**Design Patterns Integration and Explanation**

***The Factory Design Pattern:***

* It is a Creational Design Pattern.
* It is a factory class that provides an interface for creating families of related objects without specifying their concrete classes.
* *Implementation:* In classes **IFactory** and **CounterFactory**.
* *Usage:* In counter program **wcOO** to create a **CounterTemplate** class for counting chars, words, lines in a source file.
* *Advantages:* 
  + It decouples the business logic from creating objects.
  + It enhances Readability (Clean code) and Maintainability (debugging and adding code).

***The Strategy Design Pattern:***

* It is a Behavioral Design Pattern.
* It defines a family of algorithms, encapsulate each one inside a class, and make them interchangeable by selecting an algorithm at runtime.
* *Implementation:* In interface **ICounterStrategy** which is a generalization for different counting strategy templates.
* *Usage:* It is used as a super class to define other Counters; **CharCounter**, **WordCounter** and **LineCounter** and **TotalCounter**.
* *Advantages:* 
  + It's easy to switch between different algorithms (strategies) in runtime.
  + It enhances Readability (Clean code) and Maintainability (adding more new strategies in the future without breaking the code).

***The Composite Design Pattern:***

* It is a Structural Design Pattern.
* It is a tree structure of simple and composite objects, where Composite objects are made of multiple Simple objects that have a certain defined concept and functionality.
* *Implementation:* In class **TotalCounterTemplate** which is a **CounterTemplate** composed of simple counter templates **CharCounterTemplate**, **WordCounterTemplate** and **LineCounterTemplate**.
* *Usage:* In class **wcOO** and **CounterFactory** that creates a **TotalCounterTemplate** to be used for counting characters, words and lines.
* *Advantages:*
  + Easy to add new kinds of components to the design structure made of previous components.
  + It provides flexibility to the design structure by using interchangeable components.

***The Template Method Design Pattern:***

* It is a Behavioral Design Pattern.
* *Implementation:* In classes **CounterTemplate** and the sub-classes **CharCounterTemplate**, **WordCounterTemplate, LineCounterTemplate** and **TotalCouterTemplate**.
* *Usage:* In class **wcOO** and **CounterFactory** that creates a **TotalCounterTemplate** to be used for counting characters, words and lines.
* *Advantages:*
  + It lets sub-classes redefine certain steps of an algorithm without changing the algorithms structure.