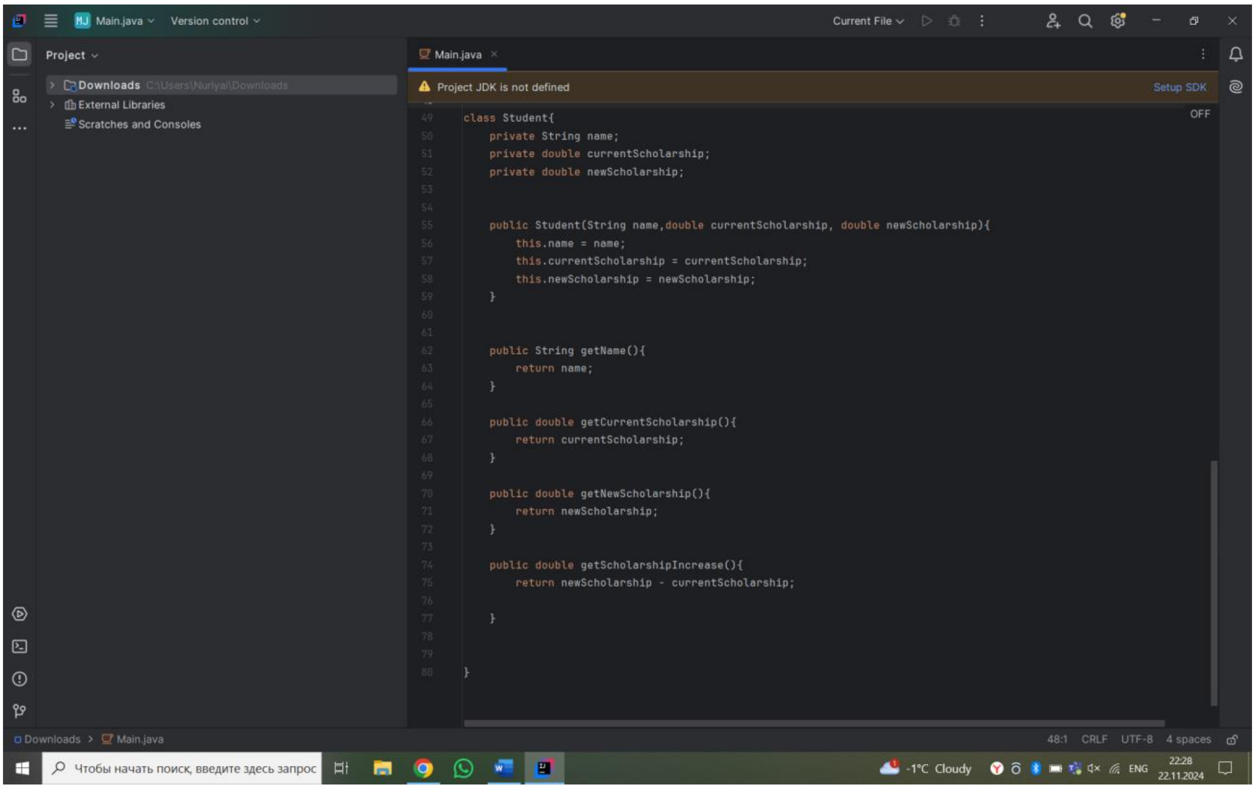
37851

Topic: Reading and writing Excel files using Apache POI

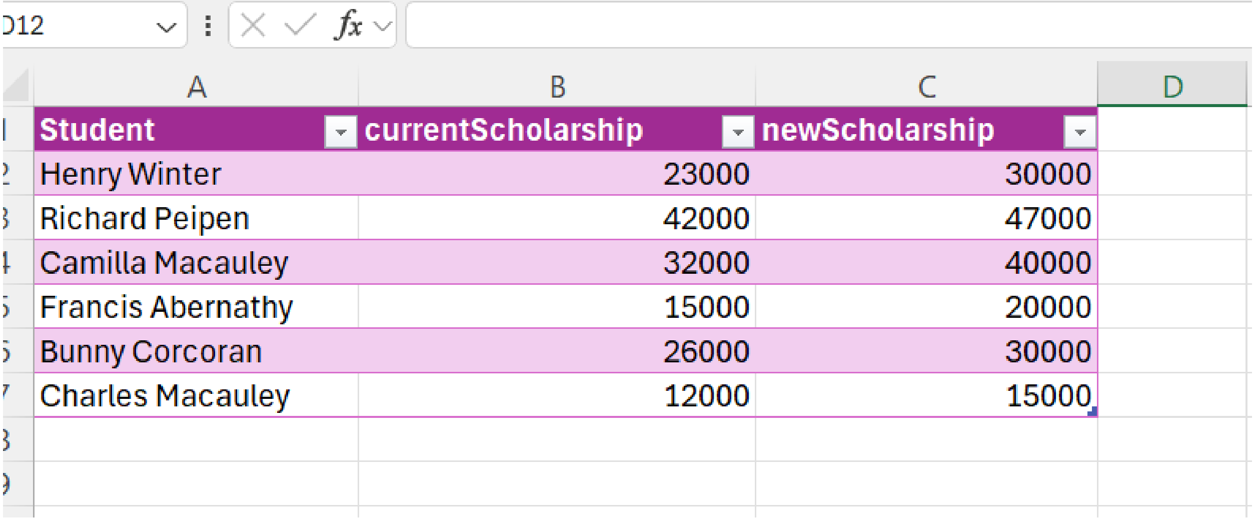
Explaining of my code (I downloaded my code from my computer so it's not defined):

1.First, I created a class Student whith the following fields: name (String), currentScholarship (double), newScholarship (double). Added getters, constructor and method getScholarshipIncrease (double).



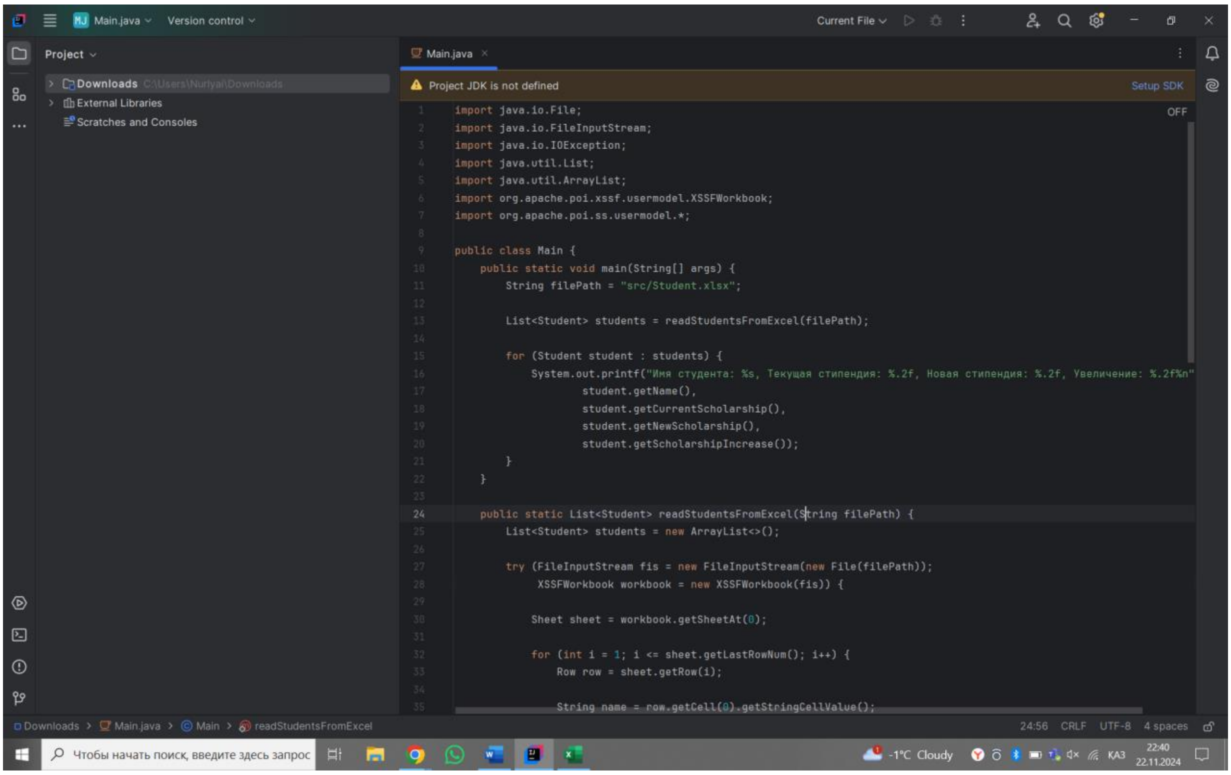
```java
class Student{
    private String name;
    private double currentScholarship;
    private double newScholarship;


    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;

    }


}
```

2. Then I created a sheet in Excel and named it Student. I entered the data I needed into the table, that is the names of the students and their scholarships.



Student	currentScholarship	newScholarship
Henry Winter	23000	30000
Richard Peipen	42000	47000
Camilla Macauley	32000	40000
Francis Abernathy	15000	20000
Bunny Corcoran	26000	30000
Charles Macauley	12000	15000

3.In this step I created Main class, where we can read data from students.xlsx file.


Main.java
Version control
Project
Downloads C:\Users\Nurlyal\Downloads
External Libraries
Scratches and Consoles
Current File
Main.java
Setup SDK
OFF
Downloads > Main.java > Main > readStudentsFromExcel
24:56 CRLF UTF-8 4 spaces
Чтобы начать поиск, введите здесь запрос
-1°C Cloudy
КАЗ
22:40
22.11.2024

Project JDK is not defined

```java
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.util.List;
import java.util.ArrayList;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;
import org.apache.poi.ss.usermodel.*;

public class Main {
    public static void main(String[] args) {
        String filePath = "src/Student.xlsx";

        List<Student> students = readStudentsFromExcel(filePath);

        for (Student student : students) {
            System.out.printf("Имя студента: %s, Текущая стипендия: %.2f, Новая стипендия: %.2f, Увеличение: %.2f%n"
                    student.getName(),
                    student.getCurrentScholarship(),
                    student.getNewScholarship(),
                    student.getScholarshipIncrease());
        }
    }

    public static List<Student> readStudentsFromExcel(String filePath) {
        List<Student> students = new ArrayList<>();

        try (FileInputStream fis = new FileInputStream(new File(filePath));
             XSSFWorkbook 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();
```

```java
String filePath = "src/Student.xlsx";
```

Here I created a string variable filePath and assigned it to the path to my Excel file that contains the students' data.

```java
List<Student> students = readStudentsFromExcel(filePath);
```

This call calls the readStudentsFromExcel() method, which reads data from the Excel file and returns a list of Student objects, which we store in the students variable.

```java
    for (Student student : students) {
        System.out.printf("Имя студента: %s, Текущая стипендия: %.2f, Новая
стипендия: %.2f, Увеличение: %.2f%n",
                student.getName(),
                student.getCurrentScholarship(),
                student.getNewScholarship(),
                student.getScholarshipIncrease());
    }
}
```

Here I'm using a for-each loop to iterate through each student in the students list.

Thus, information about each student is displayed, cuz it returns from our Excel file

```java
public static List<Student> readStudentsFromExcel(String filePath) {
    List<Student> students = new ArrayList<>();
```

This method returns a list of students. We create an empty list of students, which will be filled with data. Also, list<Student> is an interface for working with lists that is implemented by the ArrayList class.

```java
try (FileInputStream fis = new FileInputStream(new File(filePath));
     XSSFWorkbook workbook = new XSSFWorkbook(fis)) {

    Sheet sheet = workbook.getSheetAt(0);
```

This line creates a FileInputStream to read the Excel file. The path to the file is passed through the filePath parameter.

XSSFWorkbook is a class from the Apache POI library designed to work with Excel (.xlsx) files.

We use try-with-resources, which ensures that I/O streams (such as fis and workbook) are closed when they are finished.

Then we get the first sheet from the Excel workbook. Starting from 0.

```
for (int i = 1; i <= sheet.getLastRowNum(); i++) {
    Row row = sheet.getRow(i);
```

The for loop starts at index 1 because the first row in Excel usually contains the column headers and we need to skip it.

sheet.getLastRowNum() returns the last row number to determine the limits of the loop.

We get each row from the sheet using sheet.getRow(i).

```
String name = row.getCell(0).getStringCellValue();
double currentScholarship = row.getCell(1).getNumericCellValue();
double newScholarship = row.getCell(2).getNumericCellValue();
```

row.getCell(0) is to get the value from the first cell (student name).

getStringCellValue() - method to get a string value.

getCell(1) and getCell(2) - Get values from the second and third cells (current and new scholarship).

getNumericCellValue() - method for getting a numeric value from a cell.

```
students.add(new Student(name, currentScholarship, newScholarship));
            }
        } catch (IOException e) {
            e.printStackTrace();
        }

        return students;
    }
}
```

Here a new Student object is created with the extracted data and it is added to the students list.

If there is an error while reading the file, an IOException will be thrown, which we handle by printing the error stack.

After processing all the rows from the Excel file, the method returns a list of students who were read from the file.