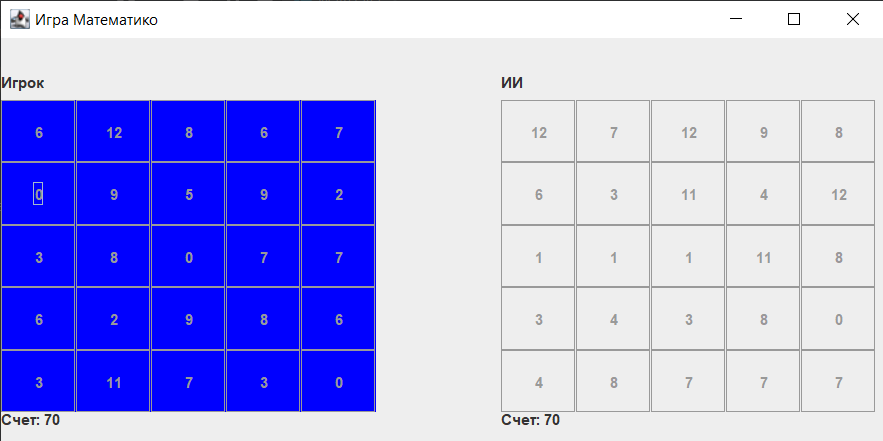


Рисунок 5.2 – конец игры «Математико»

Соединение всех задач

Код программы:

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.net.URL;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JTextField;

public class Main extends JFrame {

private JTextField textField;

public Main() {

super("Все выполненные задания");

createGUI();

}

public void createGUI() {

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

JPanel panel = new JPanel();

panel.setLayout(new FlowLayout());

ActionListener actionListener = new TestActionListener();

JButton button1 = new JButton("Задача 1");

button1.setActionCommand("https://github.com/KriLuna/up01/blob/main/task\_1/src/Main.java");

panel.add(button1);

button1.addActionListener(actionListener);

JButton button2 = new JButton("Задача 2");

button2.setActionCommand("https://github.com/KriLuna/up01/blob/main/task\_2/src/Main.java");

panel.add(button2);

button2.addActionListener(actionListener);

JButton button4 = new JButton("Задача 4");

button4.setActionCommand("https://github.com/KriLuna/up01/blob/main/task\_4/src/Main.java");

panel.add(button4);

button4.addActionListener(actionListener);

JButton button5 = new JButton("Задача 5");

button5.setActionCommand("https://github.com/KriLuna/up01/blob/main/task\_4/src/Main.java");

panel.add(button5);

button5.addActionListener(actionListener);

JButton button9 = new JButton("Задача 9");

button9.setActionCommand("https://github.com/KriLuna/up01/blob/main/task\_9/src/Main.java");

panel.add(button9);

button9.addActionListener(actionListener);

getContentPane().add(panel);

setPreferredSize(new Dimension(320, 200));

}

public class TestActionListener implements ActionListener {

public void actionPerformed(ActionEvent e) {

try {

Desktop.getDesktop().browse(new URL(e.getActionCommand()).toURI());

} catch (Exception ex) {

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

}

}

}

public static void main(String[] args) {

javax.swing.SwingUtilities.invokeLater(new Runnable() {

public void run() {

JFrame.setDefaultLookAndFeelDecorated(true);

Main frame = new Main();

frame.pack();

frame.setLocationRelativeTo(null);

frame.setVisible(true);

}

});

}

}

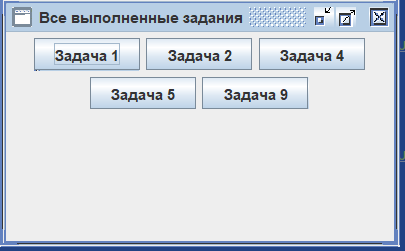


Рисунок 6 – задачи в диалоговом окне

Задача №10

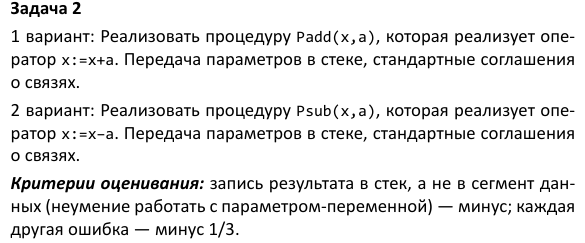


Таблица 1 - Структура программы 1:

|  |  |  |
| --- | --- | --- |
| Код команды | Мнемокод | Комментарий |
| AF | XRA A | Очистить аккумулятор |
| 3E | MVI A | Записать в аккумулятор |
| 02 |  | Число x |
| 06 | MVI B | Записать в регистр B |
| 03 |  | Число a |
| 80 | ADD B | Сложить x и a |
| AF | XRA A | Очистить аккумулятор |
| E7 | RST 7 | Прервать выполнение программы |

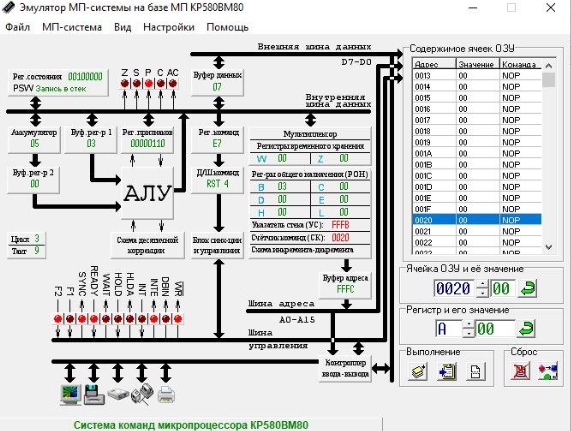


Рисунок 7.1 – Результат задачи 10

Таблица 2 - Структура программы 2

| Код команды | Мнемокод | Комментарий |
| --- | --- | --- |
| AF | XRA A | Очистить аккумулятор |
| 3E | MVI A | Записать в аккумулятор |
| 08 |  | Число x |
| 06 | MVI B | Записать в регистр B |
| 10 |  | Число a |
| 90 | SUB B | Вычесть из x и a |
| E7 | RST 7 | Прервать выполнение программы |

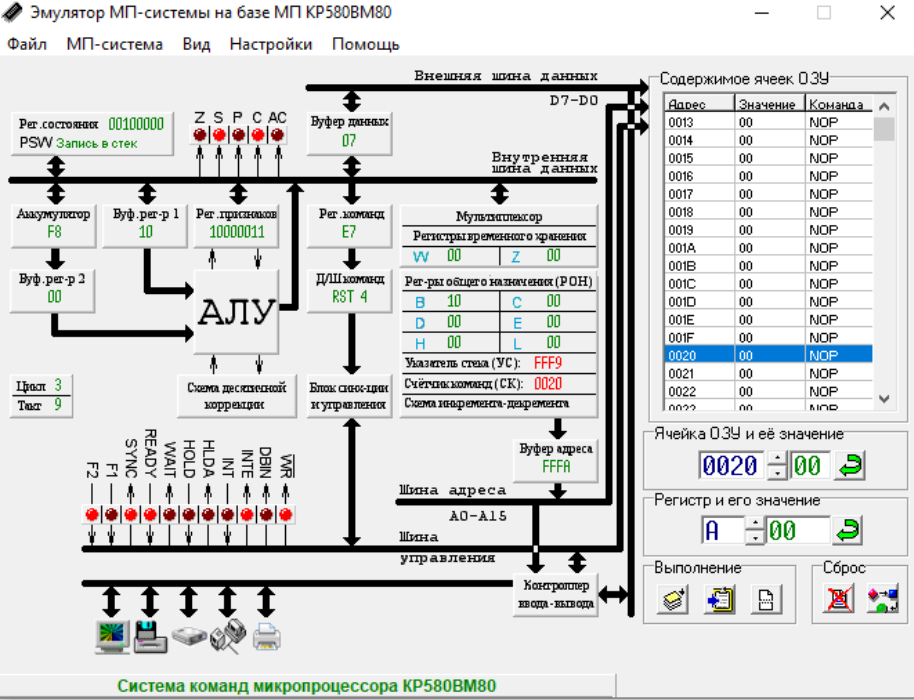


Рисунок 7.2 – Результат задачи 10

Разработка АИС «Словарь»

ER-диаграмма сущность связь. Связь один-к одному

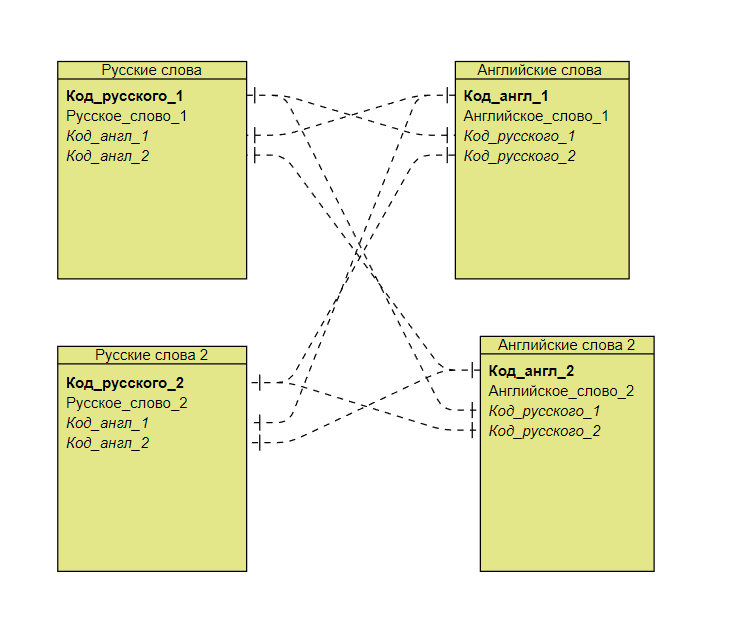


Рисунок 8 – ER-диаграмма

Создание сайта с помощью инструмента sites.google.com

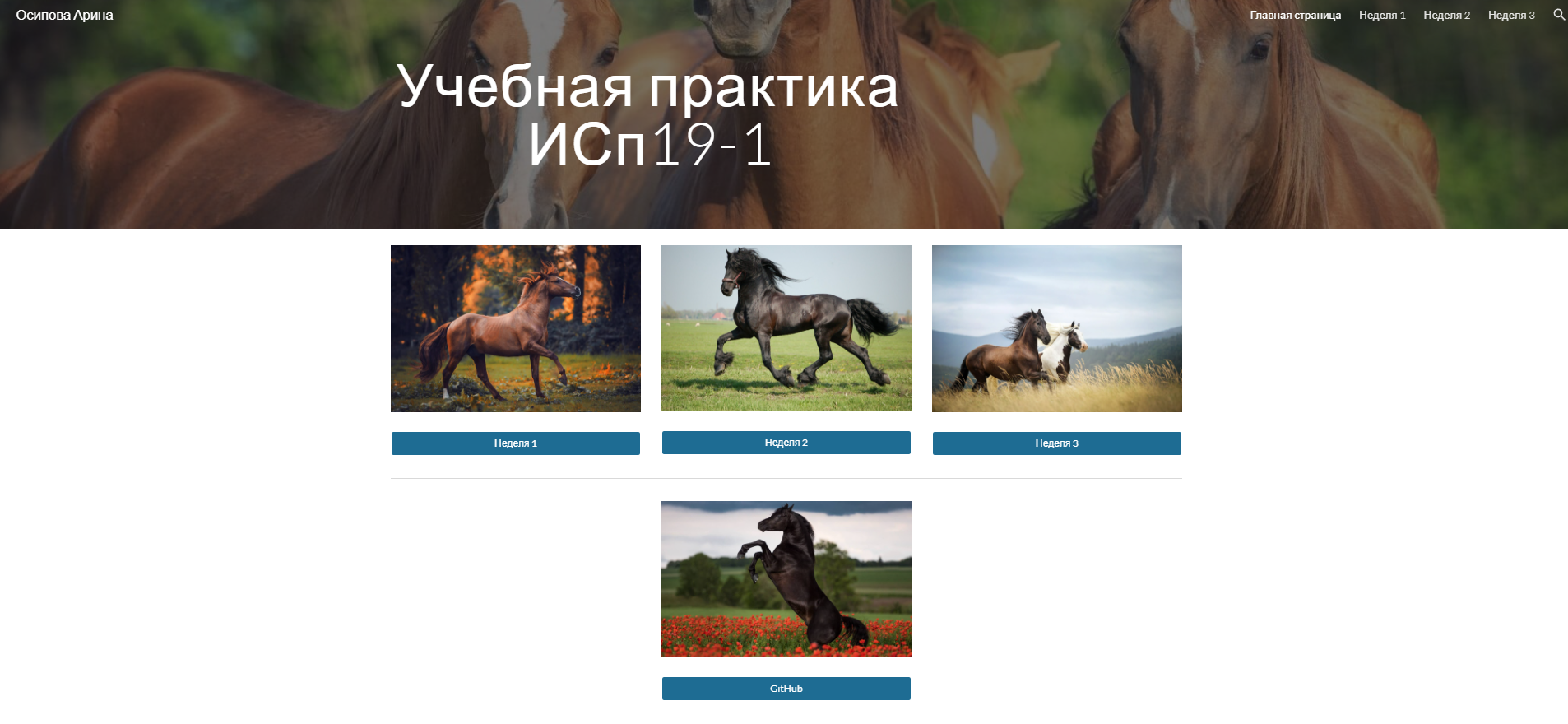


Рисунок 9 – главная страница сайта



Рисунок 10 – страница сайта первой недели

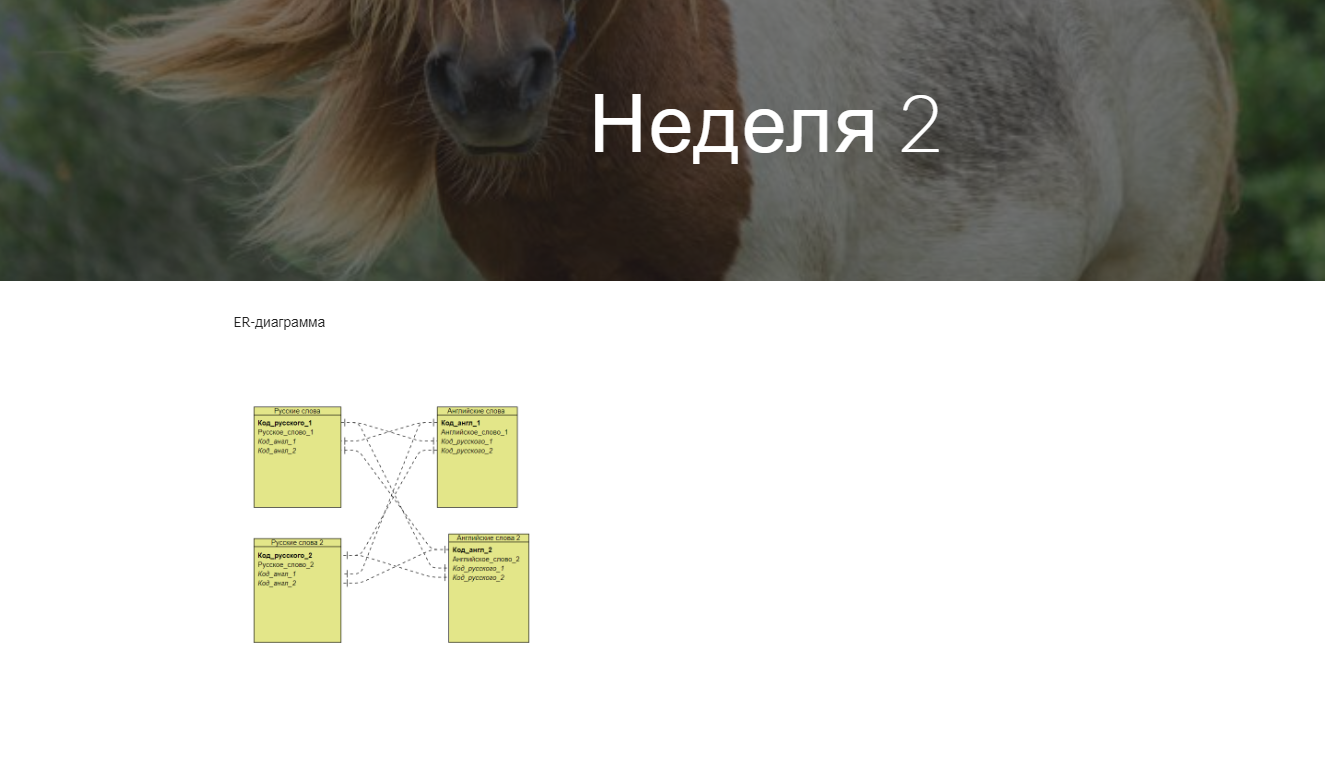


Рисунок 11 – страница сайта второй недели

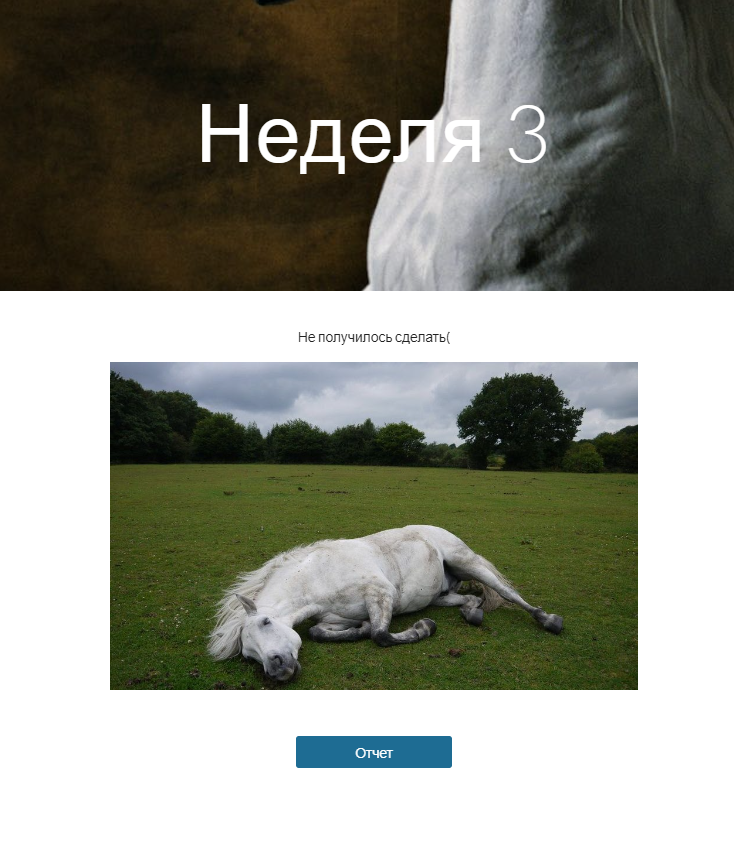


Рисунок 12 – страница сайта с третьей неделей

Заключение

В ходе прохождения практики освоены разработка приложений по работе с матрицами, классами и вложенными классами, работе с циклами, разработки визуального интерфейса, проектирование базы данных и мобильного приложения.

Цели и задачи практики достигнуты. Разработаны алгоритмы решения задач с матрицами и циклами, разработана АИС «Словарь». Для демонстрации всех разработанных приложений был сконструирован сайт с использованием хостинга Google Site.

Список литературы

1. Google Sites: <https://sites.google.com/new>
2. Учебник «Java – Методы  программирования»
3. Семакин И. Г. – «Основы алгоритмизации и программирования»
4. Google Sites – [https://sites.google.com](https://sites.google.com/)
5. Форум программистов - <https://www.cyberforum.ru>
6. Платформа для выгрузки проектов - <https://github.com>