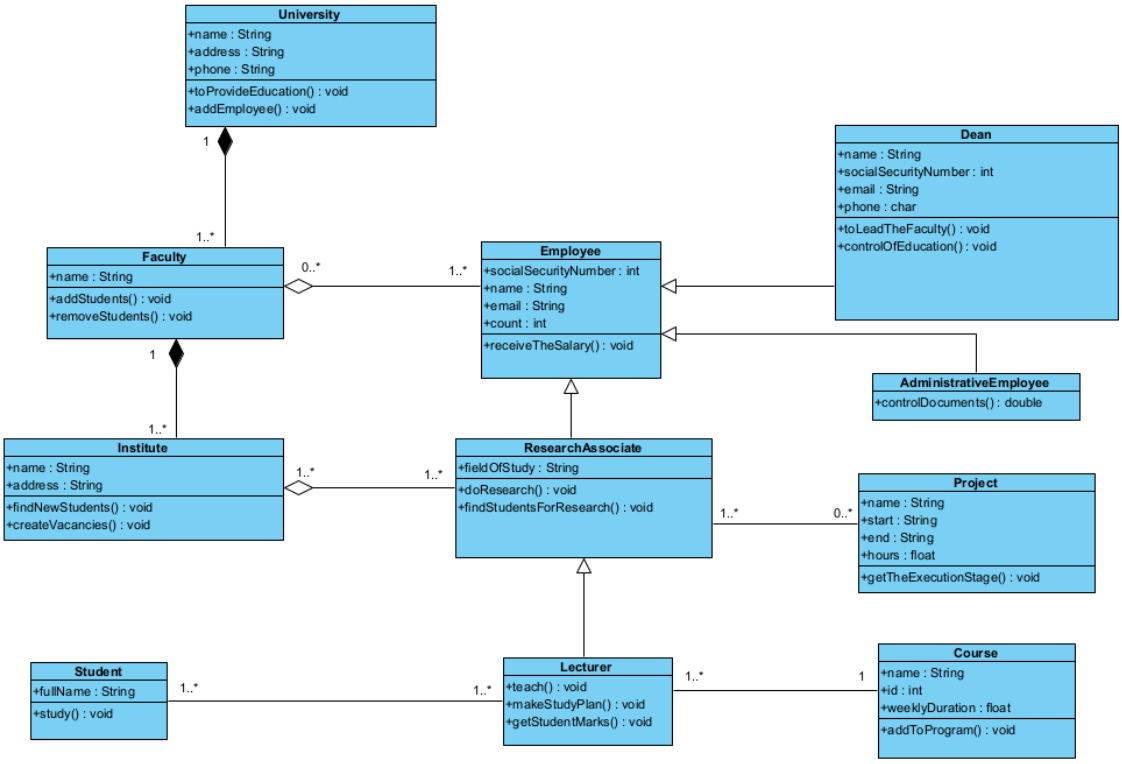
Диаграмма классов:



Сгенерированный код на Java:

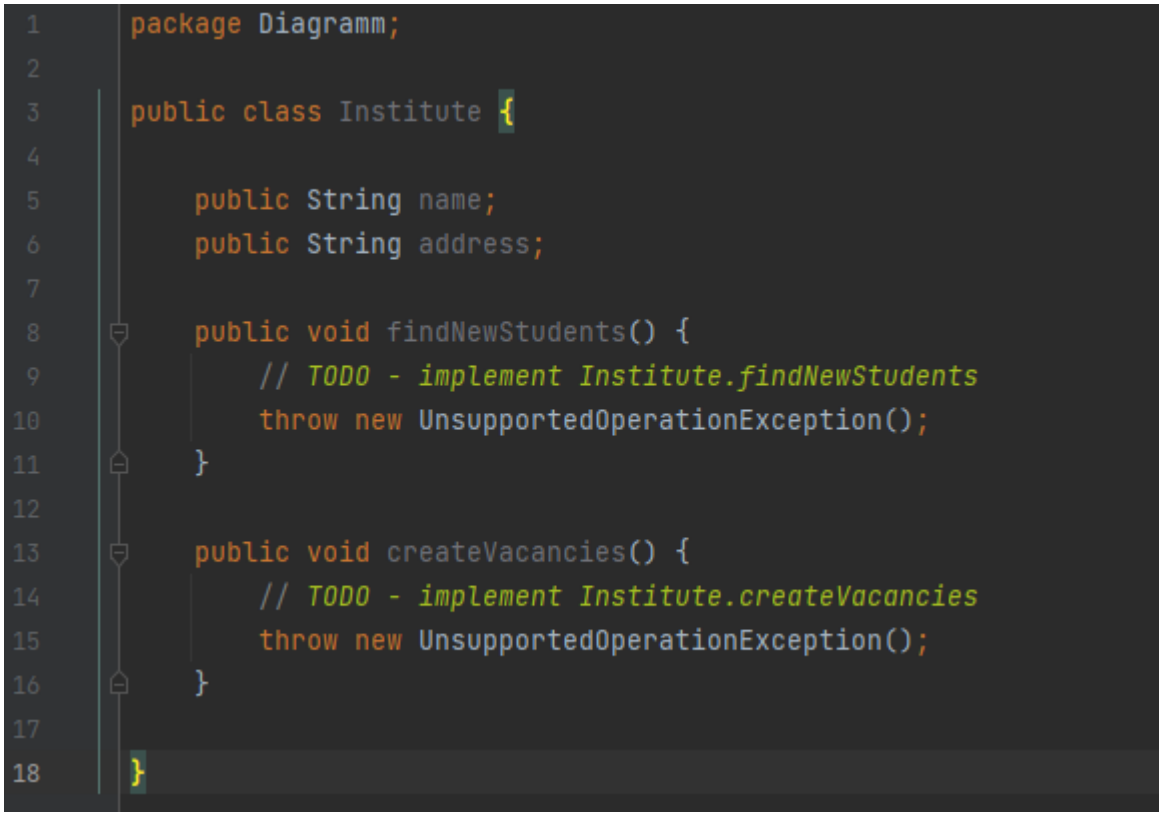
```java
package Diagramm;

public class AdministrativeEmployee extends Employee {

    public double controlDocuments() {
        // TODO - implement AdministrativeEmployee.controlDocuments
        throw new UnsupportedOperationException();
    }

}
```

```java
package Diagramm;

public class Course {

    public String name;
    public int id;
    public float weeklyDuration;

    public void addToProgram() {
        // TODO - implement Course.addToProgram
        throw new UnsupportedOperationException();
    }

}
```

```java
package Diagramm;

public class Dean extends Employee {

    public String name;
    public int socialSecurityNumber;
    public String email;
    public char phone;

    public void toLeadTheFaculty() {
        // TODO - implement Dean.toLeadTheFaculty
        throw new UnsupportedOperationException();
    }

    public void controlOfEducation() {
        // TODO - implement Dean.controlOfEducation
        throw new UnsupportedOperationException();
    }

}
```

```java
package Diagramm;

public class Employee {

    public int socialSecurityNumber;
    public String name;
    public String email;
    public int count;

    public void receiveTheSalary() {
        // TODO - implement Employee.receiveTheSalary
        throw new UnsupportedOperationException();
    }

}
```

```java
package Diagramm;

public class Faculty {

    public String name;

    public void addStudents() {
        // TODO - implement Faculty.addStudents
        throw new UnsupportedOperationException();
    }

    public void removeStudents() {
        // TODO - implement Faculty.removeStudents
        throw new UnsupportedOperationException();
    }

}
```



```java
package Diagramm;

public class Institute {

    public String name;
    public String address;

    public void findNewStudents() {
        // TODO - implement Institute.findNewStudents
        throw new UnsupportedOperationException();
    }

    public void createVacancies() {
        // TODO - implement Institute.createVacancies
        throw new UnsupportedOperationException();
    }

}
```

```java
package Diagramm;

public class Project {

    public String name;
    public String start;
    public String end;
    public float hours;

    public void getTheExecutionStage() {
        // TODO - implement Project.getTheExecutionStage
        throw new UnsupportedOperationException();
    }

}
```

```java
package Diagramm;

public class ResearchAssociate extends Employee {

    public String fieldOfStudy;

    public void doResearch() {
        // TODO - implement ResearchAssociate.doResearch
        throw new UnsupportedOperationException();
    }

    public void findStudentsForResearch() {
        // TODO - implement ResearchAssociate.findStudentsForResearch
        throw new UnsupportedOperationException();
    }

}
```

```java
package Diagramm;

public class Student {

    public String fullName;

    public void study() {
        // TODO - implement Student.study
        throw new UnsupportedOperationException();
    }

}
```

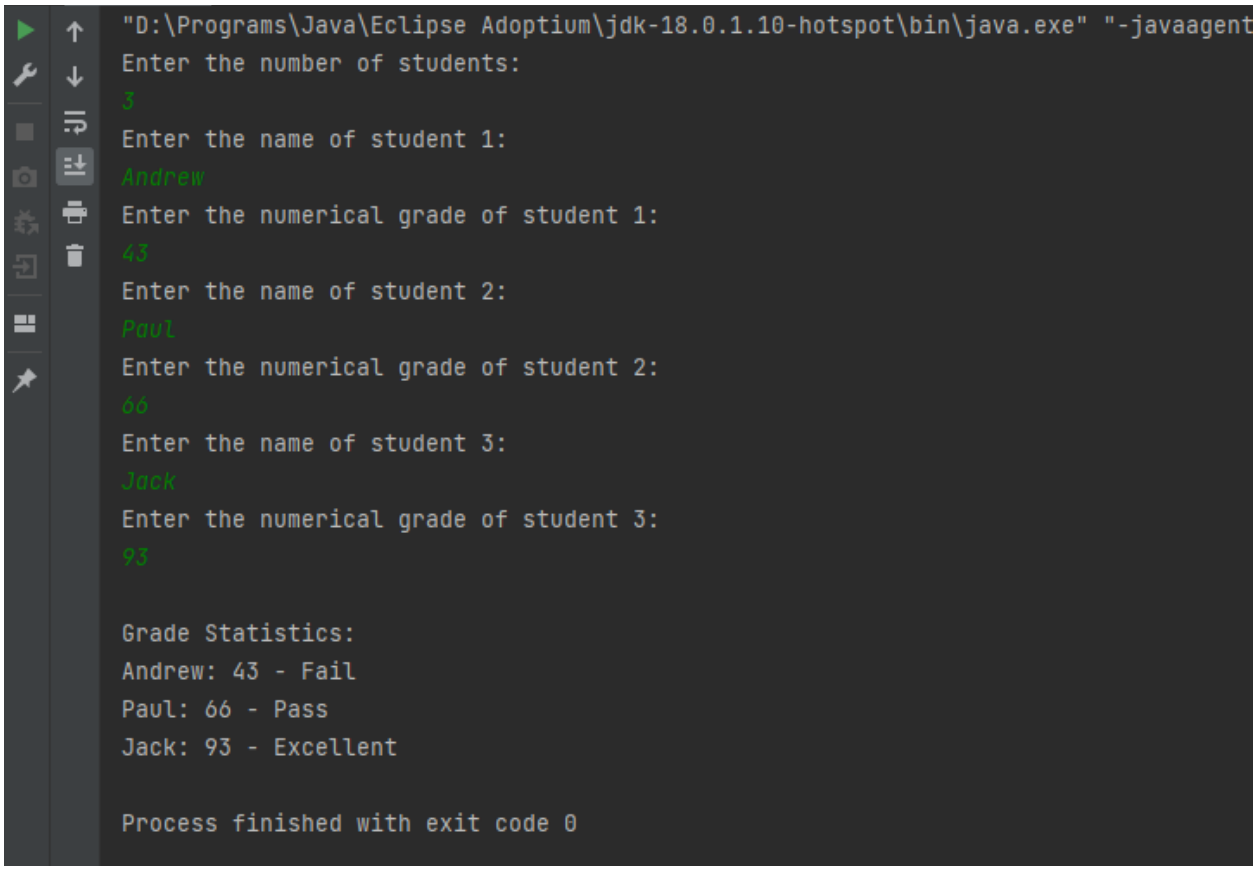
```java
package Diagramm;

public class University {

    public String name;
    public String address;
    public String phone;

    public void toProvideEducation() {
        // TODO - implement University.toProvideEducation
        throw new UnsupportedOperationException();
    }

    public void addEmployee() {
        // TODO - implement University.addEmployee
        throw new UnsupportedOperationException();
    }

}
```

```java
package Diagramm;

import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class Lecturer extends ResearchAssociate {

    public void teach() {
        // TODO - implement Lecturer.teach
        throw new UnsupportedOperationException();
    }

    public void makeStudyPlan() {
        // TODO - implement Lecturer.makeStudyPlan
        throw new UnsupportedOperationException();
    }
```

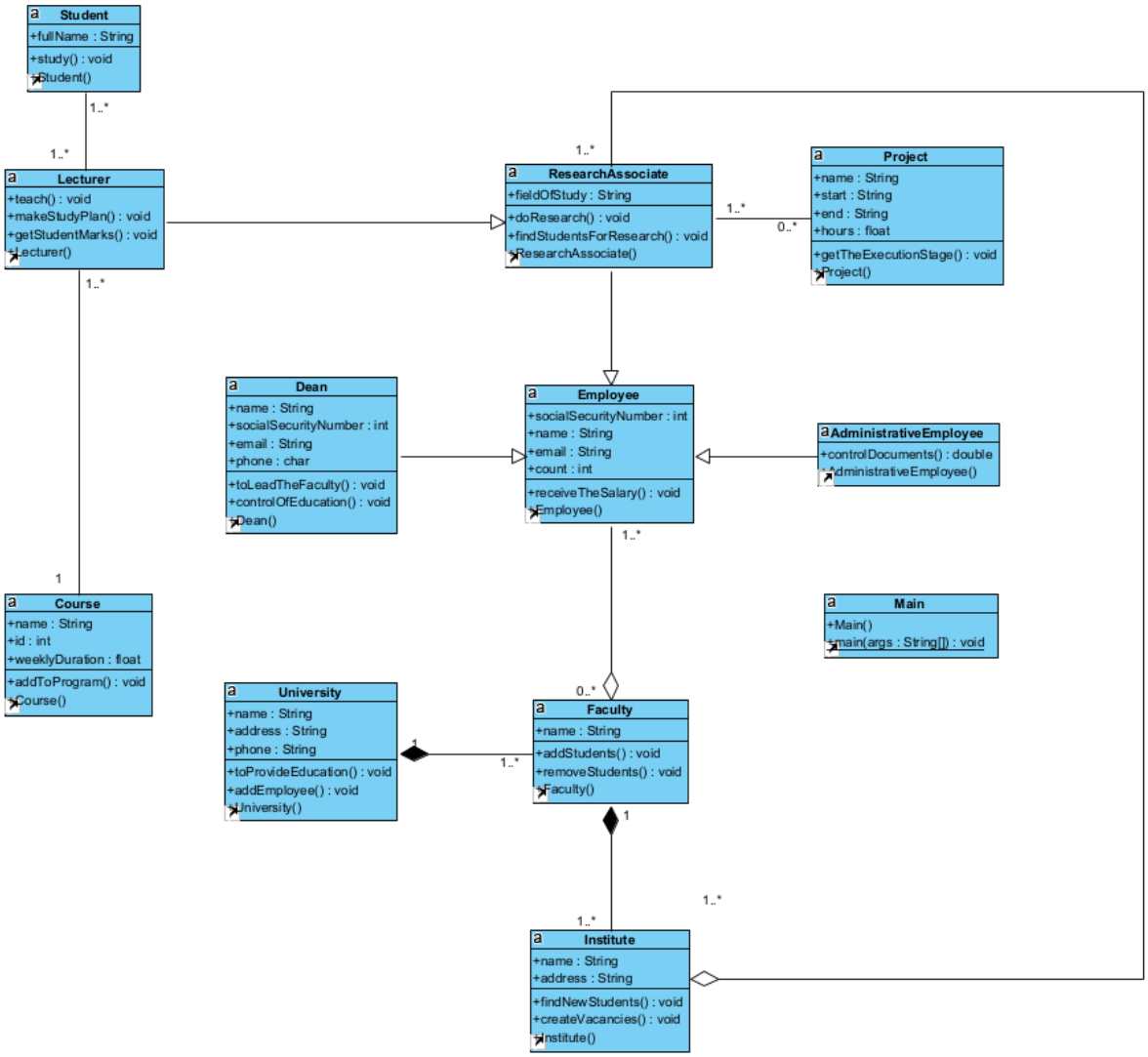
```java
    public void getStudentMarks() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter the number of students:");
        int numberOfStudents = scanner.nextInt();

        List<String> studentNames = new ArrayList<>();
        List<Integer> studentGrades = new ArrayList<>();
        List<String> gradeCategories = new ArrayList<>();

        for (int i = 0; i < numberOfStudents; i++) {
            System.out.println("Enter the name of student " + (i + 1) + ":");
            studentNames.add(scanner.next());

            System.out.println("Enter the numerical grade of student " + (i + 1) + ":");
            int studentGrade = scanner.nextInt();
            studentGrades.add(studentGrade);

            String gradeCategory;
            if (studentGrade < 50) {
                gradeCategory = "Fail";
            } else if (studentGrade <= 69) {
                gradeCategory = "Pass";
            } else if (studentGrade <= 89) {
                gradeCategory = "Good";
            } else if (studentGrade <= 100) {
                gradeCategory = "Excellent";
            } else {
                gradeCategory = "Invalid Grade";
            }

            gradeCategories.add(gradeCategory);
        }

        System.out.println("\nGrade Statistics:");

        for (int i = 0; i < numberOfStudents; i++) {
            System.out.println(studentNames.get(i) + ": " + studentGrades.get(i) + " - " + gradeCategories.get(i));
        }
    }
}
```

```java
package Diagramm;

public class Main {

    public static void main(String[] args) {
        Lecturer lecturer = new Lecturer();
        lecturer.getStudentMarks();
    }
}
```

Пример ввода и вывода:



Генерация модели:



Use case:

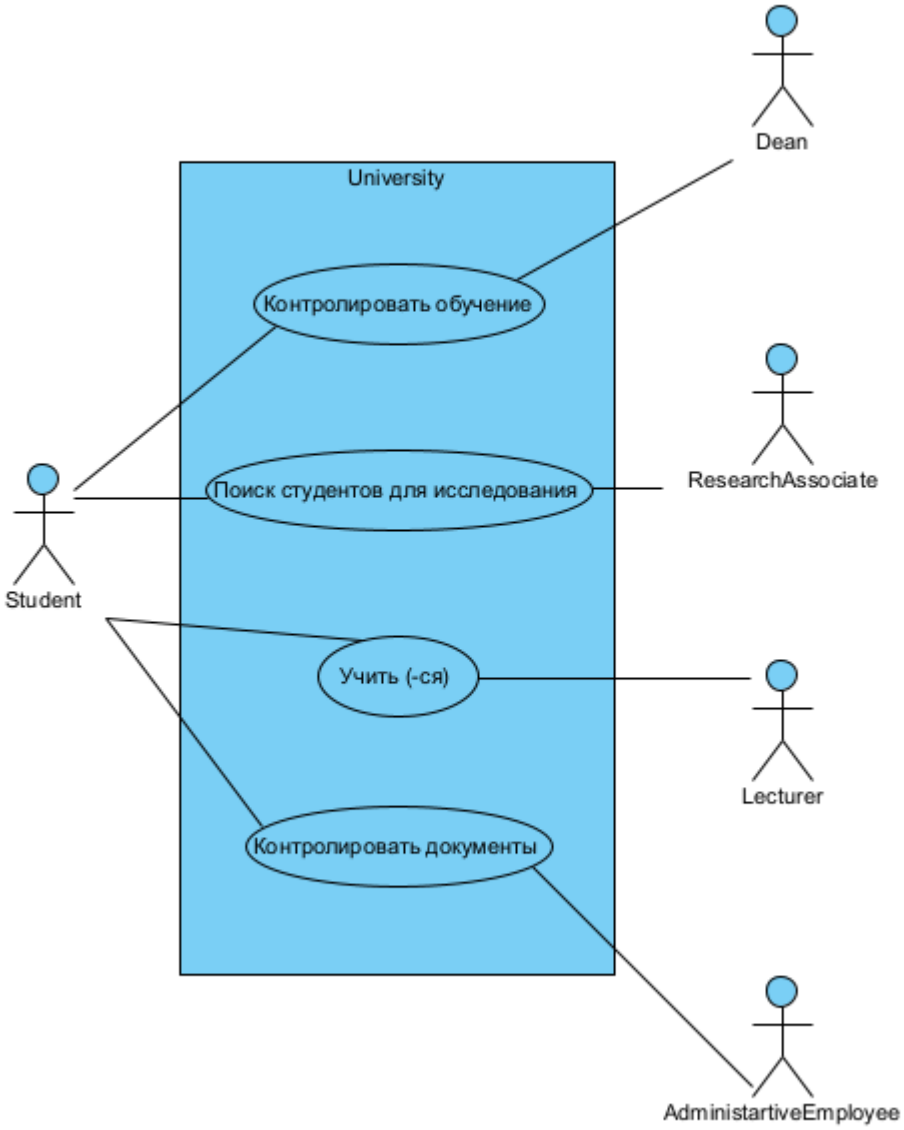

Dean
University
Контролировать обучение
ResearchAssociate
Поиск студентов для исследования
Student
Учить (-ся)
Lecturer
Контролировать документы
AdministartiveEmployee

Диаграмма последовательности обучения:

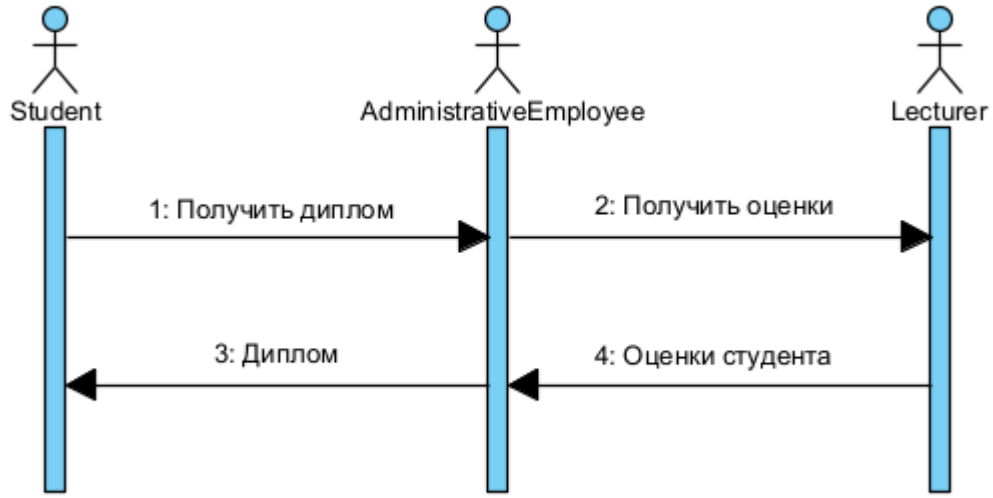


Диаграмма активностей: