



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«МИРЭА – Российский технологический университет»

**РТУ МИРЭА**

---

---

**Институт информационных технологий (ИТ)**  
**Кафедра инструментального и прикладного программного обеспечения (ИиППО)**

**Дисциплина «Программирование на языке Джава»**

**ОТЧЕТ**  
**ПО ПРАКТИЧЕСКОМУ ЗАНЯТИЮ №6**

Выполнил студент группы ИНБО-02-20

Лукияненко Д.В.

Принял

Степанов П.В.

Практическая работа выполнена «\_\_» \_\_\_\_\_ 2021 г.

«\_\_\_\_\_»  
Отметка о выполнении

«\_\_» \_\_\_\_\_ 2021 г.

**Москва – 2021 г.**

## СОДЕРЖАНИЕ

Цель работы .....	3
Задание .....	3
Репозиторий .....	3
Выполнение работы .....	3
Код выполненной работы .....	5
Вывод.....	9

## Цель работы

Цель данной практической работы – Освоить на практике сортировки различными методами.

## Задание

### Задание 1.

Написать тестовый класс, который создает массив класса Student и сортирует массив iDNumber.

### Задание 2.

Напишите класс SortingStudentsByGPA который реализует интерфейс Comparator таким образом, что она сортирует студентов с их итоговым баллом в порядке убывания.

### Задание 3.

Напишите программу, которая объединяет два списка данных о студентах в один отсортированный список.

## Репозиторий

Ссылка: [https://github.com/neluckoff/mirea-java-lessons/tree/master/src/ru/luckoff/mirea/practice\\_6](https://github.com/neluckoff/mirea-java-lessons/tree/master/src/ru/luckoff/mirea/practice_6)

## Выполнение работы

Первым делом мною был создан класс Student, в который я добавил конструктор и геттеры с сетторами.

Первое задание я реализовал двумя способами: через сортировку вставками и через интерфейс Comparable при помощи compareTo.

Далее мной опять был создан класс под названием SortingStudentsByGPA, который реализует интерфейс Comparator и сортирует список в порядке убывания.

Потом я просто объединил два массива, созданные в процессе выполнения задания 1 и 2 и отсортировал их при помощи SortingStudentsByGPA.

Ниже мною будут представлены рисунки с кодом программы, результатом запуска и UML Диаграммой.

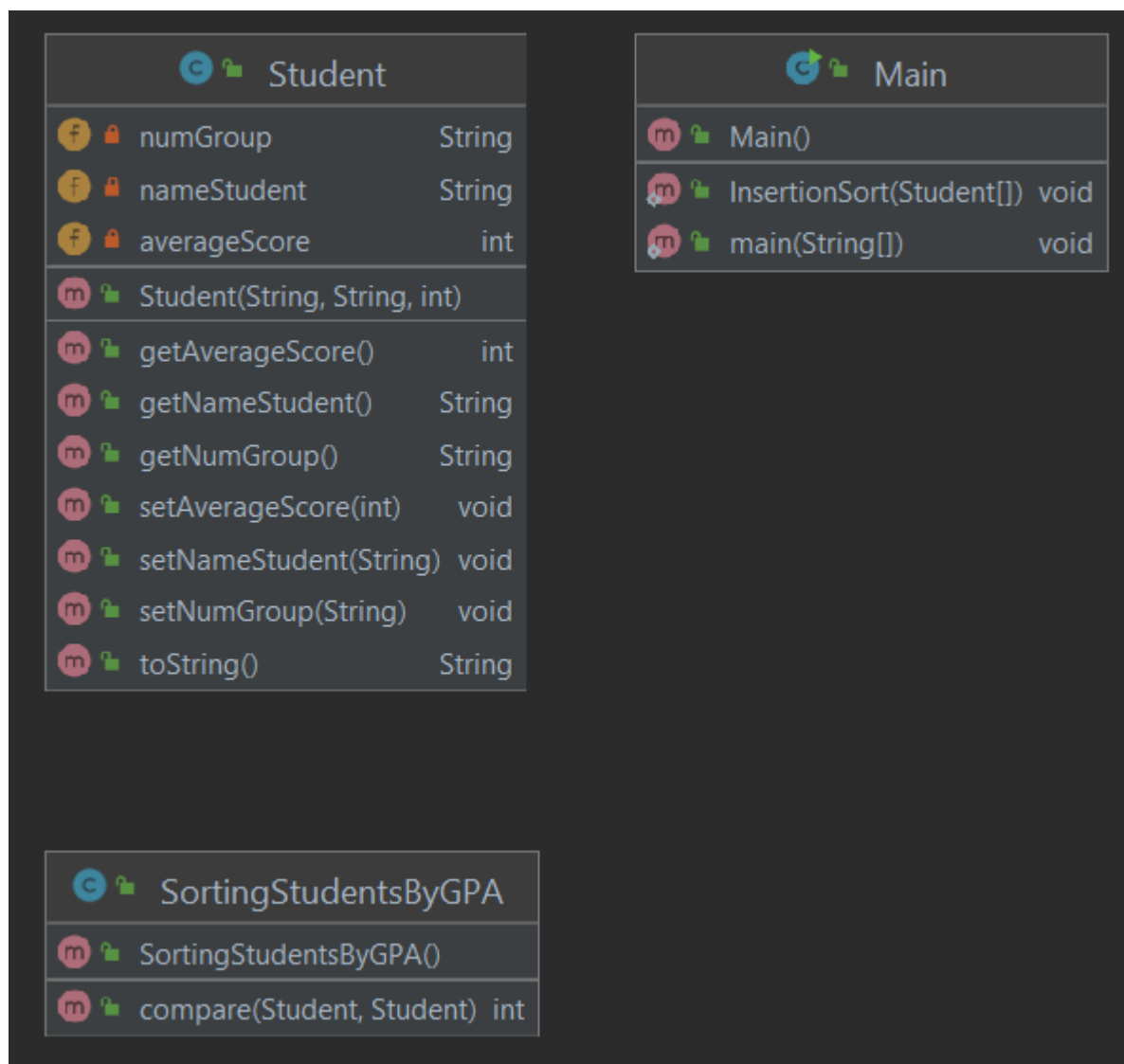


Рисунок 1 – UML Диаграмма

## Код выполненной работы

Здесь в нескольких скриншотах можно увидеть как выглядит код выполненного задания.

```
public class Student {
    private String numGroup;
    private String nameStudent;
    private int averageScore;

    public Student(String numGroup, String nameStudent, int averageScore) {
        this.numGroup = numGroup;
        this.nameStudent = nameStudent;
        this.averageScore = averageScore;
    }

    public String getNumGroup() { return numGroup; }

    public void setNumGroup(String numGroup) { this.numGroup = numGroup; }

    public String getNameStudent() { return nameStudent; }

    public void setNameStudent(String nameStudent) { this.nameStudent = nameStudent; }

    public int getAverageScore() { return averageScore; }

    public void setAverageScore(int averageScore) { this.averageScore = averageScore; }

    @Override
    public String toString() { return numGroup + " Name: " + nameStudent + " Score: " + averageScore + "\n"; }
}
```

Рисунок 2 – Класс Student

```
public static void main(String[] args) {
    //Task 1
    Student Student1 = new Student( numGroup: "INB0-02-20", nameStudent: "Tom Holland", averageScore: 4);
    Student Student2 = new Student( numGroup: "INB0-02-20", nameStudent: "Isaac Anderson", averageScore: 3);
    Student Student3 = new Student( numGroup: "INB0-02-20", nameStudent: "Hope Johnson", averageScore: 5);

    Student[] idNumber = new Student[]{Student1, Student2, Student3};
    System.out.println(Arrays.toString(idNumber));
    InsertionSort(idNumber);
    System.out.println(Arrays.toString(idNumber));
}
```

Рисунок 3 – Реализация мейна для первой задачи

```

public static void InsertionSort(Student[] idNumber) {
    for (int i = 0; i < idNumber.length - 1; i++) {
        for (int j = i + 1; j < idNumber.length; j++) {
            if (idNumber[i].getAverageScore() > idNumber[j].getAverageScore()) {
                Student temp = idNumber[i];
                idNumber[i] = idNumber[j];
                idNumber[j] = temp;
            }
        }
    }
}

```

Рисунок 4 – Сортировка вставкой

```

import java.util.Comparator;

public class SortingStudentsByGPA implements Comparator<Student> {
    @Override
    public int compare(Student o1, Student o2) { return o2.getAverageScore() - o1.getAverageScore(); }
}

```

Рисунок 5 – Класс SortingStudentByGPA

```

//Task 2
List<Student> ListTaskTwo = new ArrayList<>();
Student Student4 = new Student( numGroup: "INB0-02-20", nameStudent: "Manuel Barossa", averageScore: 5);
Student Student5 = new Student( numGroup: "INB0-02-20", nameStudent: "Slim Shady", averageScore: 3);
Student Student6 = new Student( numGroup: "INB0-02-20", nameStudent: "Pablo Escobar", averageScore: 2);
Student Student7 = new Student( numGroup: "INB0-02-20", nameStudent: "Miguel Morretti", averageScore: 5);

ListTaskTwo.add(Student4);
ListTaskTwo.add(Student5);
ListTaskTwo.add(Student6);
ListTaskTwo.add(Student7);

Comparator scoreComparator = new SortingStudentsByGPA();

Collections.sort(ListTaskTwo, scoreComparator);
System.out.println("Task 2\n" + ListTaskTwo);

```

Рисунок 6 – Мейн для второй задачи

```
//Task 3
List<Student> idNumber2 = new ArrayList<>();
idNumber2.add(Student1);
idNumber2.add(Student2);
idNumber2.add(Student3);
List<Student> realList = new ArrayList<>();
realList.addAll(idNumber2);
realList.addAll(ListTaskTwo);

Collections.sort(realList, scoreComparator);
System.out.println("Task 3\n" + realList);
```

Рисунок 7 – Мейн для третьей задачи

```
[INB0-02-20 Name: Tom Holland Score: 4
, INB0-02-20 Name: Isaac Anderson Score: 3
, INB0-02-20 Name: Hope Johnson Score: 5
]
[INB0-02-20 Name: Isaac Anderson Score: 3
, INB0-02-20 Name: Tom Holland Score: 4
, INB0-02-20 Name: Hope Johnson Score: 5
]
```

Рисунок 8 – Работоспособность первого задания

```
Task 2
[INB0-02-20 Name: Manuel Barossa Score: 5
, INB0-02-20 Name: Miguel Morretti Score: 5
, INB0-02-20 Name: Slim Shady Score: 3
, INB0-02-20 Name: Pablo Escobar Score: 2
]
```

Рисунок 9 – Работоспособность второго задания

### Task 3

```
[INB0-02-20 Name: Hope Johnson Score: 5  
, INB0-02-20 Name: Manuel Barossa Score: 5  
, INB0-02-20 Name: Miguel Morretti Score: 5  
, INB0-02-20 Name: Tom Holland Score: 4  
, INB0-02-20 Name: Isaac Anderson Score: 3  
, INB0-02-20 Name: Slim Shady Score: 3  
, INB0-02-20 Name: Pablo Escobar Score: 2  
]
```

Рисунок 10 – Работоспособность третьего задания



## **Вывод**

В результате выполнения данной практической работы я освоил сортировку различными методами и интерфейсами в языке программирования Java.