

Abstract Factory

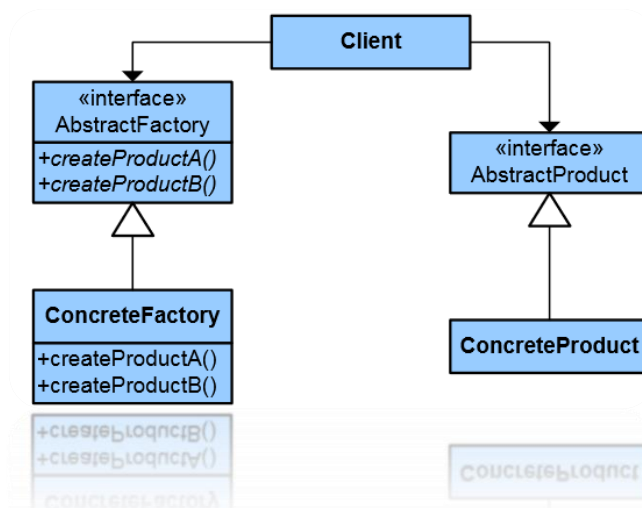
Абстрактната фабрика е създаващ шаблон за дизайн (creational design pattern). Най-голямото му предимство е изолирането на самото създаване на обекти от тяхното използване. Абстрактната фабрика предоставя на клиента възможността да получава абстрактни обекти (най-често през техните интерфейси) като оставя на самите имплементации на абстрактната фабрика да се грижат кои точно обекти да връщат.

Груб пример е абстрактна фабрика за коли, която предоставя възможност за hedge bag и sedan коли. Имплементация на тази фабрика може да е Форд – тя ще върне focus ако и се извика метода за връщане на edge bag и fiesta ако и се извика метода за връщане на sedan.

Използва се най-често в системи, в които често се налага да има промени.

Примера на C# е проекта с име Abstract Factory.

UML диаграма:



Builder

Строителя е създаващ шаблон за дизайн (creational design pattern). Най-голямото му предимство е, че отделя конструирането на сложен обект от неговата имплементация, за да може същия строителен процес да се използва и за създаването на различни обекти.

Клиента създава един директор (класа който определя кой методи на строителя, кога, в какъв ред и как да се извикат и използват). После му подава различни имплементации на строителя (абстрактния клас строител дефинира самите стъпки, които всеки негов наследник трябва да има) и вика конкретните имплементации на методите от родителския клас в определена последователност.

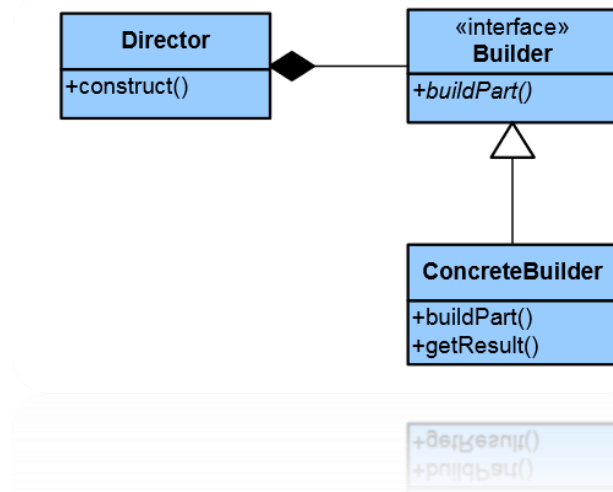
Абстракцията може да се вдигне дори още повече, като се направи базов директор и негови имплементации с различна последователност на извикване на методите от строителя.

Груб пример е директор, който казва на строител как да направи кола – първо да сложи шаши, после гуми и после мотор. Строителя (абстрактния клас) предоставя точно тези три метода. Имплементация на строителя може да е Форд – те изпълняват метода сложи мотор като поставят мотор на Форд и т.н.

Този шаблон за дизайн решава проблема с прекалено многото параметри, необходими за създаването на обекти, конкретната нужда за определна последователност при това създаване на обекти и позволява наличието на различни конструкции.

Примера на C# е проекта с име Builder.

UML диаграма:



Façade

Фасадата е структурен шаблон за дизайн (structural design pattern). Самото име подсказва идеята зад този шаблон за дизайн – улеснява потребителите като скрива сложна логика и операции зад един метод – вместо сами да бъркаме из библиотеката и да инстанциираме класове, да се гирижим за exception-и и подобни, просто използваме даден метод през фасадата.

Груб пример е четенето на файл като стринг. Първо трябва да отворим файла. После да си направим streamreader. После да запазваме някъде това което сме прочели – един stringBuilder. После в while цикъл, докато има следващ ред четем и записваме в stringBuilder-a. После връщаме резултата като викаме ToString метода на stringBuilder-a. Всичко това може да се постигне с един ред – статичния метод на File.ReadAllText представлява фасада за всичко изброено по-горе.

Зад фасадата се скриват всички сложни методи, а тези, които крайния потребител използва, се показват и се правят да се използват лесно.

Примера на C# е проекта с име Facade.

UML диаграма:

