

Introduction

The original class from the UML diagram contained several classes interlinked, making it poor code that's harder to maintain and violates the SOLID principles. I have refactored the code to enhance the design by ensuring modularity, reusability, extensibility, and eliminating as many dependencies as possible.

Key Changes

- I have added the IOperation interface for all the math operations. Also facilitates extensibility and polymorphism, allowing for easy addition of new operations and dynamic execution without altering existing code structures.
- The BaseOperation abstract class encapsulate the same functionality promoting code reusability and reducing redundancy.
- Modification in DoubleOps from inheritance to composition , allowing it to execute any operation that implements the IOperation interface.
- Absolute: This class was abstracted to form the BaseOperation class, and its method was changed to for the new IOperation interface.
- AbsoluteLogBaseBitFlipper: The functionality of this class has been incorporated into the new BitFlipOperation class. (Don't need the AbsoluteLogBaseBitFlipper)
- AbsoluteReverser: This class's functionality has been moved to the ReverseOperation class. (Don't need the AbsoluteReverser)
- Activity: This enum the same.
- LogBase: The functionality of this class has been incorporated into the LogOperation class. (Don't need LogBase)
- Modularization and the Single Responsibility Principle Each class now has a clear and unique responsibility. This modular approach ensures easier maintenance and also aids in future scalability.
- Enhanced Encapsulation: Direct dependencies have been minimized, and classes are now better encapsulated. This results in a more secure and well-defined interaction between classes.

I left the classes AbsoluteLogBase, AbsoluteLogBaseBitFlipper, AbsoluteReverser, and LogBase in the code . However they can be deleted as they are no longer required after the refactor and redesign.

Conclusion

This refactoring has transformed the software design to be more in line with industry standards and best practices. The structure is now more flexible, modular, and ready for future enhancements.

