Name: Brendan Furtado
ID: 260737867

Mini 8 Part 3

Discussion

- In the context of this problem, the Strategy design offers the simplest class interface to the user. Aside from the class object (in this case food), the only part the user/client has to code is the driver class and define the column header and items to display. The template method design requires the user subclass the abstract class and define all methods in the abstract class. Template Method uses inheritance to vary **part** of an algorithm. Strategy uses delegation to vary the **entire** algorithm. With Strategy you are calling the original algorithm. With Template you can only implement the parts you need.

- Both patterns are useful and it depends on the requirements. In this case I found the strategy method is more robust for the following reasons..
    - With the strategy pattern you have the ability to implement a generic table that can store and display different objects - Food, Car, People, etc.
    - This is not possible with the Template pattern because the Table has to be derived by the Object class itself. So Food, Car, etc. are all separate tables.
    - With the Strategy pattern you have the ability to make run time changes. You cannot do this with Template since you have to create subclasses.

- **Advantages of the Strategy pattern:**
    - The Strategy method allows more flexibility by the user for changing what columns get displayed and what don't.
        - The strategy pattern allows a certain algorithm to be chosen at runtime, for optimal dynamic input by client
        - Clear and separated concrete algorithms to be passed to the strategy (encapsulation of the algorithm)
    - Avoid lengthy else-if statements or switch statements
- **Disadvantages of the Strategy Pattern:**
    - You have to instantiate a new class object, eg Name() in order to represent all the different properties of the class.
    - You also have to code the strategy interface to contain all the methods that need to be called.
    - The flow for the strategy pattern is a little more complicated to follow in code. You need to run a debugger to trace the flow.
- **Advantages of the Template Method Pattern:**
    - Allows flexibility in the implementation of how to implement the abstract methods of the template algorithm.

    - Provides a simple skeleton of how the algorithm should run
    - Self contained and easier to read the code. Also, it's easier to divide the tasks across multiple programmers when you need to write multiple subclasses.
- **Disadvantages of the Template Method Pattern:**
    - You have to write more classes if you want to implement different kinds of tables.
    - You can only store one type of object in each table. This I feel is the biggest disadvantage.