Iterator: Here, there is an iterator interface that creates an outline of the functions and variables used in each iterator. There are subclasses that implement this interface for each type of thing being iterated over (list, set, etc.)

Observable: Here, both the DarkLord class and DeathEater class implement the observable interface. This allows them to send and receive messages to each other via object methods, and for us to intercept these messages and view them in the console. Here, the threads are synchronized to prevent two or more classes from interfering with one another.

Strategy: Here, classes are made that define problems and a “strategy” is made to solve these problems. Based on the needs of the object/program, different sets of instructions can be executed to create the desired effect and functionality. This allows for easier plug and play, as methods for each task can be called when needed, and multiple can be called in a row.

Template Method: Here, chunks of a full story are created and stored in objects in a way such that they can be used regardless of the situation. Since each of these object classes holds part of the story, they can be pieced together based on the context of the story and create a larger story from these smaller Strings.