Оглавление

[**1.** **Роли и не только** 3](#_Toc149815273)

[**2.** **Формулировка задачи** 3](#_Toc149815274)

[**3.** **Постановка задачи** 3](#_Toc149815275)

# **Роли и не только**

Как говорил французский социалист Луи Блан, от каждого по способностям, каждому по потребностям. Именно поэтому в нашем маленьком, но гордом обществе, не произошло разделения на должности. Мы выполняли те части, которые хотели. Хотя можно сказать, что Дима работал с первой частью подпрограммы (писал, тестировал и тд.), читаемостью кода и заполнением, документации, а Миша разрабатывал вторую часть подпрограммы, общался с заказчиком и делал общий запускающий скрипт. Общая концепция программы, а также тонкости работы отдельных кусков кода являются результатом совместного brain-шторма.

# **Формулировка задачи**

Заказчик выдал нам следующее задание:

1. В качестве предметной области и исходных данных выступает Файловая Система Преподавателя из Лабораторной работы №3.

2. Базовое решение для успешной сдачи лабораторной работы должно предоставлять следующие возможности по заданному пользователем номеру группы:

\* вывод имени студента, не сдавшего хотя бы один тест (с указанием номера теста);

\* вывод списка группы, упорядоченного по количеству попыток сдачи теста.

3. Для получения Отличной оценки (и потенциально дополнительных баллов) также необходимо в рамках решения реализовать:

\* вывод по фамилии студента его средней оценки по всем тестам предмета;

\* вывод по номеру группы занятий с минимальной общей посещаемостью;

\* вывод по номеру группы занятий с максимальной общей посещаемостью.

# **Постановка задачи**

Разработка началась с создания отдельного репозитория на github.

Далее было принято решение, отделить реализацию пунктов 2 и 3, а затем свести их в общем «запускающем» скрипте. Это было сделано для некого распараллеливания работы, а также удобства редактирования. Так же сразу было ясно, что будут использоваться case с возможностью возврата на уровень выше.

*Задача 1* делиться на две подзадачи. В них обеих нужен номер группы, поэтому ее считывание происходит перед вопросом о выборе подзадачи. Сразу выделили несколько критических случаев, для которых придумали исключения/обработчики.

Для решения первой подзадачи выбрали метод прохождения отдельно по предметом с поиском человека, у которого есть оценка «2», а затем провести обрезание лишней информации и сортировка. Идея использовать цикл для более аккуратного вывода информации.

Для решение второй подзадачи выбрали метод прохождения по отдельным тестам каждого предмета с использованием циклов, далее происходит обрезание лишней информации, выделение повторяющейся информации с помощью команды uniq с ключом с и итоговая сортировка.

*Задача 2* начинается с выбора подзадачи. Для первой – спрашивается фамилия студента. Решено обработать случаи, когда вводится только фамилия без инициалов, или введено с маленькой буквы. Используется массив, в который сливаются все тесты подходящих людей по одному предмету. В массив имен добавляются только уникальные имена из предыдущего массива. Для людей, входящих в массив имен ищем все успешные результаты тестов по предмету и суммируем их. Затем делим это значение на кол-во тестов (4) и выводим результат. Аналогично для другого предмета.

Для второй подзадачи мы запрашиваем номер группы (с обработанными исключениями) и затем формируем массив, содержащий кол-во студентов, пришедших на отдельно взятое занятие. Далее на основе этого массива, мы находим занятие, с максимальной посещаемостью и выводим его индекс.

Третья подзадача решается аналогичным образом, только ищется минимальная посещаемость.