

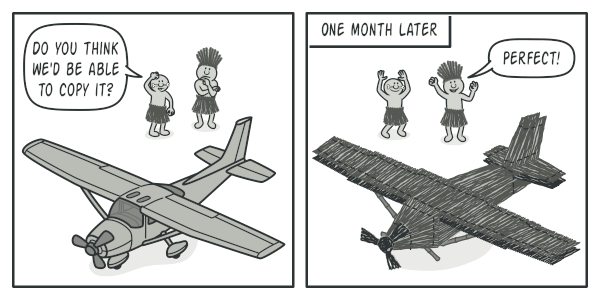
1. **Tổng quan**

Tên: Mẫu Prototype

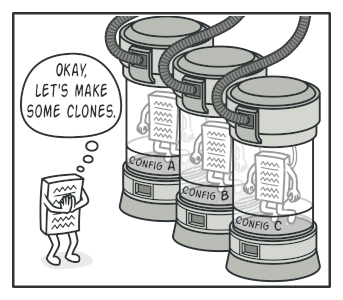
Phân loại: Thuộc phân nhóm Creational Pattern

Mục đích, ý nghĩa: Cho phép tạo ra các đối tượng mới dựa trên một nguyên mẫu đã tồn tại, mà không cần biết chi tiết về cách tạo ra chúng.

1. **Motivation**



Vấn đề: Trong một số trường hợp, việc tạo ra các đối tượng mới có thể tốn kém về tài nguyên hoặc phức tạp về logic. Đồng thời, các đối tượng này có thể chứa các thuộc tính hoặc trạng thái mà việc sao chép trực tiếp có thể gây ra sự không nhất quán hoặc ảnh hưởng đến hiệu suất.



Giải pháp: Sử dụng mẫu Prototype để tạo ra các đối tượng mới bằng cách sao chép các đối tượng đã tồn tại, giúp giảm bớt chi phí tạo đối tượng và đảm bảo tính nhất quán trong dữ liệu.

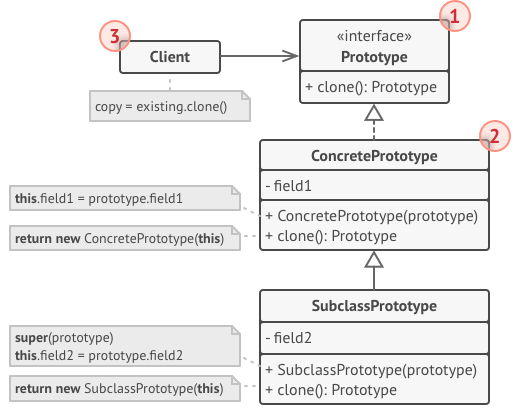
1. **Khả năng ứng dụng**

Mẫu Prototype thường được sử dụng trong các tình huống sau:

* Khi việc tạo ra các đối tượng mới đòi hỏi nhiều tài nguyên hoặc có thể làm giảm hiệu suất của ứng dụng.
* Khi cần tạo ra các đối tượng mới có cùng cấu trúc và thuộc tính với các đối tượng đã tồn tại.

1. **Đặc điểm:**
2. **Cài đặt cơ bản**

Cấu trúc mẫu:



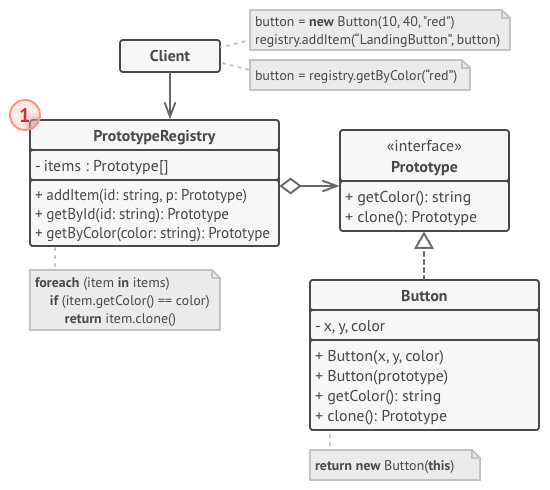
Ý nghĩa của từng thành viên:

* Prototype interface khai báo phương thức clone.
* Concrete Prototype thực hiện phương thức clone. Ngoài việc sao chép dữ liệu của đối tượng gốc, phương thức này cũng xử lý các edge case trong quá trình nhân bản như linked objects,…
* Client có thể tạo bản sao của bất kỳ đối tượng nào tuân theo interface prototype.
* Các đối tượng có thể có nhiều cấu hình khác nhau, việc sử dụng cloning có thể thay thế cho việc kế thừa.

Sự cộng tác:

Client sử dụng interface chung để làm việc với tất cả đối tượng hỗ trợ cloning, làm cho mã nguồn không phụ thuộc vào lớp cụ thể.

1. **Cài đặt bằng registry:**



Prototype Registry giúp dễ dàng truy cập các prototype thường xuyên được sử dụng. Nó lưu trữ một bộ sưu tập các đối tượng đã được xây dựng sẵn và sẵn sàng để được sao chép. Registry prototype đơn giản nhất là một bảng băm từ (tên → prototype). Tuy nhiên, có thể xây dựng một registry tốt hơn để tìm kiếm với các tiêu chí khác ngoài tên.

1. **Các hệ quả mang lại:**

* Ưu điểm:
  + Sao chép đối tượng mà không liên kết với lớp cụ thể.
  + Giảm bớt mã khởi tạo lặp đi lặp lại.
  + Tạo đối tượng phức tạp một cách thuận tiện hơn.
  + Cung cấp một phương án thay thế cho kế thừa khi xử lý các cấu hình mặc định cho đối tượng phức tạp.
* Nhược điểm:
  + Sao chép đối tượng phức tạp có tham chiếu tuần hoàn có thể rất phức tạp.

1. **Các chú ý liên quan đến cài đặt, demo, mã nguồn minh họa**

Khi triển khai mẫu Prototype, đảm bảo rằng đối tượng Prototype triển khai phương thức clone hoặc copy constructor một cách đúng đắn và đầy đủ.

Đối tượng Prototype thường cung cấp một phương thức để khởi tạo bản sao của chính nó.

Đảm bảo rằng các thuộc tính của đối tượng Prototype được sao chép một cách đầy đủ và chính xác.

1. **Nêu ra các ví dụ thực tế**

Hệ thống quản lý tài nguyên: Khi cần tạo ra nhiều bản sao của các tài nguyên như tệp tin, hồ sơ hoặc đối tượng cấu trúc dữ liệu phức tạp, việc sử dụng mẫu Prototype có thể giúp giảm bớt tài nguyên và tăng hiệu suất.

Các trò chơi điện tử: Trong phát triển trò chơi, việc tạo ra nhiều đối tượng có cùng thuộc tính hoặc cấu trúc có thể được thực hiện thông qua mẫu Prototype, giúp tăng hiệu suất và giảm thiểu tài nguyên.

1. **Các mẫu liên quan**

Nhiều thiết kế bắt đầu bằng Phương thức Factory (ít phức tạp hơn và linh hoạt hơn thông qua các lớp con) và tiến triển về hướng Abstract Factory, Prototype hoặc Builder (linh hoạt hơn, nhưng phức tạp hơn).

Các lớp Abstract Factory thường dựa trên một tập hợp các Phương thức Factory, nhưng cũng có thể sử dụng Prototype để hợp thành các phương thức trên các lớp này.

Prototype có thể giúp khi bạn cần lưu các bản sao của các Commands.

Các thiết kế sử dụng nhiều Composite và Decorator thường có lợi từ việc sử dụng Prototype, cho phép sao chép các cấu trúc phức tạp thay vì xây dựng lại chúng từ đầu.

Prototype không dựa trên kế thừa, vì vậy nó không có nhược điểm của nó nhưng Prototype yêu cầu một bước khởi tạo phức tạp của đối tượng được sao chép. Phương thức Factory dựa trên kế thừa nhưng không yêu cầu một bước khởi tạo.

Đôi khi Prototype có thể là một phương án đơn giản để thay thế Memento. Trong trường hợp trạng thái của đối tượng muốn lưu khá đơn giản và không có các liên kết với tài nguyên bên ngoài, hoặc các liên kết dễ tái thiết lập.

Abstract Factories, Builders và Prototypes có thể được triển khai như Singletons.