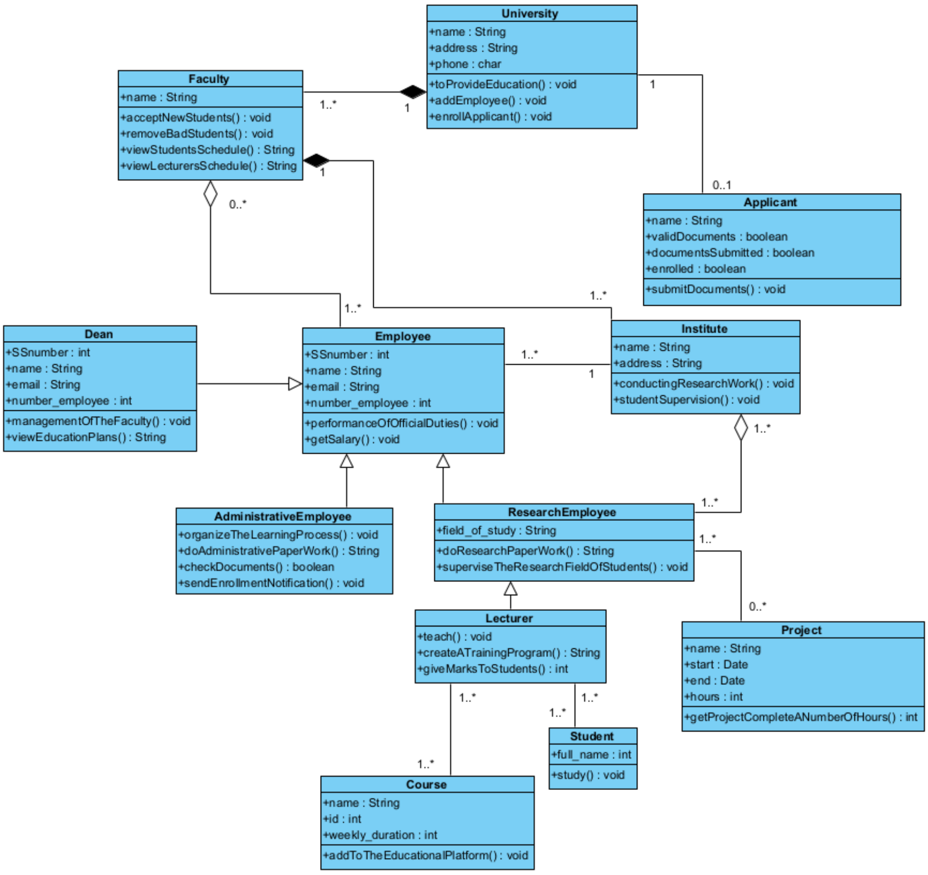
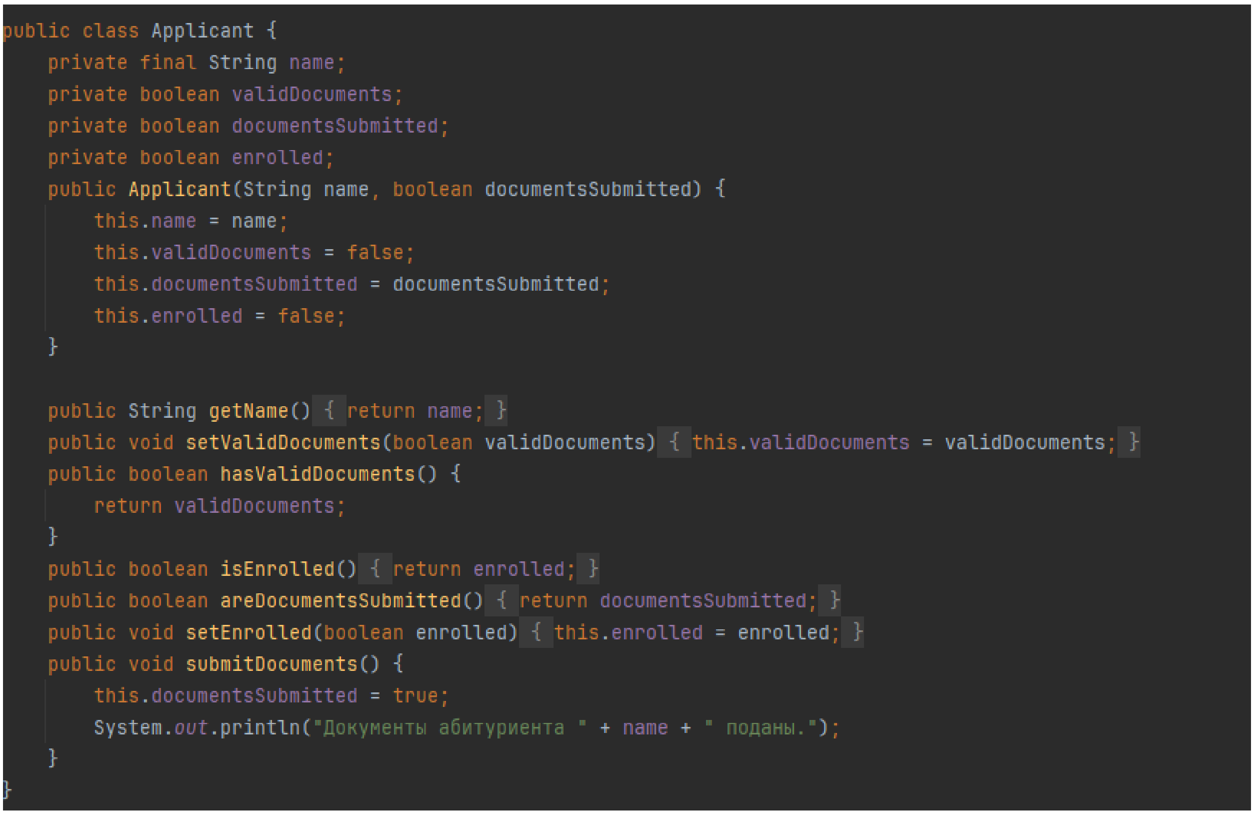
Модель:


University
+name : String
+address : String
+phone : char
+toProvideEducation() : void
+addEmployee() : void
+enrollApplicant() : void
Faculty
+name : String
+acceptNewStudents() : void
+removeBadStudents() : void
+viewStudentsSchedule() : String
+viewLecturersSchedule() : String
Applicant
+name : String
+validDocuments : boolean
+documentsSubmitted : boolean
+enrolled : boolean
+submitDocuments() : void
Dean
+SSnumber : int
+name : String
+email : String
+number_employee : int
+managementOfTheFaculty() : void
+viewEducationPlans() : String
Employee
+SSnumber : int
+name : String
+email : String
+number_employee : int
+performanceOfOfficialDuties() : void
+getSalary() : void
Institute
+name : String
+address : String
+conductingResearchWork() : void
+studentSupervision() : void
AdministrativeEmployee
+organizeTheLearningProcess() : void
+doAdministrativePaperWork() : String
+checkDocuments() : boolean
+sendEnrollmentNotification() : void
ResearchEmployee
+field_of_study : String
+doResearchPaperWork() : String
+superviseTheResearchFieldOfStudents() : void
Lecturer
+teach() : void
+createATrainingProgram() : String
+giveMarksToStudents() : int
Project
+name : String
+start : Date
+end : Date
+hours : int
+getProjectCompleteANumberOfHours() : int
Student
+full_name : int
+study() : void
Course
+name : String
+id : int
+weekly_duration : int
+addToTheEducationalPlatform() : void
1..*
1
1
0..1
1
0..*
1..*
1..*
1..*
1
1..*
1..*
1..*
0..*
1..*
1..*
1..*
1..*

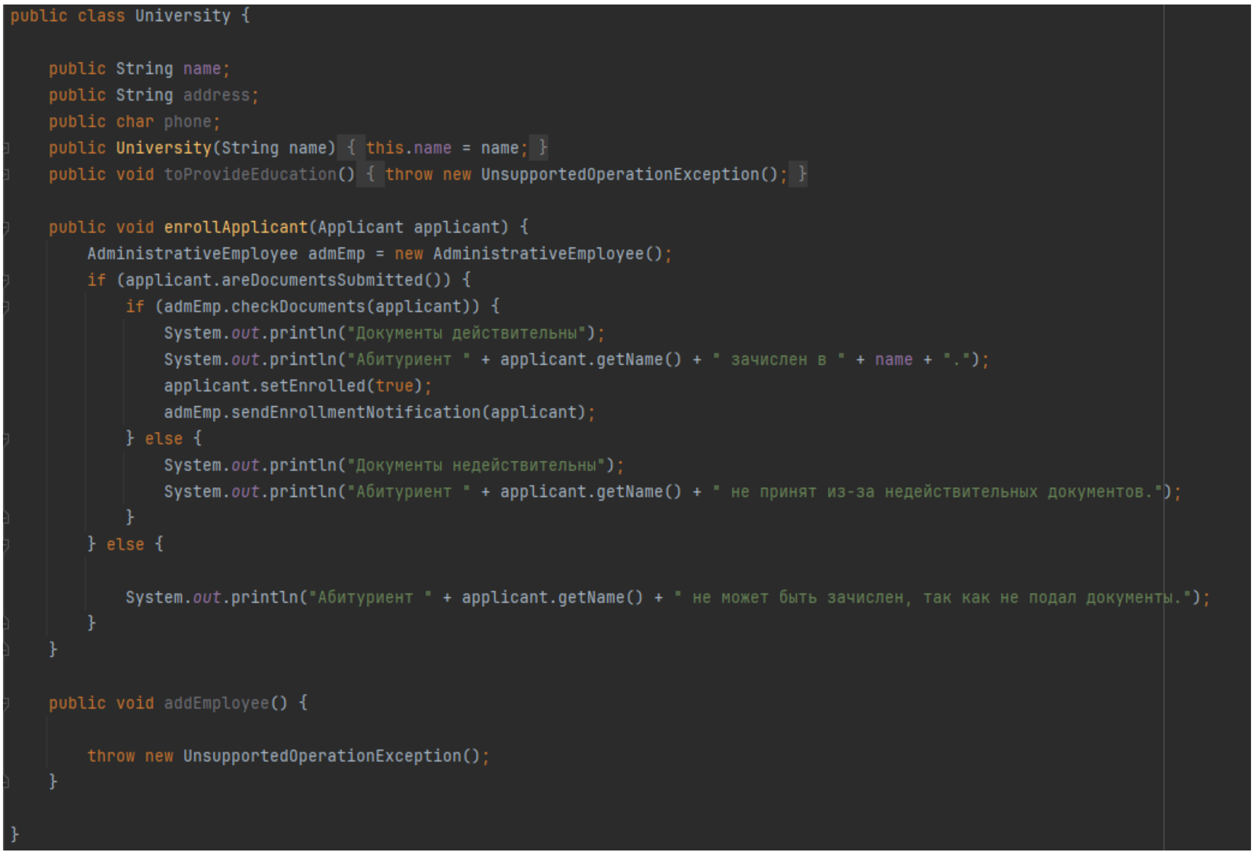
Код:

Класс Applicant



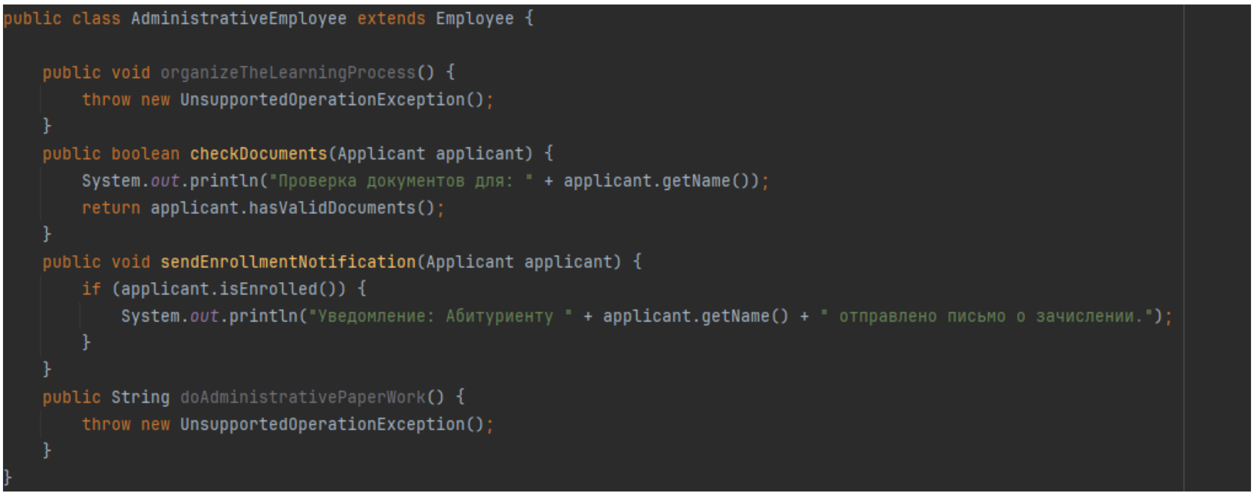
```
public class Applicant {
    private final String name;
    private boolean validDocuments;
    private boolean documentsSubmitted;
    private boolean enrolled;
    public Applicant(String name, boolean documentsSubmitted) {
        this.name = name;
        this.validDocuments = false;
        this.documentsSubmitted = documentsSubmitted;
        this.enrolled = false;
    }

    public String getName() { return name; }
    public void setValidDocuments(boolean validDocuments) { this.validDocuments = validDocuments; }
    public boolean hasValidDocuments() {
        return validDocuments;
    }
    public boolean isEnrolled() { return enrolled; }
    public boolean areDocumentsSubmitted() { return documentsSubmitted; }
    public void setEnrolled(boolean enrolled) { this.enrolled = enrolled; }
    public void submitDocuments() {
        this.documentsSubmitted = true;
        System.out.println("Документы абитуриента " + name + " поданы.");
    }
}
```

Класс University



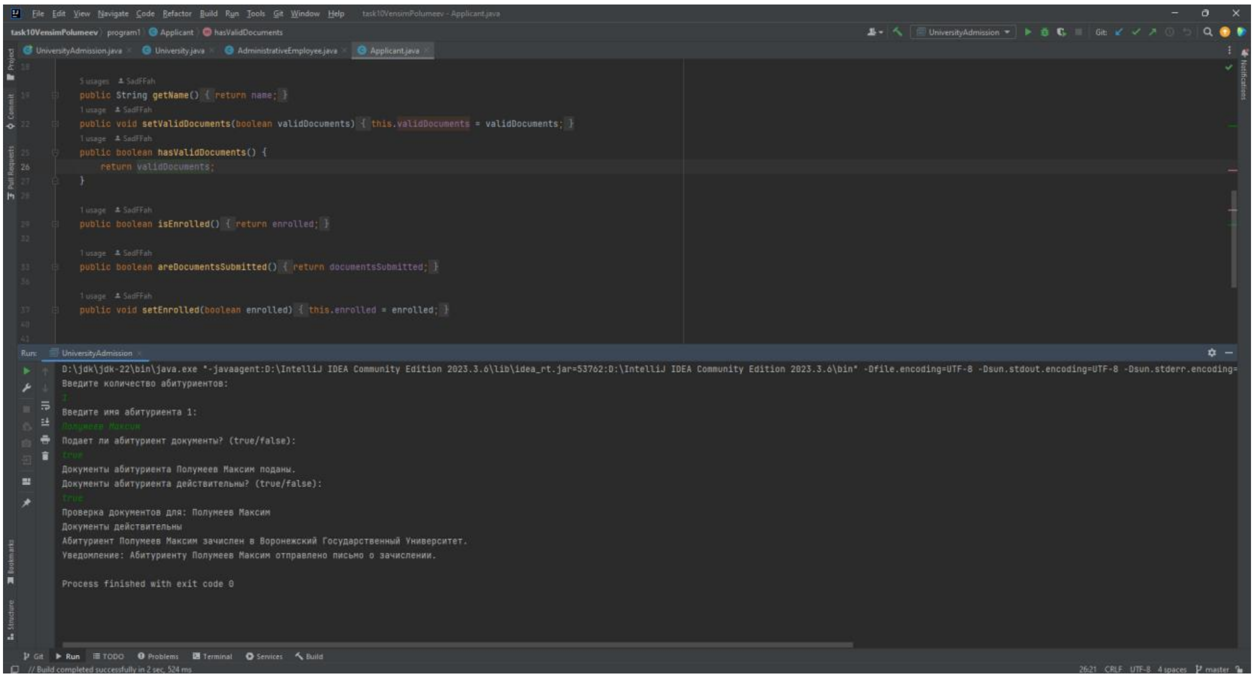
```
public class University {

    public String name;
    public String address;
    public char phone;
    public University(String name) { this.name = name; }
    public void toProvideEducation() { throw new UnsupportedOperationException(); }

    public void enrollApplicant(Applicant applicant) {
        AdministrativeEmployee admEmp = new AdministrativeEmployee();
        if (applicant.areDocumentsSubmitted()) {
            if (admEmp.checkDocuments(applicant)) {
                System.out.println("Документы действительны");
                System.out.println("Абитуриент " + applicant.getName() + " зачислен в " + name + ".");
                applicant.setEnrolled(true);
                admEmp.sendEnrollmentNotification(applicant);
            } else {
                System.out.println("Документы недействительны");
                System.out.println("Абитуриент " + applicant.getName() + " не принят из-за недействительных документов.");
            }
        } else {

            System.out.println("Абитуриент " + applicant.getName() + " не может быть зачислен, так как не подал документы.");
        }
    }

    public void addEmployee() {

        throw new UnsupportedOperationException();
    }

}
```

Класс AdministrativeEmployee



```
public class AdministrativeEmployee extends Employee {

    public void organizeTheLearningProcess() {
        throw new UnsupportedOperationException();
    }
    public boolean checkDocuments(Applicant applicant) {
        System.out.println("Проверка документов для: " + applicant.getName());
        return applicant.hasValidDocuments();
    }
    public void sendEnrollmentNotification(Applicant applicant) {
        if (applicant.isEnrolled()) {
            System.out.println("Уведомление: Абитуриенту " + applicant.getName() + " отправлено письмо о зачислении.");
        }
    }
    public String doAdministrativePaperWork() {
        throw new UnsupportedOperationException();
    }
}
```

Скриншот выполнения


File Edit View Navigate Code Refactor Build Run Tools Git Window Help task10VensimPolumeev - Applicant.java
task10VensimPolumeev program1 Applicant hasValidDocuments
UniversityAdmission
UniversityAdmission.java
University.java
AdministrativeEmployee.java
Applicant.java
Project
Commit
Pull Requests
Bookmarks
Structure
Notifications
5 usages SadFFah
1 usage SadFFah
Run: UniversityAdmission
Git
Run
TODO
Problems
Terminal
Services
Build
// Build completed successfully in 2 sec, 524 ms
26:21 CRLF UTF-8 4 spaces master

```java
    public String getName() { return name; }
    public void setValidDocuments(boolean validDocuments) { this.validDocuments = validDocuments; }
    public boolean hasValidDocuments() {
        return validDocuments;
    }

    public boolean isEnrolled() { return enrolled; }

    public boolean areDocumentsSubmitted() { return documentsSubmitted; }

    public void setEnrolled(boolean enrolled) { this.enrolled = enrolled; }
```

```
D:\jdk\jdk-22\bin\java.exe "-javaagent:D:\IntelliJ IDEA Community Edition 2023.3.6\lib\idea_rt.jar=53762:D:\IntelliJ IDEA Community Edition 2023.3.6\bin" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=
Введите количество абитуриентов:
1
Введите имя абитуриента 1:
Полумеев Максим
Подает ли абитуриент документы? (true/false):
true
Документы абитуриента Полумеев Максим поданы.
Документы абитуриента действительны? (true/false):
true
Проверка документов для: Полумеев Максим
Документы действительны
Абитуриент Полумеев Максим зачислен в Воронежский Государственный Университет.
Уведомление: Абитуриенту Полумеев Максим отправлено письмо о зачислении.

Process finished with exit code 0
```