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 «12» октября 2022 г. по «01» ноября 2022 г.

Место практики ГБУ КО ПОО «КИТиС»

|  |  |
| --- | --- |
| Выполнил: | студент 4 курса,  группы ИСп19-2к  Макеев Дмитрий Константинович  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |
| Проверила: | Большакова-Стрекалова Анна Викторовна  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (оценка)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) |

Калининград, 2022

Содержание

Введение…………………………………………………………………………...3

1.Раздел…………………………………………………………………………….4

1.1 Задача 1………………………………………………………………………...4

1.2 Задача 2………………………………………………………………………...7

1.3 Задача 3……………………………………………………………………….12

1.4 Задача 4……………………………………………………………………….16

1.5 Задача 5……………………………………………………………………….22

1.6 Задача 6……………………………………………………………………….25

2.Раздел

Разработка АИС…………………………………………………………………32

Создание Сайта……………………………………………………………….…36

Список использованных источников……………………………………….…33

Заключение………………………………………………………………………37

Введение

Данный отчет составлен по учебной практике УП.01 Разработка модулей программного обеспечения для компьютерных систем.

Цели:

1. Написать консольные задачи в среде разработки InteliJ IDEA.
2. Разработка АИС.
3. Разработка ER-диаграммы.
4. Сформировать «Техническое задание».
5. Проектирование и реализация интерфейса.
6. Разработать встроенную систему справочного руководства и программного документа «Руководство пользователя».
7. Проведение тестирования.
8. Обеспечить механизм авторизации и работы как минимум двух пользователей, разграничить права.

1 Основной раздел

1.1 Задача 1

Условие:

Вывести числа от 1 до k в виде матрицы N x N слева направо и сверху вниз.

Код программы:

package com.epam.khalii.Task2 2;

import java.util.\*;

/\*\*

\* Created by Anhelina\_Khalii on 16.04.2015.

\*/

public class ConsoleInput {

public static void printShorter (ArrayList<Integer> args) {

int size = Integer.toString(Integer.MAX\_VALUE).length(), el=Integer.MAX\_VALUE;

for (int iterator : args) {

int itsize = Integer.toString(iterator).length();

String siterator = Integer.toString(iterator);

if (itsize < size) {

size = itsize;

el = iterator;

}

}

System.out.println("Minimal length: "+size+" character in "+el+" element");

}

public static void printLonger(ArrayList<Integer> args){

int size = 0, el=0;

for (int iterator : args) {

int itsize = Integer.toString(iterator).length();

String siterator = Integer.toString(iterator);

if (itsize > size) {

size = itsize;

el = iterator;

}

}

System.out.println("Maximal length: "+size+" character in "+el+" element");

}

private static void insertSortDesc(String[] array){

for(int i=1; i<array.length;i++) {

String cur = array[i];

int j;

for(j=i-1;j>=0&&array[j].length()>cur.length();j--)

array[j+1]=array[j];

array[j+1]=cur;

}

}

private static void insertSortAsc(String[] array){

for(int i=1; i<array.length;i++) {

String cur = array[i];

int j;

for(j=i-1;j>=0&&array[j].length()<cur.length();j--)

array[j+1]=array[j];

array[j+1]=cur;

}

}

public static void printArray(String[] array){

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

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

System.out.println();

}

public static void sortDesc (ArrayList<Integer> args){

String[] array = new String[args.size()];

for(int iterator=0; iterator<args.size();iterator++) {

array[iterator] = (Integer.toString(args.get(iterator)));

}

insertSortDesc(array);

System.out.println("Array sort by count of numbers(desc): ");

printArray(array);

}

public static void sortAsc (ArrayList<Integer> args){

String[] array = new String[args.size()];

for(int iterator=0; iterator<args.size();iterator++) {

array[iterator] = (Integer.toString(args.get(iterator)));

}

insertSortAsc(array);

System.out.println("Array sort by count of numbers(asc): ");

printArray(array);

}

public static void lessLength(ArrayList<Integer> args){

ArrayList<String> result = new ArrayList<String>();

int average=0;

for(int iterator:args){

result.add(Integer.toString(iterator));

average+=Integer.toString(iterator).length();

}

average/=result.size();

System.out.println("Average is: " + average);

System.out.print("Element with lower then average size: ");

for(int i=0;i<result.size();i++){

if(result.get(i).length()<average){

System.out.print("element "+result.get(i)+" length "+result.get(i).length()+", ");

}

}

System.out.println();

}

public static void muchLength(ArrayList<Integer> args){

ArrayList<String> result = new ArrayList<String>();

int average=0;

for(int iterator:args){

result.add(Integer.toString(iterator));

average+=Integer.toString(iterator).length();

}

average/=result.size();

System.out.println("Average is: " + average);

System.out.print("Element with lower then average size: ");

for(int i=0;i<result.size();i++){

if(result.get(i).length()>average){

System.out.print("element "+result.get(i)+" length "+result.get(i).length()+", ");

}

}

System.out.println();

}

private static boolean isUnique(String st){

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

for (int j = i+1;j<st.length();j++){

if (st.charAt(i)==st.charAt(j))

return false;

}

}

return true;

}

private static int uniqueNum(String st){

int[] nums = new int[10]; // counter 0 1 2 3 4 5 6 7 8 9

int count=0;

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

nums[Character.getNumericValue(st.charAt(i))]++;

}

for(int i = 0; i<10;i++) {

if(nums[i]==1) count++;

}

return count;

}

public static void minDifDigit(ArrayList<Integer> args){

ArrayList<String> result = new ArrayList<String>();

String ="";

for(int : args){

if(Integer.toString(iterator).length()<11){

result.add(Integer.toString(iterator));

}

}

int min = 11;

for (String st: result){

if(uniqueNum(st)<min){

min=uniqueNum (st);

winner = st;

}

}

System.out.println("Number with minimal count of unique digit is "+winner);

}

private static boolean isEvenOdd (String st){

int even=0, odd=0;

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

if(Character.getNumericValue(st.charAt(i))%2==0)

even++;

else odd++;

}

if(even==odd)

return true;

else

return false;

}

public static void evenEvenOdd(ArrayList<Integer> args){

ArrayList<String> result = new ArrayList<String>();

for(int iterator:args){

if(iterator%2==0){

result.add(Integer.toString(iterator));

}

}

int count=0;

System.out.println("Numbers with equal number of even and odd digits:");

for(String st: result){

if(isEvenOdd(st)){

System.out.print(st+" ");

count++;

}

}

System.out.println("\nCount of even numbers with equal number of even and odd digits is "+count);

}

//6. Íàéòè ÷èñëî, öèôðû â êîòîðîì èäóò â ñòðîãîì ïîðÿäêå âîçðàñòàíèÿ. Åñëè òàêèõ ÷èñåë

//íåñêîëüêî, íàéòè ïåðâîå èç íèõ.

private static boolean isProgression(String st){

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

for (int j = i+1;j<st.length();j++){

if (Character.getNumericValue(st.charAt(i))>Character.getNumericValue(st.charAt(j)))

return false;

}

}

return true;

}

public static void arifmProgr(ArrayList<Integer> args){

ArrayList<String> result = new ArrayList<String>();

String winner="";

for(int iterator:args){

if (iterator>9)

result.add(Integer.toString(iterator));

}

for(String st: result){

if(isProgression(st)){

winner = st;

System.out.println("Number with digit in arithmetic progression "+winner);

break;

}

}

}

public static void difDigit(ArrayList<Integer> args){

ArrayList<String> result = new ArrayList<String>();

String winner="";

for(int iterator:args){

if(Integer.toString(iterator).length()<11&&Integer.toString(iterator).length()>1){

result.add(Integer.toString(iterator));

}

}

for(String st: result)

if (isUnique(st)) {

winner = st;

System.out.println("Number with unique digits is " + winner);

break;

}

}

//8. Âûâåñòè ÷èñëà îò 1 äî k â âèäå ìàòðèöû N x N ñëåâà íàïðàâî è ñâåðõó âíèç.

public static void printMatrix(int size){

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

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

System.out.print(i\*size+j+1+" ");

}

System.out.println();

}

}

public static void entryPoint(){

System.out.println("Please, tap some number of integers in row. ");

Scanner sc = new Scanner(System.in);

try {

if (sc.hasNext()) {

String[] nums = sc.nextLine().split("\\s+");

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

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

argum.add(Integer.parseInt(nums[i]));

}

printShorter(argum);

printLonger(argum);

sortDesc(argum);

sortAsc(argum);

lessLength(argum);

muchLength(argum);

minDifDigit(argum);

evenEvenOdd(argum);

arifmProgr(argum);

difDigit(argum);

printMatrix(nums.length);

}

} catch (NumberFormatException ex) {

System.out.println("You tap wrong arguments!");

return;

}

}

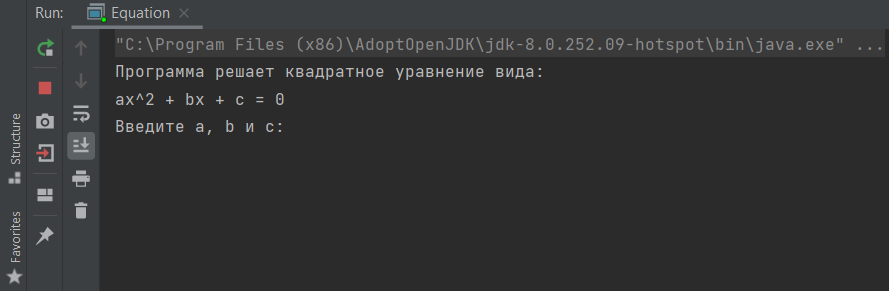
}

Рисунок 1 - Результат выполнения

1.2 Задача 2

Условие:

Создать классы, спецификации которых приведены ниже. Определить конструкторы и методы setТип(), getТип(), toString(). Определить дополнительно.

Customer: id, Фамилия, Имя, Отчество, Адрес, Номер кредитной карточки, Номер банковского счета. Создать массив объектов. Вывести: a) список покупателей в алфавитном порядке; b) список покупателей, у которых номер кредитной карточки находится в заданном интервале.

Код программы:

import java.util.ArrayList;

import java.util.Collections;

import java.util.Comparator;

import java.util.List;

public class Customers {

public static void main(String[] args) {

Shop market = new Shop("Keshenya");

market.addCustomerToShopList(new Customer("Дмитрий", "Имануилов", "Алексеевич", 60\_45\_48, 4589));

market.addCustomerToShopList(new Customer("Василий", "Тарасенко", "Петрович", 25\_45\_48, 1668));

market.addCustomerToShopList(new Customer("Алесей", "Колесник", "Андреевич", 30\_45\_89, 1598));

market.addCustomerToShopList(new Customer("Борис", "Македон", "Станиславович", 90\_47\_48, 4785));

market.addCustomerToShopList(new Customer("Дмитрий", "Турецкий", "Федорович", 40\_45\_45, 2569));

System.out.println("\nsort by name:");

List<Customer> listSortByName = market.getListName();

for (Customer c : listSortByName) {

System.out.println(c);

}

System.out.println("\nsort by surname:");

List<Customer> listSortBySurname = market.getListSurname();

for (Customer c : listSortBySurname) {

System.out.println(c);

}

System.out.println("\ndiapason card number:");

List<Customer> diapasonCardumber = market.getListDiapasonCardNumber(254548, 554556);

for (Customer c : diapasonCardumber) {

System.out.println(c);

}

}

}

class Shop {

private String ShopName;

private ArrayList<Customer> customerList = new ArrayList<>();

Shop(String ShopName) {

this.ShopName = ShopName;

}

void addCustomerToShopList(Customer customer) {

customerList.add(customer);

}

List<Customer> getListName() {

List<Customer> list = new ArrayList<>(customerList);

Collections.sort(list, new Comparator<Customer>() {

@Override

public int compare(Customer o1, Customer o2) {

return o1.getName().compareToIgnoreCase(o2.getName());

}

});

return list;

}

List<Customer> getListSurname() {

List<Customer> list = new ArrayList<>(customerList);

Collections.sort(list, new Comparator<Customer>() {

@Override

public int compare(Customer o1, Customer o2) {

return o1.getSurname().compareToIgnoreCase(o2.getSurname());

}

});

return list;

}

List<Customer> getListDiapasonCardNumber(int diapasonStart, int diapasonEnd) {

List<Customer> list = new ArrayList<>();

for (Customer c : customerList) {

if (c.getCardNumber() >= diapasonStart && c.getCardNumber() <= diapasonEnd) {

list.add(c);

}

}

return list;

}

}

class Customer {

{

idGenertor++;

}

private static int idGenertor = 0;

private int id = idGenertor;

private String surname;

private String name;

private String patronymic;

private int CardNumber;

private int BankNumber;

Customer(String surname, String name, String patronymic, int CardNumber, int BankNumber) {

this.name = name;

this.surname = surname;

this.patronymic = patronymic;

this.CardNumber = CardNumber;

this.BankNumber = BankNumber;

}

public String getName() {

return name;

}

public String getSurname() {

return surname;

}

public String getPatronymic() {

return patronymic;

}

public int getCardNumber() {

return CardNumber;

}

public int getBankNumber() {

return BankNumber;

}

public String toString() {

return String.format("ID: %d\t name: %s\t surname: %s\t patronymic %s \t CardNumber %d\t BankNumber %d",

id, surname, name, patronymic, CardNumber, BankNumber);

}

}

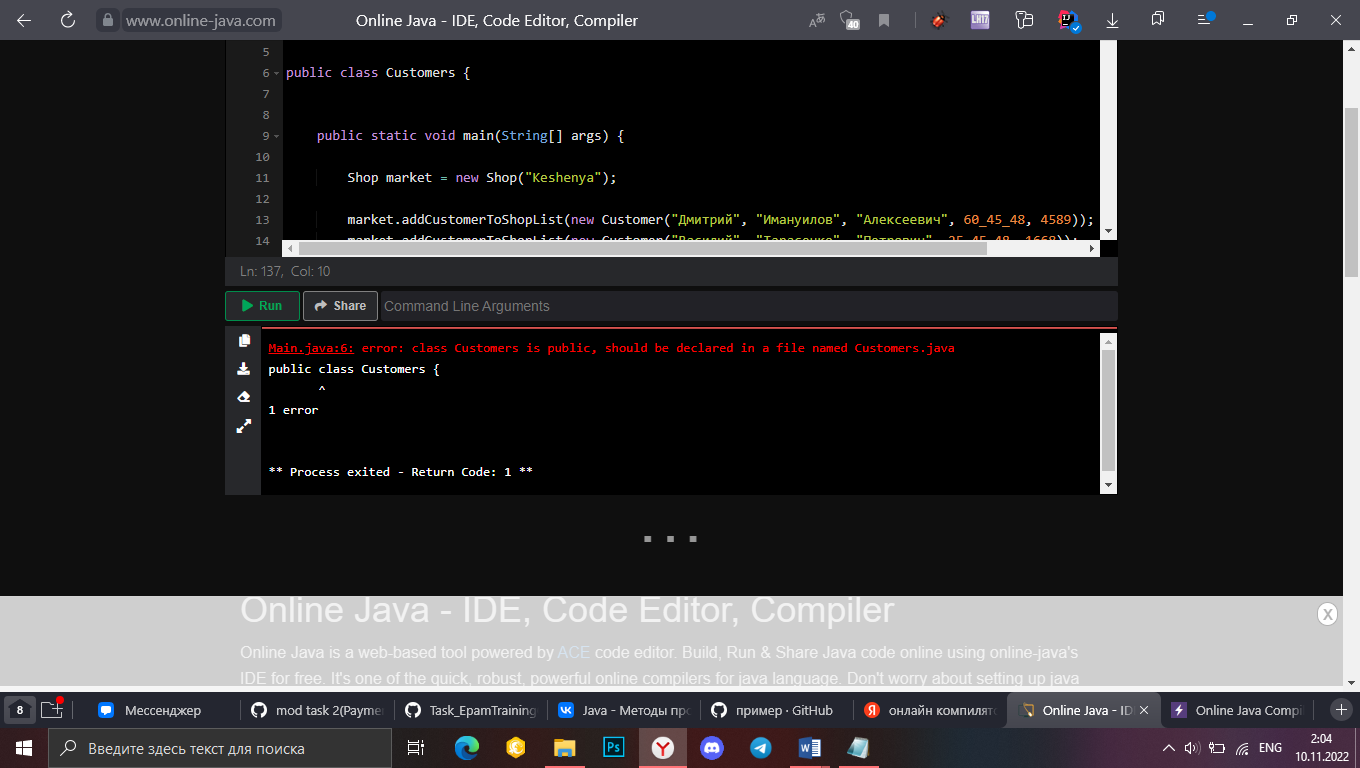


Рисунок 2 - Результат выполнения

1.3 Задача 3

Условие:

Создать объект класса Автомобиль, используя классы Колесо, Двигатель. Методы: ехать, заправляться, менять колесо, вывести на консоль марку автомобиля.

Код программы:

package by.krava.etc.unit4.aggregation\_and\_composition.t02\_car;

/\*

\* Создать объект класса Автомобиль, используя классы Колесо, Двигатель.

\* Методы: ехать, заправляться, менять колесо, вывести на консоль марку автомобиля.

\*/

public class Main {

public static void main (String[] args) {

Car = new Car ("Porsche 911");

System.out.printf("We created a car %s. Trying to drive:%n", porsche911.getModelName());

porsche911.run();

Engine engine = new Engine (370, 2.981);

porsche911.setEngine (engine);

System.out.printf("%nA %d hp engine is installed. A volume of %.3f liters. Trying to drive:%n",

engine.getPower(), engine.getVolume());

porsche911.run();

int whellDiameter = 19;

porsche911.setWheels(new Wheel (whellDiameter), new Wheel(whellDiameter), new Wheel(whellDiameter), new Wheel(whellDiameter));

System.out.printf("%nWheels are installed. Trying to drive:\n");

porsche911.run();

porsche911.getWheels().get(2).breakWheel();

System.out.println("\nSTOP! Something broke.\nTrying to drive:");

porsche911.run();

porsche911.changeWheel();

System.out.println("\nTrying to drive:");

porsche911.run();

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

porsche911.setFuel(false);

porsche911.run();

porsche911.refuel();

porsche911.run();

System.out.println("\nCar model: " + porsche911.getModelName());

}

}

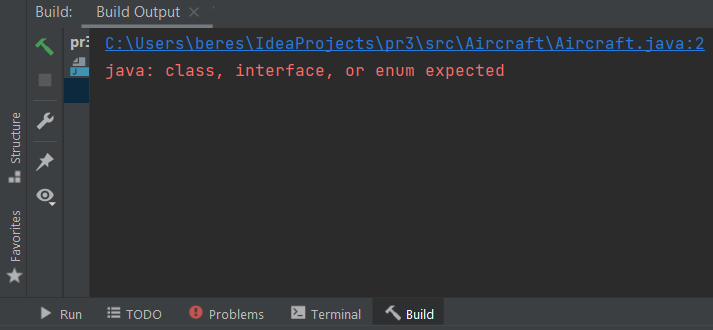


Рисунок 3 - Результат выполнения

1.4 Задача 4

Условие:

Создать класс Payment с внутренним классом, с помощью объектов которого можно сформировать покупку из нескольких товаров

Код программы:

package Task\_02\_Payment;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Main {

public static void main(String[] args) throws {

Payment payment = new Payment ();

Payment.Order = payment.new Order();

purchase.createShoppingList();

payment.amountMoneyPay(purchase.getOrderList());

payment.printShoppingList(purchase.getOrderList());

}

}

**src/Task\_02\_Payment/Payment.java**

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.HashMap;

import java.util.Map;

import static Task\_02\_Payment.DatabaseProduct.\*;

import static Task\_02\_Payment.DataUtilProduct.\*;

/\*

Создать класс Payment с внутренним классом,

с помощью объектов которого можно сформировать покупку из нескольких товаров.

\*/

public class Payment {

private int amountMoney;

//Вычисление суммы денег к оплате за список товаров

public void amountMoneyPay(Map<Product, Integer> shoppingList) {

public int amountMoneyPay(Map<Product, Integer> shoppingList) {

int sum = 0;

for (Map.Entry<Product, Integer> pair : shoppingList.entrySet()) {

sum += pair.getKey().getPrice() \* pair.getValue();

}

this.amountMoney = sum;

return sum;

}

//вывод в консоль списка покупок и суммы к оплате

@@ -29,22 +28,19 @@ public void printShoppingList(Map<Product, Integer> shoppingList) {

for (Map.Entry<Product, Integer> pair : shoppingList.entrySet()) {

System.out.println(pair.getKey().toString() + " - " + pair.getValue() + " шт.");

}

System.out.println("Сумма к оплате: " + amountMoney + " у.е.");

System.out.println("Сумма к оплате: " + amountMoneyPay(shoppingList) + " у.е.");

}

public class Purchase {

private Map<Product, Integer> shoppingList;

public Map<Product, Integer> getShoppingList() {

return shoppingList;

}

public class Order {

private Map<Product, Integer> orderList;

public void setShoppingList(Map<Product, Integer> shoppingList) {

this.shoppingList = shoppingList;

public Map<Product, Integer> getOrderList() {

return orderList;

}

//Создание списка покупок

public void createShoppingList(BufferedReader reader) throws IOException {

public void createShoppingList() throws IOException {

BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));

Map shoppingList = new HashMap<Product, Integer>();

boolean search = true;

while (search) {

@@ -61,11 +57,11 @@ public void createShoppingList(BufferedReader reader) throws IOException {

search = false;

break;

} else {

System.out.println("Давайте попробуем еще раз.");

System.out.println("Попробуйте еще раз.");

}

}

}

this.shoppingList = shoppingList;

this.orderList = shoppingList;

}

}

}

**src/Task\_02\_Payment/DataUtilProduct.java**

import java.io.IOException;

import java.util.\*;

public class DatabaseProduct {

public class DataUtilProduct {

public static List<Product> productList = new ArrayList<Product>() {

{

this.add(new Product("Принтер", "Canon", 200));

this.add(new Product("Принтер", "HP", 300));

this.add(new Product("Принтер", "Epson", 280));

this.add(new Product("Ноутбук", "Lenovo", 1000));

this.add(new Product("Ноутбук", "Apple", 2500));

this.add(new Product("Ноутбук", "Vaio", 1200));

this.add(new Product("Планшет", "Samsung", 1300));

this.add(new Product("Планшет", "Apple", 1900));

this.add(new Product("Планшет", "Huawei", 1000));

this.add(new Product("Телевизор", "Samsung", 1300));

this.add(new Product("Телевизор", "LG", 1000));

this.add(new Product("Телевизор", "Горизонт", 300));

this.add(new Product("Пылесос", "Samsung", 350));

this.add(new Product("Пылесос", "Bosh", 500));

this.add(new Product("Пылесос", "Витязь", 150));

this.add(new Product("принтер", "canon", 200));

this.add(new Product("принтер", "hp", 300));

this.add(new Product("принтер", "epson", 280));

this.add(new Product("ноутбук", "lenovo", 1000));

this.add(new Product("ноутбук", "apple", 2500));

this.add(new Product("ноутбук", "vaio", 1200));

this.add(new Product("планшет", "samsung", 1300));

this.add(new Product("планшет", "apple", 1900));

this.add(new Product("планшет", "huawei", 1000));

this.add(new Product("телевизор", "samsung", 1300));

this.add(new Product("телевизор", "lg", 1000));

this.add(new Product("телевизор", "горизонт", 300));

this.add(new Product("пылесос", "samsung", 350));

this.add(new Product("пылесос", "bosh", 500));

this.add(new Product("пылесос", "витязь", 150));

}

};

@@ -42,7 +42,7 @@ private static String readNameProduct(BufferedReader reader) throws IOException

String name;

while (true) {

name = reader.readLine();

if (createNameSet().contains(name)) {

if (createNameSet().contains(name.toLowerCase())) {

break;

} else {

System.out.print("Ничего не найдено, попробуйте еще раз: ");

@@ -70,7 +70,7 @@ private static String readManufacturerProduct(Set<String> manufacturerSet, Buffe

String manufacturer;

while (true) {

manufacturer = reader.readLine();

if (manufacturerSet.contains(manufacturer)) {

if (manufacturerSet.contains(manufacturer.toLowerCase())) {

break;

} else {

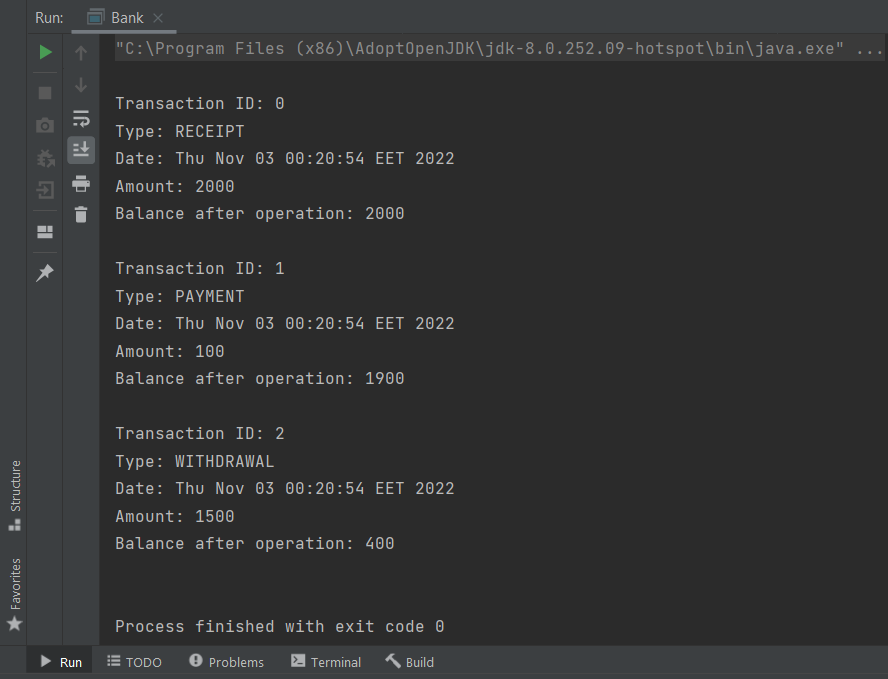


Рисунок 5 – Результат выполнения

1.5 Задача 5

Условие:

В тексте каждую букву заменить ее порядковым номером в алфавите. При выводе в одной строке печатать текст с двумя пробелами между буквами,

Код программы:

import java.util.\*;

import java.util.Map.Entry;

public class Main {

public static void main(String... args){

char i = 0;

int count = 0;

HashMap<Integer, Character> words = new LinkedHashMap<>();

for(i = '\u0430'; i < '\u044f'; i++){

count++;

words.put(count, i);

}

char[] line = "культурный презентационный материал предпосылки и развитие".toCharArray();

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

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

}

System.out.println("");

for(int k = 0; k < line.length; k++){

for(Entry<Integer, Character> abc: words.entrySet()){

int key = abc.getKey();

char value = abc.getValue();

if(line[k] == value){

System.out.print(key + " ");

}

}

}

}

}

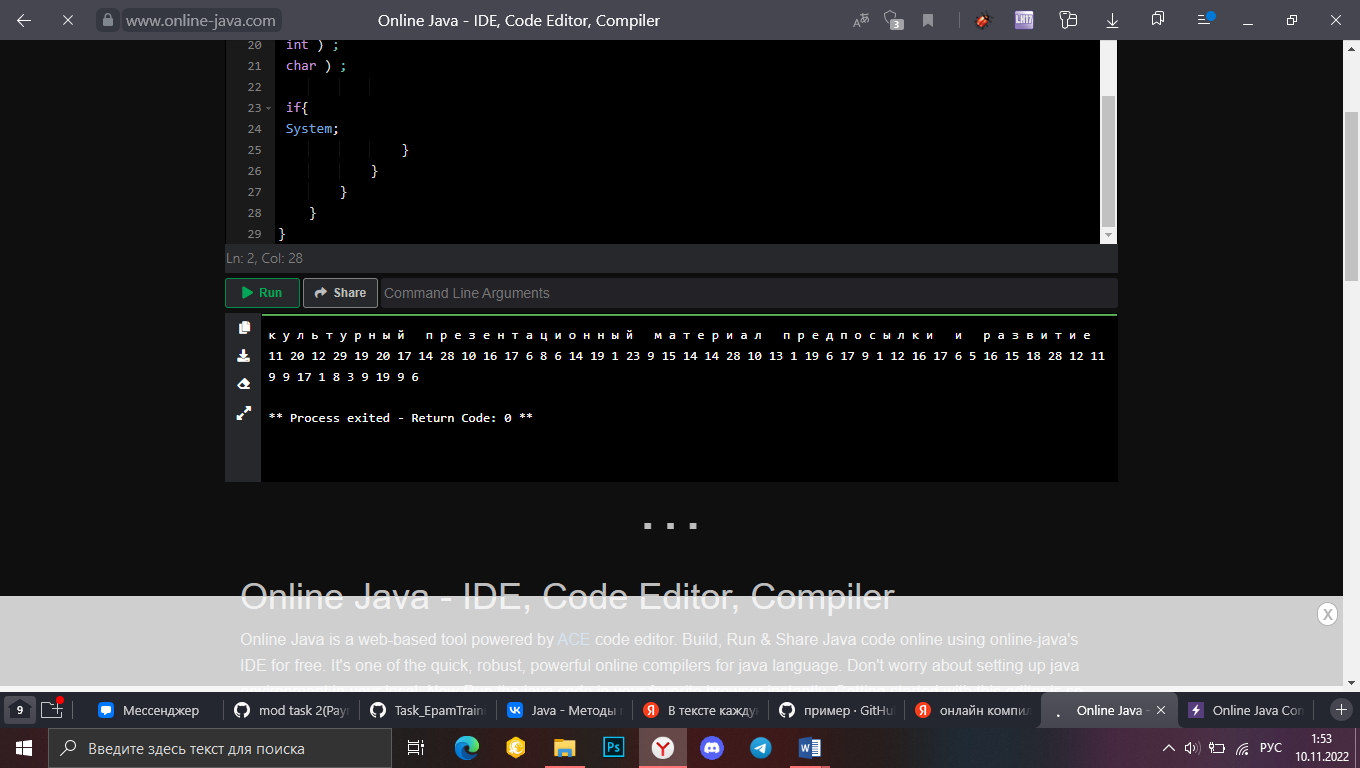


Рисунок 6 – Результат выполнения

1.6 Задача 6

Условие:

Вариант B

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

Код программы:

package com.company;

import java.io.\*;

import java.util.\*;

public class Matrix {

    private double element[][];//Двумерный массив, чтобы хранить элементы матрицы

    private double q; //Чтобы хранить знак матрицы

    private int degree;//Хранит n(размер)

    public Matrix() {

        q=1;

        degree=1;

        element=new double[degree-1][degree-1];

        element[0][0]=0;

    }

    public Matrix(int n){

        q=1;

        degree=n;

        element=new double[degree][degree];

    }

    public Matrix(String name)throws IOException{

        q=1;

        Scanner in=new Scanner(new FileReader(name));

        degree=in.nextInt();

        element=new double[degree][degree];

        for(int i=0;i<degree;++i){

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

                element[i][j]=in.nextDouble();

            }

        }

        in.close();

    }

    public Matrix(Matrix B){

        q=B.q;

        degree=B.degree;

        element=new double [degree][degree];

        for(int i=0;i<degree;++i){

            System.arraycopy(B.element[i], 0, element[i], 0, degree);

        }

    }

    int getDegree(){

        return degree;

    } //Получить n(размер матрицы)

    public void show(){

        for(int i=0;i<degree;++i){

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

                System.out.format("%6.2f",element[i][j]);//Чтобы вывести на экран

            }

            System.out.println();

        }

    }

    Matrix Transpose(){

        Matrix result=new Matrix(this);

        for(int i=0;i<degree;++i){

            for(int j=i+1;j<degree;++j){

                double buf=result.element[i][j];

                result.element[i][j]=result.element[j][i];

                result.element[j][i]=buf;

            }

        }

        return result;

    }

    double getFirstNorm(){

        double nextS = 0;

        double max = -Double.MIN\_VALUE;

        for (int i = 0; i <this.degree ; i++) {

            nextS = 0;

            for (int j = 0; j <this.degree ; j++) {

                nextS += this.element[i][j];

            }

            if (max<nextS) max = nextS;

        }

        return max;

    }

    @Override

    public String toString(){

        Formatter f=new Formatter();

        for(int i=0;i<degree;++i){

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

                f.format("%6.2f", element[i][j]);

            }

            f.format("\n");

        }

        return f.toString();

    }

    public void set(String name)throws IOException{

        q=1;

        degree=0;

        Scanner in=new Scanner(new FileReader(name));

        degree=in.nextInt();

        element=new double[degree][degree];

        for(int i=0;i<degree;++i){

            element[i]=new double [degree];

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

                element[i][j]=in.nextDouble();

            }

        }

        in.close();

    }

    boolean Equals(Matrix B){

        if(degree!=B.degree)

            return false;

        for(int i=0;i<degree;++i){

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

                if(element[i][j]!=B.element[i][j])

                    return false;

            }

        }

        return true;

    }

    Matrix Multiply(Matrix B){          //Умножение

        Matrix result=new Matrix(degree);

        double s=0;

        for(int i=0;i<degree;++i){

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

                s=0;

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

                    s+=element[i][k]\*B.element[k][j];

                }

                result.element[i][j]=s;

            }

        }

        return result;

    }

    Matrix Multiply(double n){       //умножение на число

        Matrix result=new Matrix(degree);

        for(int i=0;i<degree;++i){

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

                result.element[i][j]=element[i][j]\*n;

            }

        }

        return result;

    }

    Matrix Plus(Matrix B){                  //сложение

        Matrix result=new Matrix(degree);

        for(int i=0;i<degree;++i){

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

                result.element[i][j]=element[i][j]+B.element[i][j];

            }

        }

        return result;

    }

    Matrix Minus(Matrix B){                  //вычитание

        Matrix result=new Matrix(degree);

        for(int i=0;i<degree;++i){

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

                result.element[i][j]=element[i][j]-B.element[i][j];

            }

        }

        return result;

    }

}

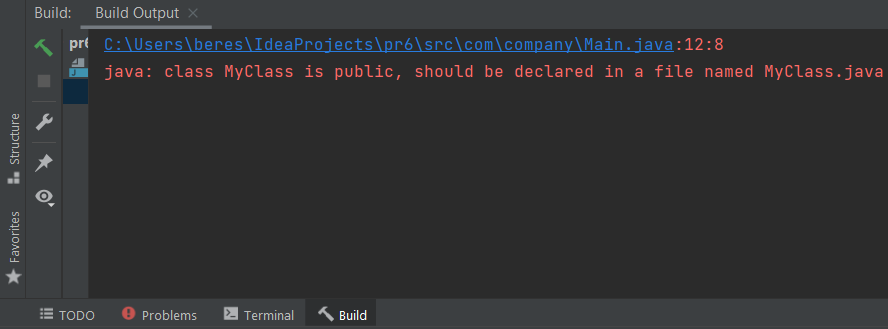
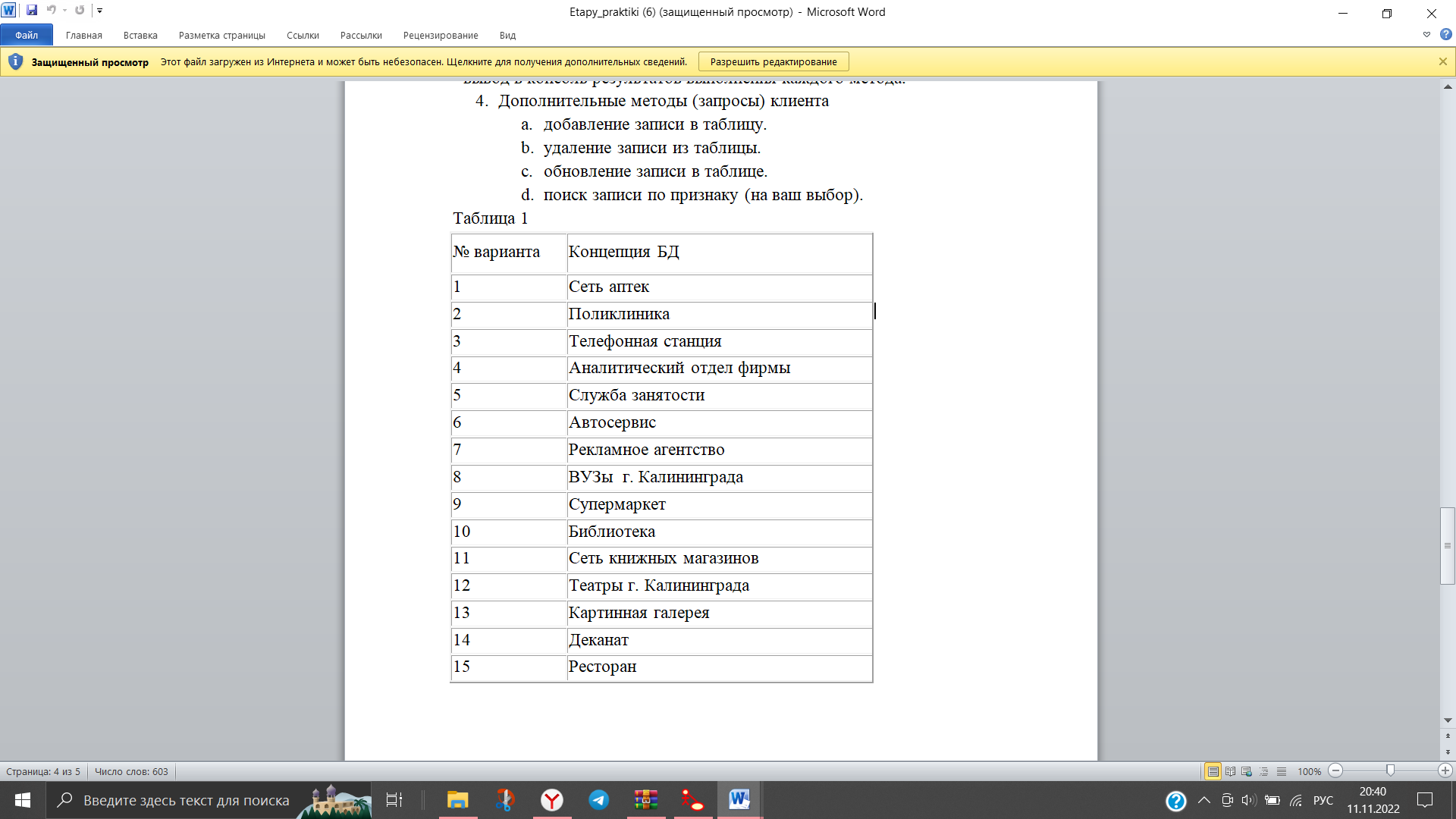


Рисунок 7 – Результат выполнения

1. Разработка АИС

Условие:



    В качестве серверной части разработать базу данных в СУБД MySQL в соответствии с вариантом. База данных содержит таблицу, состоящую из не менее чем 5-ти полей и 20-ти записей.

    Клиентская часть должна быть разработана на языке Java. Каждый запрос должен быть реализован отдельным методом. Все методы могут принадлежать одному классу. Отдельно должен быть реализован класс, содержащий метод main().

    Таким образом, структура клиента должна быть такой:

1. Класс, содержащий статический метод main() для демонстрации возможностей программы.
2. Класс, содержащий поля и методы для работы с удалённой базой данных:
3. Обязательные методы (запросы) клиента:

- выборка содержимого таблиц базы данных;

- запись в файл результата выборки содержимого таблиц БД.

- вывод в консоль результатов выполнения каждого метода.

1. Дополнительные методы (запросы) клиента
   1. добавление записи в таблицу.
   2. удаление записи из таблицы.
   3. обновление записи в таблице.
   4. поиск записи по признаку (на ваш выбор).

Техническое задание.

1) Анализ предметной области. Разработка ER-диаграммы «сущность-связь» (ErWin). Разработка диаграммы вариантов использования, диаграммы классов, диаграммы последовательностей, диаграммы кооперации, диаграммы состояний, диаграммы деятельности (MS Visio, Rational Rose и другие).

2) Разработать базу данных в СУБД MySQL «Поликлиника».

Клиентская часть должна быть разработана на языке Java. Каждый запрос должен быть реализован отдельным методом. Все методы могут принадлежать одному классу. Отдельно должен быть реализован класс, содержащий метод main().

3) Разработка структурной и функциональной схемы АИС.

4) Проектирование и реализация интерфейса.

5) Разработка встроенной системы справочного руководства и программного документа «Руководство пользователя».

6) Проведение тестирования.

ER-диаграмма:

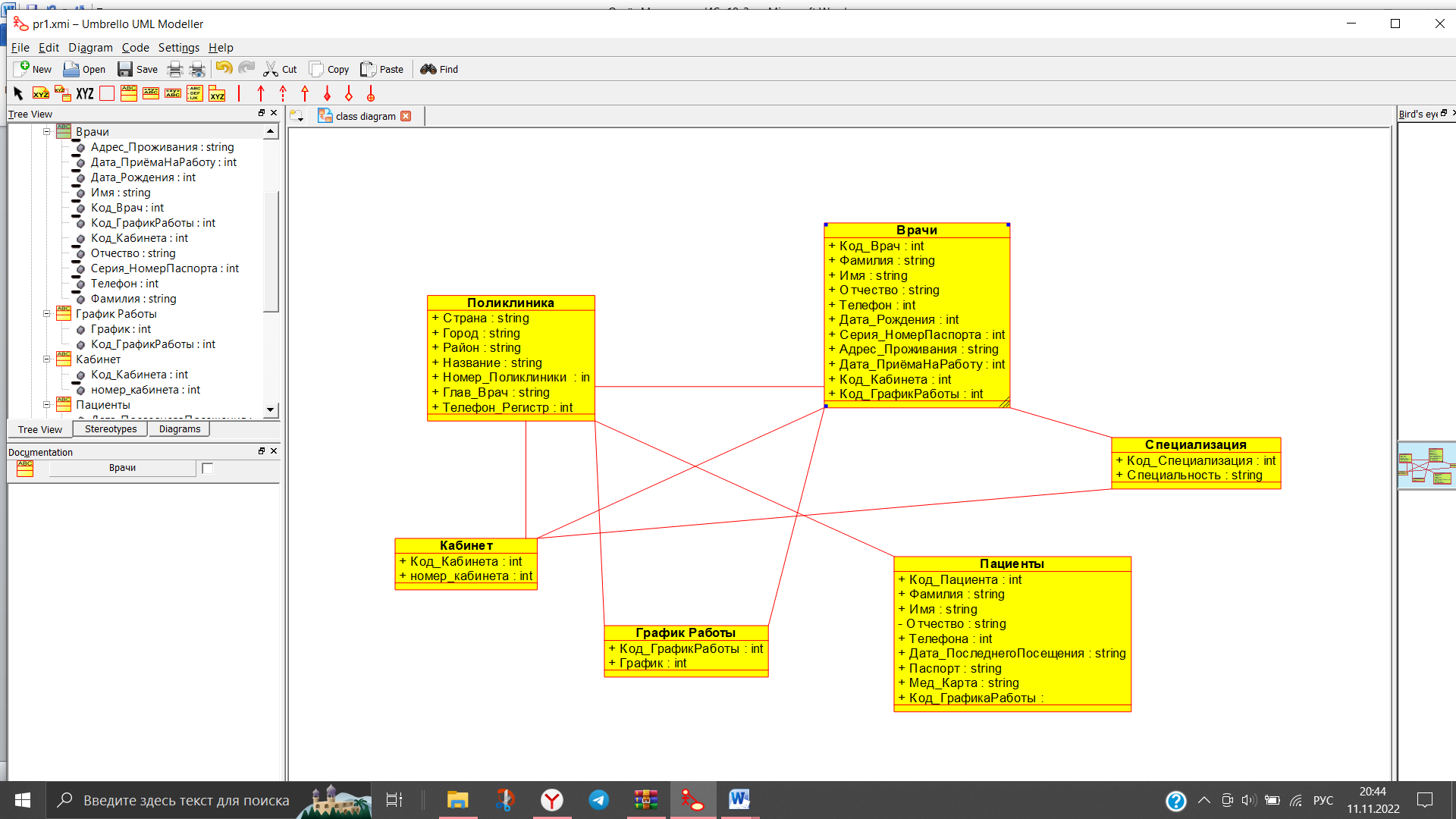


Рисунок 8 - ER-диаграмма

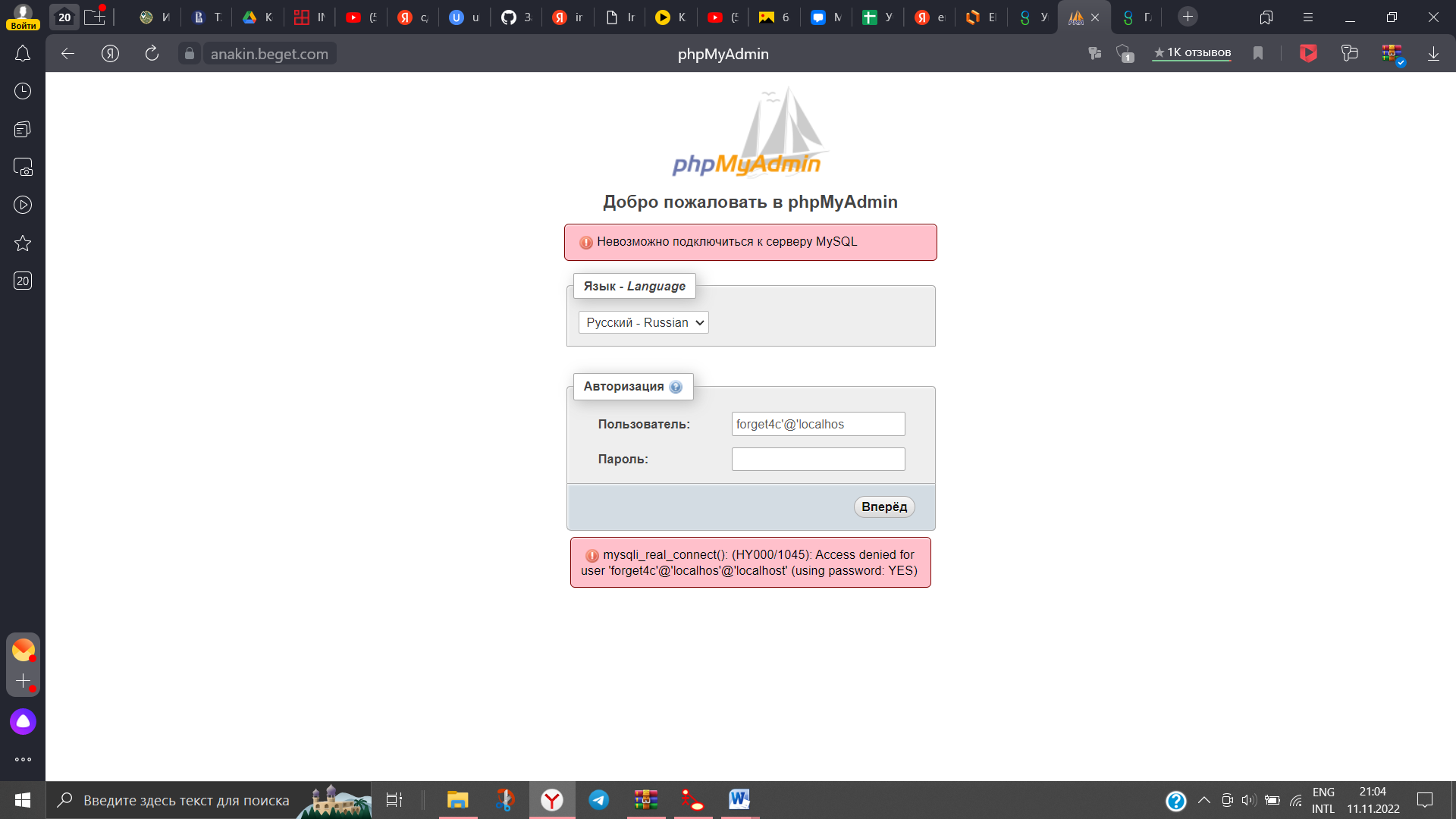


Рисунок 9 – Неудачная привязка к php

3. Создание сайта

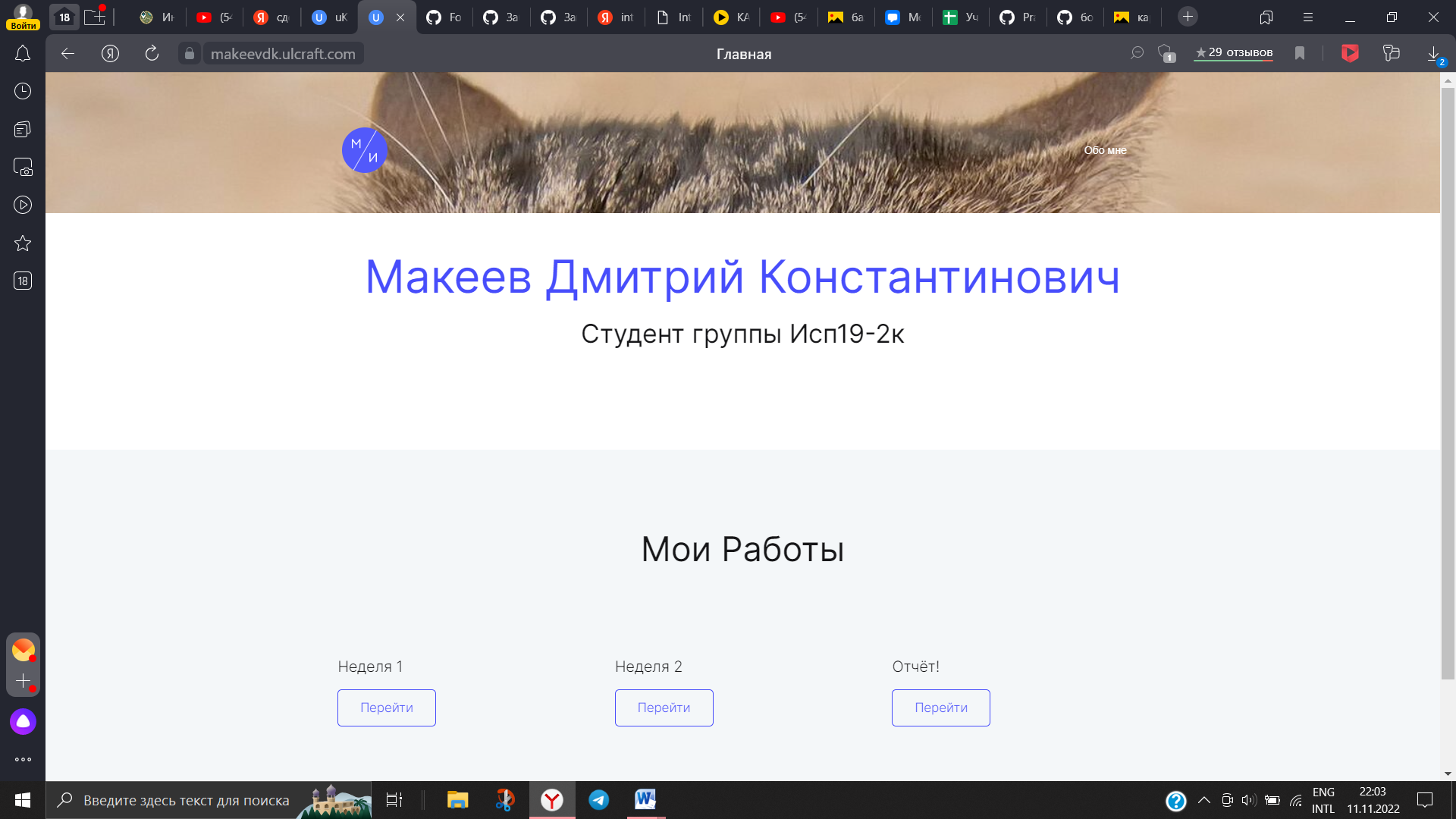


Рисунок 10 – сайт

Заключение

В ходе выполнения практической работы я научился работать с массивами. Я получил новые знания и бесценный опыт. Однако у меня получились не все задания.