# Single ton

<http://www.codeproject.com/Articles/25513/Singleton-Design-Pattern-C-implementation>

Design pattern, с който си подсигуряваме, че в рамките на приложението ни ще имаме само една инстанция на даден клас. В общия случай се постига чрез скриване на конструктора с задаване модификатор private. В самия клас се прави инстанцията (много често с lazy load (друг design pattern, който създава обект чак когато някой го поиска), която при поискаване се връща чрез публично пропърти. За по-голяма сигурност на уникалността на инстанцията, често пъти класът със single тон е sealed, което означава че не може да бъде наследяван. При многонишково програмиране при самото създаване инстанцията се lock-ва за да се предотврати създаването и няколко пъти в случай на едновременно първо поискване от няколко нишки.

# Facade

<http://www.codeproject.com/Articles/481297/UnderstandingplusandplusImplementingplusFacadeplus>

Design pattern с който осъществяваме скриване на сложна логика от потребителя при използване на нашите обекти. Например ако за изпълнението на дадена операция е неоходимо извикването на няколко метода от различни класове и взаимодействие между техните даннни, фасадата позволява скриване на всичко това зад един метод, който вътрешно в себе си реализира връзките. Стандартно това се постига чрез инициализиране на инстанции в класа- façade на всики класове, чиито методи ще се наложи да бъдат извикавани, за да се постигнат резултатите на методите представени на потребителя от фасадата.

# Proxy

<http://www.codeproject.com/Articles/492594/Understanding-and-Implementing-Proxy-Pattern-in-Cs>

Design pattern, който ни помага да имаме по-добре контролиран достъп до класовете и методите в тях, като тяхното извикване се случва през междинен клас – proxy. Proxy-то работи с реалните обекти и техните методи, но позволява добавянето на допълнителни проверки, преди да се изпълнят заявките. В общия случай това се постига чрез инстанции на реалните обекти в класа –proxy и наследяване на реалния клас с който искаме да общуваме от proxy-то. Много е важно да се обележи, че proxy-то не добавя функционалност, неговите главни функции са контролиране, валидиране, дистрибутиране и т.н, но в крайна сметка работи с оригиналните методи 1:1.

# Memento

<http://www.codeproject.com/Articles/18025/Generic-Memento-Pattern-for-Undo-Redo-in-C>

Design pattern, с който осъществяваме запазване на състояния на обекта ни в различни моменти от изпълнението на приложението. Катo цяло данните, които се запазват не представяляват целия обект, а само информацията необходима за изграждането на такъв при поискаване.