Exploring the Command Design Pattern
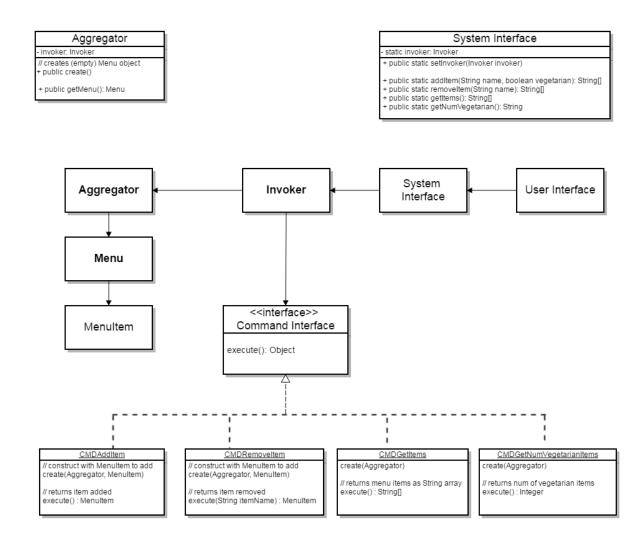
 you are to implement the Command design pattern. Design a simple MenuItem class that stores only two pieces of information - the name of a menu item, and whether it is a vegetarian dish or not. Then implement a Menu class that maintains a collection of MenuItem objects (i.e., as an aggregator). The Menu class should have methods to allow menu items to be added and removed. It should also have a method that returns an array of strings containing all of the menu items so that they can be displayed. Finally, the Menu class should have a method that returns how many of the items on the menu are vegetarian dishes.

Design a simple text-based user interface that provides the following commands,

Main Menu

1 – Add Item
2 – Remove Item
3 – Display Items
4 – Display Number of Vegetarian Items

UML Diagram on following page

Aggregator

- invoker: Invoker

// creates (empty) Menu object
+ public create()

+ public getMenu(): Menu

System Interface

- static invoker: Invoker

+ public static setInvoker(Invoker invoker)

+ public static addItem(String name, boolean vegetarian): String[]
+ public static removeItem(String name): String[]
+ public static getItems(): String[]
+ public static getNumVegetarian(): String

```
Aggregator  <---  Invoker  <---  System      <---  User Interface
    |                              Interface
    v
  Menu
    |
    v
 MenuItem
```

<<interface>>
Command Interface

execute(): Object

CMDAddItem

// construct with MenuItem to add
create(Aggregator, MenuItem)

// returns item added
execute() : MenuItem

CMDRemoveItem

// construct with MenuItem to add
create(Aggregator, MenuItem)

// returns item removed
execute(String itemName) : MenuItem

CMDGetItems

create(Aggregator)

// returns menu items as String array
execute() : String[]

CMDGetNumVegetarianItems

create(Aggregator)

// returns num of vegetarian items
execute() : Integer

MenuItem

- description: String
- vegetarian: Boolean

+ public create(String name, Boolean veg)
+ public toString(): String
+ public isVegetarian(): Boolean

Menu

- items: ArrayList

+ public addItem(MenuItem item): MenuItem
+ public removeItem(String itemName):
MenuItem
+ public numVegetarian(): Integer
+ public getItems(): String[]