Java Design Patterns

Prototype

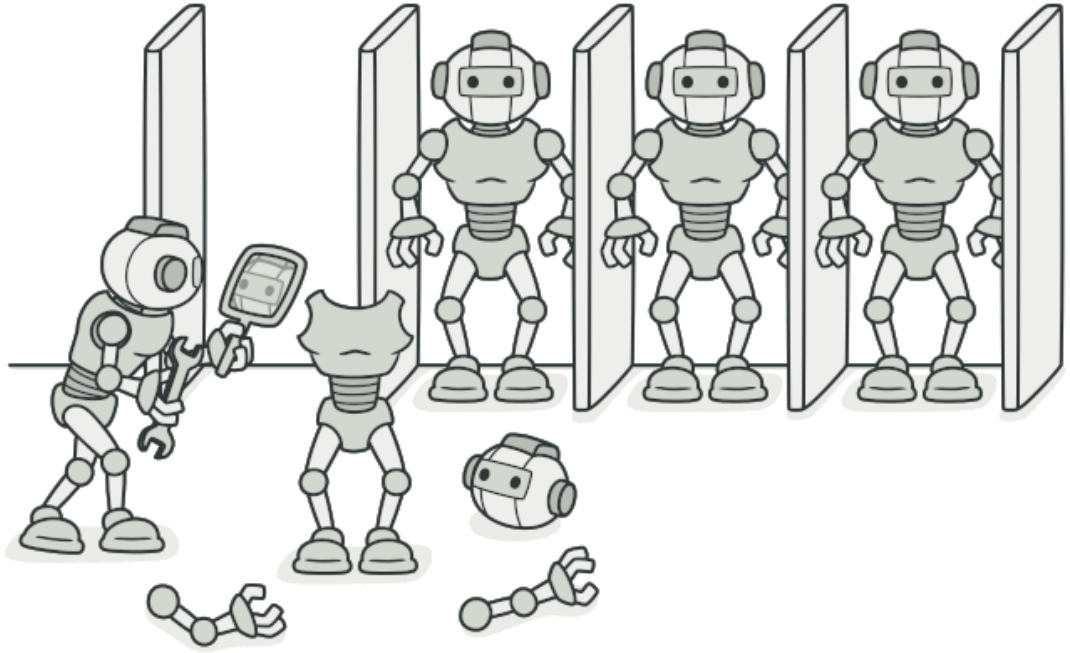
Тема

Prototype

Суть паттерна

Прототип

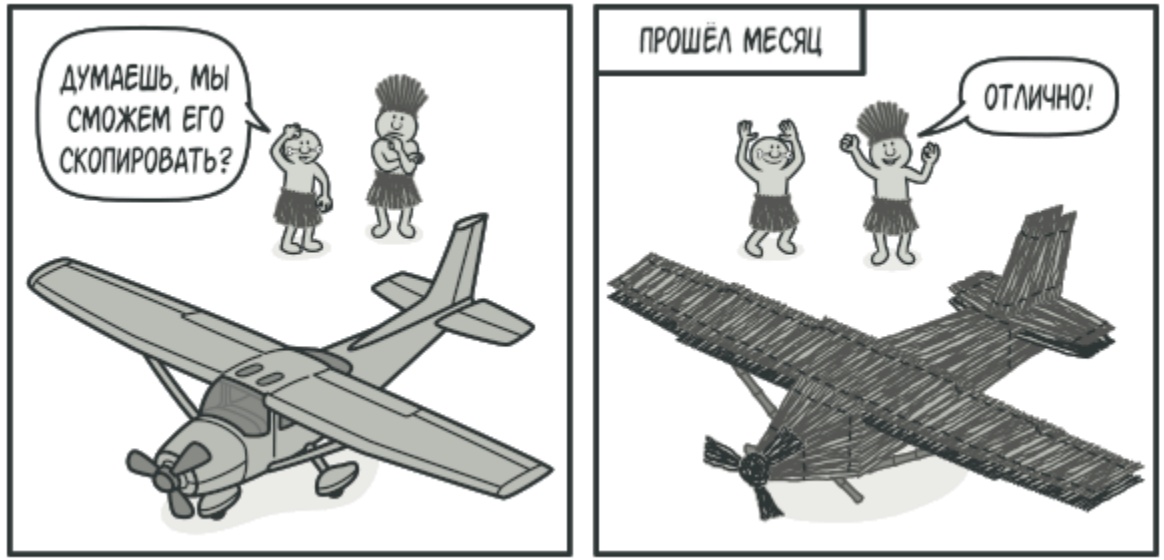
Прототип — это порождающий паттерн проектирования, который позволяет копировать объекты, не вдаваясь в подробности их реализации.



Проблема

Постановка задачи

У вас есть объект, который нужно скопировать. Как это сделать? Нужно создать пустой объект такого же класса, а затем поочерёдно скопировать значения всех полей из старого объекта в новый.



Решение задачи

Паттерн Прототип поручает создание копий самим копируемым объектам.
Он вводит общий интерфейс для всех объектов, поддерживающих клонирование. Это позволяет копировать объекты, не привязываясь к их конкретным классам. Обычно такой интерфейс имеет всего один метод clone.

Аналоги из жизни

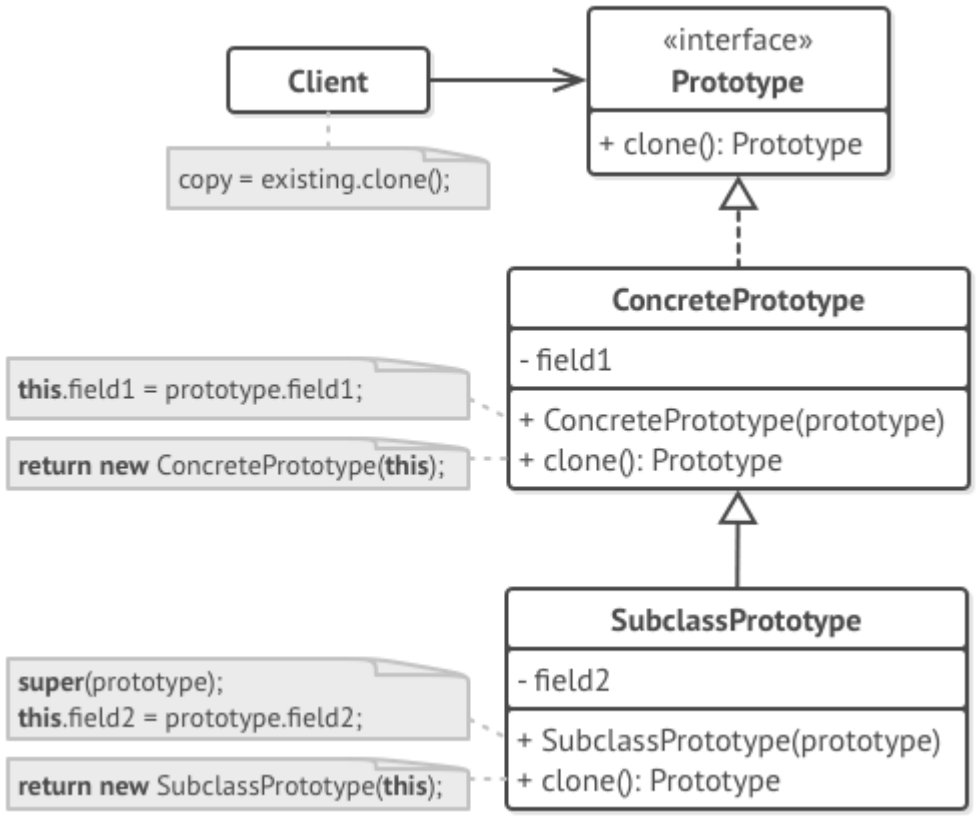
Пример

В промышленном производстве прототипы создаются перед основной партией продуктов для проведения всевозможных испытаний. При этом прототип не участвует в последующем производстве, отыгрывая пассивную роль.

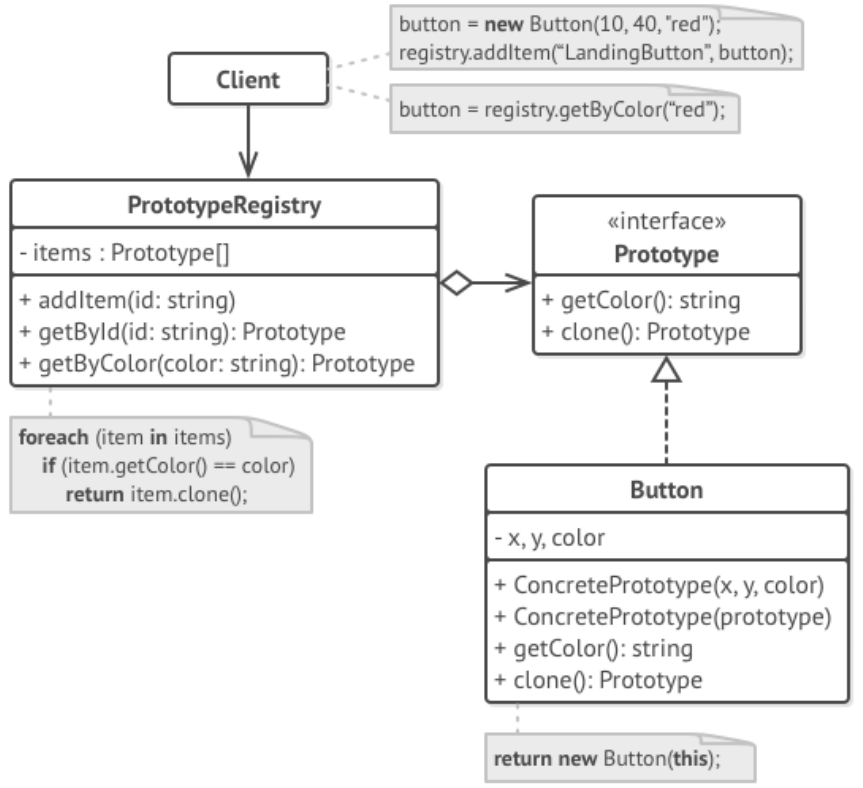
Структура

Две структуры прототипа

Базовая реализация



Реализация с общим хранилищем прототипов



Применимость

Правила применения

1. Когда ваш код не должен зависеть от классов копируемых объектов.
2. Когда вы имеете уйму подклассов, которые отличаются начальными значениями полей. Кто-то создал эти классы, чтобы быстро создавать объекты с определённой конфигурацией.

Шаги реализации

Алгоритм реализации

1. Создайте интерфейс прототипов с единственным методом clone. Если у вас уже есть иерархия продуктов, метод клонирования можно объявить непосредственно в каждом из её классов.
2. Добавьте в классы будущих прототипов альтернативный конструктор, принимающий в качестве аргумента объект текущего класса. Этот конструктор должен скопировать из поданного объекта значения всех полей, объявленных в рамках текущего класса, а затем передать выполнение родительскому конструктору, чтобы тот позаботился об остальных полях.
3. Метод клонирования обычно состоит всего из одной строки: вызова оператора new с конструктором прототипа. Все классы, поддерживающие клонирование, должны явно определить метод clone, чтобы подать собственный класс в оператор new. В обратном случае, результатом клонирования окажется объект родительского класса.
4. Опционально, создайте центральное хранилище прототипов. В нём можно хранить вариации объектов, возможно даже одного класса, но по-разному настроенных.

Преимущества и недостатки

Плюсы и недостатки

Плюсы:

- Позволяет клонировать объекты, не привязываясь к их конкретным классам.
- Меньше повторяющегося кода инициализации объектов.
- Ускоряет создание объектов.
- Альтернатива созданию подклассов для конструирования сложных объектов.

Минусы:

- Сложно клонировать составные объекты, имеющие ссылки на другие объекты.

Отношения с другими паттернами

- Многие архитектуры начинаются с применения Фабричного метода (более простого и расширяемого через подклассы) и эволюционируют в сторону Абстрактной фабрики, Прототипа или Строителя (более гибких, но и более сложных).
- Классы Абстрактной фабрики чаще всего реализуются с помощью Фабричного метода, хотя они могут быть построены и на основе Прототипа.
- Если Команду нужно копировать перед вставкой в историю выполненных команд, вам может помочь Прототип.
- Архитектура, построенная на Компоновщиках и Декораторах, часто может быть улучшена за счёт внедрения Прототипа. Он позволяет клонировать сложные структуры объектов, а не собирать их заново.
- Прототип не опирается на наследование, но ему нужна сложная операция инициализации. Фабричный метод наоборот, построен на наследовании, но не требует сложной инициализации.
- Снимок иногда можно заменить Прототипом, если объект, чьё состояние требуется сохранять в истории, довольно простой, не имеет активных ссылок на внешние ресурсы, либо их можно легко восстановить.
- Абстрактная фабрика, Строитель и Прототип могут быть реализованы при помощи Одиночки.

Информационный видеосервис для разработчиков программного обеспечения

ITVDN

IT VIDEO DEVELOPERS NETWORK

TestProvider PROMETRIC TEST CENTER CyberBionic systematics Microsoft Partner Silver Learning Windows Azure Cloud Partner EBA European Business Association