

### 15с

Считать файл 11401.txt, 11402.txt, 11405.txt (выберите файл по номеру своей группы) в удобную коллекцию для хранения и анализа информации. Кроме считывания данных учесть номера задач, сохранить в какой-нибудь коллекции их так, чтобы можно было потом использовать в решении дальнейших задач.

### 16с

По удобной коллекции, построенной в задаче 15с, подсчитать следующую статистику:

- Map<Student, Integer> - количество решенных каждым студентом задач (решенная = 1 за нее)
- Map<Student, Double> - средний балл студента по всем задачам (за точку (.) ставить 0 баллов)
- Map<String, Integer> - количество студентов, решивших каждую задачу
- Map<String, Double> - средний балл по каждой решенной задаче.

Номера задач брать именно те, которые описаны в исходном файле.

### 17с

На вход программы подается файл с английским текстом. Составить статистику с помощью Map<Character, Integer> о том, сколько раз каждая буква английского алфавита упоминается в тексте.

### 18с

В классе Vector2D (с прошлого семестра, можете использовать чужое решение) реализовать интерфейс Comparable<Vector2D>, выполняющий сравнение векторов по длине.

В методе main создать ArrayList из векторов и отсортировать их по длине с помощью метода Collections.sort.

### 19с

Реализовать класс от интерфейса Comparator<Vector2D>, сравнивающий два вектора в лексикографическом порядке.

В методе main создать ArrayList из векторов и отсортировать их с использованием реализованного компаратора с помощью метода Collections.sort.

**19b**

Вз входного файла in.txt вводятся целые числа. Они записываются в ArrayList<Integer>.

Упорядочить полученную коллекцию по возрастанию количества цифр в числах (реализовать Comparator<Integer>).

Результат сортировки вывести в выходной файл out.txt.

Запрещается конвертировать числа в строки для применения к последним length. Все сделать через арифметику цифр числа.

Для прохождения по элементам коллекции использовать итератор!