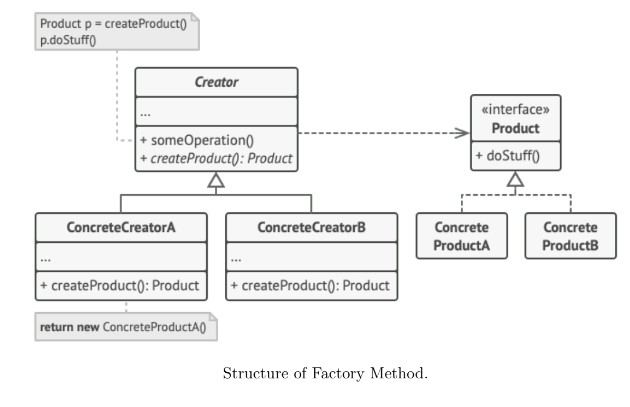
**Factory pattern**

1. **Intent**

Factory pattern là mô hình design sáng tạo (creational design pattern), cung cấp một giao diện cho việc tạo ra những đối tượng trong một superclass, nhưng cũng đồng thời cho phép các lớp phụ (sub class) sửa đổi kiểu của các đối tượng được tạo ra.

1. **Structure**



* Lớp Product gọi một **giao diện**, giao diện này là chung cho tất cả các đối tượng mà có thể được tạo ra bởi lớp creator và các lớp phụ của nó.
* Các lớp **Concrete Products** là các ứng dụng khác nhau của giao diện product.
* Lớp **Creator** gọi một phương thức factory mà trả về những đối tượng product mới. Kiểu trả về của phương thức này phải khớp với giao diện product.

**(Có thể gọi phương thức factory ở kiểu abstract để bắt tất cả các lớp phụ sử dụng dạng phương thức riêng của chúng.** Ngoài ra, phương thức factory mặc định có thể trả về một số kiểu product mặc định.)

* Các lớp **Concrete Creators** ghi đè lên phương thức facory mặc định để trả về một kiểu product khác. Ghi nhớ là phương thức factory không cần phải luôn luôn tạo ra các trường hợp mới. Nó có thể trả về các đối tượng đã có trước từ một cache, một nhóm đối tượng, hay từ một nguồn khác.

1. **When do you use it?**

* Use the Factory Method when you don’t know beforehand the exact types and depen-dencies of the objects your code should work with.
* Use the Factory Method when you want to provide users of your library or framework with a way to extend its internal components.
* Use the Factory Method when you want to save system resources by reusing existing objects instead of rebuilding them each time.