**Design Patterns**

1. Facade pattern

Facade e структурен модел, най-често използван в обектно ориентираното програмиране. Facade е обект, който предоставя опростен интерфейс за големи части от код. Той прави клас библиотеките по-лесни за разбиране, достъп и тестване. Намалява зависимостта с кода извън тях. Опакова приложения със сходна функционалност в един добре организиран интерфейс.

Facade се използва, когато се цели опростяване на основната имплементация. Предпазва потребителя от сложните детайли по функционирането на програмата. Ограничавайки зависимостта между подсистемите в нея Facade прави кода по-лесен за модифициране.

1. Bridge pattern

Bridge e структурен модел целящ намаляване на зависимостта между абстракцията и нейната имплементация, така че да могат да се изменят отделно. Създава се абстрактен интерфейс за нуждите от реализирането на различни класове .Използва се енкапсулация, агрегация и наследяване, за да се разделят отговорностите в различни класове.

Bridge е много сходен с Adapter моделa, който служи за прилагане на интерфейс на един клас върху друг.

1. Template Method Pattern

Template Method е поведенчески модел, който дефинира алгоритмичния скелет на даден метод. Една или повече стъпки могат да бъдат пренаписани от подкласовете позволявайки различно поведение.

Template Method модела се използва при:

* променящи се части на даден алгоритъм, използван в подкласове за различни поведение
* когато е направено рефакториране и съвместно поведение се среща в различни класове. Тогава трябва да се създаде абстрактен базов клас, съдържащ всички сходства между класовете в template method с цел избягване на повтаряемостта.