**ДОДАТОК А**

import java.io.\*;  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.Comparator;  
import java.util.Scanner;  
  
public class Learner {  
 public static void main(String[] args) {  
 File inputFiles = new File("C:\\Project");  
 File outputFiles = new File("C:\\Project");  
 inputFiles.mkdir();  
 outputFiles.mkdir();  
  
 File f1 = new File(inputFiles, "file1in.txt");  
 File f2 = new File(inputFiles, "file2in.txt");  
 File f3 = new File(inputFiles, "file3in.txt");  
  
 Controller.*handleFiles*(outputFiles, f1, f2, f3);  
 System.*out*.println("Files read and processed.");  
 }  
}  
  
class Student implements Comparable<Student> {  
 private String group;  
 private String name;  
 private double mark;  
  
 public Student(String group, String name, double mark) {  
 this.group = group;  
 this.name = name;  
 this.mark = mark;  
 }  
 public String getGroup() {  
 return this.group;  
 }  
  
 public String getName() {  
 return this.name;  
 }  
  
 public double getMark() {  
 return this.mark;  
 }  
  
 @Override  
 public String toString() {  
 return this.group + " " + this.name + " " + this.mark;  
 }  
  
 @Override  
 public int compareTo(Student student) {  
 int result = Double.*compare*(this.mark, student.mark);  
 if (result != 0) {  
 return result;  
 } else {  
 result = this.group.compareTo(student.group);  
 if (result != 0) {  
 return result;  
 } else {  
 return this.name.compareTo(student.name);  
 }  
 }  
 }  
}  
  
class Controller {  
 public static final ArrayList<Student> *HIGH\_MARK\_STUDENTS* = new ArrayList<>();  
 public static final ArrayList<Student> *MEDIUM\_MARK\_STUDENTS* = new ArrayList<>();  
 public static final ArrayList<Student> *LOW\_MARK\_STUDENTS* = new ArrayList<>();  
  
 public static void handleFiles(File outputFilesDir, File... files) {  
 String sortType = *inputSortType*();  
 for (File : files) {  
 try {  
 *readStudentFile*(file);  
 } catch (FileNotFoundException fnfe) {  
 System.*out*.println("File for read with path " + file.getPath() + " does not exist!");  
 } catch (IOException ioe) {  
 System.*out*.println("Exception is thrown: " + ioe.getMessage());  
 }  
 }  
 *sortStudents*(sortType);  
 *writeStudents*(outputFilesDir);  
 }  
  
 private static String inputSortType() {  
 System.*out*.println("Select student sorting type:\n1 - by Mark, Group, Name\n2 - by Group, Name, Mark\nDefault - by Mark, Group, Name\nEnter: ");  
 Scanner input = new Scanner(System.*in*);  
 String str = "";  
 try {  
 int type = input.nextInt();  
 if (type == 1) {  
 str = "mark";

} else if (type == 2) {  
 str = "group";  
 }  
 } catch (Exception e) {  
 str = "mark";  
 }  
 System.*out*.println("You choose " + str + " type of sort");  
 return str;  
 }  
  
 private static void readStudentFile(File file) throws IOException {  
 BufferedReader br = new BufferedReader(new FileReader(file));  
 String line;  
 while ((line = br.readLine()) != null) {  
 String[] parts = line.split("\\s+");  
 if (parts.length == 4 && parts[0].matches("\\d+") && parts[3].matches("\\d+(\\.\\d+)?")) {  
 String group = parts[1];  
 String name = parts[2];  
 double mark = Double.*parseDouble*(parts[3]);  
 *separateStudent*(new Student(group, name, mark));  
 } else {  
 System.*out*.println("Invalid data format in the file: " + file.getPath());  
 }  
 }  
 br.close();  
 }  
  
 private static void separateStudent(Student student) {  
 double mark = student.getMark();  
 if (mark >= 4.5) {  
 *HIGH\_MARK\_STUDENTS*.add(student);  
 } else if (mark >= 4.0 && mark < 4.5) {  
 *MEDIUM\_MARK\_STUDENTS*.add(student);  
 } else {  
 *LOW\_MARK\_STUDENTS*.add(student);  
 }  
 }  
  
 private static void writeStudents(File outputFilesDir) {  
 File[] outputFiles = {new File(outputFilesDir, "highMarkStudents.txt"),  
 new File(outputFilesDir, "mediumMarkStudents.txt"), new File(outputFilesDir, "lowMarkStudents.txt")};  
 try {  
 PrintWriter pw;  
 for (int i = 0; i < outputFiles.length; i++) {  
 pw = new PrintWriter(outputFiles[i]);  
 ArrayList<Student> studentsList = i == 0 ? *HIGH\_MARK\_STUDENTS* : (i == 1 ? *MEDIUM\_MARK\_STUDENTS* : *LOW\_MARK\_STUDENTS*);  
 for (int j = 0; j < studentsList.size(); j++) {  
 pw.println((j + 1) + " " + studentsList.get(j));  
 }  
 pw.flush();  
 pw.close();  
 }  
 } catch (FileNotFoundException fnfe) {  
 System.*out*.println("File for write not found!");  
 }  
 }  
  
 private static void sortStudents(String key) {  
 Comparator<Student> comparator;  
 switch (key) {  
 case "group":  
 comparator =

Comparator.comparing(Student::getGroup).thenComparing(Student::getName).thenComparing(Student::getMark);

break;

case "mark":

default:

comparator = Comparator.comparing(Student::getMark).thenComparing(Student::getGroup).thenComparing(Student::getName);

break;

}

Collections.sort(HIGH\_MARK\_STUDENTS, comparator);

Collections.sort(MEDIUM\_MARK\_STUDENTS, comparator);

Collections.sort(LOW\_MARK\_STUDENTS, comparator);

}

}