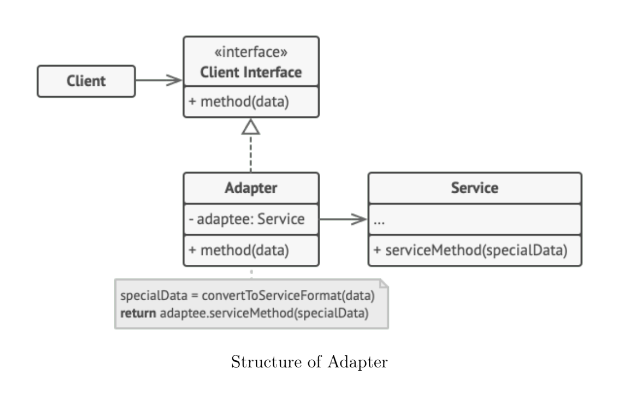
**Adapter pattern**

1. **Intent**

Pattern Adapter là mô hình cấu trúc (structural pattern) giúp các đối tượng có các (giao diện) interface khác nhau, không liên quan tới nhau, làm việc cùng nhau

1. **Structure**

****

* **Client** là lớp chứa các logic có trước của chương trình.
* **Client interface** miêu tả một giao thức mà các lớp khác phải tuân theo để có thể làm việc được với client code.
* **Service** là một lớp hữu dụng. Client không sử dụng trực tiếp được lớp này bởi nó có một interface khác.
* **Adapter** là một lớp mà có thể làm việc với lớp client và cả lớp service: nó làm việc được với giao diện client, trong khi sử bao bọc lớp service. Lớp adapter nhận “tín hiệu” (or calls) từ giao diện adapter rồi dịch chúng thành “tín hiệu” (or calls) đến đối tượng service đang được bọc trong định dạng mà đối tượng có thể hiểu được.
* Mã của client không bị lẫn với lớp adapter miễn là nó làm việc với adapter thông qua giao diện client. Nhờ vậy, ta có thể cho vào nhiều adapter mới vào chương trình mà không làm hỏng mã client có trước. Do đó, khi giao diện service bị thay đổi hay bị thay thế, ta chỉ cần tạo ra lớp Adapter mới mà không cần điều chỉnh mã client.

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

Use it when you have a bunch of existing classes that has incompatible interfaces   
Or when you want to reuse several existing subclasses that lack common functionality that can’t be added to the superclass,