|  |  |
| --- | --- |
|  |  |
| МИНОБРНАУКИ РОССИИ |  |
| Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |  |

**Институт информационных технологий**

КАФЕДРА ИНСТРУМЕТНАЛЬНОГО И ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ (ИППО)

ПРАКТИЧЕСКАЯ РАБОТА № 6

по дисциплине «Объектно-ориентированное программирование»

«Техники сортировки»

ВАРИАНТ 2

Выполнил студент группы ИНБО-01-18 *Трофимов С. М.*

Принял Батанов А.О.

Практическая работа выполнена «\_\_\_»\_\_\_\_\_\_\_2019г.

«Зачтено» «\_\_\_»\_\_\_\_\_\_\_2019г.

Москва 2019

**Цель работы**

Целью данной практической работы освоить на практике сортировки различными методами.

**Теоретическое введение**

(Quick Sort) Быстрая сортировка

* Массив A[1..n] азбивается на два непустых подмассивов по отношению к "опорному элементу”
* Два суб-массивы сортируются рекурсивно посредством Quick Sort.

(Merge Sort) Сортировка слиянием

* Разделить массив A[1..n] на 2 равные части
* Сортировка слиянием двух подмассивов (рекурсивно)
* Объединить (соединить) два отсортированных подмассива

**Выполнение практической работы**

*Задание:*

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

*Решение:*

Файл Student.java

**package** com.gradess.company;  
  
**public class** Student **implements** Comparable<Student> {  
  
 **private final int SCORE**;  
 **private final int ID**;  
  
 **public** Student(**int** id, **int** score) {  
 **SCORE** = score;  
 **ID** = id;  
 }  
  
 **public int** getScore() {  
 **return SCORE**;  
 }  
  
 **public int** getID() {  
 **return ID**;  
 }  
  
 @Override  
 **public int** compareTo(Student otherStudent) {  
 **return SCORE** > otherStudent.getScore() ? -1 : 1;  
 }  
}

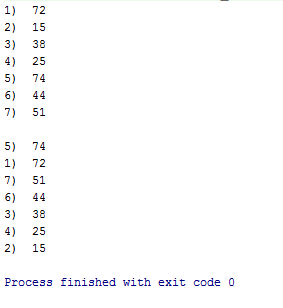
Файл SortingStudentsByGPA.java

**package** com.gradess.company;  
  
**import** java.util.Comparator;  
  
**public class** SortingStudentsByGPA **implements** Comparator<Student> {  
  
 @Override  
 **public int** compare(Student firstStudent, Student secondStudent) {  
 **return** firstStudent.compareTo(secondStudent);  
 }  
  
}

Файл Main.java

**package** com.gradess.company;  
  
**import** java.util.Arrays;  
  
**public class** Main {  
  
 **public static void** main(String[] args) {  
 **final int** SIZE = 7;  
 Student[] students = **new** Student[SIZE];  
  
 **for** (**int** id = 0; id < SIZE; id++) {  
 students[id] = **new** Student(id + 1, (**int**) (Math.*random*() \* 100));  
 }  
 *printArray*(students);  
  
 Arrays.*sort*(students, **new** SortingStudentsByGPA());  
 System.***out***.println();  
 *printArray*(students);  
 }  
  
 **private static void** printArray(Student[] students) {  
 **for** (Student student : students) {  
 System.***out***.println(student.getID() + **")\t"** + student.getScore());  
 }  
 }  
}

**Результат выполнения программы**



**Выводы по работе**

Я освоил на практике сортировки различными методами.

**Используемая литература:**

1. К. Сьерра, Б. Бэйтс. Изучаем Java, 2-е издание.
2. С. Макконнелл. Совершенный код, 3-е издание.