



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

Деревянкин Н.А.

Принял

Степанов П.В.

Практическая работа выполнена

«__»____2021 г.

«_____»

«__»____2021 г.

Отметка о выполнении

Москва – 2021 г.

СОДЕРЖАНИЕ

Цель работы	3
Задание	3
Выполнение работы	4
Код выполненной работы.....	5
Вывод.....	8

Цель работы

Целью 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.

Выполнение работы

Приступив к выполнению, я создал класс `Student`, который создает объект ученика, а также создал класс `SortingStudentsByGPA`, который имплементирует интерфейс `Comparator`. После всего я создал запускной класс `Main`, в котором проверил работоспособность своих сортировок, а также объединил существующие массивы в один, применив к нему сортировку.

Код выполненной работы

Здесь в нескольких скриншотах можно увидеть, как выглядит код полученного задания и его вывод.

```
public class Student {  
    private String name;  
    private String phone;  
    private int mark;  
    private int ID;  
  
    public Student() {  
  
    }  
  
    public int getMark() { return mark; }  
  
    public void setMark(int mark) { this.mark = mark; }  
  
    public Student(int ID, String name, String phone, int mark) {  
        this.name = name;  
        this.phone = phone;  
        this.mark = mark;  
        this.ID = ID;  
    }  
  
    public int getID() { return ID; }  
  
    public void setID(int ID) { this.ID = ID; }  
  
    public String getName() { return name; }  
  
    public void setName(String name) { this.name = name; }  
  
    public String getPhone() { return phone; }  
  
    public void setPhone(String phone) { this.phone = phone; }
```

Рисунок 2 – Класс Student

```

static Student[] merge(Student[] arr1, Student[] arr2) {
    Set<Student> set = new HashSet<>();
    set.addAll(Arrays.asList(arr1));
    set.addAll(Arrays.asList(arr2));
    return set.toArray(new Student[]{});
}

```

Рисунок 3 – Метод объединения массивов

```

public static void InsertionSort(Student[] idNumber) {
    for (int left = 0; left < idNumber.length; left++) {
        Student value = idNumber[left];

        int i = left - 1;
        for (; i >= 0; i--) {
            if (value.getMark() < idNumber[i].getMark()) {
                idNumber[i + 1] = idNumber[i];
            } else {
                break;
            }
        }
        idNumber[i + 1] = value;
    }
}

```

Рисунок 4 – Сортировка вставкой

```

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

```

Рисунок 5 – Класс SortingStudentsByGPA

Вывод

В результате выполнения данной практической работы я познакомился с интерфейсом Comparator, а также научился пользоваться и работать с массивами.

GitHub - <https://github.com/dronikosha/JavaPractice>