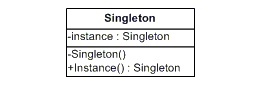
* **Select 3 design patterns**
  + **Write a short description (about half page) for each of them (prefer Bulgarian language)**
    - **Describe their motivation, intent, applicability, known uses, implementation, consequences, structure, related patterns, etc.**
  + **Provide C# examples for their use**
  + **Provide a UML diagram or image of the pattern**
    - **You can download it from the Internet**

1. Singleton –  is a [design pattern](http://en.wikipedia.org/wiki/Design_pattern_(computer_science)) that restricts the [instantiation](http://en.wikipedia.org/wiki/Instantiation_(computer_science)) of a class to one [object](http://en.wikipedia.org/wiki/Object-oriented_programming). This is useful when exactly one object is needed to coordinate actions across the system. The concept is sometimes generalized to systems that operate more efficiently when only one object exists, or that restrict the instantiation to a certain number of objects.

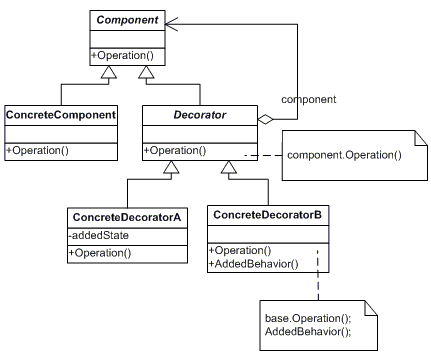
Example: SingletonExample.cs

UML diagram:   


1. Decorator – Attach additional responsibilities to an object dynamically. Decorators provide a flexible alternative to subclassing for extending functionality.

Example: DecoratorExample.cs

UML diagram



1. Mediator - Define an object that encapsulates how a set of objects interact. Mediator promotes loose coupling by keeping objects from referring to each other explicitly, and it lets you vary their interaction independently.

Example: MediatorExample.cs

UML diagram:

