1. Which design pattern defines one-to-many dependency among objects?

* **Observer**
* Facade
* Singleton
* Factory

1. Parent of concrete strategies is

* Interface only
* Abstract method only
* Interface or other class
* **Interface or abstract class**

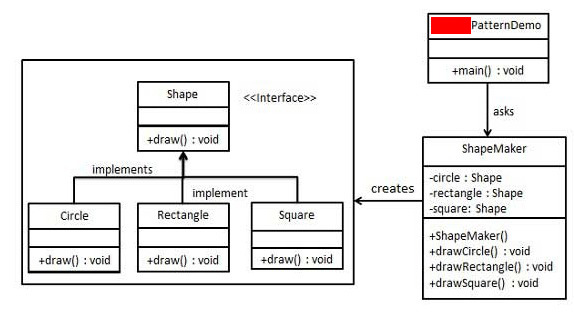
1. The State design pattern should put large, multipart conditional statements that depend on the object’s state to:

* Separated methods
* **Separated classes**
* Separated options for a Case clause
* All above are correct

1. Which is **NOT** a typical use case for the Facade design pattern?

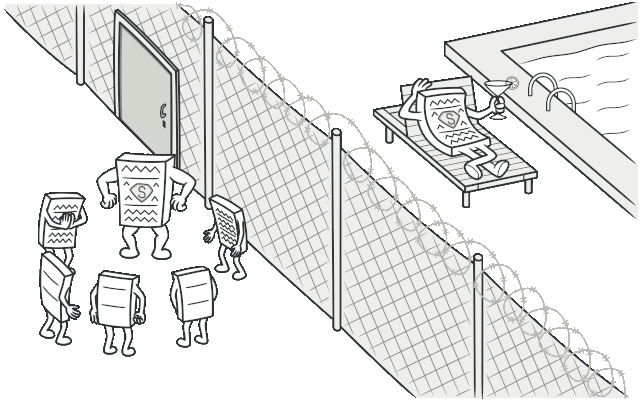
* A simple interface is required to access a complex system
* An entry point is needed to each level of layered software
* **To alter interface to match what the client is expecting**
* A system is very complex or difficult to understand

1. Identify the most suitable design pattern for the following UML diagram:

****

* Factory Method
* Abstract Factory
* **Facade**
* Builder

1. Identify the most suitable design pattern being depicted by the following funny cartoon:



* Visitor
* **Proxy**
* Memento
* Command