

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

институт

Кафедра «Информатика»

кафедра

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4

Работа с базами данных в Spring

тема

Преподаватель

дата, подпись

А.С. Черниговский

инициалы, фамилия

Студент

КИ18-16Б

номер группы, зачётной книжки

дата, подпись

К.И. Оглезнев

инициалы, фамилия

Красноярск 2021

1 Цель работы

Ознакомиться с механизмами работы с базами данных в Spring Framework.

2 Задачи работы

В каждом варианте есть сущность базы данных. Необходимо:

1) Описать класс сущности, который имеет как минимум три текстовых поля и два числовых (и, естественно, id). Она описывает некий товар (затем, эта сущность и БД пригодится нам в следующих работах).

2) Создать таблицу базы данных (студент может выбрать любую реляционную БД), соответствующую спроектированной сущности.

3) Реализовать консольное Spring приложение (должно иметь простейший консольный пользовательский интерфейс), которое должно позволять:

- *Вводить (консольный ввод) пользователю поля сущности и добавлять её в таблицу БД.*

- *Выводить в консоль все записи из таблицы БД.*

- *Редактировать запись таблицы БД по Id.*

- *Удалять запись по Id.*

- *Осуществлять поиск по любому из признаков (Студент самостоятельно выбирает поле для поиска. Например, поиск всех студентов, средний балл которых выше введенного пользователем.)*

4) Способ работы с БД (JdbcTemplate, Hibernate, JPA или др.) студентом выбирается самостоятельно, ограничение одно — должен использоваться Spring Framework.

3 Ход работы

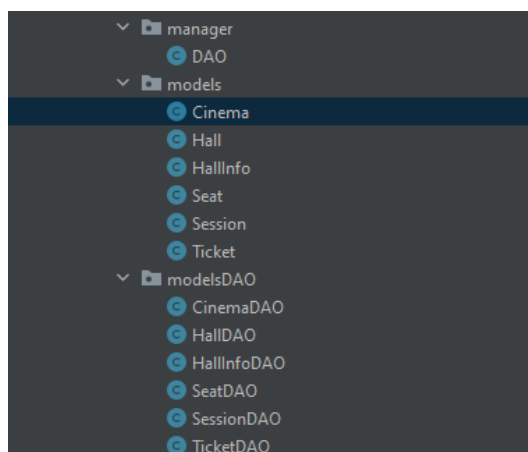
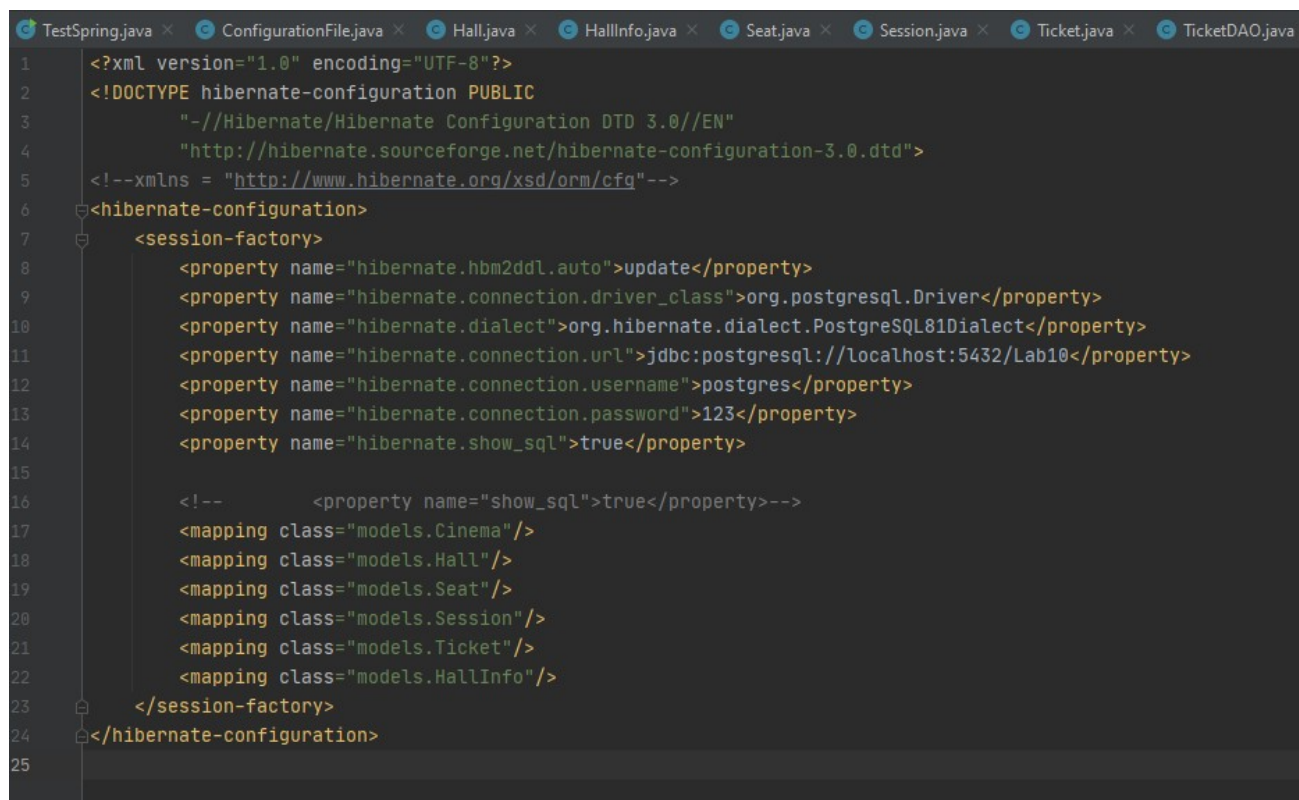


Рисунок 1 – ORM Hibernate + DAO



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE hibernate-configuration PUBLIC
3     "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
4     "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
5 <!--xmlns = "http://www.hibernate.org/xsd/orm/cfg"-->
6 <hibernate-configuration>
7     <session-factory>
8         <property name="hibernate.hbm2ddl.auto">update</property>
9         <property name="hibernate.connection.driver_class">org.postgresql.Driver</property>
10        <property name="hibernate.dialect">org.hibernate.dialect.PostgreSQL81Dialect</property>
11        <property name="hibernate.connection.url">jdbc:postgresql://localhost:5432/Lab10</property>
12        <property name="hibernate.connection.username">postgres</property>
13        <property name="hibernate.connection.password">123</property>
14        <property name="hibernate.show_sql">true</property>
15
16        <!-- <property name="show_sql">true</property>-->
17        <mapping class="models.Cinema"/>
18        <mapping class="models.Hall"/>
19        <mapping class="models.Seat"/>
20        <mapping class="models.Session"/>
21        <mapping class="models.Ticket"/>
22        <mapping class="models.HallInfo"/>
23    </session-factory>
24 </hibernate-configuration>
25
```

Рисунок 2 – hibernate.cfg.xml

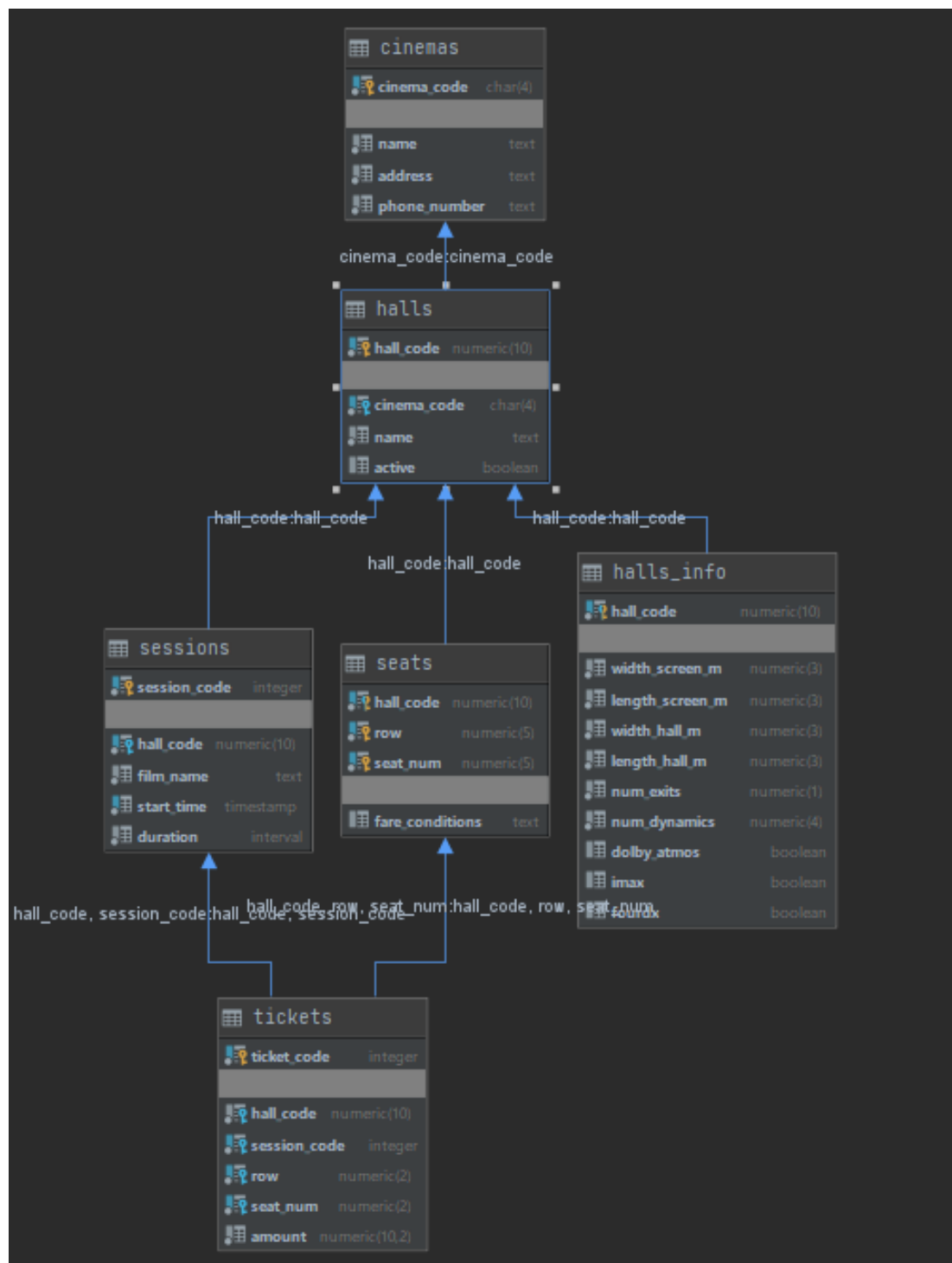


Рисунок 3 – БД

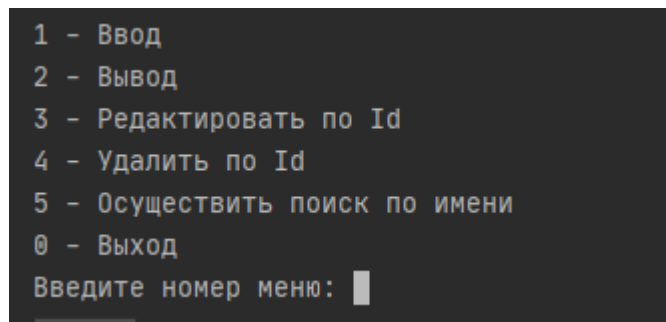


Рисунок 4 – Меню

```
Cinema(cinemaCode=A24 , name=ЛУЧ, address=Красноярск, Карла Маркса 149, phoneNumber=+7 XXX XXX-XX-XX)
Cinema(cinemaCode=A19 , name=НАУТИЛУС, address=Абакан, Тараса Шевченко, 74, phoneNumber=+7 XXX XXX-XX-XX)
Cinema(cinemaCode=B24 , name=ЛУЧ, address=Лесосибирск, 6-й микрорайон, 1, phoneNumber=+7 XXX XXX-XX-XX)
Cinema(cinemaCode=C24 , name=ЛУЧ, address=Красноярск, Что-то, 1, phoneNumber=+7 XXX XXX-XX-XX)
Cinema(cinemaCode=TEST, name=TEST, address=TEST, phoneNumber=TEST)
Cinema(cinemaCode=3 , name=3, address=3, phoneNumber=3)
```

Рисунок 5 – Вывод (меню – 2)

4 Вывод

В ходе данной лабораторной работы были изучены основы работы с базами данных в Spring Framework.

Листинг

main.java

```
package ru.myapp;

import manager.DAO;
import models.Cinema;
import modelsDAO.CinemaDAO;

import java.util.Scanner;

public class TestSpring {
    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.println("1 - Ввод");
            System.out.println("2 - Вывод");
            System.out.println("3 - Редактировать по Id");
            System.out.println("4 - Удалить по Id");
            System.out.println("5 - Осуществить поиск по имени");
            System.out.println("0 - Выход");

            System.out.print("Введите номер меню: ");

            String key = scanner.nextLine();

            switch (key) {
                case "1":
                    Cinema newObj = new Cinema();
                    System.out.print("Введите код кинотеатра: ");
                    String str = scanner.nextLine();
                    newObj.setCinemaCode(str);
                    System.out.print("Введите название кинотеатра: ");
                    str = scanner.nextLine();
                    newObj.setName(str);
                    System.out.print("Введите адрес кинотеатра: ");
                    str = scanner.nextLine();
                    newObj.setAddress(str);
                    System.out.print("Введите телефон кинотеатра: ");
                    str = scanner.nextLine();
                    newObj.setPhoneNumber(str);
                    CinemaDAO.insert(newObj);
                    DAO.transaction();
                    break;
                case "2":
                    var tmpList = CinemaDAO.findAll();
                    for (Object item : tmpList)
                        System.out.println(item);
                    break;
                case "3":
```

```

        System.out.print("Введите код кинотеатра: ");
        str = scanner.nextLine();
        Cinema edit = CinemaDAO.findById(str);
        if (edit == null) {
            System.out.println("Не найдено!");
            break;
        }
        System.out.print("Введите название кинотеатра: ");
        str = scanner.nextLine();
        edit.setName(str);
        System.out.print("Введите адрес кинотеатра: ");
        str = scanner.nextLine();
        edit.setAddress(str);
        System.out.print("Введите телефон кинотеатра: ");
        str = scanner.nextLine();
        edit.setPhoneNumber(str);
        CinemaDAO.insert(edit);
        DAO.transaction();
        break;
    case "4":
        System.out.print("Введите код кинотеатра: ");
        str = scanner.nextLine();
        Cinema del = CinemaDAO.findById(str);
        CinemaDAO.delete(del);
        DAO.transaction();
        break;
    case "5":
        System.out.print("Введите имя кинотеатра: ");
        str = scanner.nextLine();
        var byName = CinemaDAO.findByName(str);
        for (Object item : byName)
            System.out.println(item);
        break;
    case "0":
        return;
    }
}
}
}
}

```