Practiceweek12

```
// I create a Student class to represent student data. Each object of this class will
//store information about one student.
public class Student {
   private String name;
   private double currentScholarship;
   private double newScholarship;
 // These variables store information about the student:
// name — student's name.
//currentScholarship - the student's current scholarship.
//newScholarship is a new student scholarship.
//They are declared private to protect the data from direct changes from outside.
   public Student(String name, double currentScholarship, double newScholarship) {
      this.name = name;
      this.currentScholarship = currentScholarship;
      this.newScholarship = newScholarship;
   }
//The constructor is used to create objects of the Student class.
//When i create a new student, we pass three parameters:
//name — student's name.
//currentScholarship - the amount of the current scholarship.
//newScholarship — the amount of the new scholarship.
//These values are assigned to object variables using this.

   public String getName() {
      return name;
   }
 // This method returns the value of the name variable (student name).
```

```
    public double getCurrentScholarship() {
        return currentScholarship;
    }// This method returns the current scholarship amount.

    public double getNewScholarship() {
        return newScholarship;
    }// This method returns the amount of the new scholarship.

    public double getScholarshipIncrease() {
        return newScholarship - currentScholarship;
    }
}// This method returns the difference between the new and current scholarships.
//Used to immediately get the increase, instead of calculating it manually each time.
```

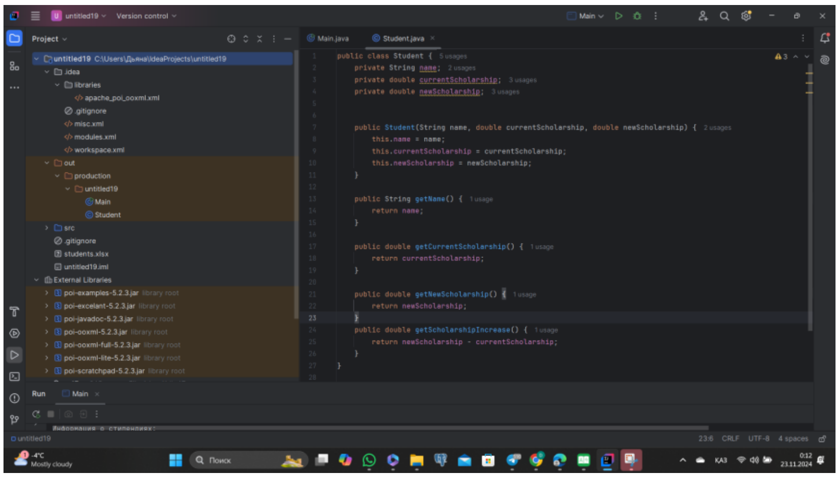
Target

This class is designed to store and process student data:

Stores information (name, current and new scholarships).

Provides access to this data through methods.

Automatically calculates scholarship growth.



```java
public class Student {
    private String name;
    private double currentScholarship;
    private double newScholarship;


    public Student(String name, double currentScholarship, double newScholarship) {
        this.name = name;
        this.currentScholarship = currentScholarship;
        this.newScholarship = newScholarship;
    }

    public String getName() {
        return name;
    }

    public double getCurrentScholarship() {
        return currentScholarship;
    }

    public double getNewScholarship() {
        return newScholarship;
    }
    public double getScholarshipIncrease() {
        return newScholarship - currentScholarship;
    }
}
```

```java
import org.apache.poi.ss.usermodel.*;

import org.apache.poi.xssf.usermodel.XSSFWorkbook;


import java.io.File;

import java.io.FileInputStream;

import java.util.ArrayList;

import java.util.List;
 // Why are these libraries needed:

//org.apache.poi.ss.usermodel.*: Used to work with Excel files. Provides classes for
//reading data from rows and cells.

//org.apache.poi.xssf.usermodel.XSSFWorkbook: Allows you to work with .xlsx files.

//java.io.*: Needed to work with files on your computer.

//java.util.*: Used to create a list of students (ArrayList).

public class Main {

    public static void main(String[] args) {

        List<Student> students = new ArrayList<>(); //I created a students list to store data
//about all students. This list is populated with objects of the Student class


        try (FileInputStream fis = new FileInputStream(new File("students.xlsx"));

             Workbook workbook = new XSSFWorkbook(fis)) {

 // I'm using FileInputStream to open an Excel file with student data.
//The Workbook class allows you to work with this file and retrieve data.
      Sheet sheet = workbook.getSheetAt(0);


      for (int i = 1; i <= sheet.getLastRowNum(); i++) {
        Row row = sheet.getRow(i);
 // First, I get the first sheet from the Excel file, since it contains the necessary data.
//The loop starts from the second row (i = 1) to skip the table header.
        String name = row.getCell(0).getStringCellValue();
        double currentScholarship = row.getCell(1).getNumericCellValue();
        double newScholarship = row.getCell(2).getNumericCellValue();
 // From each line I read:
//Student name (row.getCell(0)).
//Current scholarship (row.getCell(1)).
//New scholarship (row.getCell(2)).
        students.add(new Student(name, currentScholarship, newScholarship));
      }// For each student, I create an object of the Student class and add it to the
students list.
This allows you to conveniently store and process information about each student.
    } catch (Exception e) {
      e.printStackTrace();
    }// If something goes wrong, the program displays an error in the console. This is
necessary to make it easier to find problems.


    displayScholarshipInfo(students);
  }// After processing all the data, I call the displayScholarshipInfo method to display
the student information.


  private static void displayScholarshipInfo(List<Student> students) {
    System.out.println("Scholarship Information:");
```

```
        for (Student student : students) {
            System.out.println("Name: " + student.getName());
            System.out.println("Current Scholarship: " + student.getCurrentScholarship());
            System.out.println("New Scholarship: " + student.getNewScholarship());
            System.out.println("Increase: " + student.getScholarshipIncrease());
            System.out.println("-----------------------------------");
        }
    }
}
```

In this method I iterate through the list of students.

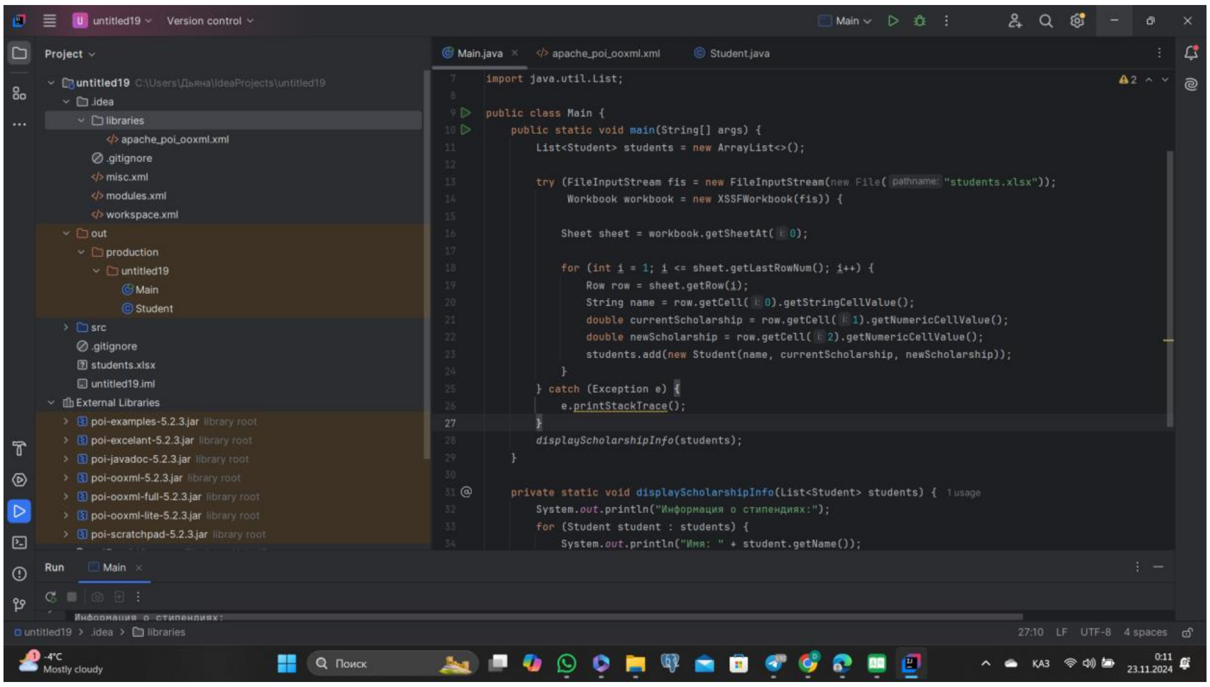
For each student I output:

Name.

Current scholarship.

New scholarship.

Scholarship increase, which is calculated by the getScholarshipIncrease() method in the Student class.



```
import java.util.List;

public class Main {
    public static void main(String[] args) {
        List<Student> students = new ArrayList<>();

        try (FileInputStream fis = new FileInputStream(new File( pathname: "students.xlsx"));
             Workbook workbook = new XSSFWorkbook(fis)) {

            Sheet sheet = workbook.getSheetAt(0);

            for (int i = 1; i <= sheet.getLastRowNum(); i++) {
                Row row = sheet.getRow(i);
                String name = row.getCell(0).getStringCellValue();
                double currentScholarship = row.getCell(1).getNumericCellValue();
                double newScholarship = row.getCell(2).getNumericCellValue();
                students.add(new Student(name, currentScholarship, newScholarship));
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        displayScholarshipInfo(students);
    }

    private static void displayScholarshipInfo(List<Student> students) {
        System.out.println("Информация о стипендиях:");
        for (Student student : students) {
            System.out.println("Имя: " + student.getName());
```

```
Информация о стипендиях:
```