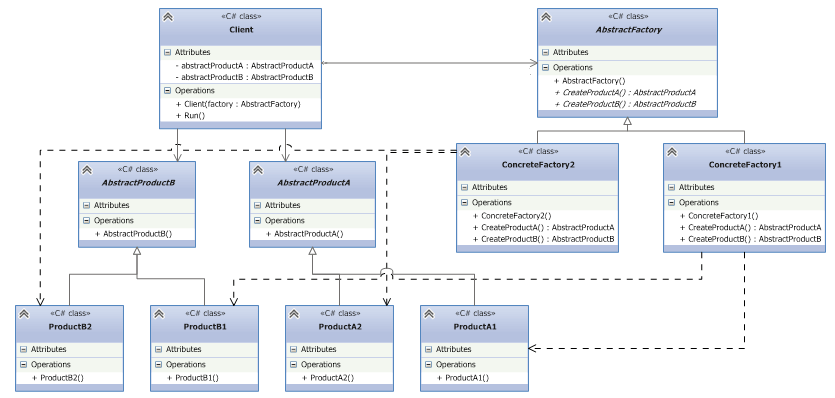
Паттерн Abstract Factory (Абстрактная фабрика)

Паттерн «Абстрактная фабрика» является порождающим паттерном проектирования. Порождающие паттерны предназначены для создания объектов, позволяя системе оставаться независимой как от самого процесса порождения, так и от типов порождаемых объектов.

Сухое определение звучит так: «Абстрактная фабрика» (Abstract Factory) предоставляет интерфейс для создания **семейств взаимосвязанных объектов** с определенными интерфейсами без указания конкретных типов данных объектов. Паттерн можно представить при помощи следующей UML диаграммы:



Теперь рассмотрим все более подробно:

Итак, ключевыми словами в определении «Абстрактной фабрики» являются «создание семейства взаимосвязанных объектов». Рассмотрим такой пример: допустим, мы моделируем работу завода по производству бутылок для различных напитков и у нас есть несколько типов напитков, под которые мы и производим наши бутылки (пускай это будет кола и пепси). Мы объединим их в сущность «Бутылка» и зададим несколько общих составляющих: пускай это будет «крышка» и «этикетка».

Теперь вернемся к диаграмме: клиентом у нас будет бутылка, которая состоит (имеет такие свойства) из абстрактной крышки (Cap) и настолько же абстрактной этикетки (Label). В конструкторе Bottle принимает некую реализацию абстрактной фабрики (BottleFactory). Абстрактная фабрика имеет два метода: создать крышку и создать этикетку. Примечание: крышка и этикетка и есть то самое семейство взаимосвязанных объектов.

Теперь настало время перейти от абстракций к конкретике. Создадим две фабрики: ColaFactory и PepsiFactory, которые будут создавать соответственно крышки и этикетки для колы и пепси. Методы создать крышку и создать этикетку этих фабрик будут возвращать уже не абстрактные крышки и этикетки, а вполне конкретные. Создадим таковые: для колы у нас будет ColaCap и ColaLabel, для пепси – PepsiCap и PepsiLabel.

Готово! Теперь, мы можем создавать бутылки для этих двух напитков просто передавая в конструктор класса Bottle нужную фабрику. Вся прелесть в том, что сам класс Bottle, то есть клиент нашей фабрики никак не зависит от конкретной реализации своих свойств и мы можем с легкостью подменять их реализацию, не затрагивая класс-клиент. Так же нам не составит труда добавить новую фабрику, которая будет «порождать», скажем, бутылки для фанты.

Не смотря на все свои прелести, паттерн «Абстрактная фабрика» все же имеет существенный недостаток, касающийся расширяемости. Если нам захочется добавить в конфигурацию бутылки новый объект, например, форму бутылки, то придется переделывать все классы фабрик, а также и сам класс бутылки.