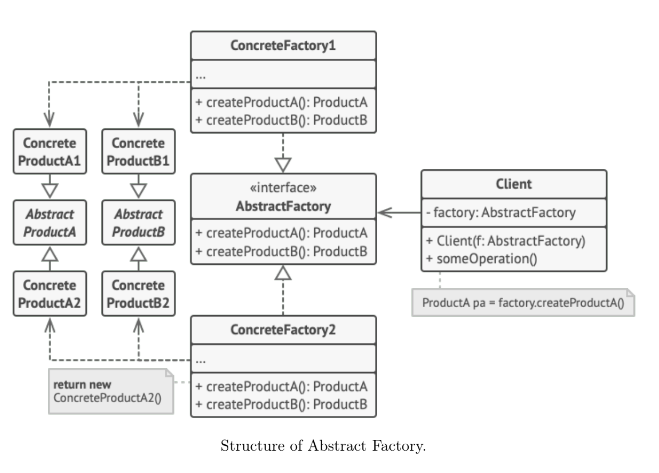
**Abstract Factory pattern**

1. **Intent**

Abstract Factory là một mô hình sáng tạo (creational pattern), cho phép người dùng cho ra một loạt các đối tượng có liên quan với nhau mà không phân biệt cụ thể các lớp concrete của chúng.

1. **Structure**



* Các **Abstract Products** gọi các giao diện cho một bộ các products (mặt hàng ig) khác nhau nhưng lại liên quan tới nhau, tạo ra một product family (idk how to translate this)
* Các lớp **Concrete Products** là một loạt các triển khai (implementations) của các abstract products.
* Giao diện **Abstract factory** gọi một bộ các phương thức cho việc tạo ra các abstract products.
* Các lớp **Concrete Factories** thực hiện các phương thức tạo của lớp abstract factory. Mỗi concrete factory tương ứng với một kiểu product cụ thể và chỉ tạo ra các kiểu product đó.
* Tuy các concrete factory khởi tạo (instantiate) các concrete products, các phương thức tạo của chúng phải trả về abstract products tương ứng. Như vậy, client code mà sử dụng một factory không bị lẫn với một kiểu product cụ thể được tạo ra bởi một factory. Lớp **client** có thể làm việc với bất cứ kiểu factory/product nào, miễn khi là nó giao tiếp được với các đối tượng của nó thông qua giao diện abstract

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

* Use the Abstract Factory when your code needs to work with various families of related products, but you don’t want it to depend on the concrete classes of those products—they might be unknown beforehand or you simply want to allow for future extensibility.
* The Abstract Factory provides you with an interface for creating objects from each class of the product family. As long as your code creates objects via this interface, you don’t have to worry about creating the wrong variant of a product which doesn’t match the products already created by your app.