

Результаты моделирования

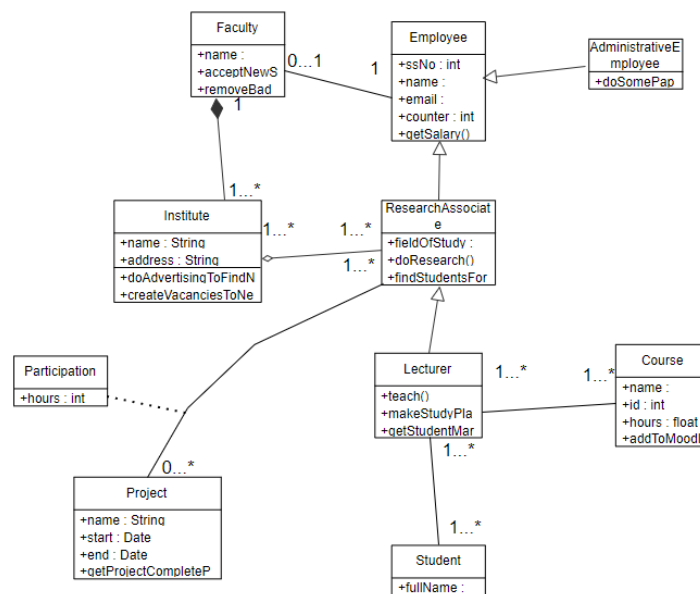


Рис.1 Диаграмма классов

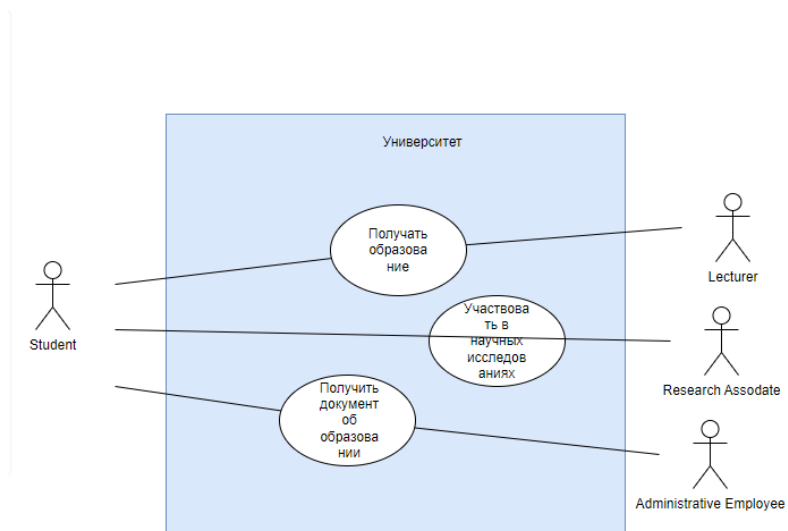


Рис. 2 Диаграмма прецедентов

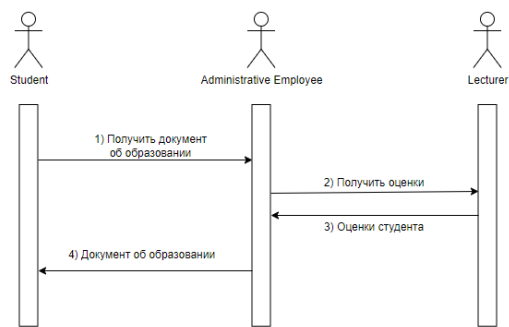


Рис. 3 Диаграмма последовательности

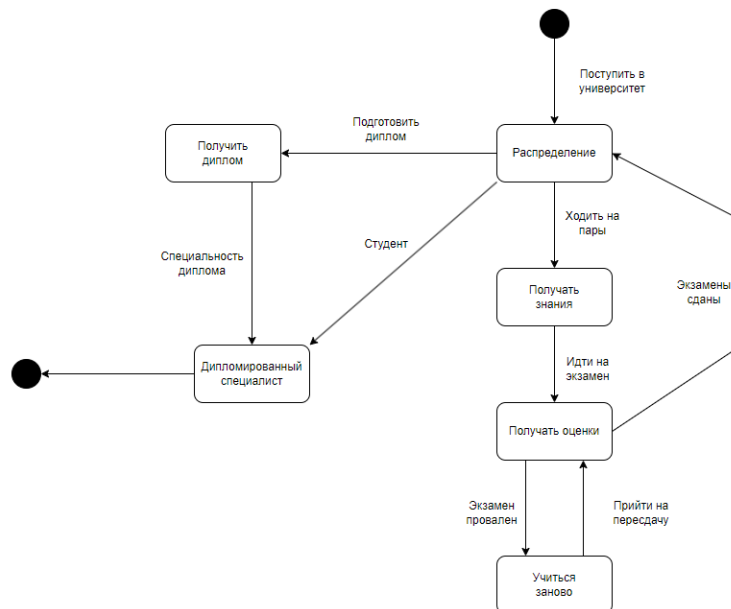


Рис. 4 Диаграмма состояний

```

class University {
    2 usages
    String name;
    1 usage
    List<Faculty> faculties;

    1 usage
    University(String name) {
        this.name = name;
        this.faculties = new ArrayList<>();
    }
}

3 usages
class Faculty {
    2 usages
    String name;
    1 usage
    Dean dean;
    2 usages
    List<Institute> institutes;

    1 usage
    Faculty(String name) {
        this.name = name;
        this.institutes = new ArrayList<>();
    }
}

3 usages
class Institute {
    2 usages
    String name;
    2 usages
    String address;

    1 usage
    Institute(String name, String address) {
        this.name = name;
        this.address = address;
    }
}

```

1. “Университет (university) состоит из множества факультетов (faculty), которые в свою очередь разделены на кафедры (institute).”

```

3 usages
class Dean extends Employee {
    1 usage
    Dean(String socialSecurityNumber, String name, String email) { super(socialSecurityNumber, name, email); }
}

3 usages: 4 inheritors
class Employee {
    1 usage
    String socialSecurityNumber;
    3 usages
    String name;
    1 usage
    String email;

    3 usages
    Employee(String socialSecurityNumber, String name, String email) {
        this.socialSecurityNumber = socialSecurityNumber;
        this.name = name;
        this.email = email;
    }
}

3 usages: 1 inheritor
class Researcher extends Employee {
    2 usages
    String researchArea;
    2 usages
    List<Project> projects;

    2 usages
    Researcher(String socialSecurityNumber, String name, String email, String researchArea) {
        super(socialSecurityNumber, name, email);
        this.researchArea = researchArea;
        this.projects = new ArrayList<>();
    }
}

```

2. “Руководителем каждого факультета является декан (dean), который является сотрудником (employee) университета и научным работником(researcher)”

```

3 usages
class Project {
    2 usages
    String name;
    1 usage
    Date startDate;
    1 usage
    Date endDate;
    2 usages
    int hours;

    1 usage
    Project(String name, Date startDate, Date endDate, int hours) {
        this.name = name;
        this.startDate = startDate;
        this.endDate = endDate;
        this.hours = hours;
    }
}

```

4. “Кроме того, каждый научный сотрудник может быть вовлечён в проекты (project) на определённое количество часов ...”

```

class Lecturer extends Researcher {
    1 usage
    List<Course> courses;

    no usages
    Lecturer(String socialSecurityNumber, String name, String email, String researchArea) {
        super(socialSecurityNumber, name, email, researchArea);
        this.courses = new ArrayList<>();
    }
}

1 usage
class Course {
    1 usage
    int uniqueNumber;
    1 usage
    String name;
    1 usage
    int weeklyDuration;

    no usages
    Course(int uniqueNumber, String name, int weeklyDuration) {
        this.uniqueNumber = uniqueNumber;
        this.name = name;
        this.weeklyDuration = weeklyDuration;
    }
}

```

5. “Некоторые научные сотрудники могут вести учебные курсы (course). Тогда они называются преподавателями (lecturer).”

```

// Связываем декана с факультетом
faculty.dean = dean;

```

```

// Связываем институт с факультетом
faculty.institutes.add(institute);

```

```
// Ввод данных для университета
System.out.println("Enter University name:");
String universityName = scanner.nextLine();
University university = new University(universityName);

// Ввод данных для факультета
System.out.println("Enter Faculty name:");
String facultyName = scanner.nextLine();
Faculty faculty = new Faculty(facultyName);

// Ввод данных для декана
System.out.println("Enter Dean's social security number:");
String deanSSN = scanner.nextLine();
System.out.println("Enter Dean's name:");
String deanName = scanner.nextLine();
System.out.println("Enter Dean's email:");
String deanEmail = scanner.nextLine();
Dean dean = new Dean(deanSSN, deanName, deanEmail);
```

```
// Вывод информации
System.out.println("University: " + university.name);
System.out.println("Faculty: " + faculty.name);
System.out.println("Dean: " + dean.name);
System.out.println("Institute: " + institute.name + ", Address: " + institute.address);
System.out.println("Researcher: " + researcher.name + ", Research Area: " + researcher.researchArea);
System.out.println("Project: " + project.name + ", Start Date: " + startDate + ", End Date: " + endDate + ", Hours: " + project.hours);
```

```
Enter University name:
VSD
Enter Faculty name:
FKN
Enter Dean's social security number:
111
Enter Dean's name:
XXXXXXXX
Enter Dean's email:
@email
Enter Institute name:
IOK
Enter Institute address:
xxxx
Enter Researcher's social security number:
IXC
Enter Researcher's name:
ING
Enter Researcher's email:
@email
Enter Researcher's research area:
CCC
Enter Project name:
SD
Enter Project start date (yyyy-MM-dd):
1023-12-22
```

Пример ввода