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 ИС 19-2к  Заиончевский Максим Александрович  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |
| --- | --- |
| Проверила: | Большакова-Стрекалова Анна Викторовна  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (оценка)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) |

Калининград, 2022

Введение

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.

Основная часть

1.1 Задача 1

Условие:

Вычислить норму матрицы.

Результат:

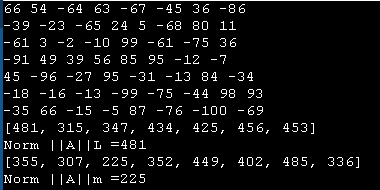


Рисунок 1 - Вывод задачи 1

1.2 Задача 2

Условие:

Train: Пункт назначения, Номер поезда, Время отправления, Число мест

(общих, купе, плацкарт, люкс).

Создать массив объектов. Вывести:

а) список поездов, следующих до заданного пункта назначения;

b) список поездов, следующих до заданного пункта назначения и отправляющихся после заданного часа;

c) список поездов, отправляющихся до заданного пункта назначения и имеющих общие места.

Решение:

public class Train {

private String destination;

private String number;

private String timeOfDeparture;

private EnumMap<CarType, Integer> seatCountHolder;

public enum CarType {

COMMON, COMPARTMENT, RESERVED\_SEAT, LUXURY;

}

public Train(String destination, String number, String timeOfDeparture,

final int commonSeatsCount, final int compartmentSeatsCount,

final int reservedSeatCount, final int luxurySeatCount ) {

this.destination = destination;

this.number = number;

this.timeOfDeparture = timeOfDeparture;

this.seatCountHolder = new EnumMap<CarType, Integer>(CarType.class);

seatCountHolder.put(CarType.COMMON, commonSeatsCount);

seatCountHolder.put(CarType.COMPARTMENT, compartmentSeatsCount);

seatCountHolder.put(CarType.RESERVED\_SEAT, reservedSeatCount);

seatCountHolder.put(CarType.LUXURY, luxurySeatCount);

}

public String getDestination() {

return destination;

}

public void setDestination(String destination) {

this.destination = destination;

}

public String getNumber() {

return number;

}

public void setNumber(String number) {

this.number = number;

}

public String getTimeOfDeparture() {

return timeOfDeparture;

}

public void setTimeOfDeparture(String timeOfDeparture) {

this.timeOfDeparture = timeOfDeparture;

}

public int getSeatCountByCarType(final CarType type) {

return seatCountHolder.get(type);

}

public void setSeatCountByCarType(final CarType type, final int newCount) {

seatCountHolder.put(type, newCount);

}

public int getTotalSeatCount() {

int count = 0;

for (CarType type : CarType.values())

count = count + seatCountHolder.get(type);

return count;

}

@Override

public String toString() {

return "Train :\n" +

" destination : '" + destination + "\'\n" +

" number : '" + number + "\'\n" +

" time of departure : '" + timeOfDeparture + "\'\n" +

" seats : " + getTotalSeatCount() + "\n" +

" common seats : " + seatCountHolder.get(CarType.COMMON) + "\n" +

" compartment seats : " + seatCountHolder.get(CarType.COMPARTMENT) + "\n" +

" reserved seats : " + seatCountHolder.get(CarType.RESERVED\_SEAT) + "\n" +

" luxury seats : " + seatCountHolder.get(CarType.LUXURY);

}

}

public class Trains {

private static final String[] CITIES = {"Moscow", "St.Petersburg", "Ekaterinburg", "Paris", "London"};

private static final ThreadLocalRandom RNG = ThreadLocalRandom.current();

public static List<Train> findTrainsWithSameDestination(final Collection<Train> trains, final String destination) {

List<Train> result = new ArrayList<>(trains);

return result.stream()

.filter(train -> Objects.equals(train.getDestination(), destination))

.collect(Collectors.toList());

}

public static List<Train> findTrainsWithSameDestinationAndAfterTime(final Collection<Train> trains,

final String destination,

final String givenTime) {

List<Train> result = new ArrayList<>(trains);

return result.stream()

.filter(train -> Objects.equals(train.getDestination(), destination))

.filter(train -> DatatypeConverter.parseTime(train.getTimeOfDeparture()).after(

DatatypeConverter.parseTime(givenTime)))

.collect(Collectors.toList());

}

public static List<Train> findTrainsWithSameDestinationAndHaveCommonSeats(final Collection<Train> trains,

final String destination) {

List<Train> result = new ArrayList<>(trains);

return result.stream()

.filter(train -> Objects.equals(train.getDestination(), destination))

.filter(train -> train.getSeatCountByCarType(Train.CarType.COMMON) > 0)

.collect(Collectors.toList());

}

public static List<Train> newRandomTrainList(final int trainCount) {

List<Train> trains = new ArrayList<>(trainCount);

for (int i = 0; i < trainCount; i++)

trains.add(newRandomTrain());

return trains;

}

public static Train newRandomTrain() {

final String city = randomCity();

final String number = randomNumber();

final String time = randomTime();

Train train = new Train(city, number, time, 0, 0, 0, 0);

for (Train.CarType type : Train.CarType.values())

train.setSeatCountByCarType(type, RNG.nextInt(21));

return train;

}

private static String randomCity() {

return CITIES[RNG.nextInt(CITIES.length)];

}

private static String randomTime() {

return String.format("%02d:%02d:%02d", RNG.nextInt(13), RNG.nextInt(60), RNG.nextInt(60));

}

private static String randomNumber() {

return String.format("%03d", RNG.nextInt(1000));

}

}

public class Main {

public static void main(String[] args) {

List<Train> trainsList = Trains.newRandomTrainList(10);

System.out.println("Original list:");

trainsList.forEach(System.out::println);

List<Train> trainsWithGivenDestination = Trains.findTrainsWithSameDestination(trainsList, "Moscow");

System.out.println("Trains for Moscow:");

trainsWithGivenDestination.forEach(System.out::println);

List<Train> trainsWithGivenDestinationAndTime = Trains

.findTrainsWithSameDestinationAndAfterTime(trainsList, "Paris", "09:00:00");

System.out.println("Trains for Paris and 09:00:");

trainsWithGivenDestinationAndTime.forEach(System.out::println);

List<Train> trainsWithGivenDestinationAndSeats = Trains

.findTrainsWithSameDestinationAndHaveCommonSeats(trainsList, "Ekaterinburg");

System.out.println("Trains for Ekaterinburg and seats:");

trainsWithGivenDestinationAndSeats.forEach(System.out::println);

}

}

1.3 Задача 3

Условие:

Создать объект класса Щенок, используя классы Животное, Собака.

Методы: вывести на консоль имя, подать голос, прыгать, бегать, кусать.

Решение:

public class Animal {

protected String name;

public Animal(String name) {

this.name = name;

}

public String getName() {

return name;

}

}

public class Dog extends Animal {

public Dog(String name) {

super(name);

}

public String votesCast(){

return "Gav";

}

public String jump() {

return "Jump";

}

public String run() {

return "Run";

}

public String bite() {

return "Bite";

}

}

public class Puppy extends Dog {

public Puppy(String name) {

super(name);

}

}

public class PuppyRunner {

public static void main(String[] args) {

Puppy puppy = new Puppy("Ralf");

System.out.println("Name: "+puppy.getName());

System.out.println("Votes Cast: " + puppy.votesCast());

System.out.println("Jump: " + puppy.jump());

System.out.println("Run: "+ puppy.run());

System.out.println("Bite: "+ puppy.bite());

}

}

1.4 Задание 4

Условие:

Создать класс Mobile с внутренним классом, с помощью объектов которого можно хранить информацию о моделях телефонов и их свойствах.

Решение:

public class Mobile {

private String brand;

private String model;

private ModelProperties modelProperties;

// constructors, getters, setters, toString, etc...

private class ModelProperties {

private String model;

private long accumCapacity;

private double screenSize;

// constructors, getters, setters, toString, etc...

}

}

1.5 Задание 5

Условие:

В тексте найти и напечатать n символов (и их количество), встречающихся наиболее часто.

Решение:

public class FindSymbolsInTheText {

public static String calculateWordsEnding(String string) {

int count1 = 0; // а

int count2 = 0; // е

int count3 = 0; // в

int count4 = 0; // и

int count5 = 0; // к

int count6 = 0; // л

int count7 = 0; // м

int count8 = 0; // о

string = string.toLowerCase();

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

if (string.charAt(i) == 'а')

count1++;

if (string.charAt(i) == 'е')

count2++;

if (string.charAt(i) == 'в')

count3++;

if (string.charAt(i) == 'и')

count4++;

if (string.charAt(i) == 'к')

count5++;

if (string.charAt(i) == 'л')

count6++;

if (string.charAt(i) == 'м')

count7++;

if (string.charAt(i) == 'о')

count8++;

}

System.out.println("а = " + count1 + "\n" + "е = " + count2 + "\n" + "в = " + count3 + "\n" + "и = " + count4 + "\n" +

"к = " + count5 + "\n" + "л = " + count6 + "\n" + "м = " + count7 + "\n" + "о = " + count8 + "\n");

return string;

}

private static String readingFromFile(String path) throws IOException {

byte[] encoded = Files.readAllBytes(Paths.get(path));

return new String(encoded, "utf-8");

}

}

1.6 Задача 6

1.7 Задача 7

1.8 Задача 8

Условие:

Составить описание класса для работы с цепными списками строк (строками произвольной длины). Обеспечить при этом выполнение операций включения в список, удаления из списка элемента с заданным значением, удаления всего списка или конца списка, начиная с указанного элемента.

Решение:

static void Main(string[] args)

{

Console.Title = "Задача 2.9";

MyStringBuilder stringList = new MyStringBuilder();

while (true)

{

int menu;

Console.Clear();

Console.WriteLine("Меню:");

Console.WriteLine("1. Добавить строку.");

Console.WriteLine("2. Вывод строки на экран.");

Console.WriteLine("3. Удаление элемента из строки.");

Console.WriteLine("4. Удаление всей строки.");

Console.WriteLine("5. Удаление конца строки с заданного элемента.");

Console.WriteLine("6. Вывести весь список строк.");

Console.WriteLine("7. Удалить весь список.");

Console.WriteLine("8. Выход.");

Console.Write("\nВыберите пункт меню: ");

try

{

menu = Convert.ToInt32(Console.ReadLine());

switch (menu)

{

case 1:

String str;

Console.WriteLine("Введите строку: ");

str = Console.ReadLine();

stringList.Add(str);

break;

case 2:

Console.Write("\nВведите номер строки: ");

int pos = Convert.ToInt32(Console.ReadLine());

stringList.ShowString(pos);

Console.WriteLine("\n\nPress any key...");

Console.ReadKey();

break;

case 3:

Console.Write("Введите номер строки из которой хотите удалить символ: ");

int posLine = Convert.ToInt32(Console.ReadLine());

Console.Write("Введите номер элемента, который необходимо удалить: ");

int posChar = Convert.ToInt32(Console.ReadLine());

stringList.DeleteChar(posLine, posChar);

break;

case 4:

Console.Write("Введите номер строки которую хотите удалить: ");

posLine = Convert.ToInt32(Console.ReadLine());

stringList.DeleteString(posLine);

break;

case 5:

Console.Write("Введите номер строки конец которой вы хотите удалить: ");

posLine = Convert.ToInt32(Console.ReadLine());

Console.Write("Введите номер элемента, от начала которого вы хотите удалить символы: ");

posChar = Convert.ToInt32(Console.ReadLine());

stringList.DeleteEndOfString(posLine, posChar);

break;

case 6:

stringList.ShowAll();

Console.WriteLine("\n\nPress any key...");

Console.ReadKey();

break;

case 7:

stringList.DeleteAll();

break;

case 8:

return;

}

}

catch (FormatException)

{

Console.Clear();

Console.WriteLine("Неверный пункт. Нажмите любую клавишу и повторите ввод.");

Console.ReadKey();

}

}

}

⦁ Разработка ER-диаграммы

Условие:

Рекламное агентство

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:

⦁ Класс, содержащий статический метод main() для демонстрации возможностей программы.

⦁ Класс, содержащий поля и методы для работы с удалённой базой данных:

⦁ Обязательные методы (запросы) клиента:

- выборка содержимого таблиц базы данных;

- запись в файл результата выборки содержимого таблиц БД.

- вывод в консоль результатов выполнения каждого метода.

⦁ Дополнительные методы (запросы) клиента

⦁ добавление записи в таблицу.

⦁ удаление записи из таблицы.

⦁ обновление записи в таблице.

⦁ 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 «Театры г. Калининграда».

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.

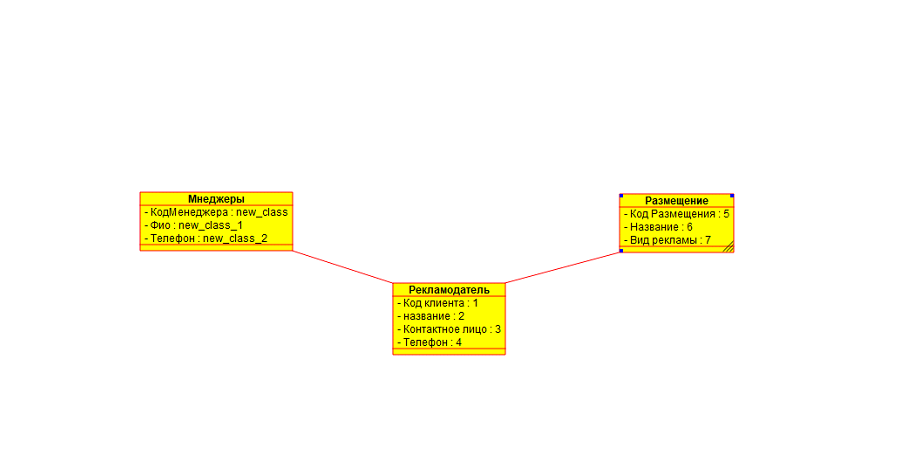


Рисунок – 1(ER Диаграма)

Выгрузка базы данных:

-- phpMyAdmin SQL Dump

-- version 4.9.7

-- https://www.phpmyadmin.net/

--

-- Хост: localhost

-- Время создания: Ноя 12 2022 г., 23:41

-- Версия сервера: 5.7.21-20-beget-5.7.21-20-1-log

-- Версия PHP: 5.6.40

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

SET AUTOCOMMIT = 0;

START TRANSACTION;

SET time\_zone = "+00:00";

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8mb4 \*/;

--

-- База данных: `deputasy\_practik`

--

-- --------------------------------------------------------

--

-- Структура таблицы `Manager`

--

-- Создание: Ноя 12 2022 г., 20:33

-- Последнее обновление: Ноя 12 2022 г., 20:36

--

DROP TABLE IF EXISTS `Manager`;

CREATE TABLE `Manager` (

`IDManager` int(11) NOT NULL,

`FIO` text NOT NULL,

`TLF` text NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Дамп данных таблицы `Manager`

--

INSERT INTO `Manager` (`IDManager`, `FIO`, `TLF`) VALUES

(1, 'Артур Магомедов', '+79998789889'),

(2, 'Галина Паловна', '+79994567887'),

(3, 'Петр Нормандов', '+799845387');

-- --------------------------------------------------------

--

-- Структура таблицы `Placed`

--

-- Создание: Ноя 12 2022 г., 20:28

-- Последнее обновление: Ноя 12 2022 г., 20:37

--

DROP TABLE IF EXISTS `Placed`;

CREATE TABLE `Placed` (

`IDPlaced` int(11) NOT NULL,

`Names` text NOT NULL,

`VidReklama` text NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Дамп данных таблицы `Placed`

--

INSERT INTO `Placed` (`IDPlaced`, `Names`, `VidReklama`) VALUES

(1, 'Площадь победы', 'Банер'),

(2, 'Улица Горького 34', 'Объявления на столбах'),

(3, 'Советский проспект 132', 'Банер');

-- --------------------------------------------------------

--

-- Структура таблицы `Reklamodatel`

--

-- Создание: Ноя 12 2022 г., 20:38

-- Последнее обновление: Ноя 12 2022 г., 20:40

--

DROP TABLE IF EXISTS `Reklamodatel`;

CREATE TABLE `Reklamodatel` (

`IDKlietn` int(11) NOT NULL,

`Names` text NOT NULL,

`KontaktName` text NOT NULL,

`TLF` text NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Дамп данных таблицы `Reklamodatel`

--

INSERT INTO `Reklamodatel` (`IDKlietn`, `Names`, `KontaktName`, `TLF`) VALUES

(1, 'Олег Палыч', 'Юлия Секретарь', '+79985432321'),

(2, 'Валентина Олеговна ', '-', '+79975435645'),

(3, 'Артур Шарифов', 'Лаура Василькова', '+79763568754');

--

-- Индексы сохранённых таблиц

--

--

-- Индексы таблицы `Manager`

--

ALTER TABLE `Manager`

ADD PRIMARY KEY (`IDManager`);

--

-- Индексы таблицы `Placed`

--

ALTER TABLE `Placed`

ADD PRIMARY KEY (`IDPlaced`);

--

-- Индексы таблицы `Reklamodatel`

--

ALTER TABLE `Reklamodatel`

ADD PRIMARY KEY (`IDKlietn`);

--

-- AUTO\_INCREMENT для сохранённых таблиц

--

--

-- AUTO\_INCREMENT для таблицы `Manager`

--

ALTER TABLE `Manager`

MODIFY `IDManager` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=4;

--

-- AUTO\_INCREMENT для таблицы `Placed`

--

ALTER TABLE `Placed`

MODIFY `IDPlaced` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=4;

--

-- AUTO\_INCREMENT для таблицы `Reklamodatel`

--

ALTER TABLE `Reklamodatel`

MODIFY `IDKlietn` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=4;

COMMIT;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;