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  Попов Ярослав Дмитриевич  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (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 | 10 |
| 1.5 Задача 6 | 11 |
| 1.6 Задача 7 | 13 |
| 1.7 Задача 10 | 15 |
| 2 Разработка АИС «Издательство» | 18 |
| 2.1 Техническое задание | 19 |
| 2.2 ER-диаграмма сущность-связь | 20 |
| 2.3 Функциональная система АИС | 21 |
| 3 Разработка мобильного приложения | 26 |
| 4 Разработка сайта | 27 |
| Заключение | 59 |
| Список использованных источников | 60 |

Введение

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

Цели практики является закрепление знаний в области разработки автоматических информационных систем, разработки мобильных приложений, развитие профессиональных навыков в области проектирования баз данных и проектирования программных интерфейсов.

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

14. Округлить все элементы матрицы до целого числа.

Таблица 1 – Таблица переменных кода задачи 1

| Название | Тип | Назначение |
| --- | --- | --- |
| array | array | Изначальная матрица |
| xx | double | Получить значение из матрицы, округлить его и заменить старое значение матрицы на данное |

Блок схема

нет

да

Конец

Начало

I от 1 до n

I от 1 до j

array[i][j] = Math.random

I от 1 до j

Array[i][j] = Double.valueOf(String.format("%.0f",xx));

I от 1 до n

Код программы

import java.util.Scanner;

public class Main {

public static void main (String[] args) {

double[][] array = new double[4][4];

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

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

array[i][j] = Math.random();

}

}

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

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

double xx = array[i][j];

array[i][j] = Double.valueOf(String.format("%.0f",xx));

}

}

}

}

Скриншот результата

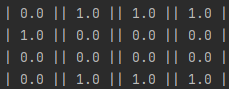


Рисунок 1 – Результат выполнения задачи 1

Задание 2

6. Определить класс Точка на плоскости (в пространстве) и во времени. Задать движение точки в определенном направлении. Создать методы по определению скорости и ускорения точки. Проверить для двух точек возможность пересечения траекторий. Определить расстояние между дву- мя точками в заданный момент времени.

Таблица 2 – Таблица переменных кода задачи 2

| Название | Тип | Назначение |
| --- | --- | --- |
| t | int | Задать прошедшее время |

Блок схема

Начало

Вычисление координат

Вывод данных о точке

конец

Ввод t

Код программы

package com.company;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

int t = scan.nextInt();

point first = new point("первая точка",0,0,5);

point second = new point("вторая точка",1,1,2);

first.xcord = first.xcord + first.speed\*t;

first.ycord = first.ycord + first.speed\*t;

second.xcord = second.xcord + second.speed\*t;

System.out.println(first.name+" " +first.speed+" "+first.xcord+" "+first.ycord+ " " + t);

System.out.println(second.name+" " +second.speed+" "+second.xcord+" "+second.ycord+ " " + t);

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

System.out.println("Расстояние между точками: ("+Math.abs(first.xcord - second.xcord)+";"+Math.abs(first.ycord- second.ycord)+")");

}

}

class point {

String name;

int xcord;

int ycord;

int speed;

point(String q, int x, int y, int v){

name = q;

xcord = x;

ycord = y;

speed = v;

}

}

Скриншот результата

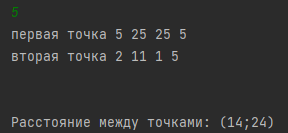


Рисунок 2 – Результат выполнения задачи 2

Задание 3

18. Создать объект класса Пианино, используя классы Клавиша, Педаль. Методы: настроить, играть на пианино, нажимать клавишу.

Код программы

import java.util.ArrayList;

import java.util.List;

public class Main {

public static void main(String[] args) {

pedal ped = new pedal(1);

button ba = new button(1);

pianino pi = new pianino(ped,ba);

pi.addbutt(ba);

pi.addPed(ped);

pi.pressbutton(ba);

pi.nastroyka();

pi.presspedal(ped);

}

}

class pianino {

List<pedal> pedals = new ArrayList<>() ;

List<button> buttons = new ArrayList<>();

void nastroyka(){

System.out.println("пианино настроено");

}

void addPed(pedal pedal){

pedals.add(pedal);

}

void addbutt(button button){

buttons.add(button);

}

void presspedal(pedal pedal){

System.out.println("Нажата педаль: "+pedal.name);

}

void pressbutton(button button){

System.out.println("Нажата кнопка: "+button.name);

}

pianino(pedal pedals, button buttons){

this.pedals.add(pedals);

this.buttons.add(buttons) ;

}

}

class button{

int name;

public button(int number){

this.name=number;

}

}

class pedal{

int name;

pedal(int name){

}

public void setName(int name) {

this.name = name;

}

}

Скриншот результата

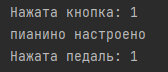


Рисунок 3 – Результат выполнения задачи 3

Задание 4

18. Создать класс Программа Передач с внутренним классом, с помощью объектов которого можно хранить информацию о названии телеканалов и программ.

Код программы

package com.company;

import java.util.ArrayList;

import java.util.List;

public class Main {

public static void main(String[] args) {

programmaperedax.raspisanie first = new programmaperedax.raspisanie("РОССИЯ 1","Новости");

programmaperedax.raspisanie second = new programmaperedax.raspisanie("РОССИЯ 1","Следствие вели");

programmaperedax.raspisanie third = new programmaperedax.raspisanie("РПЦ","просвящение");

programmaperedax.raspisanie fourth = new programmaperedax.raspisanie("Условно","ДаДа");

}

class programmaperedax {

static class raspisanie {

String nameTV;

String namePeredax;

ArrayList<raspisanie> aboba = new ArrayList<>();

public raspisanie(String nameTV, String namePeredax) {

this.nameTV = nameTV;

this.namePeredax = namePeredax;

aboba.add(this);

}

}

}}

Задание 6

6. Вариант А

Выполнить задания на основе варианта А задачи 3, контролируя состояние потоков ввода/вывода. При возникновении ошибок, связанных с корректностью выполнения математических операций, генерировать п обрабатывать исключительные ситуации. Предусмотреть обработку исключений, возникающих при нехватке памяти, отсутствии требуемой записи (обьекта) в файле, недопустимом значении поля и т. д.

Вариант в

Выполнить задания из варианта в задачи 3, реализуя собственные обработчики исключений и исключения ввода/вывода.

Код программы

package com.company;

import java.util.ArrayList;

import java.util.List;

public class Main {

public static void main(String[] args) {

pedal ped = new pedal(null);

button ba = new button(null);

pianino pi = new pianino(ped,ba);

pi.addbutt(ba);

pi.addPed(ped);

pi.pressbutton(ba);

pi.nastroyka();

pi.presspedal(ped);

}

}

class pianino {

List<pedal> pedals = new ArrayList<>() ;

List<button> buttons = new ArrayList<>();

void nastroyka(){

System.out.println("пианино настроено");

}

void addPed(pedal pedal){

pedals.add(pedal);

}

void addbutt(button button){

buttons.add(button);

}

void presspedal(pedal pedal){

try {

if(pedal.name == Integer.parseInt(null)){

}

System.out.println("Нажата педаль: "+pedal.name);

} catch (Exception e) {

System.out.println("Требуется дать номер педали для работы");

}

}

void pressbutton(button button){

try {

if(button.name == Integer.parseInt(null)){

}

System.out.println("Нажата клавиша: "+button.name);

} catch (Exception e) {

System.out.println("Требуется дать номер клавишы для работы");

}

}

pianino(pedal pedals, button buttons){

this.pedals.add(pedals);

this.buttons.add(buttons) ;

}

}

class button{

int name;

public button(int number){

this.name=number;

}

}

class pedal{

int name;

pedal(Integer name){

}

public void setName(int name) {

this.name = name;

}

}

Скриншот результата

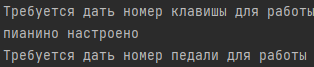


Рисунок 4 – Результат выполнения задачи 6

Задание 7

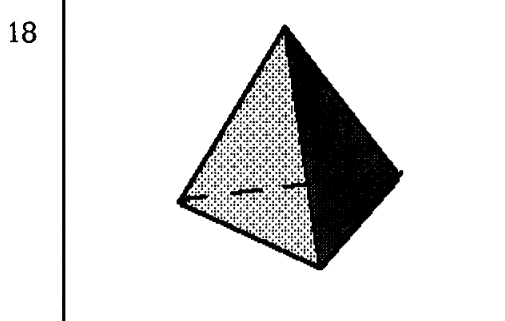


Рисунок 5 – Задача 7

Код программы

package com.company;

import java.awt.\*;

import java.awt.geom.Rectangle2D;

import java.awt.image.BufferedImage;

import javax.swing.\*;

class Gr1 extends JFrame{

Gr1(String s){

super(s);

setLayout(null);

setSize(500,500);

setVisible(true);

this.setDefaultCloseOperation(EXIT\_ON\_CLOSE);

}

public void paint(Graphics my\_picture){

my\_picture.drawLine(150, 50, 300, 200);

my\_picture.drawLine(300, 200, 225, 275);

my\_picture.drawLine(150, 50, 225, 275);

my\_picture.drawLine(225, 275, 100, 225);

my\_picture.drawLine(100, 225, 150, 50);

my\_picture.setColor(Color.black);

Polygon p = new Polygon();

p.addPoint(150,50);

p.addPoint(300,200);

p.addPoint(225,275);

my\_picture.drawPolygon(p);

my\_picture.fillPolygon(p);

my\_picture.drawLine(100,225,125,220);

my\_picture.drawLine(135,218,155,215);

my\_picture.drawLine(165,215,185, (int) 213.5);

Polygon z = new Polygon();

z.addPoint(150,50);

z.addPoint(225,275);

z.addPoint(100,225);

}

public static void main(String[] args) {

new Gr1("");

}

}

Скриншот результата

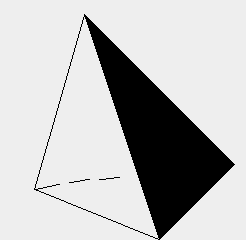


Рисунок 6 – Результат выполнения задачи 7

Задание 10

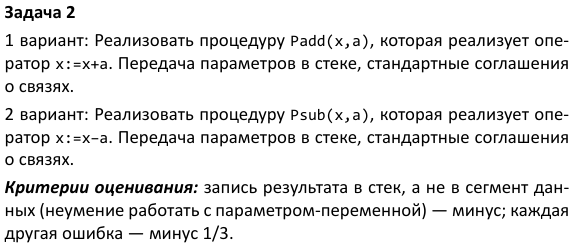


Таблица 3 – Структура программы 1 вариант

| Код команды | Мнемокод | Комментарий |
| --- | --- | --- |
| AF | XRA A | Очистить аккумулятор |
| 3E | MVI A | Записать в аккумулятор |
| 05 |  | Число x |
| 06 | MVI B | Записать в регистр B |
| 01 |  | Число a |
| 80 | ADD B | Сложить x и a |
| E7 | RST 7 | Прервать выполнение программы |

Скриншот результата

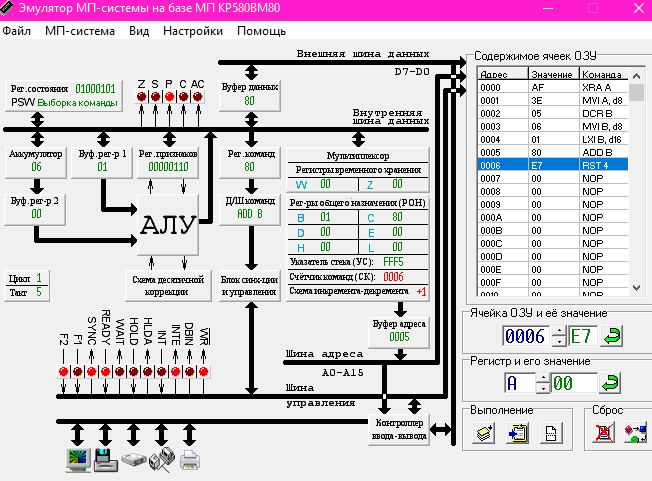


Рисунок 7 – Результат выполнения программы 1 вариант

После выполнения программы в аккумуляторе появляется число 6

Таблица 4 – Структура программы 2 вариант

| Код команды | Мнемокод | Комментарий |
| --- | --- | --- |
| AF | XRA A | Очистить аккумулятор |
| 3E | MVI A | Записать в аккумулятор |
| 05 |  | Число x |
| 06 | MVI B | Записать в регистр B |
| 01 |  | Число a |
| 90 | SUB B | Вычесть из x и a |
| E7 | RST 7 | Прервать выполнение программы |

Скриншот результата

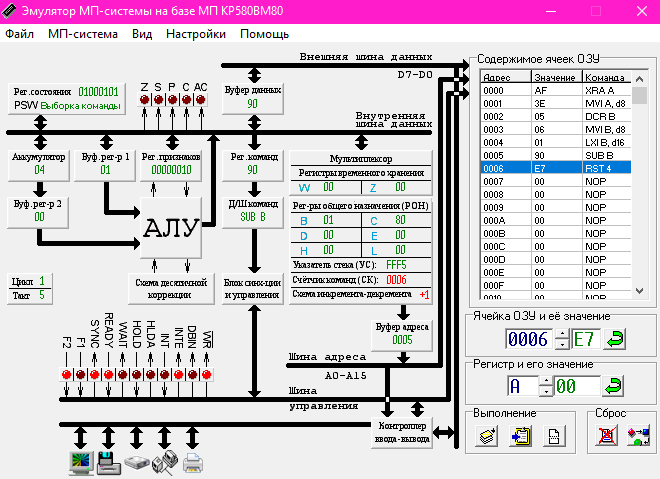


Рисунок 8 – Результат выполнения программы 2 вариант

После выполнения программы в аккумуляторе появляется число 4.

2. Разработка АИС «Издательство»

2.1 Анализ предметной области

Издательство

|  |  |
| --- | --- |
| Функциональная область | Процесс |
| Кадры | Зарплатная политика |
| Бухгалтерский учет | Учет основных средств |
|  | Учет торговых операций |
|  | Учет оплаты труда |
| Финансы | Управление финансовыми ресурсами |
| Торговля | Управление заказами |

Таблица 5 – анализ предметной области

Запишем основные действия в виде следующей таблицы.

|  |
| --- |
| Прием заказа |
| Выписка счета |
| Оплата товара |
| Проверка оплаты |
| Заказ права на печать |
| Проверка прав на печать |

Таблица 6 – основные действия

Исследуем динамику процесса товарооборота с рассмотрением участников операций, используемой информации и документов.

К торговому менеджеру обращается клиент, желающий выпустить в печать свою книгу. Может появиться лично или позвонить.

Менеджер собирает всю информацию, которая нужна для отправки данной книги на печать (будет ли интересно это аудитории, окупится ли в итоге данное действие, какую плату запросить и т.д.).

Далее после согласования каких-либо рабочих моментов происходит оплата. После оплаты согласовывается дата начала печати и выпуска книги в свет.

2.2 Техническое задание:

2.2.1 Требования к функциональным характеристикам

2.2.1.1 Состав выполняемых функций

Разрабатываемое ПО должно обеспечивать:

* сбор и анализ информации о заказах;
* сбор и анализ отзывов;
* возможность видеть отзыв на ту или иную книгу в издательстве;
* возможность увидеть какой работник ответственен за издательство;
* выдачу рекомендаций по дальнейшей работе;
* отображение заказов всех гостей;

В конце каждого месяца система должна архивировать данные.

2.2.1.2. Организация входных и выходных данных

Исходные данные в систему поступают в виде информации от менеджера, который сообщает что клиент подал заявку на издательство своей книги. Эти значе­ния отображаются на компьютере менеджера и оператора. После анализа по­ступившей информации оператор сообщает работнику, чтоб он проверил возможность осуществления издания данного материала (хватит ресурсов ли, окупится ли и т.д.).

Основной режим использования системы - ежедневная работа.

2.2.2 Требования к надежности

Для обеспечения надежности необходимо проверять кор­ректность введенных данных.

2.2.2.3.Условия эксплуатации и требования к составу и параметрам технических средств

Для работы системы должен быть выделен ответственный менеджер.

Требования к составу и параметрам технических средств уточняются на этапе эскизного проектирования системы.

2.2.4 Требования к информационной и программной совмес­тимости

Программа должна работать на платформах Windows 8/10/11.

2.2.5 Требования к транспортировке и хранению. Программа поставляется на дисковом носителе информации или отправляется на электронную почту.

Программная документация поставляется в электронном и пе­чатном виде.

2.2.6 Специальные требования

Интерфейс должен быть понятен для людей с средним уровнем знаний компьютера.

Требования к программной документации

Основными документами, регламентирующими разработку будущих программ, должны быть документы Единой Системы Программной Документации (ЕСПД); руководство пользовате­ля, руководство администратора, описание применения.

Технико-экономические показатели

Эффективность системы определяется удобством и скоростью работы данного ПО.

Порядок контроля и приемки

После передачи Исполнителем отдельного функционального модуля программы Заказчику, последний имеет право тестиро­вать модуль в течение 5 дней. После тестирования Заказчик дол­жен принять работу по данному этапу или в письменном виде изложить причину отказа от принятия. В случае обоснованного отказа Исполнитель обязуется доработать модуль

2.3 ER-диаграмма сущность-связь

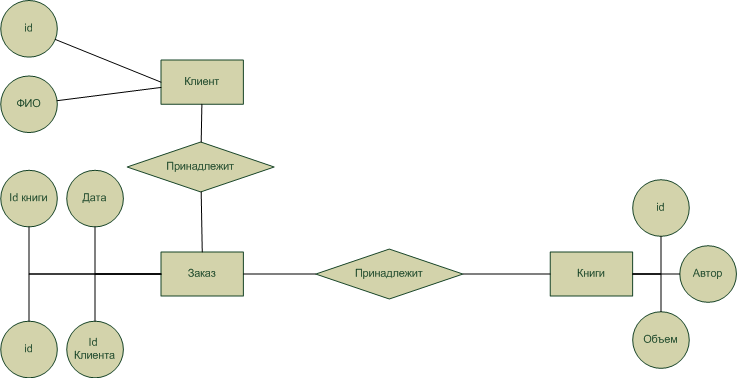


Рисунок 9 – ER-диаграмма сущность-связь

2.4 Функциональная система АИС

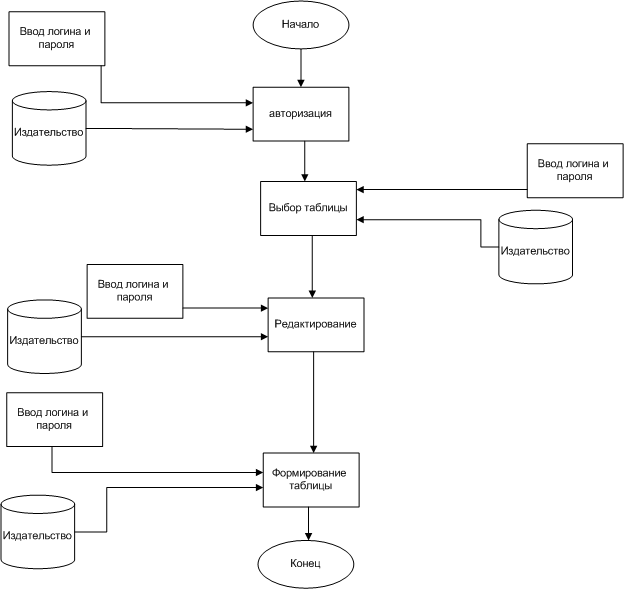


Рисунок 10 – Функциональная система АИС

Код АИС

package com.company;  
  
import java.sql.\*;  
import java.util.Scanner;  
public class Main {  
 static String username = "root";  
 static String password = "123456789";  
 static String url = "jdbc:mysql://127.0.0.1:3306/mydb";  
 static Scanner scan = new Scanner(System.in);  
 public static void main(String[] args) {  
  
 }  
 public static void vvod(){  
 try (Connection conn = DriverManager.getConnection(url, username, password)){  
 Statement add = conn.createStatement();  
 System.out.println("Введите автора книги");  
 String a = scan.next();  
 System.out.println("Введите название книги");  
 String b = scan.next();  
 System.out.println("Введите количество страниц книги");  
 int c = scan.nextInt();  
 System.out.println("Введите язык книги");  
 String d = scan.next();  
 int rows = add.executeUpdate("INSERT tabl(avtor, name, obyem, language) VALUES ('"+a + "'," +"'"+b+ "'," +"'" +c+ "',"+"'"+d+"')");  
 } catch (SQLException e) {  
 System.out.println("Connection failed...");  
 System.out.println(e);  
 }  
 }  
 public static void vivodtabl(){  
 try (Connection conn = DriverManager.getConnection(url, username, password)){  
 Statement statement = conn.createStatement();  
 ResultSet resultSet = statement.executeQuery("SELECT \* FROM tabl");  
 while(resultSet.next()){  
 int id = resultSet.getInt(1);  
 String avtor = resultSet.getString(2);  
 String name = resultSet.getString(3);  
 int obyem = resultSet.getInt(4);  
 String language = resultSet.getString(5);  
 System.out.println(id + " " + avtor+ " " + name+ " " + obyem+ " " +language +"\n");  
 }  
 }  
 catch (SQLException ex) {  
 System.out.println("Connection failed...");  
 System.out.println(ex);  
 }  
 }  
 public static void vivodpolanguage(){  
 try (Connection conn = DriverManager.getConnection(url, username, password)){  
 Statement statement = conn.createStatement();  
 System.out.println("Введите язык книги");  
 String z = scan.next();  
 ResultSet resultSet = statement.executeQuery("SELECT \* FROM tabl WHERE language='" +z+"'");  
 while(resultSet.next()){  
 int id = resultSet.getInt(1);  
 String avtor = resultSet.getString(2);  
 String name = resultSet.getString(3);  
 int obyem = resultSet.getInt(4);  
 String language = resultSet.getString(5);  
 System.out.println(id + " " + avtor+ " " + name+ " " + obyem+ " " +language +"\n");  
 }  
 }  
 catch (SQLException ex) {  
 System.out.println("Connection failed...");  
 System.out.println(ex);  
 }  
 }  
 public static void udalenie(){  
 try (Connection conn = DriverManager.getConnection(url, username, password)){  
 Statement statement = conn.createStatement();  
 System.out.println("Введите id строчки для удаления");  
 int b = scan.nextInt();  
 int rows = statement.executeUpdate("DELETE FROM tabl WHERE Id ="+b);  
 } catch (SQLException e) {  
 System.out.println("Connection failed...");  
 System.out.println(e);  
 }  
 }  
}

В коде программы есть несколько методов, метод vvod, чтоб добивать новую запись в таблицу, vivodtabl, позволяющий вывести данные присутствующие в базе данных, udalenie, позволяющий удалить запись из базы данных, vivodpolanguage, позволяющий вывести БД с выборкой по языку книги.

Вывод изначальной таблицы с помощью метода vivodtabl

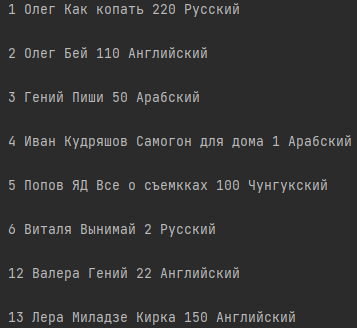


Рисунок 11 – Вывод изначальных данных в бд

Добавления строки в таблицу и последующий вывод таблицы с помощью методов vvod и vivodtabl

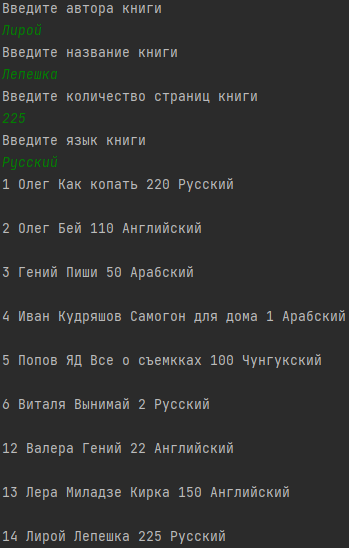


Рисунок 12 – добавление строки и последующий вывод данных их БД

Удаление строки по введеному id с помощью метода udalenie и последующий вывод БД методом vivodtabl

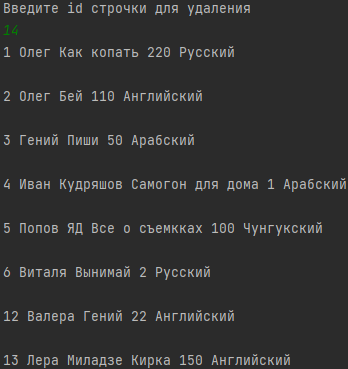


Рисунок 13 – удаление строки из БД

3 Разработка мобильного приложения[2]

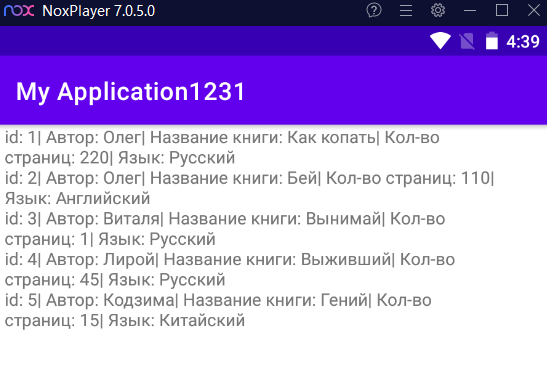


Рисунок 14 – Интерфейс мобильного приложения

Получившийся программный продукт крайне эффективно использует ресурсы устройства, в связи с низким весом программы и отсутствием сложных вычислений. Приложение доступно для любого устройства с версией Android 4.0 и выше.

4 Создание сайта с использованием CMS

Сайт разрабатывался при помощи бесплатного хостинга Google Sites [5] с встроенным конструктором.

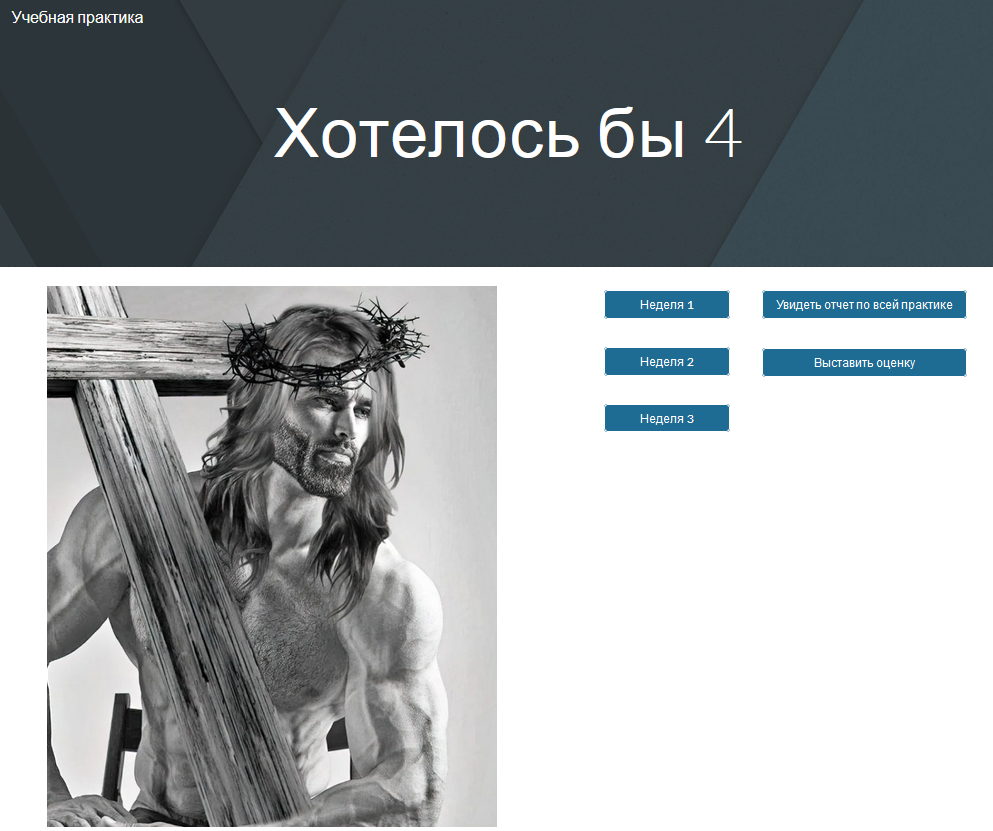


Рисунок 15 – Главная страница сайта

Ссылка на сайт

<https://sites.google.com/view/pomogite325/%D0%B3%D0%BB%D0%B0%D0%B2%D0%BD%D0%B0%D1%8F-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0>

Заключение

В ходе прохождения практики я освоил навыки по разработке автоматических информационных систем, мобильных приложений, закрепил знания в области проектирования программных интерфейсов и баз данных.

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

Список использованных источников

1. Гост TIA/EIA-569 <https://www.system-administrator.pro/files/documents/TIA-EIA-569-A.pdf>
2. Ссылка на скачивание Android studio: <https://developer.android.com/studio>
3. Документация по MySQL: <https://dev.mysql.com/doc/>
4. Ссылка на скачивание IntelijIdea: <https://www.jetbrains.com/ru-ru/idea/>
5. Google Sites: <https://sites.google.com/new>