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  Кашин Дмитрий Игоревич  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |
| Проверила: | Большакова-Стрекалова Анна Викторовна  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (оценка)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) |

Калининград, 2022

Содержание

Введение

1. Основная часть
   1. Задача 1
   2. Задача 2
   3. Задача 3
   4. Задача 4
   5. Задача 5
   6. Задача 6
   7. Задача 7
   8. Задача 8
   9. Задача 9
   10. Задача 10

2. Разработка АИС

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. Задача 1

Условие:

8. Вычислить определитель матрицы.

public class Main {  
 public static void main(String[] args) {  
 double [][] mat4 = {  
 {1.0,1.0,-1.0,-1.0},  
 {2.0,1.0,2.0,3.0},  
 {3.0,4.0,5.0,6.0},  
 {5.0,7.0,8.0,9.0}  
 };  
 double [][] mat3 = {  
 {1.0,2.0,3.0},  
 {4.0,5.0,6.0},  
 {7.0,8.0,9.0}  
 };  
 double [][] mat2 = {  
 {1,2},  
 {3,4},  
 };  
 Determin dd = new Determin(mat4);  
 dd.getValue();  
 Determin aa = new Determin(mat2);  
 aa.getValue();  
 Determin bb = new Determin(mat3);  
 bb.getValue();  
 }  
}

//

class Determin {  
 private double summ;  
  
 public Determin(double[][] matrix) {  
 getReduction(matrix, 1);  
 }  
  
  
 public void getValue() {  
 System.out.printf("%.2f", this.summ);  
 System.out.println("");  
  
 }

Результат:

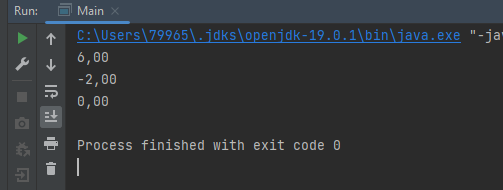


Рисунок – 1

* 1. Задача 2

Условие:

Airline: Пункт назначения, Номер рейса, Тип самолета, Время вылета, Дни недели. Создать массив объектов. Вывести:

1. список рейсов для заданного пункта назначения;
2. список рейсов для заданного дня недели;
3. список рейсов для заданного дня недели, время вылета для которых больше заданного.

package arline\_11;  
  
import java.text.ParseException;  
import java.util.ArrayList;  
import java.util.Date;  
import java.util.Iterator;  
  
public class Main {  
 public Main() {  
 }  
  
 public static void main(String[] args) throws ParseException {  
 FlightSchedule flightSchedule = new FlightSchedule("Minsk (MSQ)", new ArrayList());  
 flightSchedule.addAirline("Берлин (SXF)", "B2 891", "Embraer RJ-175", Airline.ft.parse("12:15"), new DayOfWeek[]{DayOfWeek.TUESDAY, DayOfWeek.WEDNESDAY, DayOfWeek.THURSDAY, DayOfWeek.FRIDAY, DayOfWeek.SUNDAY});  
 flightSchedule.addAirline("Москва (ZIA)", "B2 971", "Boeing 737-500", Airline.ft.parse("8:15"), new DayOfWeek[]{DayOfWeek.EVERYDAY});  
 flightSchedule.addAirline("Москва (ZIA)", "B2 973", "Boeing 737-300", Airline.ft.parse("18:05"), new DayOfWeek[]{DayOfWeek.EVERYDAY});  
 flightSchedule.addAirline("Москва (DME)", "B2 993", "Boeing 737-500", Airline.ft.parse("19:45"), new DayOfWeek[]{DayOfWeek.MONDAY, DayOfWeek.TUESDAY, DayOfWeek.WEDNESDAY, DayOfWeek.THURSDAY, DayOfWeek.FRIDAY, DayOfWeek.SATURDAY});  
 flightSchedule.addAirline("Москва (SVO)", "SU 1833", "Sukhoi Superjet 100-95", Airline.ft.parse("20:30"), new DayOfWeek[]{DayOfWeek.EVERYDAY});  
 flightSchedule.addAirline("Москва (VKO)", "UT 776", "Boeing 737-500", Airline.ft.parse("17:55"), new DayOfWeek[]{DayOfWeek.EVERYDAY});  
 flightSchedule.addAirline("Брюссель (CRL)", "B2 817", "Embraer 175", Airline.ft.parse("8:40"), new DayOfWeek[]{DayOfWeek.WEDNESDAY});  
 flightSchedule.addAirline("Брюссель (CRL)", "B2 817", "Embraer 175", Airline.ft.parse("9:35"), new DayOfWeek[]{DayOfWeek.THURSDAY, DayOfWeek.SATURDAY});  
 flightSchedule.addAirline("Брюссель (CRL)", "B2 817", "Embraer 175", Airline.ft.parse("13:00"), new DayOfWeek[]{DayOfWeek.MONDAY});  
 flightSchedule.addAirline("Париж (CDG)", "B2 863", "Embraer 195", Airline.ft.parse("14:50"), new DayOfWeek[]{DayOfWeek.MONDAY, DayOfWeek.WEDNESDAY, DayOfWeek.FRIDAY, DayOfWeek.SUNDAY});  
 System.out.println("Список рейсов с пунктом назначения Брюссель (CRL):");  
 ArrayList<Airline> airlinesToBrussels = flightSchedule.airlinesToDestination("Брюссель (CRL)");  
 Iterator var3 = airlinesToBrussels.iterator();  
  
 while(var3.hasNext()) {  
 Airline airline = (Airline)var3.next();  
 System.out.println(airline);  
 }  
  
 System.out.println("\nСписок рейсов, вылетающих в воскресенье:");  
 ArrayList<Airline> airlinesOnSunday = flightSchedule.flightDays(DayOfWeek.SUNDAY);  
 Iterator var9 = airlinesOnSunday.iterator();  
  
 while(var9.hasNext()) {  
 Airline airline = (Airline)var9.next();  
 System.out.println(airline);  
 }  
  
 System.out.println("\nСписок рейсов, вылетающих после 13:00 вторника:");  
 Date time = Airline.ft.parse("13:00");  
 ArrayList<Airline> airlinesOnTuesday = flightSchedule.flightDays(DayOfWeek.MONDAY, time);  
 Iterator var6 = airlinesOnTuesday.iterator();  
  
 while(var6.hasNext()) {  
 Airline airline = (Airline)var6.next();  
 System.out.println(airline);  
 }  
  
 }  
}

Результат:

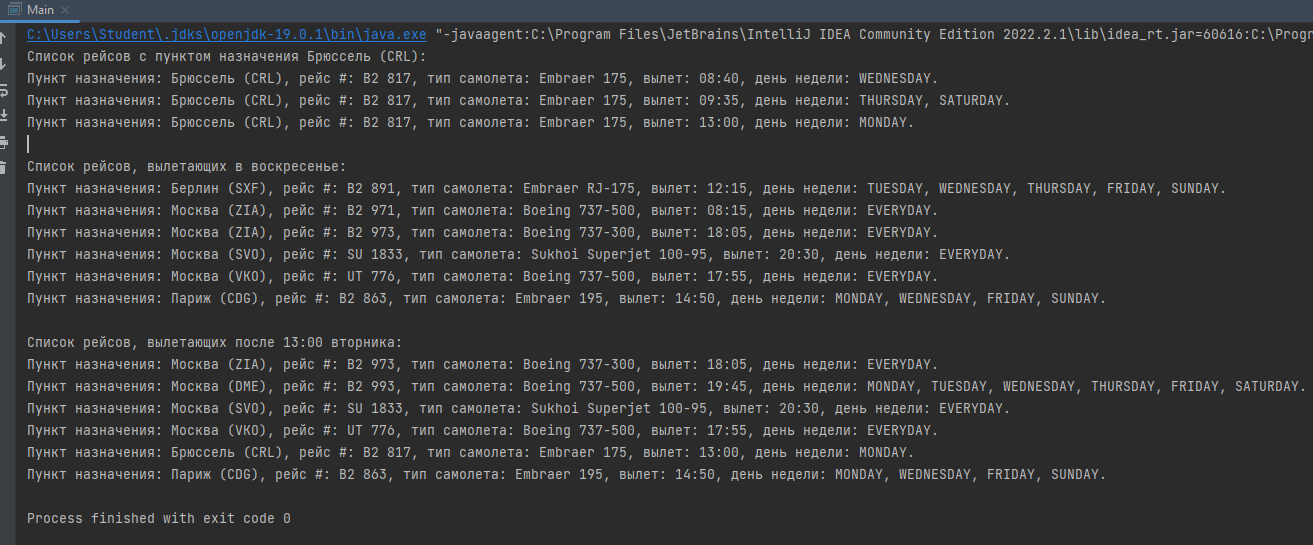


Рисунок – 2

* 1. Задача 3

Условие:

Создать объект класса Текстовый файл, используя классы Файл, Директория. Методы: создать, переименовать, вывести на консоль содержимое, дополнить, удалить.

import zadanie.utilit.entity.Directory;  
import zadanie.utilit.entity.File;  
import zadanie.utilit.entity.TextFile;  
import zadanie.utilit.utilits.DirectoryUtilit;  
  
  
  
public class Runner {  
  
 public static void main(String[] args) {  
  
 DirectoryUtilit directoryUtils = new DirectoryUtilit(new Directory("Books"));  
  
 File book1 = new TextFile("Играть");  
 File book2 = new TextFile("Изучение Java");  
 File book3 = new TextFile("Алгоритмы");  
 File book4 = new TextFile("Изучение MySql");  
 File book5 = new TextFile("Пока");  
  
 directoryUtils.addFile(book1);  
 directoryUtils.addFile(book2);  
 directoryUtils.addFile(book3);  
 directoryUtils.addFile(book4);  
 directoryUtils.addFile(book5);  
  
 directoryUtils.printFiles();  
  
 System.out.println(book1.getCreated());  
  
 book1.read();  
 System.out.println(book1.getOpened());  
  
 book1.printContent();  
  
 book1.addContent("Привет");  
 System.out.println(book1.getModified());  
  
 book1.getSize();  
  
 book1.printContent();  
  
  
  
 }  
}

Результат:

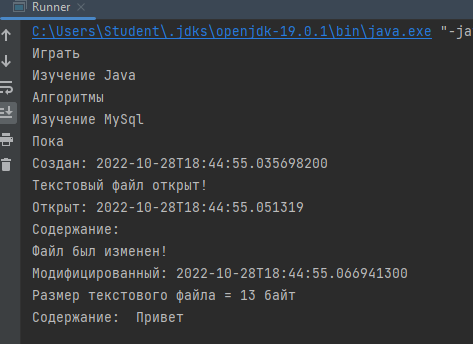


Рисунок – 3

1.4 Задача 4

Условие:

Создать класс Календарь с внутренним классом, с помощью объектов которого можно хранить информацию о выходных и праздничных днях.

package zadanie4.entity;  
  
import java.time.DayOfWeek;  
import java.time.LocalDate;  
import java.util.ArrayList;  
import java.util.List;  
  
public class Calendar {  
  
 private int year;  
 private List<Holiday> holidays;  
 private LocalDate date;  
  
 public Calendar(int year) {  
 setYear(year);  
 date = LocalDate.ofYearDay(getYear(), 1);  
 holidays = new ArrayList<>();  
 }  
  
 public void setYear(int year) {  
 if (year < 0) {  
 this.year = 2000;  
 } else {  
 this.year = year;  
 }  
 }  
  
 public int getYear() {  
 return year;  
 }  
  
 public void getAllHolidays() {  
 for (int i = 1; i < date.lengthOfYear(); i++) {  
 date = LocalDate.ofYearDay(year, i);  
 if (date.getDayOfWeek() == DayOfWeek.SATURDAY || date.getDayOfWeek() == DayOfWeek.SUNDAY) {  
 addHoliday(date.getYear(), date.getMonthValue(), date.getDayOfMonth());  
 }  
 }  
 }  
  
 private class Holiday {  
  
 private LocalDate date;  
  
 private Holiday(LocalDate date) {  
 this.date = date;  
 }  
  
 @Override  
 public String toString() {  
 return "Праздник{" + date.getDayOfMonth() + ", " + date.getMonth() + ", " + date.getYear() + '}';  
 }  
 }  
  
 public boolean addHoliday(int year, int month, int dayOfMonth) {  
 LocalDate date = LocalDate.of(year, month, dayOfMonth);  
 Holiday holiday = new Holiday(date);  
 return holidays.add(holiday);  
 }  
  
 public boolean removeHoliday(int year, int month, int dayOfMonth) {  
 LocalDate date = LocalDate.of(year, month, dayOfMonth);  
 Holiday holiday = new Holiday(date);  
 return holidays.remove(holiday);  
 }  
  
 public void printHolidays() {  
 if (holidays.isEmpty()) {  
 System.out.println("У него нет праздников!");  
 return;  
 }  
  
 for (Holiday holiday : holidays) {  
 System.out.println(holiday);  
 }  
 }  
}

package zadanie4;  
  
import zadanie4.entity.Calendar;  
  
  
public class CalendarRunner {  
  
 public static void main(String[] args) {  
  
 Calendar calendar = new Calendar(2019);  
  
 calendar.addHoliday(2022, 10, 1);  
 calendar.addHoliday(2021, 1, 12);  
  
 calendar.getAllHolidays();  
  
 calendar.printHolidays();  
 }  
}

Результат:

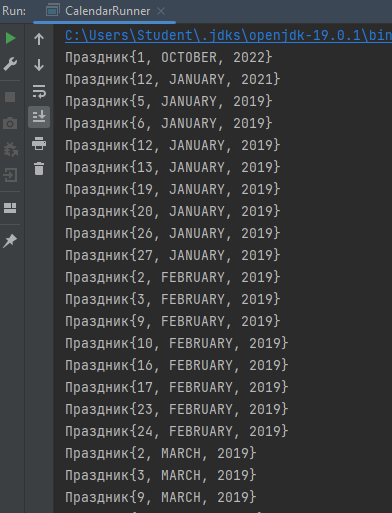


Рисунок – 4

1.5 Задача 5

Условие:

В стихотворении найти количество слов, начинающихся и заканчивающихся гласной буквой.

package com.epam.String;  
  
import java.io.\*;  
import java.util.regex.Matcher;  
import java.util.regex.Pattern;  
  
public class A12 {  
  
 public void A12(String path) {  
  
 int n=0;  
  
 try{  
  
 FileInputStream fs = new FileInputStream(path);  
 BufferedReader br = new BufferedReader(new InputStreamReader(fs));  
 String line;  
  
 while ((line = br.readLine()) != null){  
  
 String sbLowercase=line.toLowerCase();  
 String[] str = sbLowercase.split("[ —,;:.!?]");  
  
 Pattern pattern = Pattern.compile("^[уеёыаоэяию].\*[уеёыаоэяию]$");  
  
 for (String word : str) {  
 Matcher matcher = pattern.matcher(word);  
 if (matcher.find()) {  
 n++;  
 System.out.println(word);  
 }  
 }  
 }  
  
 System.out.println(n);  
  
 fs.close();  
 br.close();  
 }catch (IOException e){  
 System.out.println("Ошибка");  
 }  
  
  
  
 }  
}

package com.epam.String;  
  
public class Main {  
 public static void main(String[] args){  
  
 A12 a12 = new A12();  
 System.out.println("\n\nA12: ");  
 a12.A12("Stih.txt");  
  
 }  
  
}

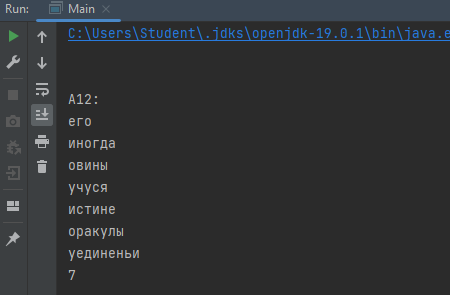
Результат:

Рисунок – 5

1.6 Задача 6

Условие:

Вариант A

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

Вариант B

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

package zadanie.utilit;  
  
import zadanie.utilit.entity.Directory;  
import zadanie.utilit.entity.File;  
import zadanie.utilit.entity.TextFile;  
import zadanie.utilit.utilits.DirectoryUtilit;  
  
  
  
public class Runner {  
  
 public static void main(String[] args) {  
  
 DirectoryUtilit directoryUtils = new DirectoryUtilit(new Directory("Books"));  
  
 File book1 = new TextFile("Играть");  
 File book2 = new TextFile("Изучение Java");  
 File book3 = new TextFile("Алгоритмы");  
 File book4 = new TextFile("Изучение MySql");  
 File book5 = new TextFile("Пока");  
  
 directoryUtils.addFile(book1);  
 directoryUtils.addFile(book2);  
 directoryUtils.addFile(book3);  
 directoryUtils.addFile(book4);  
 directoryUtils.addFile(book5);  
  
 directoryUtils.printFiles();  
  
 System.out.println(book1.getCreated());  
  
 book1.read();  
 System.out.println(book1.getOpened());  
  
 book1.printContent();  
  
 book1.addContent("Привет");  
 System.out.println(book1.getModified());  
  
 book1.getSize();  
  
 book1.printContent();  
  
 }  
}

Результат:

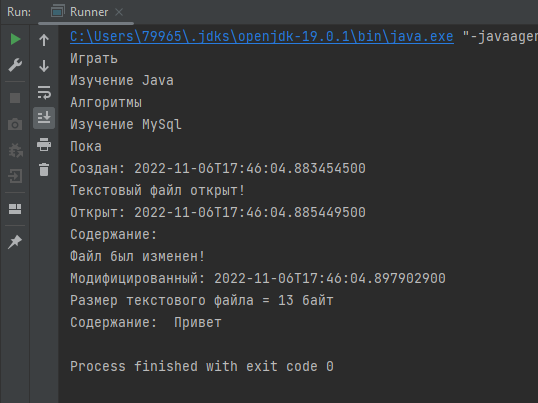


Рисунок – 6

1.8 Задача 8

Условие:

12. Составить описание класса прямоугольников со сторонами, параллельными осям координат. Предусмотреть при этом возмож­- ность перемещения прямоугольников на плоскости, изменения их размеров, построения наименьшего прямоугольника, содержаще­го два заданных прямоугольника, и прямоугольника, являющегося общей частью (пересечением) двух прямоугольников.

public class Rectangle {  
  
  
 private double width;  
  
 private double height;  
  
 public Rectangle(double width, double height) {  
 this.width = width;  
 this.height = height;  
 }  
  
 public double getPerimeter() {  
 return 2 \* (width + height);  
 }  
  
 public double getArea() {  
 return width \* height;  
 }  
  
 public double getDiagonal() {  
 return Math.sqrt(width \* width + height \* height);  
 }  
  
 public boolean isSquare() {  
 return width == height;  
 }  
  
 public static void main(String[] args) {  
  
 Rectangle r = new Rectangle(4, 5);  
 System.out.println(r.getPerimeter()); /\* 18 \*/  
 System.out.println(r.getArea()); /\* 20 \*/  
 System.out.println(r.getDiagonal()); /\* 6.4 ... \*/  
 System.out.println(r.isSquare()); /\* false \*/  
  
 }  
}

Результат:

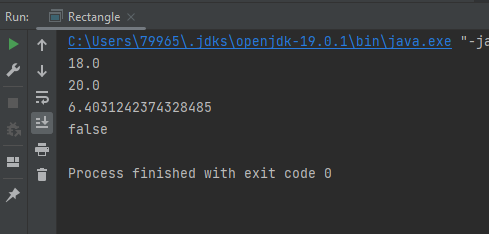


Рисунок – 8

1. Разработка ER-диаграммы

Условие:

|  |  |
| --- | --- |
| 12 | Театры г. Калининграда |

    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 «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.

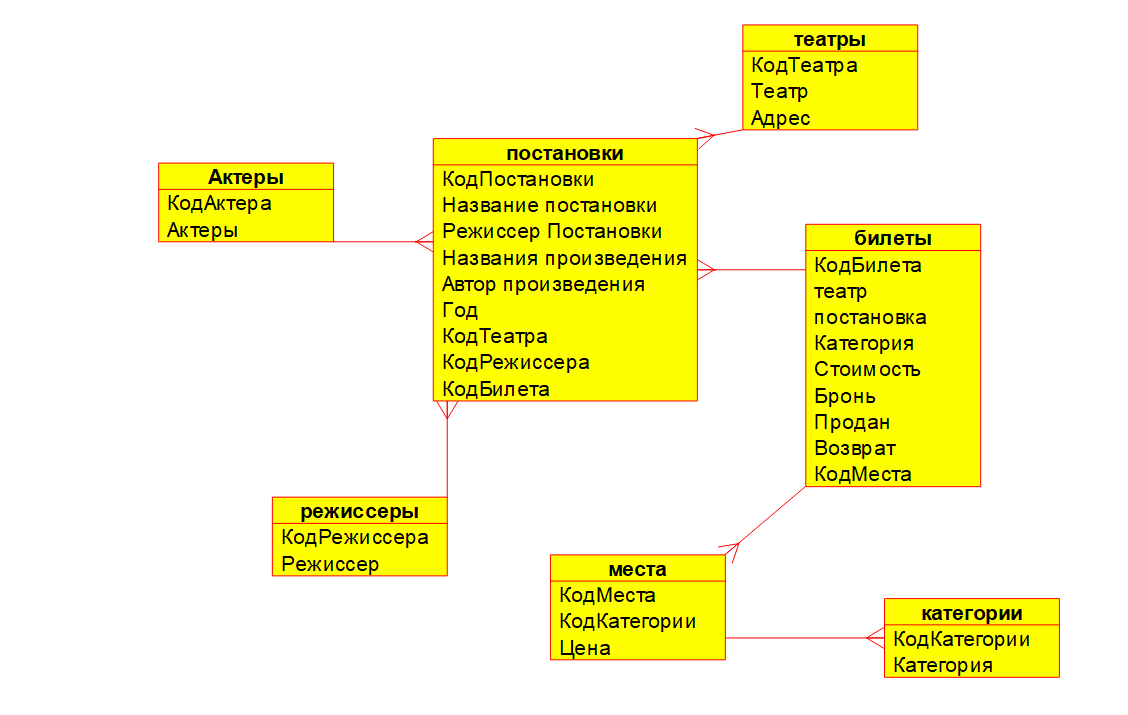


Рисунок – 9