

Анализ успеваемости студентов

Что нужно сделать?

Нужно написать скрипт для обработки csv-файла.

Скрипт читает файлы с данными (см. примеры ниже) об успеваемости студентов и формирует отчеты. Нужно сформировать всего один отчёт `student-performance` (см. пример ниже). Отчёт включает в себя список студентов и среднюю оценку студента, среднее считается по всем предметам, студенты сортируются по успеваемости. Название файлов (может быть несколько) и название отчета передается в виде параметров `--files` и `--report` (см. пример ниже). Отчёт формируется по всем переданным файлам, а не по каждому отдельно.

Чтобы сфокусироваться на функционале формирования отчёта и не отвлекаться на рутинные задачи (обработку параметров скрипта, чтения файлов и вывод), можно использовать стандартную библиотеку `argparse` и `csv`, а для отображения в консоли - библиотеку `tabulate`.

Примеры файлов можно посмотреть [здесь](#), выглядят так:

```
student_name,subject,teacher_name,date,grade
Семенова Елена,Английский язык,Ковалева Анна,2023-10-10,5
Титов Владислав,География,Орлов Сергей,2023-10-12,4
Власова Алина,Биология,Ткаченко Наталья,2023-10-15,5
```

Пример запуска скрипта:

```
$ python3 main.py --files students1.csv stidents2.csv --report students-performance
+-----+
|  | student_name | grade |
+-----+
| 1 | Соколов Павел | 4.8 |
| 2 | Власова Алина | 4.7 |
| 3 | Иванов Алексей | 4.5 |
| 4 | Калинина Ольга | 4.5 |
| 5 | Кудрявцев Глеб | 4.5 |
| 6 | Савельева Анастасия | 4.3 |
| 7 | Захарова Дарья | 3.7 |
| 8 | Фомина Ксения | 3.7 |
| 9 | Кузнецова Анна | 3.5 |
+-----+
$
```

Какие функциональные требования?

- можно передать пути к файлам
- можно указать название отчета
- в консоль выводится отчёт в виде таблице

Какие не функциональные требования?

- для всего кроме тестов и вывода в консоль, можно использовать только стандартную библиотеку, например:
 - для обработки параметров скрипта нельзя использовать click, но можно использовать argparse
 - для чтения файлов нельзя использовать pandas, но можно использовать csv
- в архитектуру заложена возможность добавления новых отчётов, например, если захотим посмотреть успеваемость в разрезе преподавателей или успеваемость по курсам, то отчет можно будет добавить быстро
- код покрыт тестами которые написаны на pytest
- для тестов можно использовать любые дополнительные библиотеки
- код соответствует:
 - общепринятым стандартам написания проектов на python
 - общепринятому стилю

Как сдавать задание?

- присылайте ссылку на git репозиторий, ссылки на google drive или yandex не подходят
- присылайте примеры запуска скрипта, например:
 - можно сделать скриншот запуска скрипта и добавить его репозиторий, для примера работы можно использовать [эти](#) файлы. За приложенные примеры запуска ревьюер скажет вам спасибо и добавит баллы.
- перед отправкой ссылки на репозиторий проверьте, пожалуйста, что репозиторий публичный и его можно посмотреть

FAQ

- Можно ли использовать нейросети?
 - Рекомендуем не использовать. Сталкиваемся со случаями, когда кандидаты увлекаются нейросетями, чтобы сделать тестовое, а потом не проходят техническое интервью, потому что не понимают, почему нейросеть написала тот или иной код.
- Будет ли приниматься задание без тестов?
 - Нет, приниматься не будет. Наличие тестов входит в основные требования.
- Код покрыт тестами - это какой процент покрытия?

- Можно ориентироваться на 80% покрытия по [pytest-cov](#), можно больше, можно меньше, но главное чтобы был протестирован критически важный функционал.
- Можно ли использовать какие-то дополнительные библиотеки к pytest?
 - Да, всё что помогает вам тестировать код можно использовать.
- Можно ли пользоваться линтерами или форматтерами кода?
 - Да, можно использовать любой линтер и любой форматтер - это хорошая практика.
- Можно ли использовать что-то кроме pip для управления зависимостями?
 - Да, можно, приветствуется.
- Можно ли менять API, название и формат параметров скрипта?
 - Нет, менять нельзя, используем `files` и `report`.
- Нужно ли учитывать случаи, когда пользователь при запуске скрипта ввёл что-то не то?
 - Да, нужно, считаем что содержимое файлов всегда валидно, но пользователь может запустить скрипт с любыми аргументами.
- Нужно ли учитывать случаи, когда файл с данными большой, например несколько Гб?
 - Нет, не нужно, ограничений по памяти нет, всё можно читать в память.
- Нужно ли писать комментарии в коде?
 - Если вы считаете что они нужны, то пишите.
- На что будет смотреть ревьюер при проверки задания?
 - Будет проверяться выполнение функциональных и нефункциональных требований, соблюдение общепринятых стандартов написания кода на python и архитектура, которая позволит добавлять новые отчёты.
- Новые отчёты которые будут добавляться будут использовать тот же формат файлов?
 - Да, формат будет такой же, новые колонки добавляться не будут.
- Нужно ли писать readme.md?
 - Да, но достаточно пара абзацев в несколько предложений. В него можно положить примеры запуска скрипта и написать то, что на ваш взгляд важно знать ревьюеру, например описание того как добавить новый отчет. Если будете писать readme, то пусть он будет небольшим, но по делу, чем большим и сгенерированным нейросетью.