Software Design Pattern Lab
Task Two

Submission: 10th July, 2021

Shamim Bin Zahid
Roll: SH-43

The Problem asks to make a program that generates a sentence in three algorithms from a list of words that a user enters.

1. Random Sentence Generator
2. Sorted Sentence Generator
3. Ordered Sentence Generator

Each of these have their own algorithm for creating a sentence. The Problem Statement requires us to use a "Strategy Design Pattern" to solve this. Strategy design pattern by definition defines a family of algorithms, encapsulates each one, and makes them interchangeable.

In this case each of the sentence generators are the three algorithms for doing the same abstract thing, making a sentence from a bunch of words. Strategy allows the algorithms to vary independently from the clients that use it.

To implement this problem into a strategy we capture the abstraction in an interface called SentenceGenerationStrategy. Which only has some abstract methods such as addWordToList, getWordsForTheSetence etc with no actual implementation details. Which will be buried deep in the derived classes of the interface, which are RandomSentenceGenerator, SortedSentenceGenerator, and the OrderedSentenceGenerator. These derived classes implement the interface and define their own rules / algorithms to generate sentences within themselves.

UML Class Diagram on LucidCharts:
https://lucid.app/lucidchart/a4886364-750e-442a-a85f-f32131de32ed/view