

Трекер задач. Финал второго курса.

Создадим персональный трекер задач, который будет общаться с пользователем через стандартные потоки ввода-вывода. Трекер хранит данные в виде пар: дата - событие. Для успешного выполнения данного проекта вам необходимо реализовать весь функционал, изложенный далее, а так же учесть важные моменты, без которых функционал вашего трекера будет неполным:

- **Логирование** (оно же - сохранение) всех задач должно проходить в отдельной реляционной БД (любой), чтобы при последующих запусках программы-трекера все предыдущие задачи пользователя были сохранены и не утеряны. В данном проекте подразумевается, что вы - единственный пользователь.
- **Команды CLI** (терминальные команды) - необходимы для быстрого вывода всех имеющихся у пользователя задач. Необходимо реализовать команды `--show_current` - показать все текущие невыполненные задачи, `--show_done` - показать все выполненные задачи, `--database` - путь до используемой базы данных в текущей сессии.
- **Class Task, Class User** - ваш код проекта должен быть построен вокруг двух сущностей. Первая - сущность `Task` - хранит в себе все необходимые методы для создания, сохранения, изменения (изменения состояния) задач, а так же их удаления (пометки "выполнено"). Вторая - сущность `User` - обладает всеми необходимыми пользовательскими методами, например показать задачи пользователя, отсортировать по приоритету и т.д.
- **Наличие модульных тестов** в данном проекте - один из самых важных аспектов. Каждый класс, метод, функция - обязаны иметь для себя как минимум один тест с не менее, чем 10 тест-кейсами. Так же необходимо протестировать все обращения к БД и создать проверку ее наличия/отсутствия.

Далее будет изложен весь внешний интерфейс вашего проекта:

- **Дата** - строка следующего вида: год-месяц-день, где год, месяц и день - целые числа.
- **Событие** - строка из печатных символов без внутренних разделителей. Событие не может быть пустой строкой, а также в одну и ту же дату может произойти несколько событий. Трекер должен суметь их все сохранить. Одинаковые события произошедшие в один и тот же день сохранять не нужно, достаточно сохранить только лишь одно из них.

Трекер должен уметь поддерживать следующие команды:

- `Add Дата Событие` - добавление события
- `Del Дата Событие` - удаление события (перемещение в список выполненных событий).
- `Del Дата` - удаление всех событий за конкретную дату
- `Find Дата` - поиск событий за конкретную дату
- `Print` - печать всех событий за все даты
- `StartApp` - команда, символизирующая начало работы с трекером
- `Quit` - команда завершения работы трекера. Дальнейший ввод становится невозможен и трекер его игнорирует

Все команды, даты и события при вводе разделены пробелами. Команды считываются из стандартного ввода. В одной строке может быть ровно одна команда, но можно ввести несколько команд в несколько строк. На вход также могут поступать пустые строки — их следует игнорировать и продолжать обработку новых команд в последующих строках.

Ниже описаны особенности работы каждой из команд за исключением StartApp и Quit:

- **Add Дата Событие** - добавление события.

При добавлении события трекер должен его запомнить и затем показывать его при поиске (командой Find) или печати (командой Print). Если указанное событие для данной даты уже существует, повторное его добавление нужно игнорировать.

- **Del Дата Событие** - удаление события (перемещение в список выполненных событий)

Команда должна удалить добавленное ранее событие из списка активных (переместить в список выполненных) с указанным именем в указанную дату, если оно существует. Если событие найдено и удалено, программа должна вывести строку «Deleted successfully» (без кавычек). Если событие в указанную дату не найдено, программа должна вывести строку «Event not found» (без кавычек).

- **Del Дата** - удаление всех события за конкретную дату (перемещение в список выполненных событий).

Команда удаляет все ранее добавленные события из списка активных событий за указанную дату. Программа всегда должна выводить строку вида «Deleted N events», где N — это количество всех найденных и удалённых событий. N может быть равно нулю, если в указанную дату не было ни одного события.

- **Find Дата** - поиск событий за конкретную дату.

Найти и распечатать ранее добавленные события в указанную дату. Программа должна вывести на печать только сами события, по одному на строке. События должны быть отсортированы по возрастанию в лексикографическом порядке.

- **Print** - печать всех событий за все даты.

С помощью этой команды можно показать полное содержимое трекера. Программа должна вывести на печать все пары Дата Событие по одной на строке. Все пары должны быть отсортированы по дате, а события в рамках одной даты должны быть отсортированы по возрастанию в лексикографическом порядке. Даты должны быть отформатированы специальным образом: ГГГГ-ММ-ДД, где Г, М, Д — это цифры чисел года, месяца и дня соответственно. Если какое-то число имеет меньше разрядов, то оно должно дополняться нулями, например, 0001-01-01 — первое января первого года.

Гарантируется, что числовое значение года - число неотрицательное, значение месяц лежит в пределах от 1 до 12, а дни заключены в отрезке от 1 до 31.