

Диаграмма классов (Visual Paradigm).

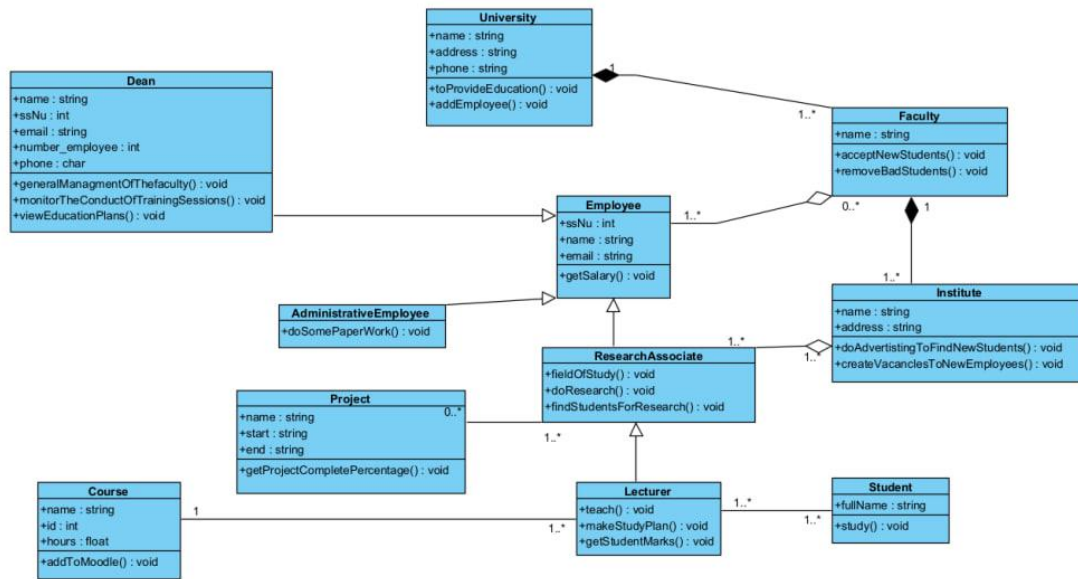
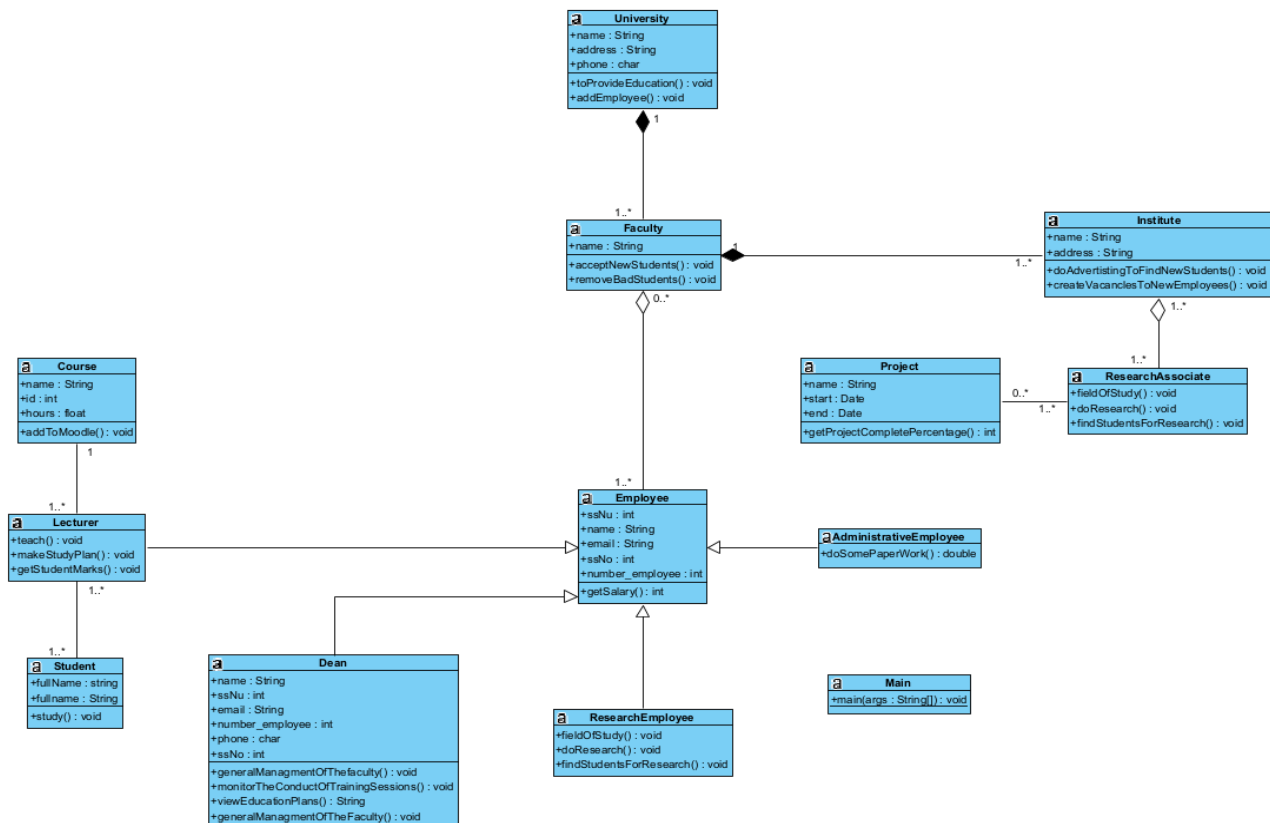


Диаграмма классов, с помощью кода (Visual Paradigm).



Use case.

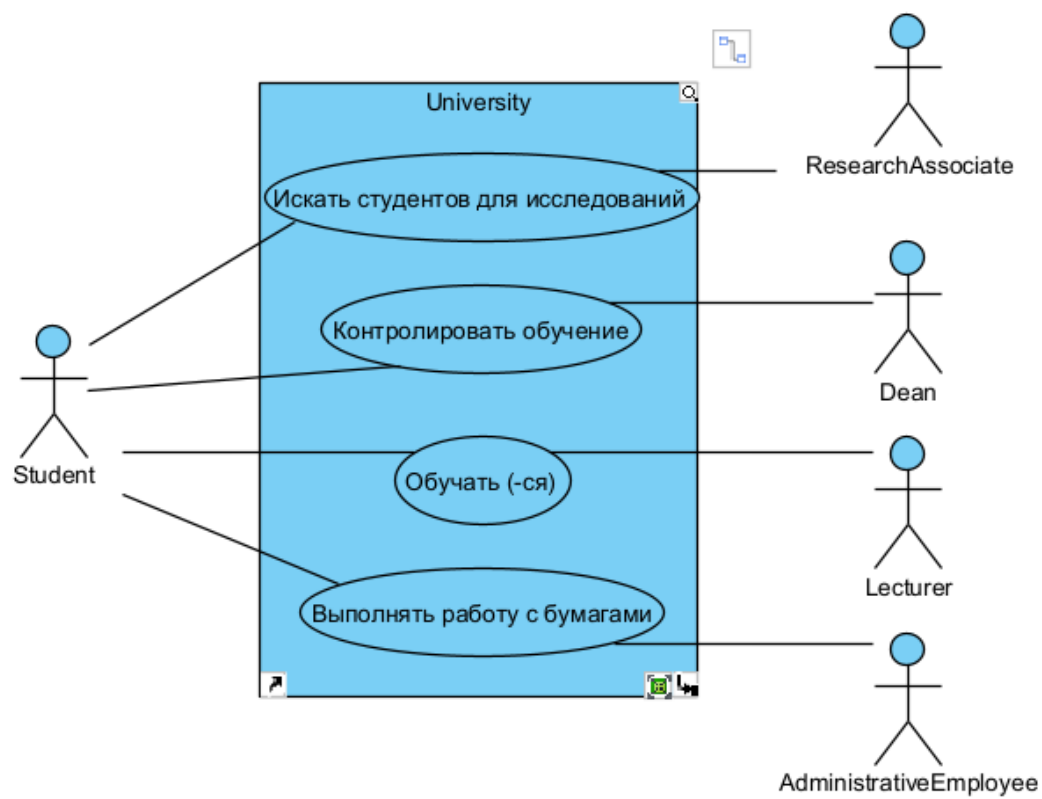


Диаграмма последовательности обучения.

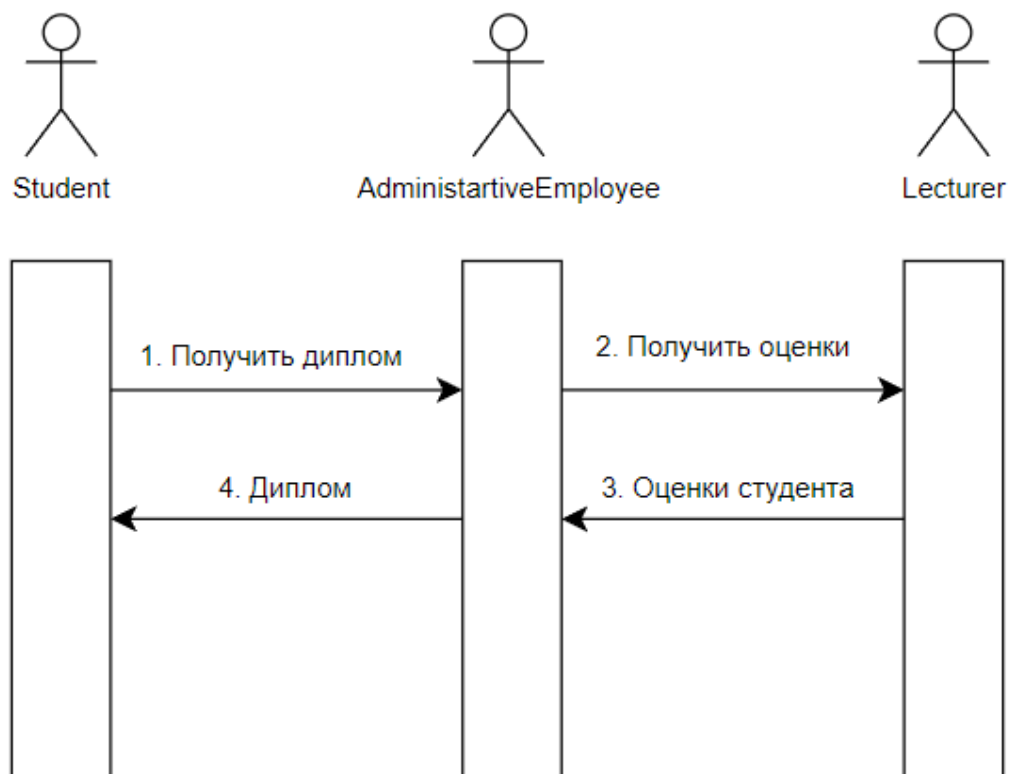
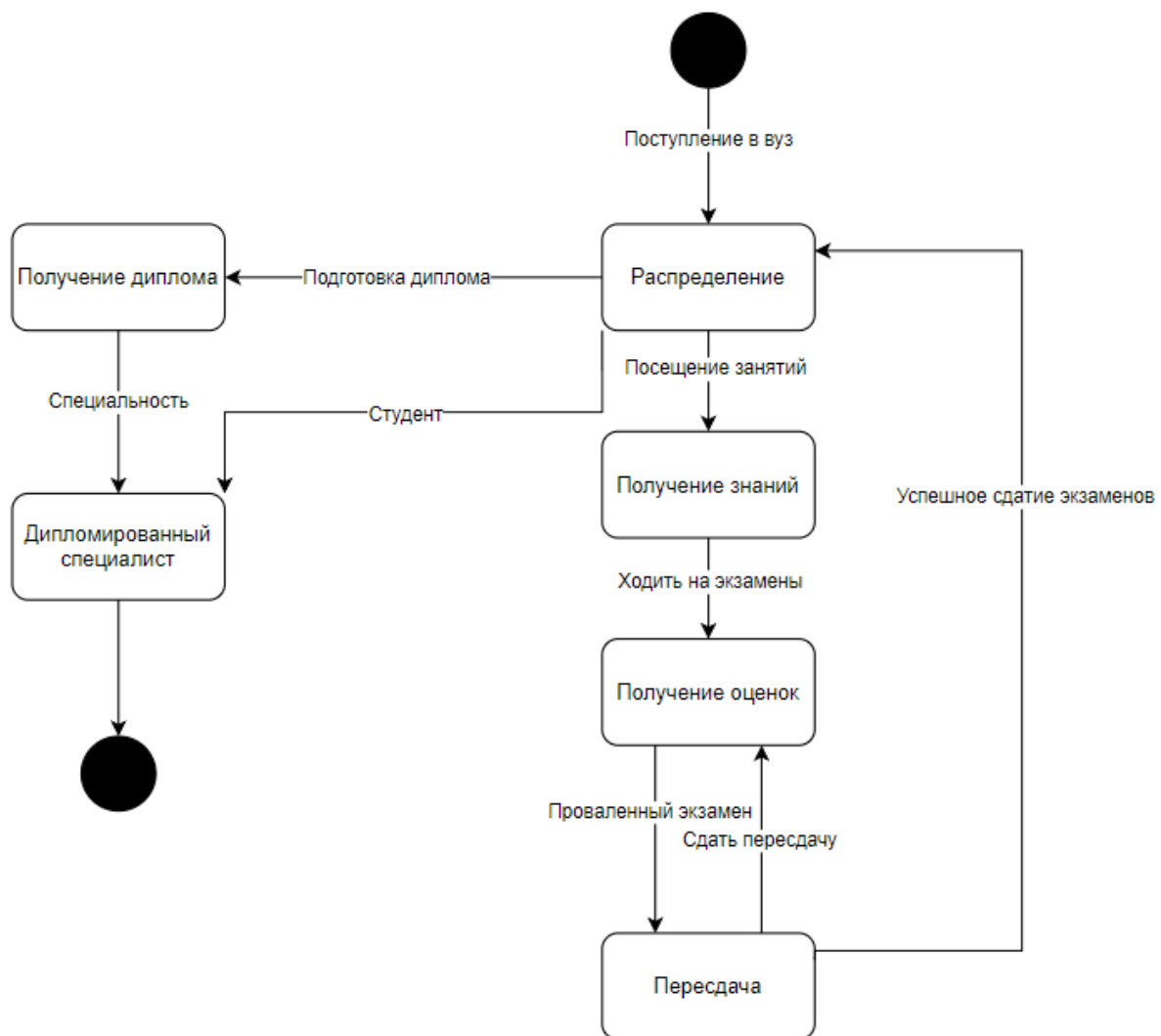


Диаграмма состояний.



Код и результат компиляции.

```
no usages
public class Course {

    no usages
    public String name;|
    no usages
    public int id;
    no usages
    public float hours;

    no usages
    public void addToMoodle() {
        // TODO - implement Course.addToMoodle
        throw new UnsupportedOperationException();
    }
}
```

no usages

```
public class Dean extends Employee {
```

no usages

```
public String name;
```

no usages

```
public int ssNu;
```

no usages

```
public String email;
```

no usages

```
public int number_employee;
```

no usages

```
public char phone;
```

no usages

```
public void generalManagmentOfThefaculty() {
```

```
    // TODO - implement Dean.generalManagmentOfThefaculty
```

```
    throw new UnsupportedOperationException();
```

```
}
```

no usages

```
public void monitorTheConductOfTrainingSessions() {
```

```
    // TODO - implement Dean.monitorTheConductOfTrainingSessions
```

```
    throw new UnsupportedOperationException();
```

```
}
```

no usages

```
public void viewEducationPlans() {
```

```
    // TODO - implement Dean.viewEducationPlans
```

```
    throw new UnsupportedOperationException();
```

```
}
```

3 usages 4 inheritors

```
public class Employee {
```

no usages

```
public int ssNu;
```

no usages

```
public String name;
```

no usages

```
public String email;
```

no usages

```
public void getSalary() {
```

```
    // TODO - implement Employee.getSalary
```

```
    throw new UnsupportedOperationException();
```

```
}
```

```
}
```

no usages

```
public class Faculty {
```



no usages

```
    public String name;
```

no usages

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

no usages

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

```
}
```


no usages

```
public class Institute {
```

no usages

```
    public String name;
```

no usages

```
    public String address;
```

no usages

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

no usages

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

```
}
```

no usages

```
public class Lecturer extends ResearchAssociate {
```



no usages

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

no usages

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

no usages

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

```
}
```

no usages

```
public class Project {
```

no usages

```
    public String name;
```

no usages

```
    public String start;
```

no usages

```
    public String end;
```

no usages

```
    public void getProjectCompletePercentage() {
```

```
        // TODO - implement Project.getProjectCompletePercentage
```

```
        throw new UnsupportedOperationException();
```

```
    }
```

```
}
```

1 usage 1 inheritor

```
public class ResearchAssociate extends Employee {
```



no usages

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

no usages

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

no usages

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

```
}
```

no usages

```
public class Student {
```



no usages

```
public String fullName;
```

no usages

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

```
}
```

no usages

```
public class University {
```

no usages

```
    public String name;
```

no usages

```
    public String address;
```



no usages

```
    public String phone;|
```

no usages

```
    public void toProvideEducation() {
```

```
        // TODO - implement University.toProvideEducation
```

```
        throw new UnsupportedOperationException();
```

```
    }
```

no usages

```
    public void addEmployee() {
```

```
        // TODO - implement University.addEmployee
```

```
        throw new UnsupportedOperationException();
```

```
    }
```

```
}
```

```

import java.util.Scanner;

no usages
public class AdministrativeEmployee extends Employee {

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

1 usage
public double doSomePaperWork(){
    Scanner scanner = new Scanner(System.in);

    System.out.println("Введите количество листов для работы - ");
    int list = scanner.nextInt();

    System.out.println("Введите количество листов обрабатываемых в час - ");
    int speed = scanner.nextInt();

    return (double) list / speed;
}
}

```

```

C:\Users\Erop\.jdk\openjdk-21.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2023.2.5\lib\idea_rt.jar=50657:C:\Program Files\JetBrains\IntelliJ IDEA 2
Введите количество листов для работы -
10
Введите количество листов обрабатываемых в час -
2
Количество часов на бумажную работу = 5.0 часов.

Process finished with exit code 0

```