Priscila Fry

December 5, 2023

**IMotivational - Project Summary**

MotiVateMe is a C# application developed as a motivational tool for users in their daily work routines. The project encompasses various elements such as user interaction, data storage using SQLite, polymorphism, and a well-structured class hierarchy. Below is a detailed summary of the different aspects of the project.

**Project Requirements:**

**Realistic Input/Output:**

*Design:* Interaction with the application through the console window. Output includes motivational quotes, goal progress updates, and reminders.

*Implementation:* Implemented using Console.WriteLine and Console.ReadLine statements to facilitate user interaction.

**Documentation:**

*Design:* Planned use of in-code comments to provide explanations for classes, methods, and functionalities.

*Implementation:* Extensive use of in-code comments to document the code and enhance its clarity.

**Interface Class (IMotivational):**

*Design:* Specification of an IMotivational interface to ensure that classes implementing it will have a method to display motivational quotes.

*Implementation:* The User class implements the IMotivational interface, providing the DisplayMotivationalQuote () method.

**Abstract Class (Goal):**

*Design:* Definition of an abstract Goal class to serve as a base class for specific goal types.

*Implementation:* The Goal class is implemented as an abstract class, with concrete subclasses like DailyTask providing specific implementations.

**Composition:**

*Design:* Planning to use composition to model the relationship between a User and their Goals.

*Implementation:* The User class contains a list of Goal objects, demonstrating composition.

**Polymorphism:**

*Design:* Incorporation of polymorphism through the use of an abstract class (Goal) with concrete subclasses providing specific implementations.

*Implementation:* Methods like DisplayProgress () are overridden in subclasses to demonstrate polymorphic behavior.

**Constructors:**

*Design:* Planning to include constructors for various classes to ensure proper initialization of objects.

*Implementation:* Constructors are implemented for classes like User, Goal, and DailyTask to initialize object instances.

**Access Specifiers:**

*Design:* Consideration of appropriate access specifiers for encapsulation and information hiding.

*Implementation:* Classes and methods use access specifiers like public and private for proper encapsulation.

**SQLite Database Operations (CRUD):**

*Design:* Planning to perform CRUD operations on an SQLite database, including creating tables, inserting data, updating records, and deleting entries.

*Implementation:* The DatabaseManager class handles the creation of tables, insertion of sample data, and methods for CRUD operations.

**Demonstration of Class Hierarchy:**

*Design:* Designating a class hierarchy with the User class at the top, Goal as an abstract base class, and concrete subclasses like DailyTask.

*Implementation:* The implemented class hierarchy includes the User class containing a list of Goal objects, showcasing composition.

**Output Format:**

*Design:* Planning to display output in the console window through appropriate methods.

*Implementation:* Output is displayed using Console.WriteLine statements, and ToString () methods are overridden for formatted string representation.

**What Went Well:**

The project successfully integrates various OOP principles, making the code modular and maintainable.

Database operations are well-implemented, showcasing proficiency in handling data storage.

**What Went Poorly:**

While the basic functionality is implemented, there might be room for additional features to enhance user experience.

Further optimization and error handling can be implemented for a more robust application.

**Conclusion**

In summary, the project requirements were translated into a well-designed structure and implemented accordingly, adhering to principles of object-oriented programming and database management. MotiVateMe demonstrates a solid understanding of object-oriented programming, database interactions, and adherence to best practices. The project satisfies the course project requirements by incorporating key concepts and principles. Future enhancements could include a graphical user interface (GUI) for a more user-friendly experience and additional motivational features. Overall, the project provides a strong foundation for further development and expansion.