SOFTWARE DESIGN PATTERN

Assumptions of Project

Implementation of Strategy Pattern

Submitted by:
Fabliha Anber
Roll- RH-56
Registration number: 2017-415-019

Submitted to:

Dr. Md. Samiullah
Assistant Professor,
Department of Computer Science and Engineering

University of Dhaka

Assumptions relating to strategy pattern implementation

Three Sentence Generators RSG(Random Sentence Generator), SSG(Sorted Sentence Generator), OSG(Ordered Sentence Generator) was implemented using the strategy pattern where the varying behavior of these three generators are altered during runtime associating it with different objects which can perform tasks in different ways. The strategy pattern was implemented using three interfaces and the behaviors of each of these classes are switched during runtime –

- **<<interface>> IStoreWordsBehavior** which is implemented by the classes StoreLowerCase and StoreUpperCaseAndReverse. The algorithm of the behavior StoreLowerCase is used by RSG and SSG when the words are stored in the internal memory given by the user. And the words are stored using the behaviors of StoreUpperCaseAndReverse in OSG.

- **<<interface>> IConcatenateWordsBehavior** which is implemented by the class ConcatenateUsingSpace and this behavior is similar for all the generators when the sentence is generated using a space in between the words.

- **<<interface>> IPickWordsBehavior** which is implemented by the class PickWordsRandomly, PickWordsRandomlyAndSort and PickWordsRandomlyAndSortInOrder used by the classes RSG,SSG and OSG respectively.

The algorithms are implemented using these concrete strategies. The strategy pattern has helped to reduce code duplication as StoreLowerCase was used by both RSG and SSG. And by isolating the class ConcatenateUsingSpace, it is made common to all the classes and thus reducing duplicated code. The Context Strategy class SentenceGeneratorContext implements the variations of an algoritm the context uses which is passed by the Client class(SentenceGenerator56). This Client class creates a specific strategy given by the user as input from the menu of Sentence Generator options. The context class shows a method for setting the strategy which lets Clients replace the strategy associated with the context of the sentence generators(RSG, SSG or OSG) at runtime.

Assumption relating to code implementation

- The Internal Vocabulary Strings of (RSG,SSG and OSG) are initially empty and is updated when the user gives word input and the words are stored in the internal vocabulary of each of the three generators.
- Each time a loop is executed until the user presses Exit button to exit.
- After first execution of the loop when there are words stored in the vocabulary, the user can generate words.
- It is assumed that the sentence length will be from 1-15 words and the words are chosen randomly for each of the three generators.

Conclusion

The program runs smoothly when executed. Some bug fixes are required for full performance of the program