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: | студент 4 курса,  группы ИСп 19-1  Шпак Александр Дмитриевич  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |
| Проверила: | Большакова-Стрекалова Анна Викторовна  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (оценка)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) |

Калининград, 2022

Содержание

|  |  |
| --- | --- |
| Введение | 3 |
| 1 Решение задач | 4 |
| 1.1 Задача 3 | 4 |
| 1.2 Задача 4 | 6 |
| 1.3 Задача 9 | 10 |
| 1.4 Задача 10 | 16 |
| 2 Разработка АИС «Письма» | 18 |
| 2.1 ER-диаграмма сущность-связь | 18 |
| 3 Разработка сайта | 19 |
| Заключение | 21 |
| Список использованных источников | 22 |

Введение

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

Целями практики являются закрепление знаний в области 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 №3

Создать объект класса Птица, используя классы Крылья, Клюв. Методы: летать, садиться, питаться, атаковать

Код программы:

Main.java

public class Main {

public static void main(String[] args) {

Bird bird1 = new Bird("raven");

Bird bird2 = new Bird("pigeon");

bird1.fly();

bird2.eat();

bird1.eat();

bird2.atack();

System.out.println(bird1.equals(bird2));

System.out.println(bird1);

System.out.println(bird2);

}

}

Wings.java

public class Wings {

public void use() {

System.out.println(" --Using wings");

}

}

Beak.java

public class Beak {

public void use() {

System.out.println("--Using beak");

}

}

Bird.java

public class Bird {

private Wings wings;

private String name;

private Beak beak;;

public Bird(String name) {

this.name = name;

this.wings = new Wings();

this.beak = new Beak();

}

public void fly() {

System.out.print("Bird " + name + " is flying");

wings.use();

}

public void atack() {

System.out.print("Bird " + name + " is atack");

beak.use();

}

public void sit\_down() {

System.out.print("Bird " + name + " is sit");

wings.use();

}

public void eat() {

System.out.println("Bird " + name + " is eating");

beak.use();

}

@Override

public boolean equals(Object o) {

if (this == o) return true;

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

Bird bird = (Bird) o;

if (!wings.equals(bird.wings)) return false;

return name.equals(bird.name);

}

@Override

public int hashCode() {

int result = wings.hashCode();

result = 31 \* result + name.hashCode();

return result;

}

@Override

public String toString() {

return "Bird : '" + name + "'";

}

}

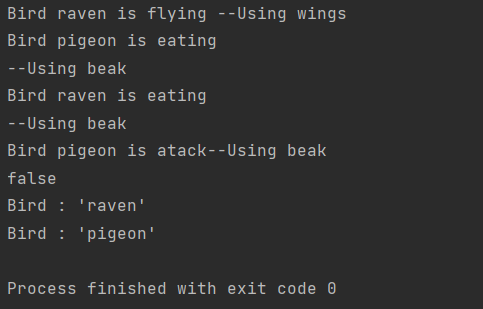


Рисунок 1 – результат выполнения 3 задания.

Задание №4

Создать класс Account с внутренним классом, с помощью объектов которого можно хранить информацию обо всех операциях со счетом (снятие, платежи, поступления).

Код программы:

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

public class Main {

public static void main(String[] args) {

Account accaunt = new Account();

accaunt.newTransaction(Account.typeTransaction.RECEIPT, 5000);

accaunt.newTransaction(Account.typeTransaction.PAYMENT, 2500);

accaunt.newTransaction(Account.typeTransaction.WITHDRAWAL, 1000);

System.out.println(accaunt.print());

}

}

class Account {

enum typeTransaction {

WITHDRAWAL, PAYMENT, RECEIPT

}

int accountBalance = 0;

int nextId = 0;

List<Transaction> history = new ArrayList<>();

abstract class Transaction {

int id = 0;

int amount = 0;

int balance = 0;

Date date = new Date();

typeTransaction type;

public Transaction() {

this.id = nextId++;

}

@Override

public String toString() {

String string;

string = '\n' + "Transaction ID: " + this.id + '\n';

string += "Type: " + this.type + '\n';

string += "Date: " + this.date + '\n';

string += "Amount: " + this.amount + '\n';

string += "Balance after operation: " + this.balance + '\n';

return string;

}

}

class Withdrawal extends Transaction {

public Withdrawal(int amount) {

this.balance = accountBalance -= amount;

this.amount = amount;

}

}

class Payment extends Transaction {

public Payment(int amount) {

this.balance = accountBalance -= amount;

this.amount = amount;

}

}

class Receipt extends Transaction {

public Receipt(int amount) {

this.balance = accountBalance += amount;

this.amount = amount;

}

}

public void newTransaction(typeTransaction type, int amount)

throws UnsupportedOperationException {

Transaction newTransaction;

switch (type) {

case WITHDRAWAL:

newTransaction = new Withdrawal(amount);

break;

case PAYMENT:

newTransaction = new Payment(amount);

break;

case RECEIPT:

newTransaction = new Receipt(amount);

break;

default:

throw new UnsupportedOperationException("Not supported yet.");

}

newTransaction.type = type;

history.add(newTransaction);

}

public String print() {

String string = "";

for( Transaction t : history)

{

string += t.toString();

}

return string;

}

}

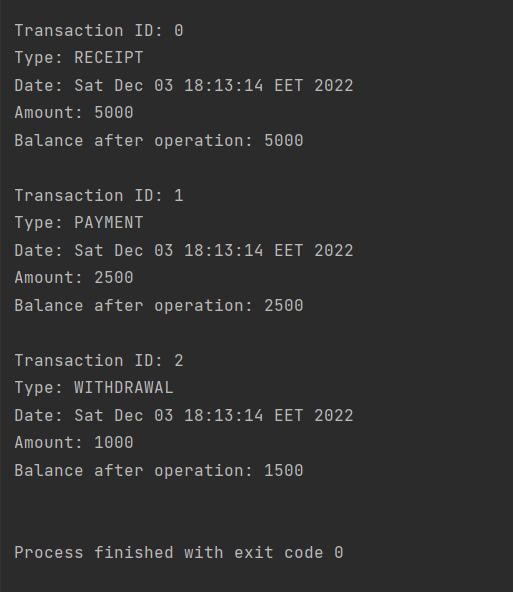
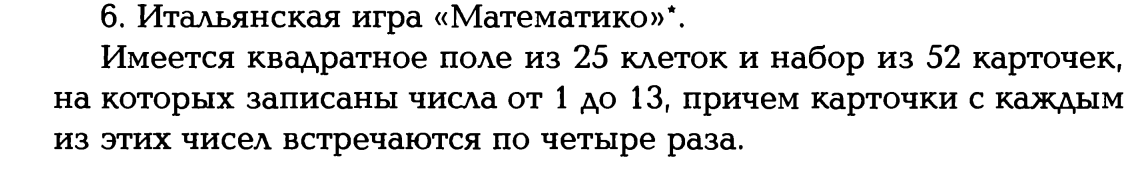
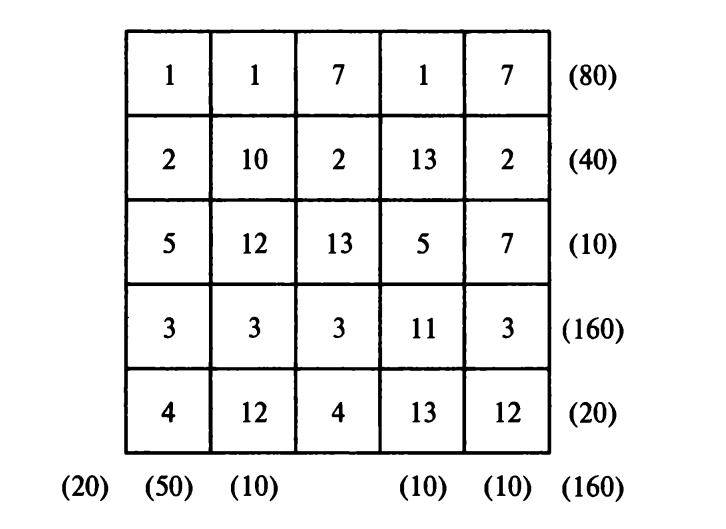


Рисунок №2 – результат выполнения задания 4.

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();

JPanel panel = new JPanel();

JLabel player = new JLabel("Игрок");

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

JLabel bot = new JLabel("Компьютер");

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.RED);

button1.setForeground(Color.PINK);

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.setBackground(Color.YELLOW);

button1.setForeground(Color.CYAN);

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("Очки: " + result[0]);

bot\_score\_label.setVisible(true);

bot\_score\_label.setText("Очки: " + 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 3 – начальное поле игры «Математико»

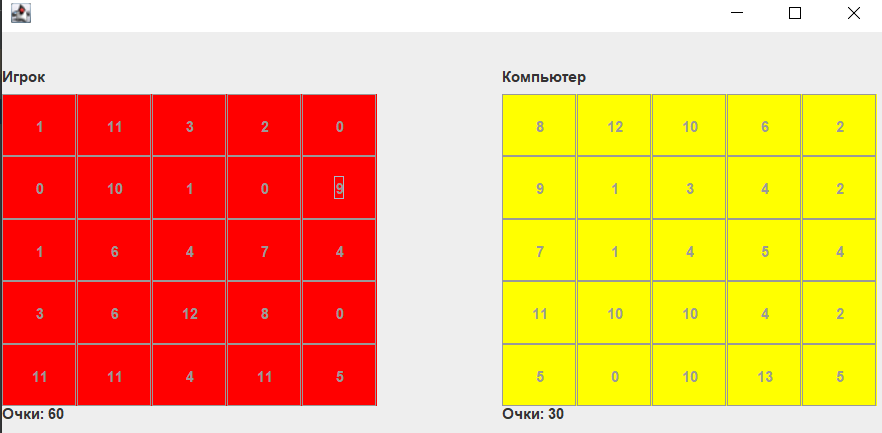


Рисунок 4 – конец игры «Математико»

Все задания в одном приложении

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 задания");

createGUI();

}

public void createGUI() {

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

JPanel panel = new JPanel();

panel.setLayout(new FlowLayout());

ActionListener actionListener = new TestActionListener();

JButton button3 = new JButton("Задача 3");

button3.setActionCommand("https://github.com/Euros2308/-1-.-19-1-1/tree/main/%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B03");

panel.add(button3);

button3.addActionListener(actionListener);

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

button4.setActionCommand("https://github.com/Euros2308/-1-.-19-1-1/tree/main/%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B04");

panel.add(button4);

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

button9.setActionCommand("https://github.com/Euros2308/-1-.-19-1-1/tree/main/%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B09");

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);

}

});

}

}

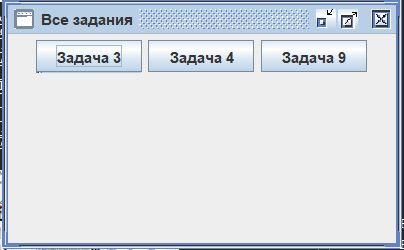


Рисунок 5 – Программа с заданиями

Задача №10

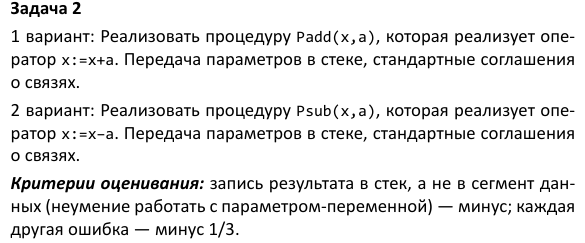


Таблица 1 - Структура программы 1:

|  |  |  |
| --- | --- | --- |
| 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 |

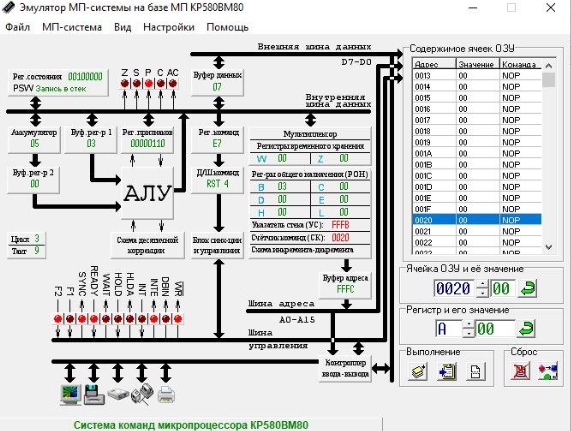


Рисунок 6 – Результат задачи 10.1

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 |

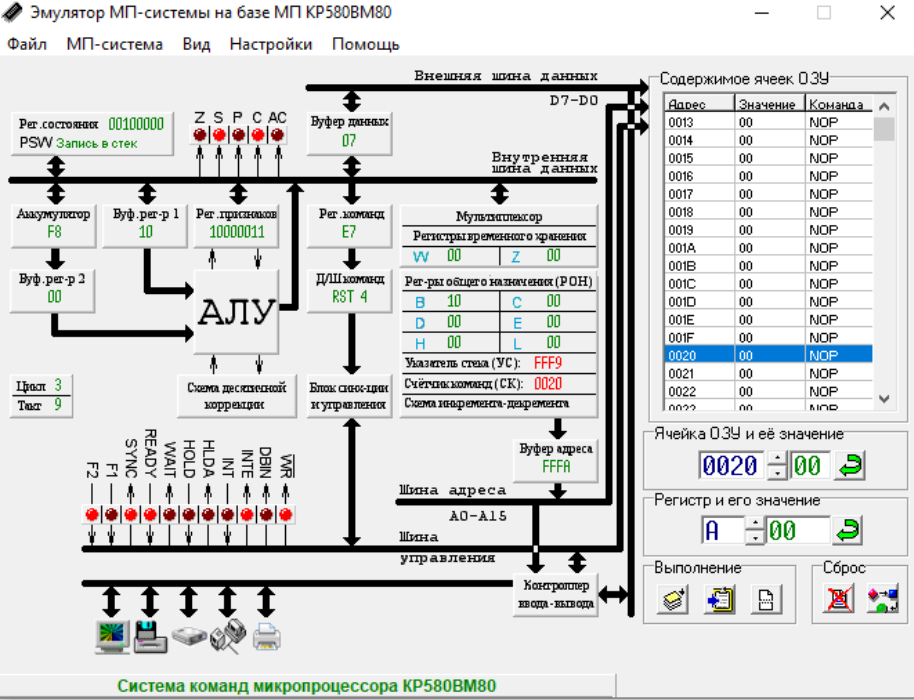


Рисунок 7 – Результат задачи 10.2

Разработка АИС «Письма»

ER-диаграмма сущность связь

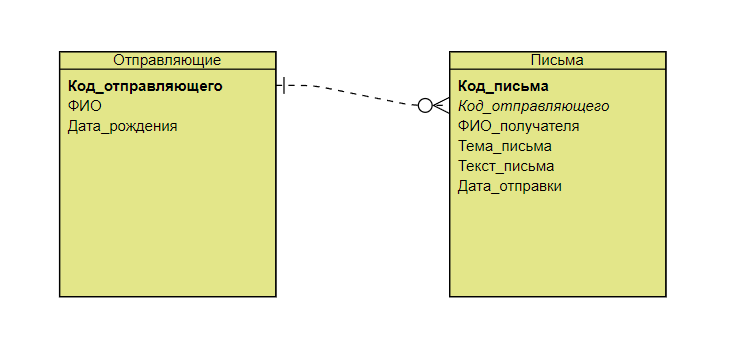


Рисунок 8 – ER-диаграмма

Разработка сайта

Для разработки сайта был использован сайт Google Sites, который имеет все необходимые инструменты и шаблоны для создания любого типа сайта.

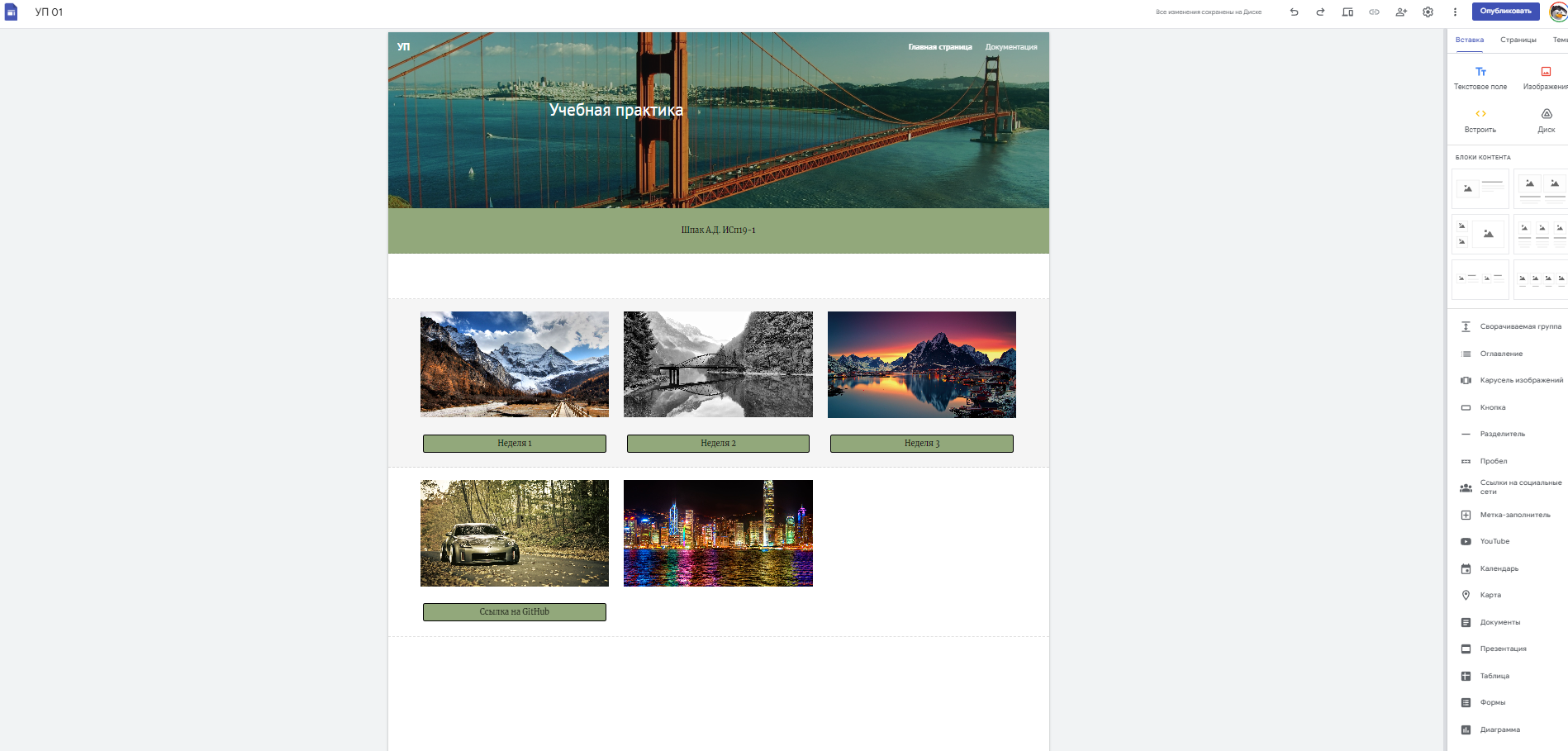


Рисунок 9 – конструктор сайта

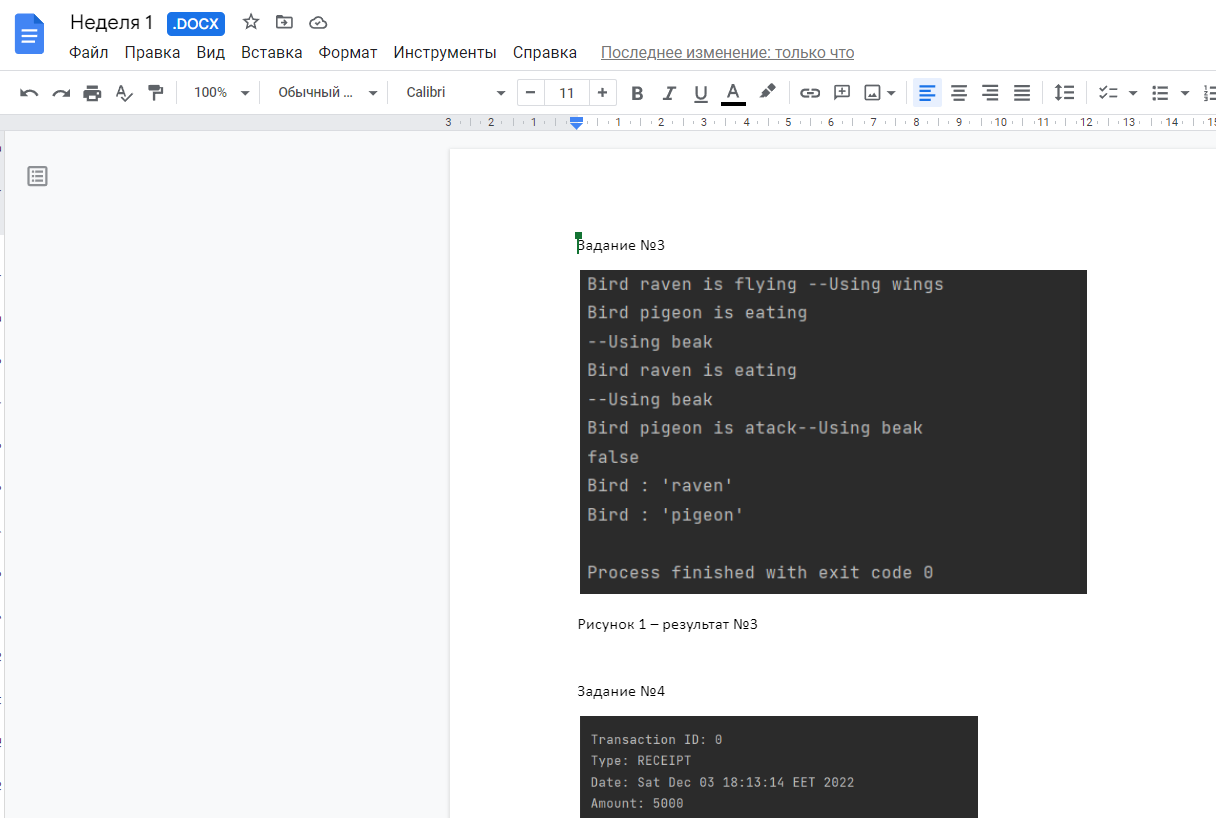


Рисунок 10 – Переход к документу 1 недели

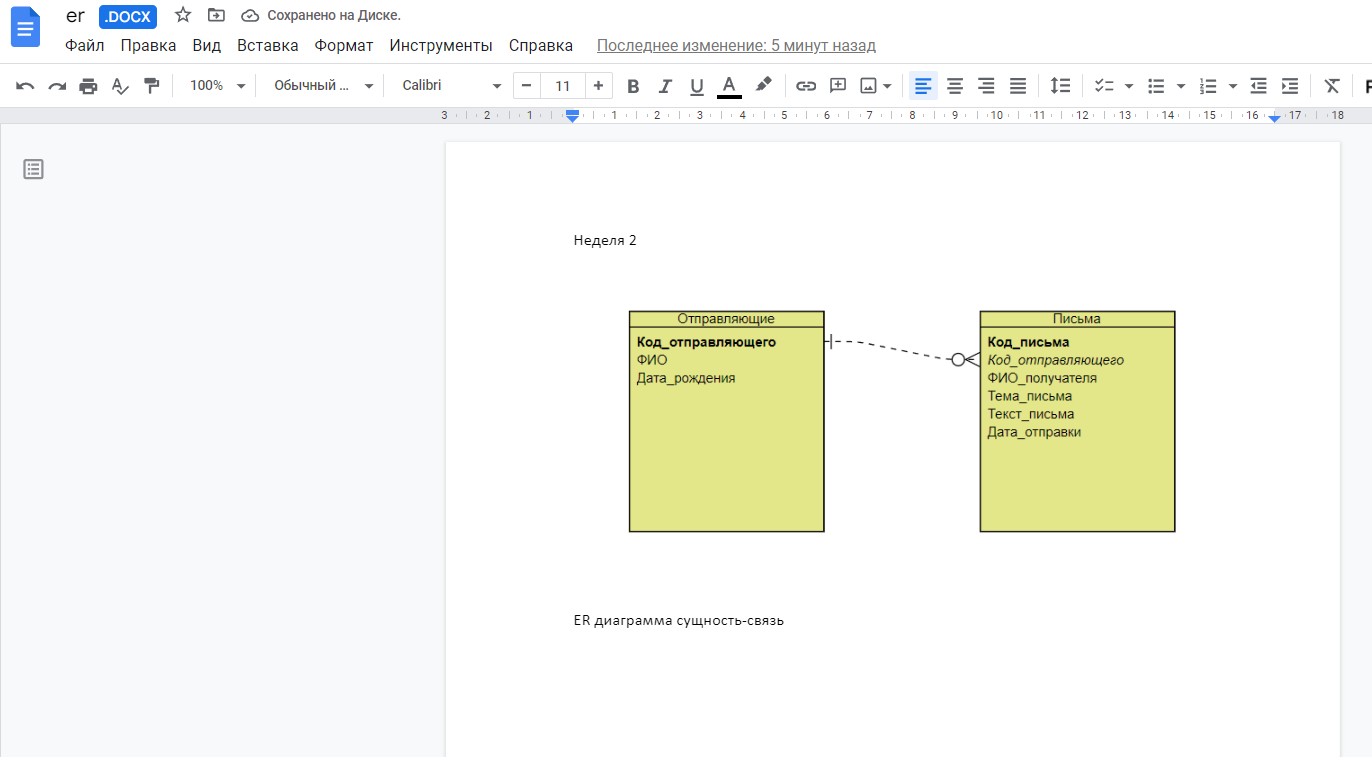


Рисунок 11 – переход к документу 2 недели

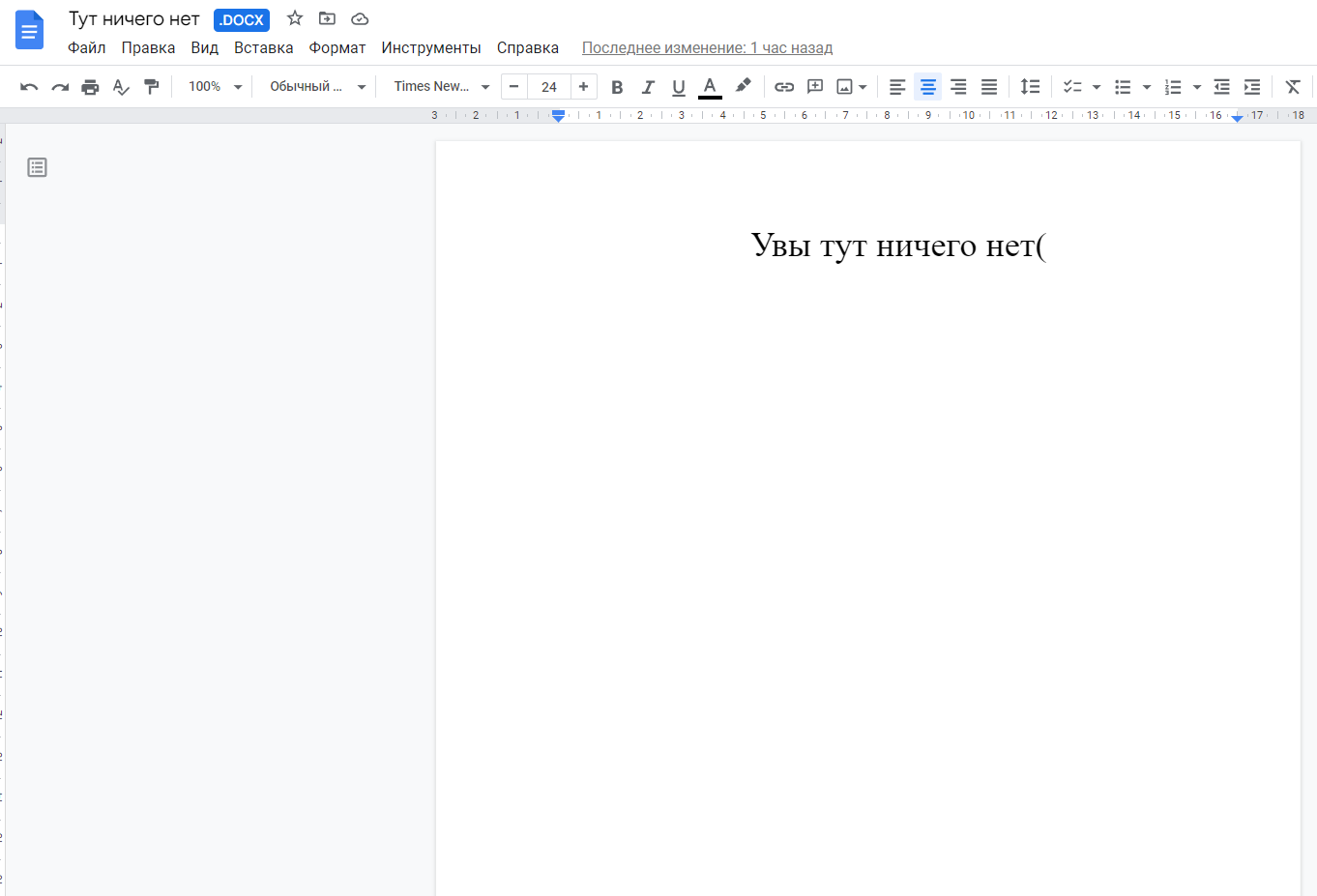


Рисунок 12 – переход к файлу 3 недели

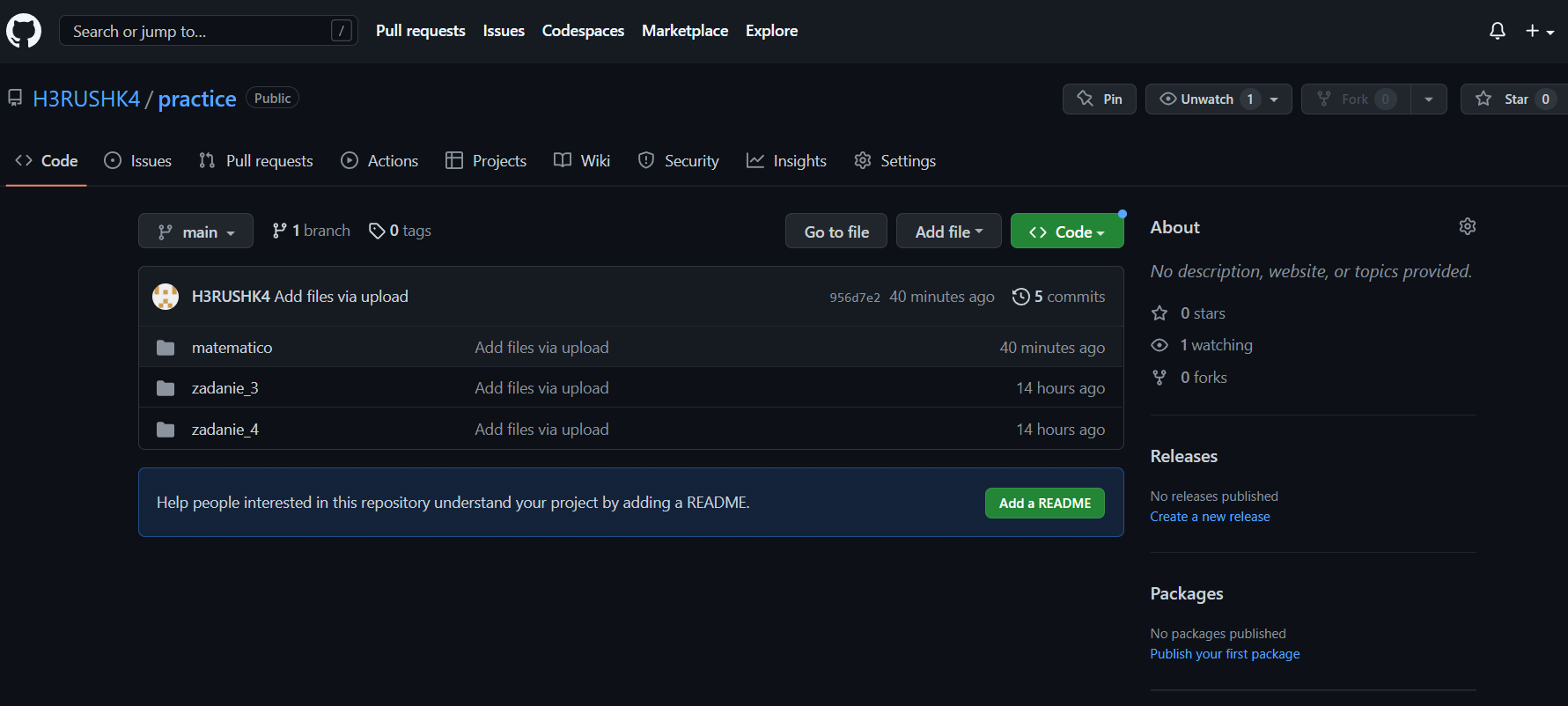


Рисунок 13 – переход по кнопке на сайт GitHub

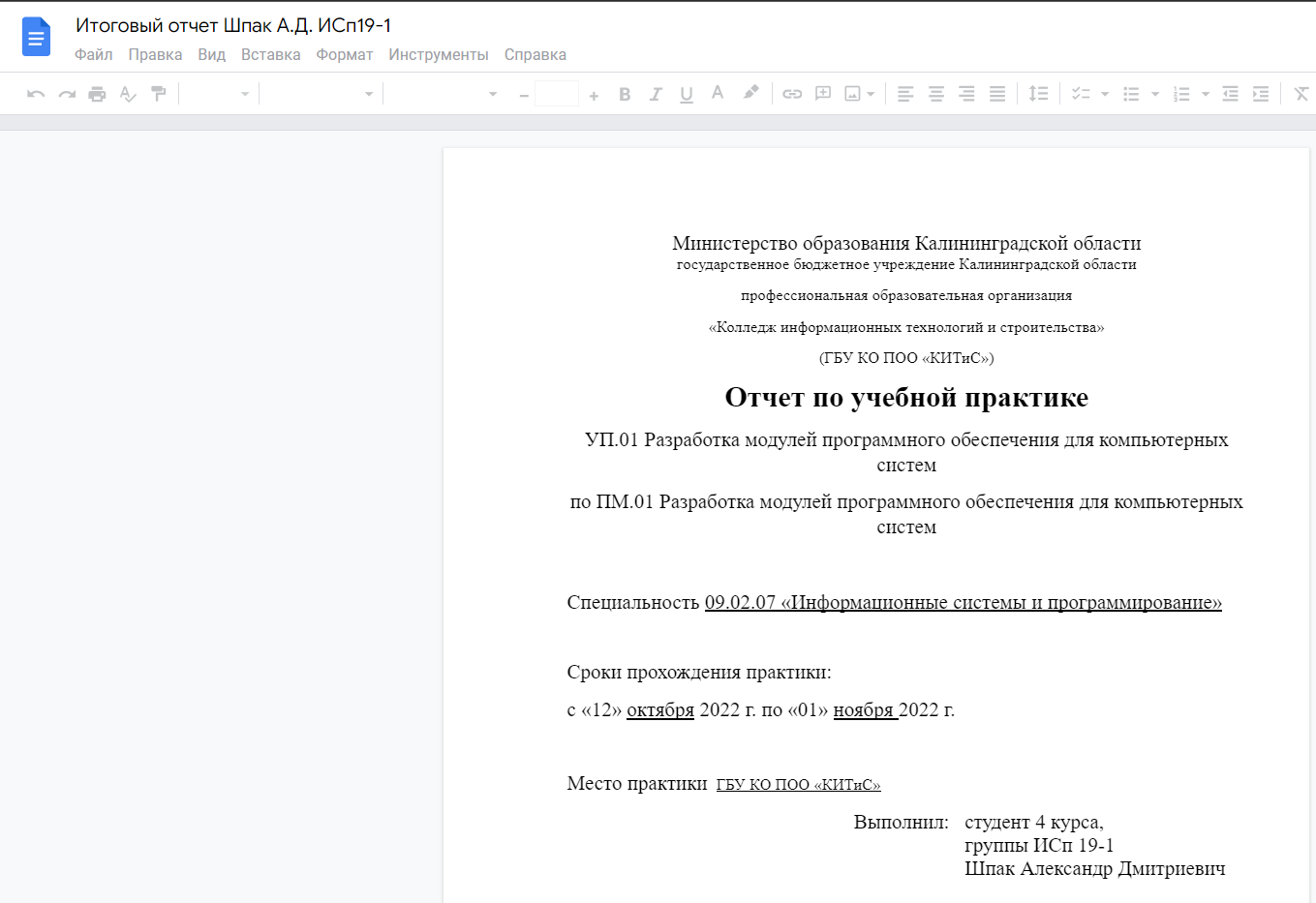


Рисунок 14 – Переход к документу «Отчет по учебной практике»

Заключение

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

Цели и задачи практики достигнуты. Разработаны алгоритмы решения задач с матрицами и циклами, разработана АИС «Письма». Для демонстрации выполненной практики был создан сайт с использованием хостинга Google Site.

Используемая литература

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