**ПРОТОТИП**: позволяет копировать объекты, не вдаваясь в подробности их реализации.

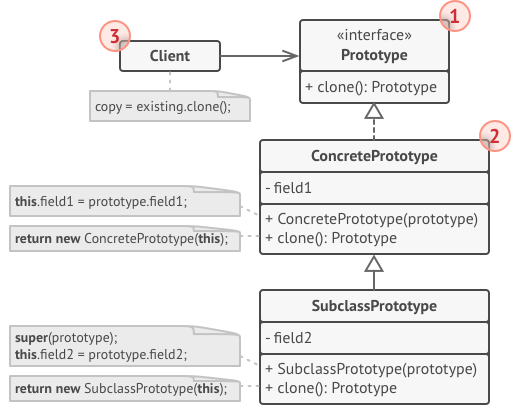
ПРИМЕНЕНИЕ:

- когда код не должен зависеть от классов копируемых объектов

- когда вы имеете уйму подклассов, которые отличаются начальными значениями полей

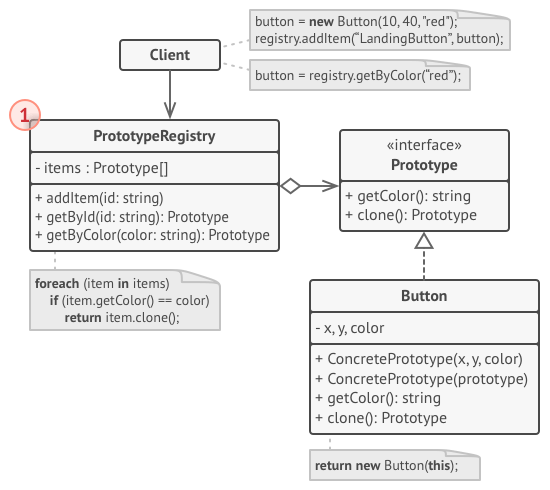
- кто-то мог создать все эти классы, чтобы иметь возможность легко порождать объекты с определённой конфигурацией

РЕАЛИЗАЦИЯ (БАЗОВАЯ):

**1. ИНТЕРФЕЙС ПРОТОТИПОВ**: описывает операции клонирования. В большинстве случаев — это единственный метод clone

**2. КОНКРЕТНЫЙ ПРОТОТИП**: реализует операцию клонирования самого себя. Помимо банального копирования значений всех полей, здесь могут быть спрятаны различные сложности, о которых не нужно знать клиенту. Например, клонирование связанных объектов, распутывание рекурсивных зависимостей и т.д.

**3. КЛИЕНТ**: создаёт копию объекта, обращаясь к нему через общий интерфейс прототипов



РЕАЛИЗАЦИЯ (С ОБЩИМ ХРАНИЛИЩЕМ):

**1. ХРАНИЛИЩЕ ПРОТОТИПОВ**: облегчает доступ к часто используемым прототипам, храня предсозданный набор эталонных, готовых к копированию объектов. Простейшее хранилище может быть построено с помощью хеш-таблицы вида "имя-прототипа → прототип". Но для удобства поиска прототипы можно маркировать и другими критериями, а не только условным именем.

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