

Problem Statement:

- Develop a solution to handle JSON file type effectively while ensuring the code is reusable and extensible.

Patterns:

- Implemented the FileFormatter interface to abstract the process of parsing and writing data to files.
- This utilizes the **Strategy Pattern**, where implementations are provided for each file type (e.g., JSON, XML).
- AbstractFileFormatter acts as a **template method**, providing a structured flow for concrete classes to implement specific logic.
- The **Factory Pattern** is employed by the FileFormatterFactory, which fetches the appropriate ConcreteFileFormatter based on the file format.

Consequences of above logic:

- The code exhibits high reusability since the CurrencyConversion logic is separated and can be used by all FileFormatter .
- It's easily extensible to handle different file formats; creating a new implementation only requires writing logic for reading and writing data for that particular file format