

Домашнее задание:

Реализуйте с помощью Hibernate следующее:

Добавление нового курса:

Добавьте данные о новом курсе, такие как название, описание, продолжительность, стоимость и преподаватель.

Работа с преподавателями:

Добавьте возможность добавления, просмотра и удаления преподавателей.

Свяжите курсы с преподавателями, чтобы каждый курс имел преподавателя.

Реализуйте вывод о студентах, зарегистрированных на курс, и общей стоимости обучения:

Например, вывод о студентах, зарегистрированных на курс "Java программирование", с их именами, возрастами и датами регистрации.

Проект с занятия:

```

import composite.SubscriptionIdKey;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.boot.Metadata;
import org.hibernate.boot.MetadataSources;
import org.hibernate.boot.registry.StandardServiceRegistry;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;

import java.util.HashMap;
import java.util.List;

public class Main {
    public static void main(String[] args) {
        // Отображения id для курсов и студентов
        HashMap<String, Integer> courseMap = new HashMap<>();
        HashMap<String, Integer> studentMap = new HashMap<>();

        // Конфигурация Hibernate
        StandardServiceRegistry registry = new StandardServiceRegistryBuilder()
            .configure("hibernate.cfg.xml").build();
        Metadata metadata = new MetadataSources(registry).getMetadataBuilder().build();
        SessionFactory sessionFactory = metadata.getSessionFactoryBuilder().build();
        Session session = sessionFactory.openSession();

        Transaction transaction = session.beginTransaction();

        //
        // // Получить список курсов и список студентов
        // List<Course> courseList = session.createQuery("From " +
Course.class.getSimpleName()).getResultList();
        // List<Student> studentList = session.createQuery("From " +
Student.class.getSimpleName()).getResultList();
        //
        // // Заполняем отображения id`s
        // for (int i = 0; i < courseList.size(); i++) {
        //     courseMap.put(courseList.get(i).getName(), courseList.get(i).getId());
        // }
        // for (int i = 0; i < studentList.size(); i++) {
        //     studentMap.put(studentList.get(i).getName(), studentList.get(i).getId());
        // }
        //
        // List<Purchase> purchaseList = session.createQuery("From " +
Purchase.class.getSimpleName()).getResultList();
        // for (int i = 0; i < purchaseList.size(); i++) {
        //     LinkedPurchaseList lpl = new LinkedPurchaseList();
        //     String studentName = purchaseList.get(i).getStudentName();
        //     String courseName = purchaseList.get(i).getCourseName();
        //     Integer tempStudentId = studentMap.get(studentName);
        //     Integer tempCourseId = courseMap.get(courseName);
        //
        //     if (tempCourseId != null && tempStudentId != null) {
        //         SubscriptionIdKey key = new SubscriptionIdKey(tempStudentId,
tempCourseId);
        //         lpl.setId(key);
        //         session.saveOrUpdate(lpl);
        //     } else {
        //         System.out.println("Ошибка: id студента или курса не найден!");
        //     }
        // }

        // todo Поиск курсов студента по имени

        String studentName = "Блаженков Артем";

        Student student = session.createQuery("FROM Student WHERE name = :name",
Student.class)
            .setParameter("name", studentName)

```

```

        .uniqueResult();
    if (student != null) {
        List<Course> courses = student.getCourses();
        if (courses.isEmpty()) {
            System.out.println("Student " + studentName + "not found!");
        } else {
            System.out.println("Courses for student " + studentName + ":");
            courses.forEach(course -> System.out.println(course.getName()));
        }
    }
}

//-----
//      todo Поиск студентов подписанных на курс по имени

//      String courseName = "Java-разработчик";
//
//      Course course = session.createQuery("FROM Course WHERE name = :name",
Course.class)
//          .setParameter("name", courseName)
//          .uniqueResult();
//
//      if (course != null) {
//          List<Student> students = course.getStudents();
//          if (students.isEmpty()) {
//              System.out.println("No students enrolled in course " +
courseName);
//          } else {
//              System.out.println("Students for course " + courseName + ":");
//              for (Student student : students) {
//                  System.out.println(student.getName());
//              }
//          }
//      } else {
//          System.out.println("Course not found: " + courseName);
//      }
//-----
//      todo Обновление цены и описания на выбранном курсе

//      int courseId = 1;
//      String newDescription = "Новое описание";
//      Integer newPrice = 2000;
//
//      Course course = session.get(Course.class, courseId);
//
//      if (course != null) {
//          course.setDescription(newDescription);
//          course.setPrice(newPrice);
//
//          session.saveOrUpdate(course);
//          System.out.println("Course information updated successfully.");
//      } else {
//          System.out.println("Course not found: " + courseId);
//      }
//-----
//      todo Получение средней продолжительности курсов

//      Double averageDuration = session.createQuery("SELECT AVG(duration) FROM
Course", Double.class)
//          .uniqueResult();
//
//      System.out.println("Average course duration: " + (averageDuration != null ?
averageDuration : "N/A"));
//-----
//      todo Удаление студента с выбранного курса

```

```

//      String studentName = "Блаженков Артем";
//      String courseName = "Анимация интерфейсов";
//
//      Student student = session.createQuery("FROM Student WHERE name = :name",
Student.class)
//          .setParameter("name", studentName)
//          .uniqueResult();
//
//      // Поиск курса
//      Course course = session.createQuery("FROM Course WHERE name = :name",
Course.class)
//          .setParameter("name", courseName)
//          .uniqueResult();
//
//      // Если студент и курс найдены
//      if (student != null && course != null) {
//          // Удаление студента из списка студентов курса
//          List<Student> students = course.getStudents();
//          students.remove(student);
//          course.setStudents(students);
//
//          // Сохранение или обновление курса
//          session.saveOrUpdate(course);
//
//          // Создание новой связи в LinkedPurchaseList
//          LinkedPurchaseList linkedPurchaseList = new LinkedPurchaseList();
//          SubscriptionIdKey key = new SubscriptionIdKey(student.getId(),
course.getId());
//          linkedPurchaseList.setId(key);
//          session.saveOrUpdate(linkedPurchaseList);
//      } todo Выполнить в базе данных перед удалением:
//      ALTER TABLE Subscriptions MODIFY COLUMN subscription_date TIMESTAMP DEFAULT
CURRENT_TIMESTAMP;
//
//      System.out.println("Student " + studentName + " removed from course " +
courseName);
//      } else {
//          System.out.println("Student or course not found.");
//      }
//-----
//-----
//      transaction.commit();
//      session.close();
//      sessionFactory.close();
//  }
}

```