

# Порождающие паттерны

**Задачи:** быстрая и безопасная генерация объектов и групп объектов

Представители порождающих паттернов:

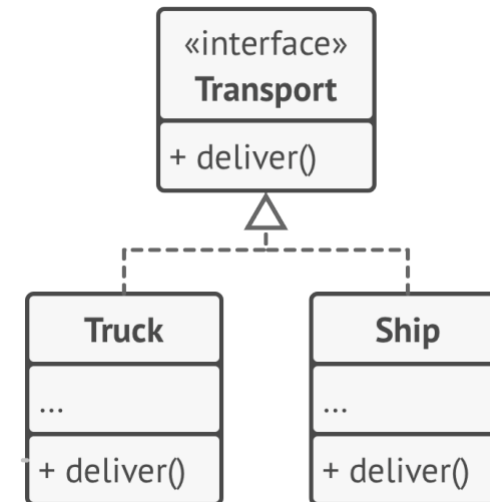
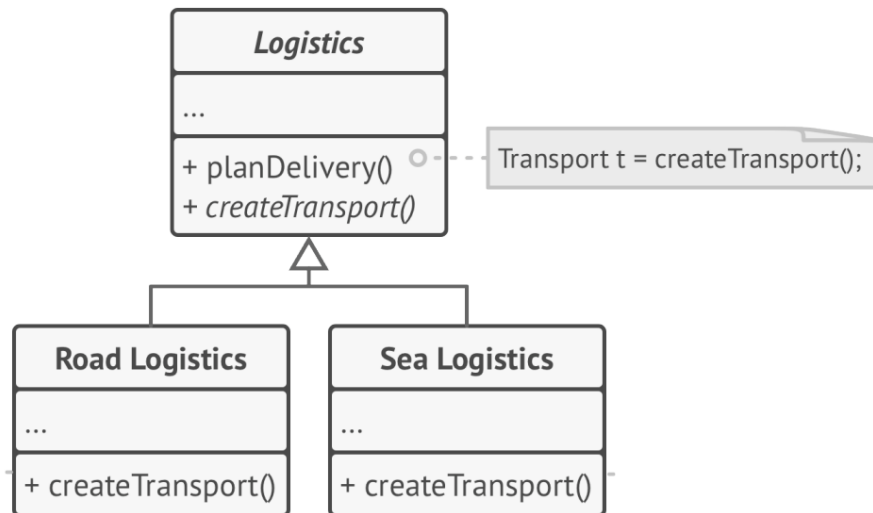
- Фабричный метод
- Абстрактная фабрика
- Строитель
- Одиночка
- Прототип

# Фабричный метод

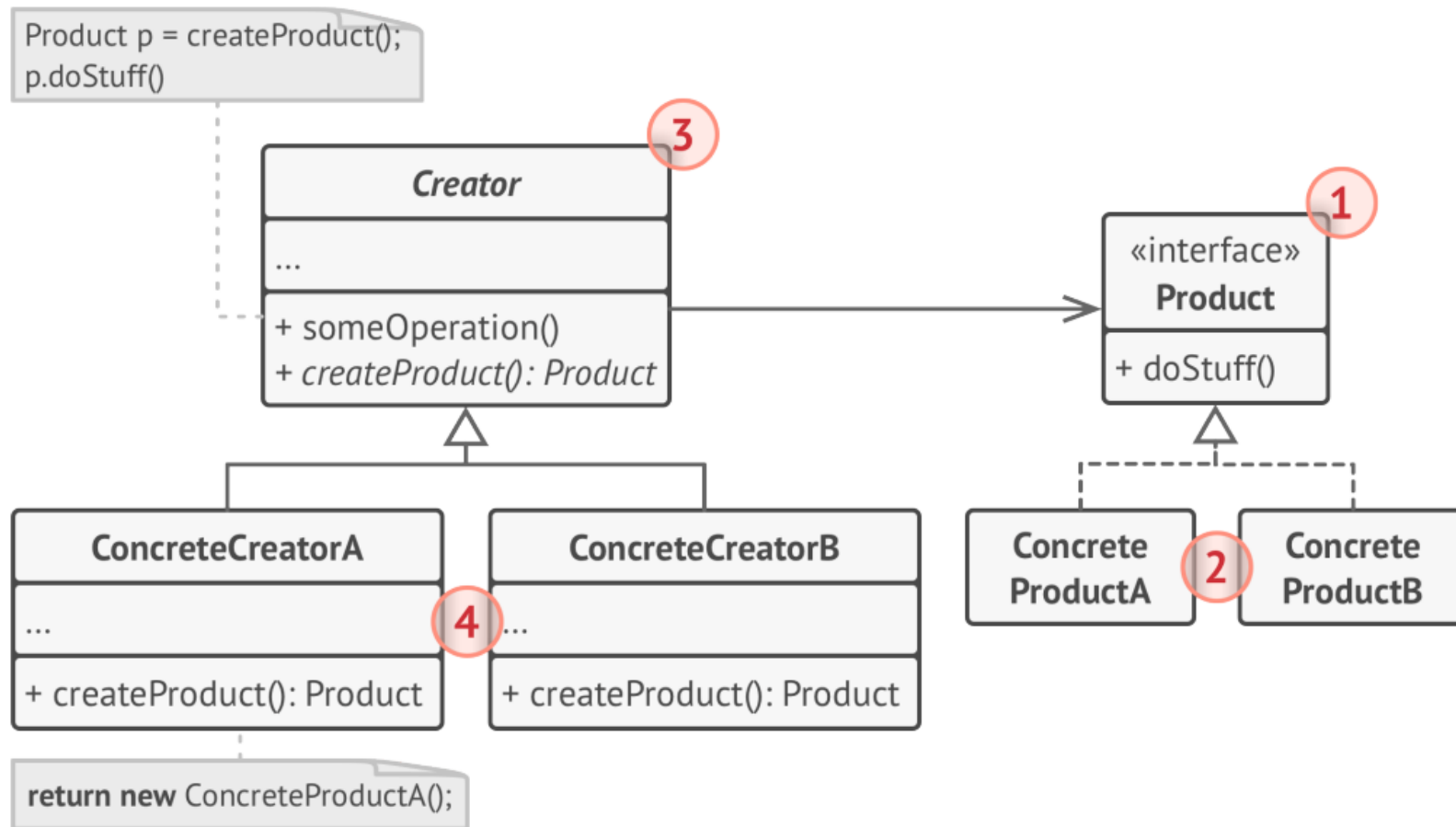
Определяет интерфейс для создания объектов суперкласса, но непосредственное решение о том, объект какого класса создавать происходит в подклассах

Особенности:

- Объекты создаются не напрямую через new а через вызов фабричного метода
- Подклассы сами определяют тип создаваемого объекта



# Структура фабричного метода



1. Интерфейс «продукт» нужен для однотипного определения генерируемых объектов
2. Определенная реализация конкретных классов
3. Содержит фабричный метод для генерации объектов
4. Подклассы, которые по своему реализуют фабричный метод

## Особенности применения

1. При неопределенности типов будущих объектов
2. При необходимости расширения другими пользователями функциональности ПО
3. Для экономии системных ресурсов (повторно используются ранее созданные объекты)

## Достоинства и недостатки

+

- Нет привязки к конкретным классам продуктов
- Код производства в одном месте
- Добавление новых продуктов легче
- Поддержка **принципа открытости/закрытости**

-

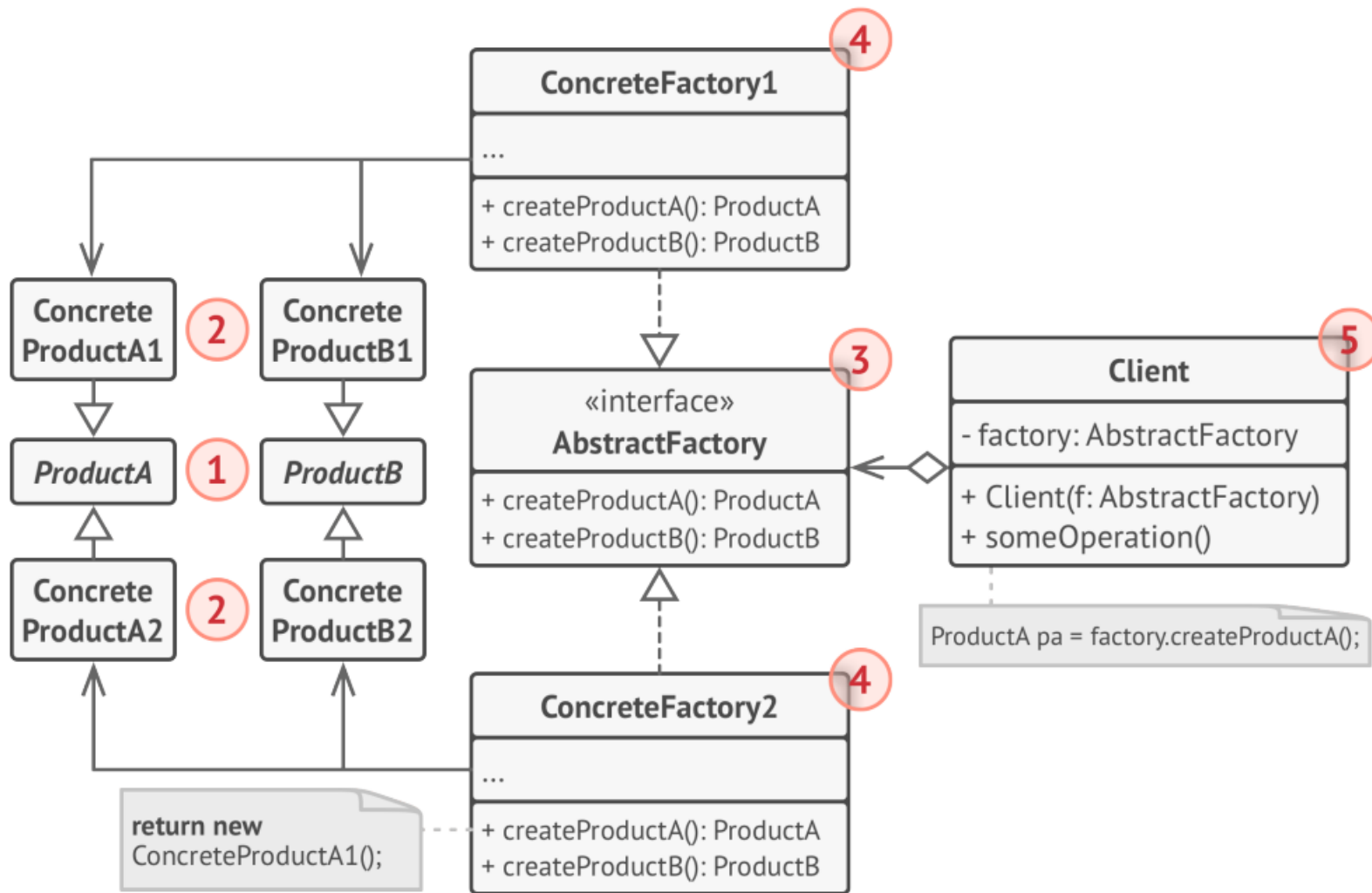
- Возможно возникновение параллельных иерархий классов

- определяет интерфейс для создания групп связанных объектов с определёнными интерфейсами без указания их конкретных типов

Особенности:

- Выделяются общие интерфейсы для отдельных продуктов группы
- Создается абстрактная фабрика – общий интерфейс, определяющий методы создания всех продуктов группы

# Структура абстрактной фабрики



1. Абстрактные продукты
2. Конкретные продукты
3. Абстрактная фабрика
4. Конкретные фабрики

## Особенности применения

1. Программа должна работать с разными видами связанных друг с другом продуктов, не завися от конкретных классов продуктов
2. В программе уже используется Фабричный метод, но очередные изменения предполагают введение новых типов продуктов

## Достоинства и недостатки

+

- Код избавлен от привязки к конкретным классам продуктов
- Код производства объектов выделен в одно место
- Простое добавление новых продуктов в программу
- Реализует **принцип открытости/закрытости**

-

- Усложняется код, увеличивается количество классов
- Требуется наличие всех типов продуктов во всех группах