**Вариант 13 конкурсных заданий *практического* этапа Конкурса**

Добрый день! Сегодня Вам предстоит побывать в роли стажера в компанию по разработке стриминговых сервисов. В один из солнечных дней Вашей работы пришла новость о том, что один из junior-разработчиков случайно стер рабочую базу данных, но у него остался бэкап в виде csv файла. Вам предстоит проявить волю к победе и восстановить минимальный функционал для работы приложения. Думаю, что Вы готовы приступать к задачам, но сначала необходимо **создать репозиторий** для проекта и задач, которые Вы будете выполнять. Обязательно сделайте его **PUBLIC**, а то мы не сможем проверить решение вашей последней задачи. Каждую задачу вам необходимо будет правильно оформить и залить в репозиторий. *Код, который Вы напишете необходимо задокументировать, чтобы другие программисты могли понять, что делает код и за что отвечает. Примеры документирования приведены ниже.*

**Пример документирования кода на языке С++**

/\*\*

\* Это описание функции foo

\*

\* @param str это описание аргумента str

\* @param pattern это описание аргумента pattern

\* @return это описание того, что вернет функция

\*/

int foo(std::string str, std::string& pattern)

{

...

};

**Пример документирования на языке Python**

def complex(real=0.0, imag=0.0):

"""Описание функции complex.

Описание аргументов:

real – описание аргумента

imag – описание аргумента

"""

if imag == 0.0 and real == 0.0: return complex\_zero

...

Теперь можете приступать к решению задач!

К задачам прикреплен файл *songs.txt,* который хранит в себе информацию о названиях песен, артистов, дат выхода и даже количество прослушиваний.

Столбцы: *streams(кол-во прослушиваний - целое число), artist\_name, track\_name, date(дата в формате ДД.ММ.ГГГГ).*

Разделитель *«?».*

***Задача 1.***

***Формат ввода: стандартный ввод***

***Вывод: файл songs\_artst.csv***

Клиенты нашей компании хотят получить доступ к песням определенных исполнителей. Для этого вам выдали задачу написать программу, на вход в которую будет подаваться имя артиста, а на выход вы будете создавать файл *songs\_artst.csv* со столбцами *track\_name, streams, date.*

*В задаче запрещено использование сторонних библиотек(Pandas и др)*

*Не забудьте сделать комментарии к коду согласно стандартам документирования кода выбранного языка (для языка Python – PEP 257). После выполнения необходимо сделать локальные и удаленные изменения Вашего репозитория.*

***Задача 2***

Чтобы в дальнейшем нам было удобнее работать с данными из нашего файла(songs.txt) отсортируйте их по имени артиста в алфавитном порядке с помощью сортировки со сложностью (в задаче нельзя использовать встроенные функции сортировок!). Из полученных данных выведите топ-5 первых песен. Формат каждой строки: “<Название песни>, <Артист>, <дата выхода>”.

*Не забудьте сделать комментарии к коду согласно стандартам документирования кода выбранного языка. После выполнения необходимо сделать локальные и удаленные изменения Вашего репозитория*

***Задача 3***

**Ввод: стандартный ввод**

**Вывод: стандартный вывод**

Пришло время сделать наработку для интерфейса, который будет взаимодействовать с базой данных. Для этого Вам необходимо написать консольную программу, которая будет запрашивать у пользователей название песни, а на выход будет выдавать автора этой песни, если ничего не найдено будет выводить: “К сожалению, ничего не удалось найти”. Программа должна всегда запрашивать название. Прекратить свою работу она сможет только после ввода “0”.

Формат ответа на запрос пользователя: “Песня <*track\_name*> принадлежит <*artist\_name*>”

Поиск необходимо осуществить с помощью двоичного поиска**.**

Поиск необходимо осуществлять в файле songs.txt

***Не забудьте сделать комментарии к коду согласно стандартам документирования кода выбранного языка. После выполнения необходимо сделать локальные и удаленные изменения Вашего репозитория***

***Задача 4***

Найдите всех артистов, чьи песни были выпущены до 1990 года, и выведите их список. Посчитайте среднее количество прослушиваний песен для каждого артиста и выведите результат в порядке убывания среднего количества прослушиваний.

*Последним этапом необходимо записать полученный список в новый файл – songs\_average.txt. Не забудьте сделать комментарии к коду согласно стандартам документирования кода выбранного языка. После выполнения необходимо сделать локальные и удаленные изменения Вашего репозитория.*

***Задача 5***

Вам предстоит создать программу, которая будет обрабатывать данные из файла songs.txt, содержащего информацию о названиях песен, артистах, датах выхода и количестве прослушиваний. Для обработки данных используйте хеш-таблицы.

Вам необходимо:

Прочитать данные из файла songs.txt и создать хеш-таблицу, где ключом будет название песни, а значением - словарь с информацией об артисте, дате выхода и количестве прослушиваний. В случае, если песни с одинаковыми названиями исполняют несколько артистов, то необходимо вывести информацию обо всех из них.

*Не забудьте сделать комментарии к коду согласно стандартам документирования кода выбранного языка. После выполнения необходимо сделать локальные и удаленные изменения Вашего репозитория.*

***Задача 6***

Ваш код будет использоваться программистом, которого возьмут на работу, поэтому он должен быть правильно оформлен и выложен на GitHub. Весь написанный код должен быть задокументирован согласно стандартам документирования кода выбранного языка.

Также необходимо оформить README.md для Вашего репозитория. Пункты, которые должны быть описаны:

1. Название проекта

2. Описание проекта

3. Оглавление (необязательно)

4. Как установить и запустить проект

5. Как использовать проект