Одиночка (Singleton)

Шаблон Одиночка гарантирует создание единственного экземпляра объекта некоторого класса. 2 реализации.

Рис. 12 иллюстрирует указанные шаги.

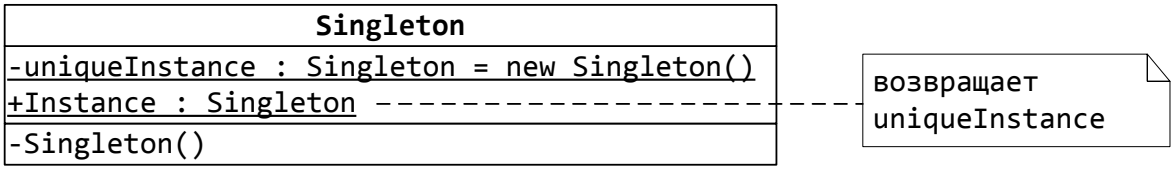

Рис. 12. UML-диаграмма шаблона Одиночка.

Фабричный метод (Factory method)

Фабричный метод занимается созданием объектов. Каждый такой объект принадлежит некому классу, однако все классы либо имеют общего предка, либо реализуют общий интерфейс. Фабричный метод сам решает, какой конкретный класс нужно использовать для создания очередного объекта. Решение принимается либо на основе информации, предоставленной клиентом, либо на основе внутреннего состояния метода.

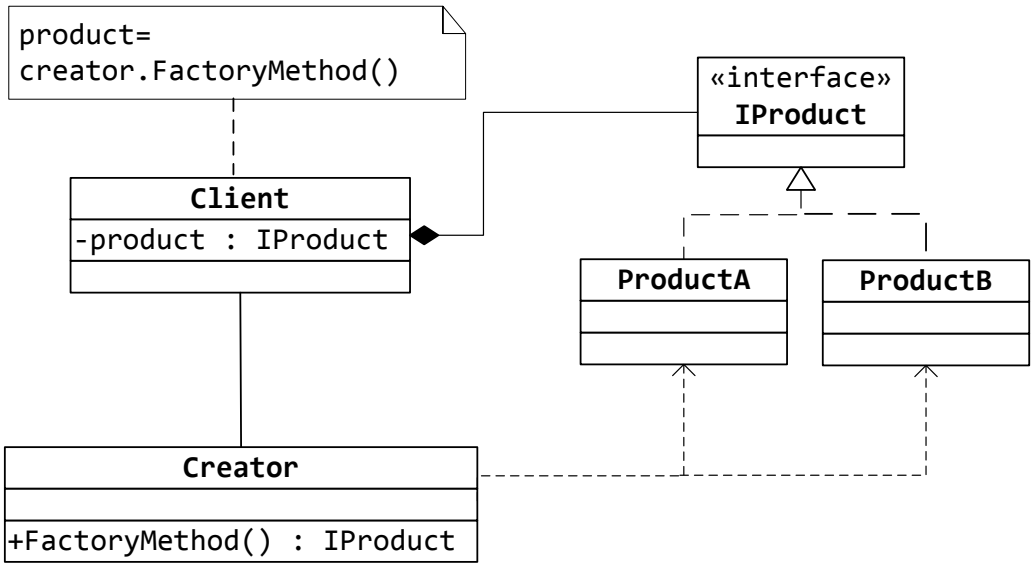

Рис. 11. UML-диаграмма шаблона Фабричный метод.

Прототип (Prototype)

Шаблон Прототип позволяет создавать специальные объекты, ничего не знаю об их классе или деталях создания. Механизм можно описать так: клиенту, инициирующему создание объектов, предоставляются объекты-прототипы. Затем целевые объекты создаются путем клонирования этих объектов-прототипов.

Абстрактная фабрика (Abstract factory)

Шаблон Абстрактная фабрика предназначен для создания объектов, принадлежащих одному набору классов и используемых совместно. Абстрактная фабрика переопределяется в конкретных классах-фабриках, создающих объекты одного набора. Этот шаблон изолирует имена классов и их определения от клиента: единственный способ для клиента получить необходимый объект – это воспользоваться одной из реализаций абстрактной фабрики.

Дизайн шаблона Абстрактная фабрика включает большое число участников, однако он достаточно прост (рис. 13). Клиент хранит ссылку на конкретную фабрику, реализующую интерфейс `IFactory`. Диаграмма показывает, что один из вариантов установки связи клиента и фабрики – это передача объекта-фабрики в конструкторе клиента. Клиент работает с определенным набором объектов, но

использует для этого только реализуемые классами объектов интерфейсы (IProductA и IProductB). Для получения нужного объекта клиент вызывает один из методов фабрики.

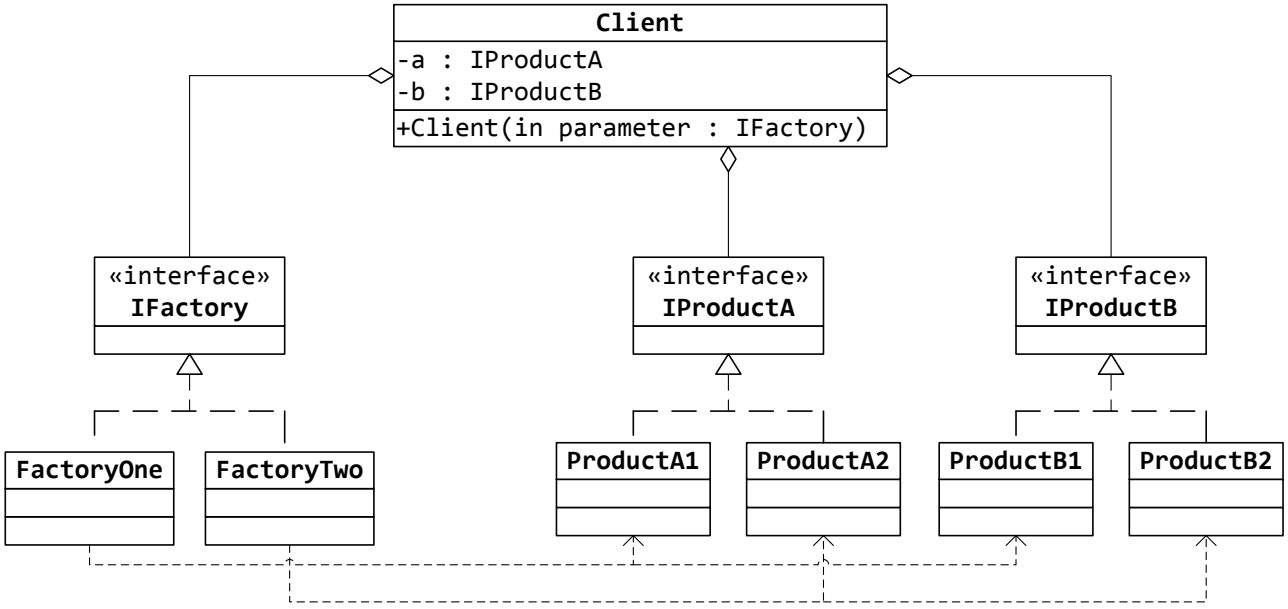

Рис. 13. Шаблон абстрактная фабрика.

Рассмотрим пример кода с реализацией шаблона абстрактная фабрика. Отметим одну важную отличительную особенность представленной реализации шаблона. Вместо набора классов-фабрик и классов для объектов клиента используются универсальные шаблоны (generics).

Строитель (Builder)

Шаблон Строитель позволяет клиенту создавать сложный объект, задавая для него только тип и содержимое. Одинаковый процесс создания способен порождать различные объекты.

Рассмотрим дизайн шаблона Строитель. В его основе лежит компонент Director (*режиссер*), который вызывает объекты-строители. Количество объектов-строителей произвольно, однако все они реализуют интерфейс IBuilder. Строители поставляют элементы для создания финального объекта Product. Как только объект Product будет создан, компонент Director предоставляет его клиенту.

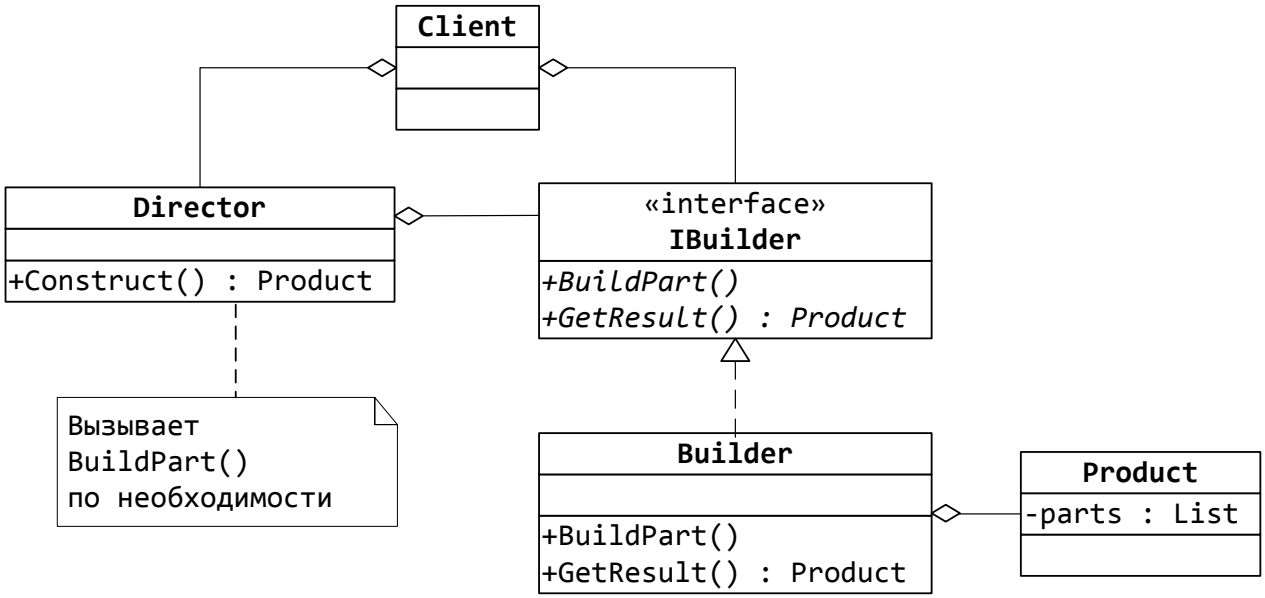

Рис. 14. Дизайн шаблона Строитель.