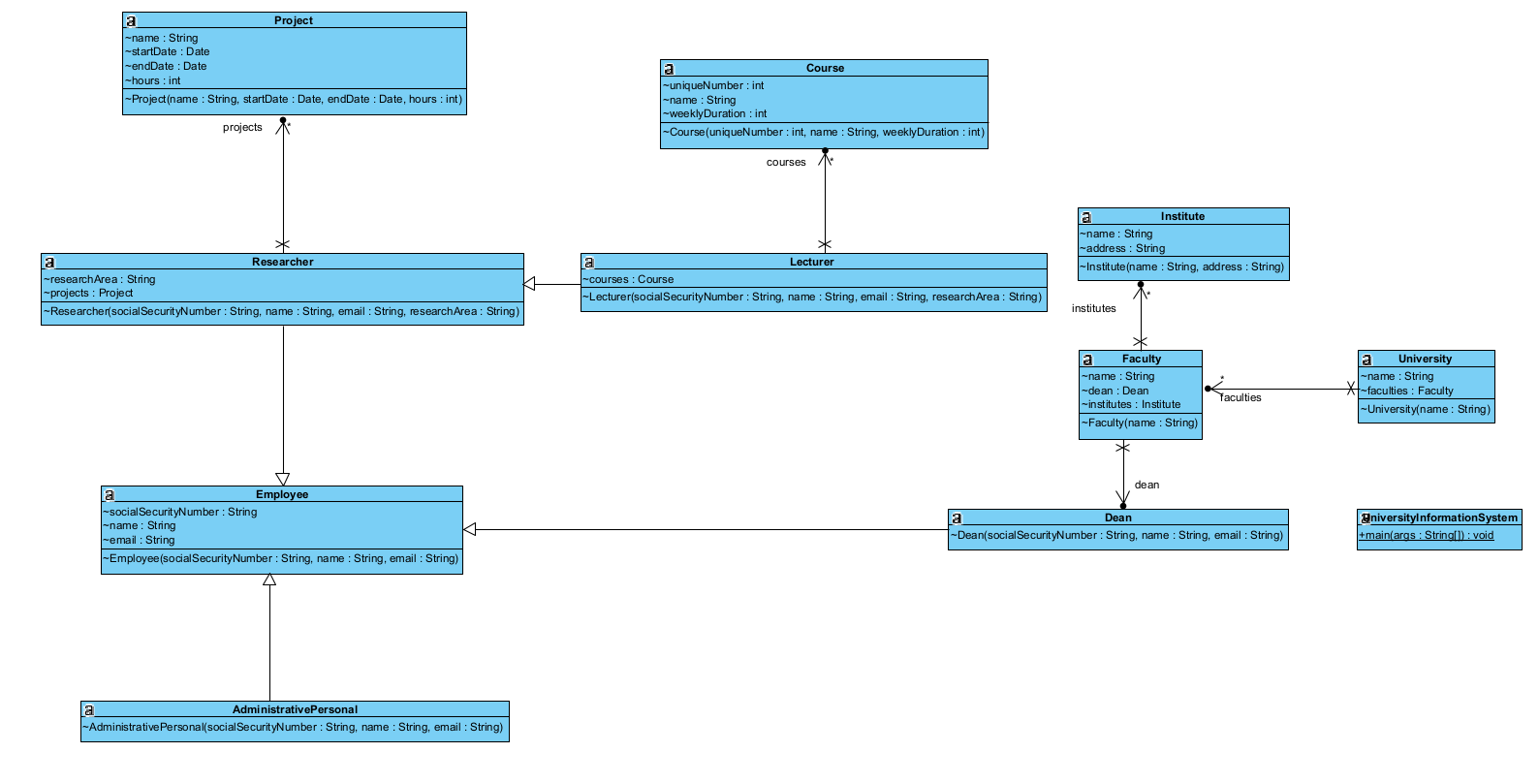
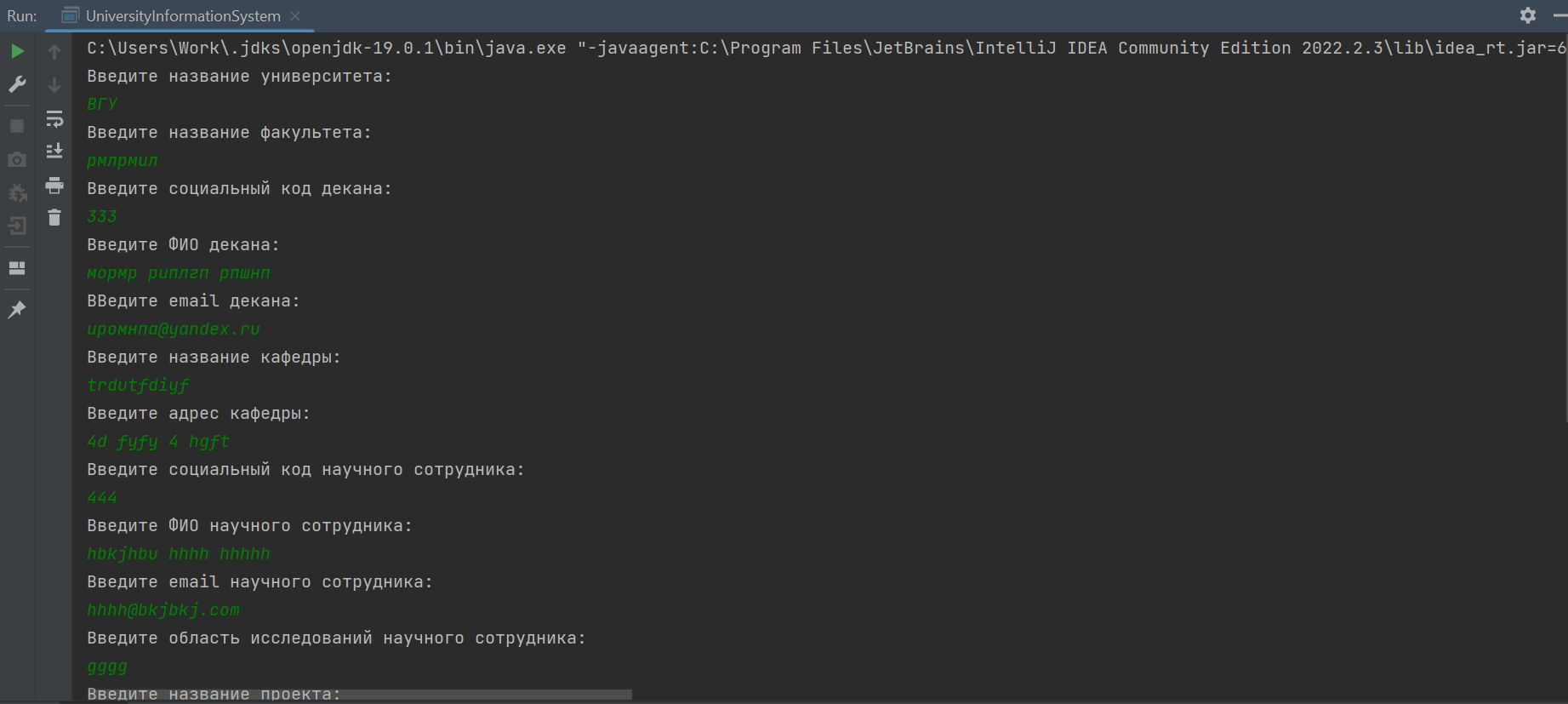
Диаграмма классов, сгенерированная по Java коду



Код:

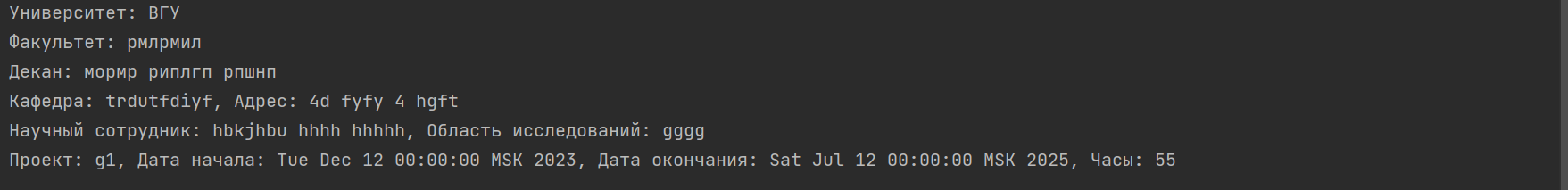
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.ArrayList;  
import java.util.Date;  
import java.util.List;  
import java.util.Scanner;  
  
class University {  
 String name;  
 List<Faculty> faculties;  
  
 University(String name) {  
 this.name = name;  
 this.faculties = new ArrayList<>();  
 }  
}  
  
class Faculty {  
 String name;  
 Dean dean;  
 List<Institute> institutes;  
  
 Faculty(String name) {  
 this.name = name;  
 this.institutes = new ArrayList<>();  
 }  
}  
  
class Institute {  
 String name;  
 String address;  
  
 Institute(String name, String address) {  
 this.name = name;  
 this.address = address;  
 }  
}  
  
class Employee {  
 String socialSecurityNumber;  
 String name;  
 String email;  
  
 Employee(String socialSecurityNumber, String name, String email) {  
 this.socialSecurityNumber = socialSecurityNumber;  
 this.name = name;  
 this.email = email;  
 }  
}  
  
class Dean extends Employee {  
 Dean(String socialSecurityNumber, String name, String email) {  
 super(socialSecurityNumber, name, email);  
 }  
}  
  
class Researcher extends Employee {  
 String researchArea;  
 List<Project> projects;  
  
 Researcher(String socialSecurityNumber, String name, String email, String researchArea) {  
 super(socialSecurityNumber, name, email);  
 this.researchArea = researchArea;  
 this.projects = new ArrayList<>();  
 }  
}  
  
class Project {  
 String name;  
 Date startDate;  
 Date endDate;  
 int hours;  
  
 Project(String name, Date startDate, Date endDate, int hours) {  
 this.name = name;  
 this.startDate = startDate;  
 this.endDate = endDate;  
 this.hours = hours;  
 }  
}  
  
class AdministrativePersonal extends Employee {  
 AdministrativePersonal(String socialSecurityNumber, String name, String email) {  
 super(socialSecurityNumber, name, email);  
 }  
}  
  
class Lecturer extends Researcher {  
 List<Course> courses;  
  
 Lecturer(String socialSecurityNumber, String name, String email, String researchArea) {  
 super(socialSecurityNumber, name, email, researchArea);  
 this.courses = new ArrayList<>();  
 }  
}  
  
class Course {  
 int uniqueNumber;  
 String name;  
 int weeklyDuration;  
  
 Course(int uniqueNumber, String name, int weeklyDuration) {  
 this.uniqueNumber = uniqueNumber;  
 this.name = name;  
 this.weeklyDuration = weeklyDuration;  
 }  
}  
  
public class UniversityInformationSystem {  
 public static void main(String[] args) throws ParseException {  
 Scanner scanner = new Scanner(System.*in*);  
  
 System.*out*.println("Введите название университета:");  
 String universityName = scanner.nextLine();  
 University university = new University(universityName);  
  
 System.*out*.println("Введите название факультета:");  
 String facultyName = scanner.nextLine();  
 Faculty faculty = new Faculty(facultyName);  
  
 System.*out*.println("Введите социальный код декана:");  
 String deanSSN = scanner.nextLine();  
 System.*out*.println("Введите ФИО декана:");  
 String deanName = scanner.nextLine();  
 System.*out*.println("ВВедите email декана:");  
 String deanEmail = scanner.nextLine();  
 Dean dean = new Dean(deanSSN, deanName, deanEmail);  
  
 // Связываем декана с факультетом  
 faculty.dean = dean;  
  
 System.*out*.println("Введите название кафедры:");  
 String instituteName = scanner.nextLine();  
 System.*out*.println("Введите адрес кафедры:");  
 String instituteAddress = scanner.nextLine();  
 Institute institute = new Institute(instituteName, instituteAddress);  
  
 // Связываем кафедру с факультетом  
 faculty.institutes.add(institute);  
  
 System.*out*.println("Введите социальный код научного сотрудника:");  
 String researcherSSN = scanner.nextLine();  
 System.*out*.println("Введите ФИО научного сотрудника:");  
 String researcherName = scanner.nextLine();  
 System.*out*.println("Введите email научного сотрудника:");  
 String researcherEmail = scanner.nextLine();  
 System.*out*.println("Введите область исследований научного сотрудника:");  
 String researchArea = scanner.nextLine();  
 Researcher researcher = new Researcher(researcherSSN, researcherName, researcherEmail, researchArea);  
  
 System.*out*.println("Введите название проекта:");  
 String projectName = scanner.nextLine();  
 System.*out*.println("Введите дату начала проекта (yyyy-MM-dd):");  
 String startDateString = scanner.nextLine();  
 Date startDate = new SimpleDateFormat("yyyy-MM-dd").parse(startDateString);  
 System.*out*.println("Введите дату окончания проекта (yyyy-MM-dd):");  
 String endDateString = scanner.nextLine();  
 Date endDate = new SimpleDateFormat("yyyy-MM-dd").parse(endDateString);  
 System.*out*.println("Введите кол-во часов на проект:");  
 int projectHours = scanner.nextInt();  
 Project project = new Project(projectName, startDate, endDate, projectHours);  
  
 // Связываем научного сотрудника с проектом  
 researcher.projects.add(project);  
  
 System.*out*.println("Университет: " + university.name);  
 System.*out*.println("Факультет: " + faculty.name);  
 System.*out*.println("Декан: " + dean.name);  
 System.*out*.println("Кафедра: " + institute.name + ", Адрес: " + institute.address);  
 System.*out*.println("Научный сотрудник: " + researcher.name + ", Область исследований: " + researcher.researchArea);  
 System.*out*.println("Проект: " + project.name + ", Дата начала: " + startDate + ", Дата окончания: " + endDate +  
 ", Часы: " + project.hours);  
  
 scanner.close();  
 }  
}

Пример ввода:





Пример вывода:



Use case

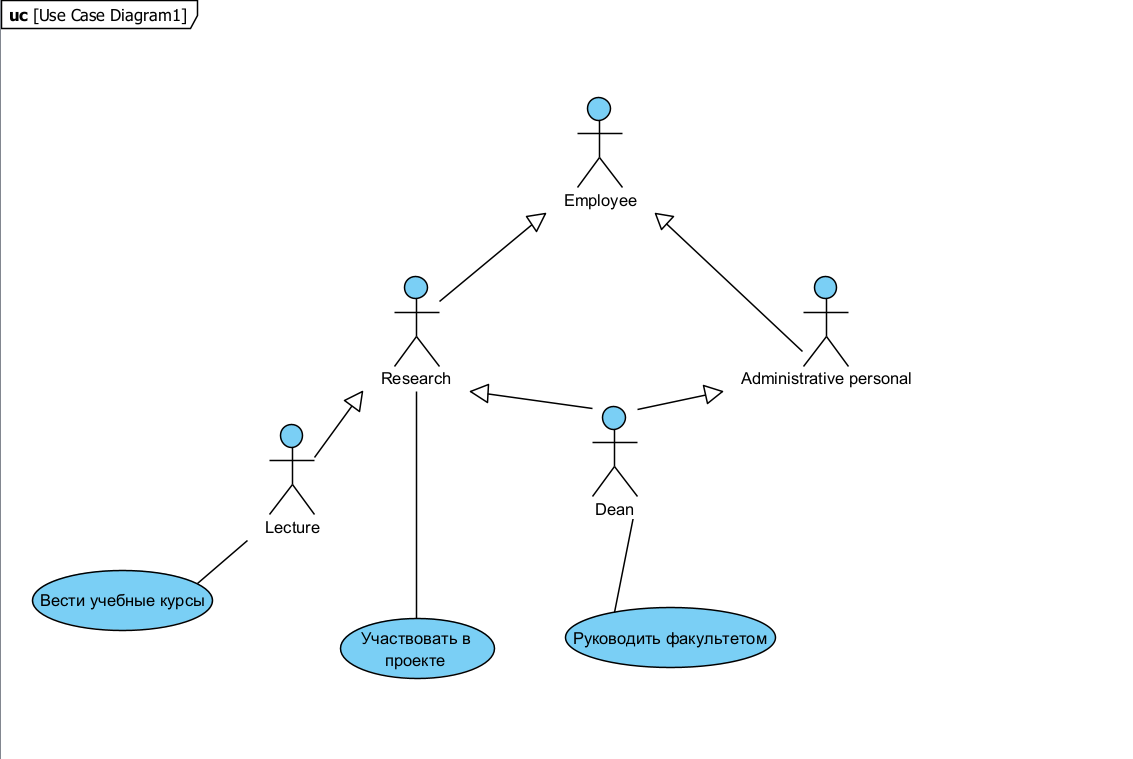


Диаграмма последовательностей

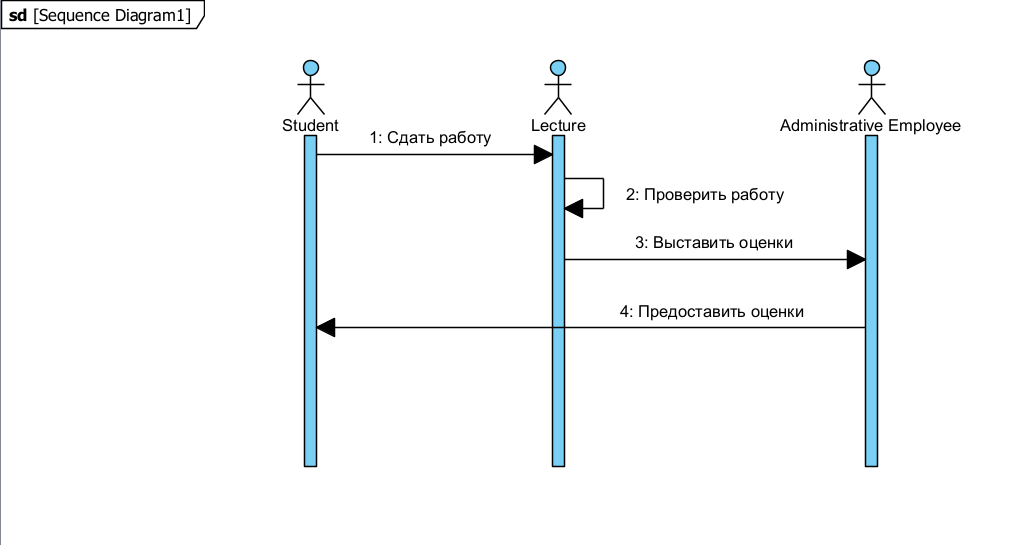


Диаграмма активностей

