Michael Jachim, “Object Oriented and Functional Programming with Python”

Project (Abstract) - My Habit Tracking App

(Github-link : https://github.com/MichalJJ25/My-Habit-Tracking-App)

Objective :

Developing a habit-tracking app, in Python, to help users monitor their habits (with the help of statistics), and help to develop consistency in finishing tasks.
Structured into 3 phases - Concept, Development and Finalization; both my concept- and development-reports have been uploaded to the GitHub-directory, and the development-report includes diagrams and code examples.
A Readme file is available as well.

Technical Approach

- Used PyCharm as my programming or integrated development environment (IDE)
- Decided to use a simple CLI for keyboard-input
- Using SQLite for the database operations
- The project has been uploaded to GitHub (see the link above)

App structure

Implemented 6 classes for the functionality, and 2 test classes :

- Habit Class - Represents a Habit with attributed like name and frequency
- User Class - Manages a list of habits for a specific user
- Analytics Class - Calculates streaks and missed habits
- Database Class - Stores habits and completion dates
- CLI (Command Line Interface) Class - Handles user interaction with the app
- Main Class - Includes the main constructor to run the app

—

- Unittest-App Class - Tests the functionality of the app, specifically the analytics
- Unittest-CLI Class - Tests the functionality of the CLI using “mock” inputs

Key features

- Flexible habit creation - 5 predefined habits + custom options,
  with differing frequencies (days, weeks and months)
- Can “mark” habits as done, streak-tracking-options (general overview or for specific habits) and shows the amount of missed dates
- CLI provides error messages for wrong inputs

Challenges and lessons learned :

I was wondering at first whether it would be feasible to put everything into 1 class, but quickly learned that there is no point once the amount of code reaches a certain volume, it becomes too cluttered.
It’s also easier to correct, expand or reduce the amount of coding once you’ve implemented several classes, better oversight makes classes and methods easily (re)usable across a multitude of other classes or even applications.

I also learned that for me personally, it’s important to try different approaches before deciding on a definitive concept. Especially without much project experience in a specific field, I think you have to take your time to understand how to approach a task or problem from different angles, to find out what works, what works well, and what doesn’t.

This project definitely also allowed (or forced ;)) me to learn entirely new approaches and libraries in Python, whether it was handling dates in general or different approaches in days, weeks and months, creating a database using SQLite, so handling SQL commands inside a Python environment, creating a CLI and trying to make the user input intuitive (whilst also thinking about possible error inputs), but also working with unittest and develop test environments, both through more direct input, but also through mock inputs to test user inputs into the CLI.

It was definitely a great learning experience and who knows, maybe I’ll get to work on similar projects again in the future.

And for anyone who stumbles across this app (and this report) - I hope it’ll provide you something interesting or useful :-)

Best regards

Michael Jachim