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  Денисов Евгений 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 | 7 |
| 1.3 Задача 3 | 14 |
| 1.4 Задача 4 | 18 |
| 1.5 Задача 5 | 21 |
| 2 Разработка АИС «Сувениры» | 23 |
| 2.1 ER-диаграмма сущность-связь | 23 |
| 3 Разработка сайта | 23 |
| Заключение | 26 |
| Список использованных источников | 27 |

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.

Решение задач

Задание1

Ввести с консоли n-размерность матрицы a [n] [n]. Задать значения элементов матрицы в интервале значений от -n до n с помощью датчика случайных чисел.

Упорядочить строки (столбцы) матрицы в порядке возрастания значений

элементов k-го столбца (строки).

Maim.java

package com.company;

import java.util.Random;

public class Main {

public static int getRandom(int min, int max) {

Random ran = new Random();

int n = max - min + 1;

int i = ran.nextInt() % n;

if (i < 0)

i = -i;

return min + i;

}

public static void main(String[] args)

{

int n = getRandom(2, 10), m = getRandom(2, 10), q = 0;

System.out.println("n = " + n + " ;m = " + m);

int[][] a = new int[n][m];

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

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

a[i][j] = getRandom(-100, 100);

}

}

System.out.println("Созданная матрица");

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

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

System.out.print(a[i][j] + "\t");

}

System.out.println();

}

// bubble sort

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

for (int j = i; j < m \* n; j++) {

if (a[ i / m][ i % m] > a[ j / m][ j % m]) {

q = a[ i / m][ i % m];

a[ i / m][ i % m] = a[ j / m][ j % m];

a[ j / m][ j % m] = q;

}

}

}

System.out.println("Отсортированная матрица");

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

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

System.out.print(a[i][j] + "\t");

}

System.out.println();

}

}

}

Блок-схема:

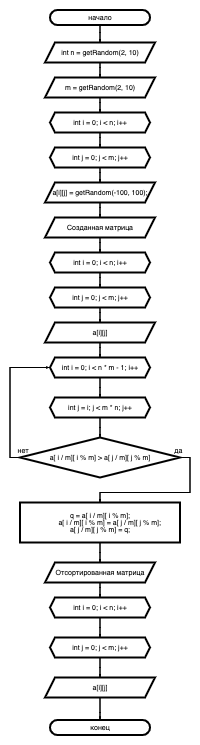


Рисунок 1 – Блок-схема к задаче1

Скриншот результата:

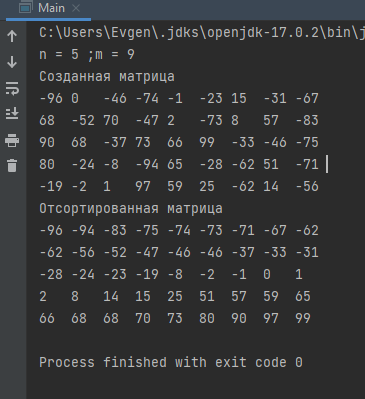


Рисунок 2 – Результат выполнения задачи 1

Задание 2

Book: id, Название, Автор (ы), Издательство, Год издания, Количество страниц, Цена, Тип переплета.

Создать массив объектов. Вывести:

a) список книг заданного автора;

b) список книг, выпущенных заданным издательством;

c) список книг, выпущенных после заданного года.

Main.java

package com.company;

import java.lang.reflect.Array;

import java.util.Arrays;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

////////////////++++++++++++

book[] book\_list = books(); ///// ручное заполнение

//book[] book\_list = bo(); ////// автозаполнение

System.out.println("----------a)Вывести список книг заданного автора;");

System.out.print("Введите автора: ");

check\_autor(book\_list,scanner.next());

System.out.println("----------b)Вывести список книг, выпущенных заданным издательством;");

System.out.print("Введите издателя: ");

check\_publisher(book\_list,scanner.next());

System.out.println("----------c)Вывести список книг, выпущенных после заданного года.");

System.out.print("Введите год: ");

check\_year(book\_list,scanner.nextInt());

}

///////////////// 5 Book

static book[] books() {

Scanner scanner1 = new Scanner(System.in);

System.out.print("Введите количество книг: ");

book[] books = new book[scanner1.nextInt()];

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

books[i] = new book();

System.out.println();

System.out.print("Введите id: ");

books[i].setId(scanner1.nextInt());

System.out.print("Введите название книги: ");

books[i].setName(scanner1.next());

System.out.print("Введите количество авторов: ");

int n = scanner1.nextInt();

String[] mass = new String[n];

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

System.out.print("Введите автора: ");

mass[j] = scanner1.next();

}

books[i].setAutor(mass);

System.out.print("Введите издательство: ");

books[i].setPublisher(scanner1.next());

System.out.print("Введите год издания: ");

books[i].setYear(scanner1.nextInt());

System.out.print("Введите количество страниц: ");

books[i].setPages(scanner1.nextInt());

System.out.print("Введите цену: ");

books[i].setPrice(scanner1.nextInt());

System.out.print("Введите тип переплета: ");

books[i].setCover(scanner1.next());

}

return books;

}

static book[] bo(){

book[] book\_list = new book[3];

String[] autors = {"Толкин"};

book\_list[0]= new book(0,"Хоббит",autors,"Единорог",1976,322,560,"Твердый переплет");

String[] autors1 = {"Пупкин"};

book\_list[1]= new book(1,"Книга",autors1,"Общество Атеистов",1994,752,1260,"Твердый переплет");

String[] autors2 = {"Пупкин","Семенов"};

book\_list[2]= new book(2,"Мифы",autors2,"Единорог",1960,152,320,"Твердый переплет");

return book\_list;

}

static void check\_autor(book[] books,String autor){

for (book book : books) {

String[] autors = book.getAutor();

for (String s : autors) {

if(s.equals(autor)){

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

break;

}

}

}

}

static void check\_publisher(book[] books, String publisher){

for (book book : books) {

if(book.getPublisher().equals(publisher)){

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

}

}

}

static void check\_year(book[] books, int year){

for (book book : books) {

if(year<book.getYear()){

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

}

}

}

}

Book.java

package com.company;

import java.util.Arrays;

public class book {

private int id;

private String name;

private String[] autor;

private String publisher;

private int year;

private int pages;

private int price;

private String cover;

public book(){

this.id = 0;

this.name = "";

this.autor = new String[0];

this.publisher = "";

this.year = 0;

this.pages = 0;

this.price = 0;

this.cover = "";

}

public book(int id,String name, String[] autor, String publisher, int year, int pages, int price, String cover) {

this.id = id;

this.name = name;

this.autor = autor;

this.publisher = publisher;

this.year = year;

this.pages = pages;

this.price = price;

this.cover = cover;

}

void setId(int id){

this.id = id;

}

int getId(){

return id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

void setAutor(String[] autor){

this.autor = autor;

}

String[] getAutor(){

return autor;

}

void setPublisher(String publisher){

this.publisher = publisher;

}

String getPublisher(){

return publisher;

}

void setYear(int year){

this.year = year;

}

int getYear(){

return year;

}

void setPages(int pages){

if(pages>0) {

this.pages = pages;

} else System.out.println("Количестов не может быть меньше 1!");

}

int getPages(){

return pages;

}

void setPrice(int price){

if(price>0) {

this.price = price;

} else System.out.println("Цена неможет быть отрицательной и нулевой!");

}

int getPrice(){

return price;

}

void setCover(String cover){

this.cover = cover;

}

String getCover(){

return cover;

}

@Override

public String toString() {

return "Книга --- " +

"id=" + id +

", Название=" + name +

", Автор(ы)=" + Arrays.toString(autor) +

", Издатель=" + publisher +

", Год издания=" + year +

", Количество страниц=" + pages +

", Цена=" + price +

", Переплет=" + cover;

}

}

Скриншот результата:

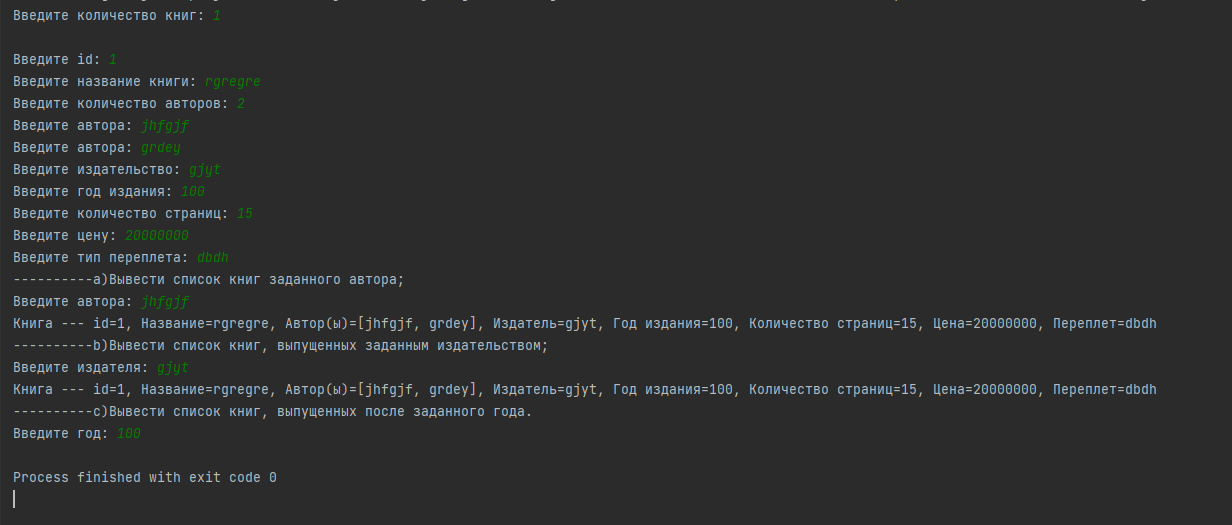


Рисунок 3 – Результат выполнения задачи 2

Задание 3

Создать объект класса Планета, используя классы Материк,Океан,Остров.

Методы: вывести на консоль название материка, планеты, количество материков.

PlanetRunner.java

import java.util.List;

import java.util.ArrayList;

public class PlanetRunner {

public static void main(String[] args) {

Island island = new Island("Barbados");

Planet planet = new Planet("Earth");

planet.addContinent(new Continent("Eurasia"));

planet.addContinent(new Continent("Africa"));

planet.addOcean(new Ocean("Atlantic"));

planet.addOcean(new Ocean("Pacific"));

planet.addIsland(new Island("Barbados"));

System.out.println("Planet name: "+planet.getName());

System.out.println("Continent name: "+getFirstContinentName(planet.getContinentList()));

System.out.println("Count continents:" + planet.getContinentList().size());

}

public static String getFirstContinentName(List<Continent> continentList) {

String result = null;

for (Continent continent: continentList) {

result = continent.getName();

}

return result;

}

}

Planet.java

import java.util.ArrayList;

import java.util.List;

public class Planet {

private String name;

private List<Ocean> oceanList = new ArrayList<>();

private List<Continent> continentList = new ArrayList<>();

private List<Island> islandList = new ArrayList<>();

public Planet(String name) {

this.name = name;

}

public void addOcean(Ocean ocean) {

oceanList.add(ocean);

}

public void addContinent(Continent continent) {

continentList.add(continent);

}

public void addIsland(Island island) {

islandList.add(island);

}

public String getName() {

return name;

}

public List<Ocean> getOceanList() {

return oceanList;

}

public List<Continent> getContinentList() {

return continentList;

}

public List<Island> getIslandList() {

return islandList;

}

}

Ocean.java

public class Ocean {

private String name;

public Ocean(String name) {

this.name = name;

}

public String getName() {

return name;

}

}

Island.java

public class Island {

private String name;

public Island(String name) {

this.name = name;

}

public String getName() {

return name;

}

}

Continent.java

public class Continent {

private String name;

public Continent(String name) {

this.name = name;

}

public String getName() {

return name;

}

}

Скриншот результата:

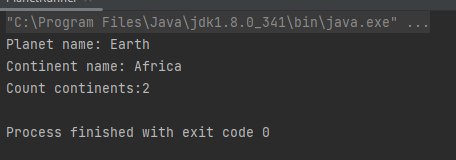


Рисунок 4 – Результат выполнения задачи 3

Задача 4

Создать класс Department с внутренним классом, с помощью объектов

которого можно хранить информацию обо всех должностях отдела и обо

всех сотрудниках, когда-либо занимавших конкретную должность

Main.java

import java.util.Scanner;public class Main {

public static void main(String[] args){

Department ct = new Department(); ct.info.print(); }

public static class Department {

public static Scanner ob=new Scanner(System.in); public info info; public String city; public Department(){

System.out.print("введите название отдела фирмы "); setDepartment(ob.next()); info=new info(); }

public String getDepartment(){

return city; }

public void setDepartment(String city){

this.city=city; }

public class info{

int pr,st,sq; public info(){

System.out.print("Введите количество должностей отдела: "); setPr(ob.nextInt()); System.out.print("количество сотрудников:"); setSt(ob.nextInt()); System.out.print("введите количество филиалов:"); setSq(ob.nextInt()); }

public int getPr(){

return pr; }

public void setPr(int pr){

this.pr=pr; }

public int getSt(){

return st; }

public void setSt(int st){

this.st=st; }

public int getSq(){

return sq; }

public void setSq(int sq){

this.sq=sq; }

public void print(){

System.out.print("На фирме "+" "+getDepartment()+" "+getPr()+" сотрудников "+getSt()+" должностей "+getSq()+" филиалов "); }

}

}

}

Блок-схема:

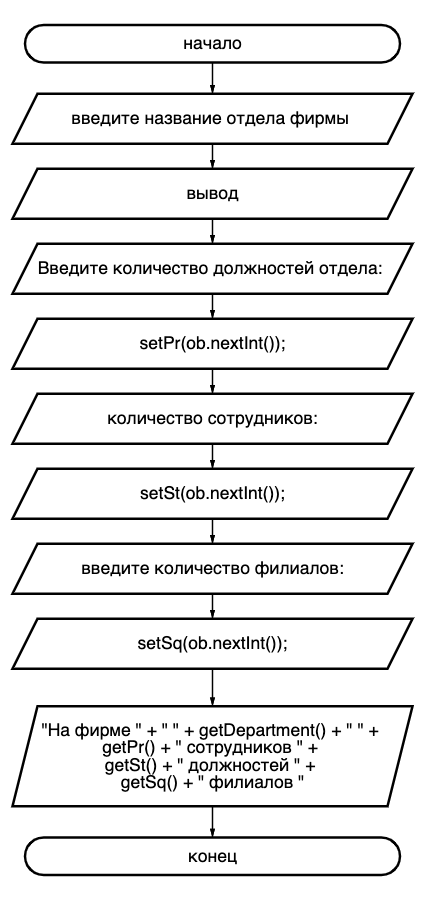


Рисунок 4 – Блок-схема к задаче 4

Скриншот результата:

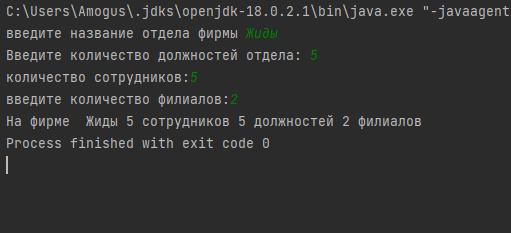


Рисунок 5 – Результат выполнения задачи 4

Задание 5

После каждого слова текста, заканчивающегося заданной подстрокой,

вставить указанное слово.

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);

}

}

Скриншот результата:

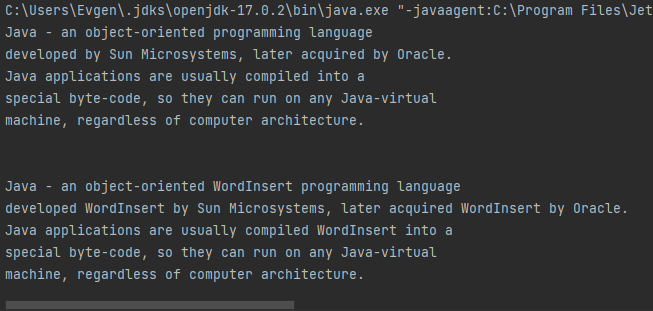


Рисунок 6 – Результат выполнения задачи 5

Разработка АИС «Сувениры»

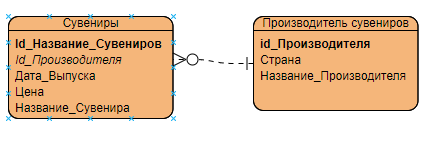
ER-диаграмма сущность-связь

Рисунок 7 - ER-диаграмма сущность-связь

Разработка сайта

Для разработки сайта был использован сайт Google Sites, который в своём арсенале имеет все необходимые инструменты для создания сайта.

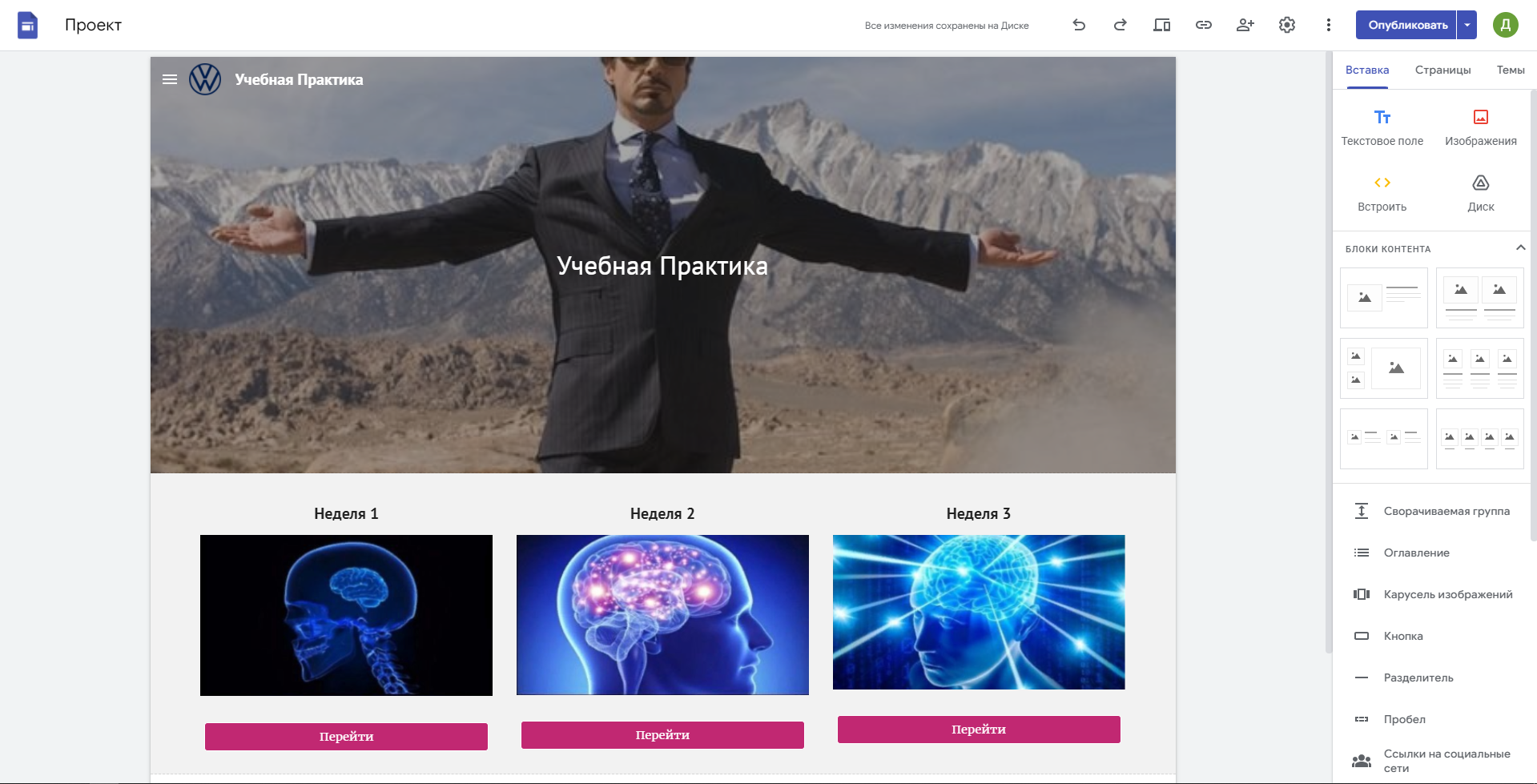


Рисунок 8 – Главная страница сайта

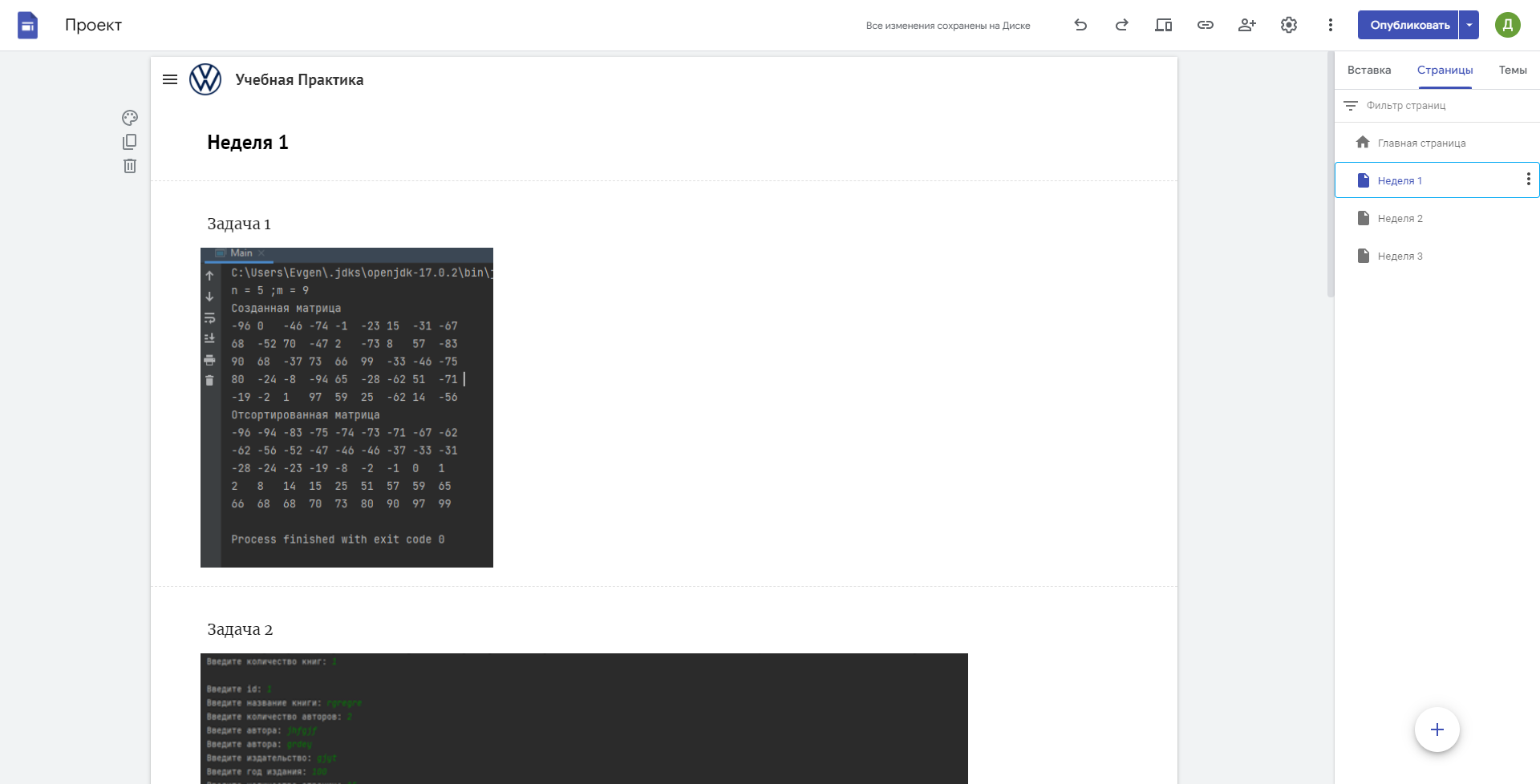


Рисунок 9 – Страница с заданиями за неделю 1

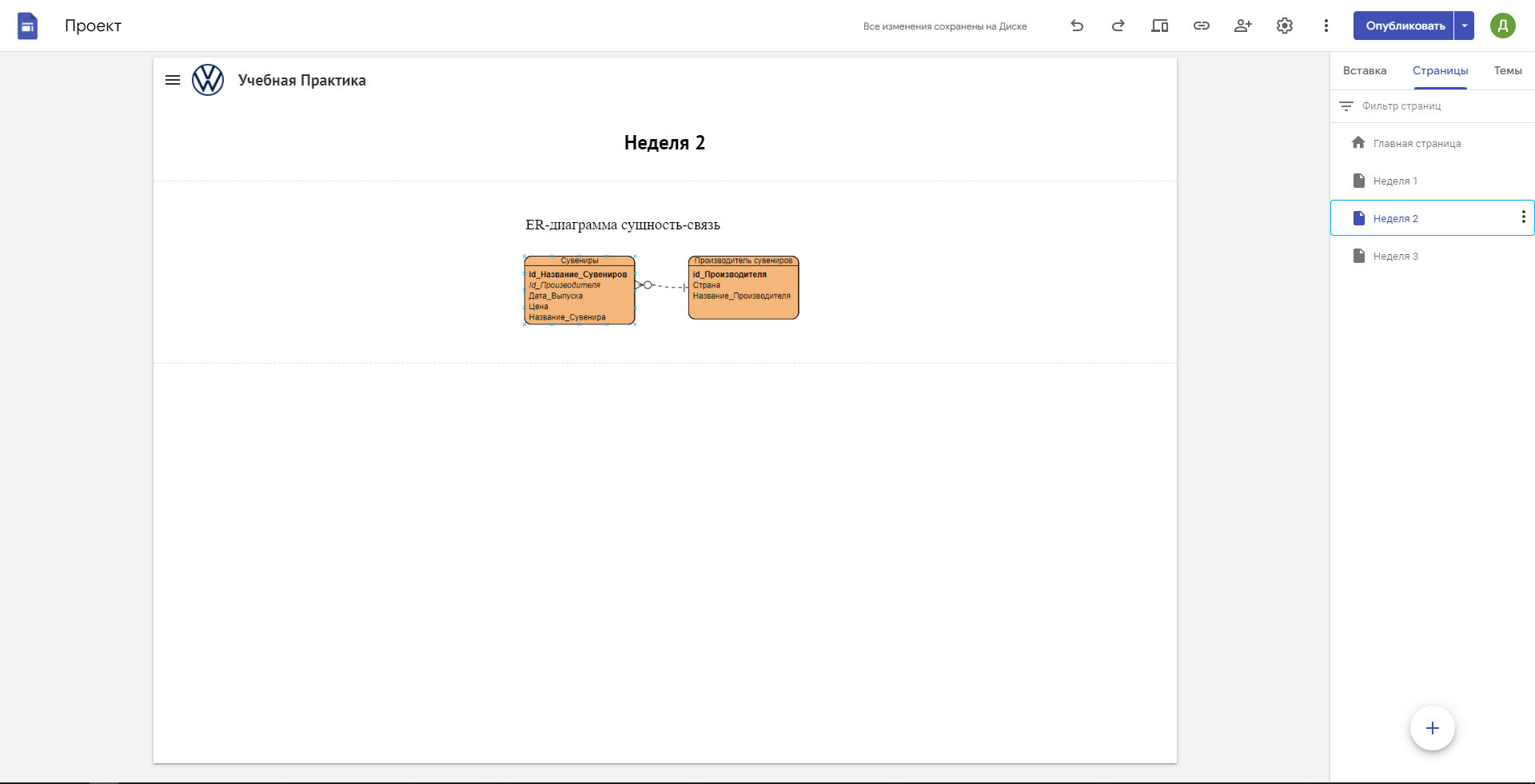


Рисунок 10 – Страница с заданиями за неделю 2

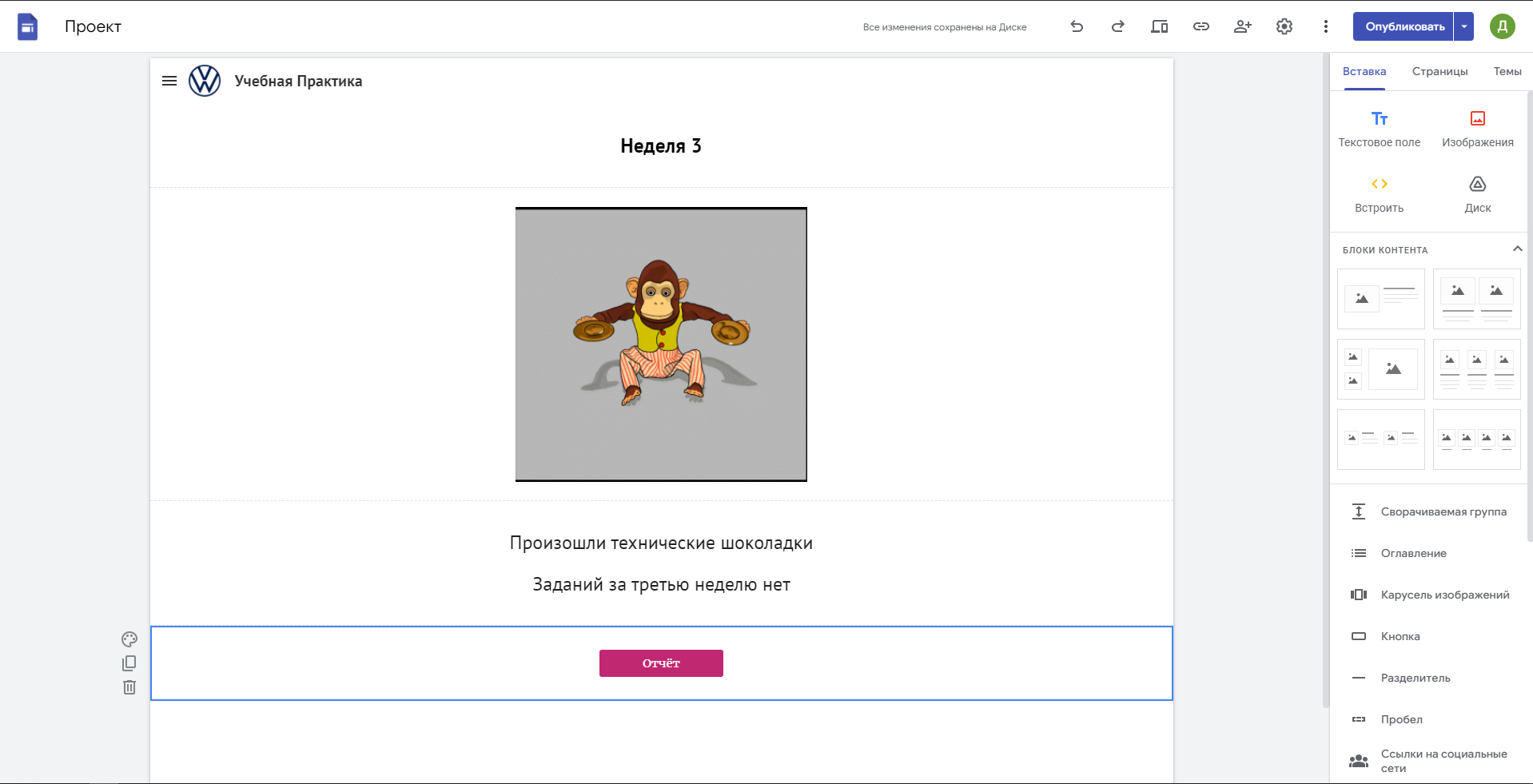


Рисунок 11 – Страница с заданиями за неделю 3

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, разработана АИС «Сувениры». Для демонстрации выполненных заданий был создан сайт с использованием хостинга Google Site.

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/>