**Protocol**

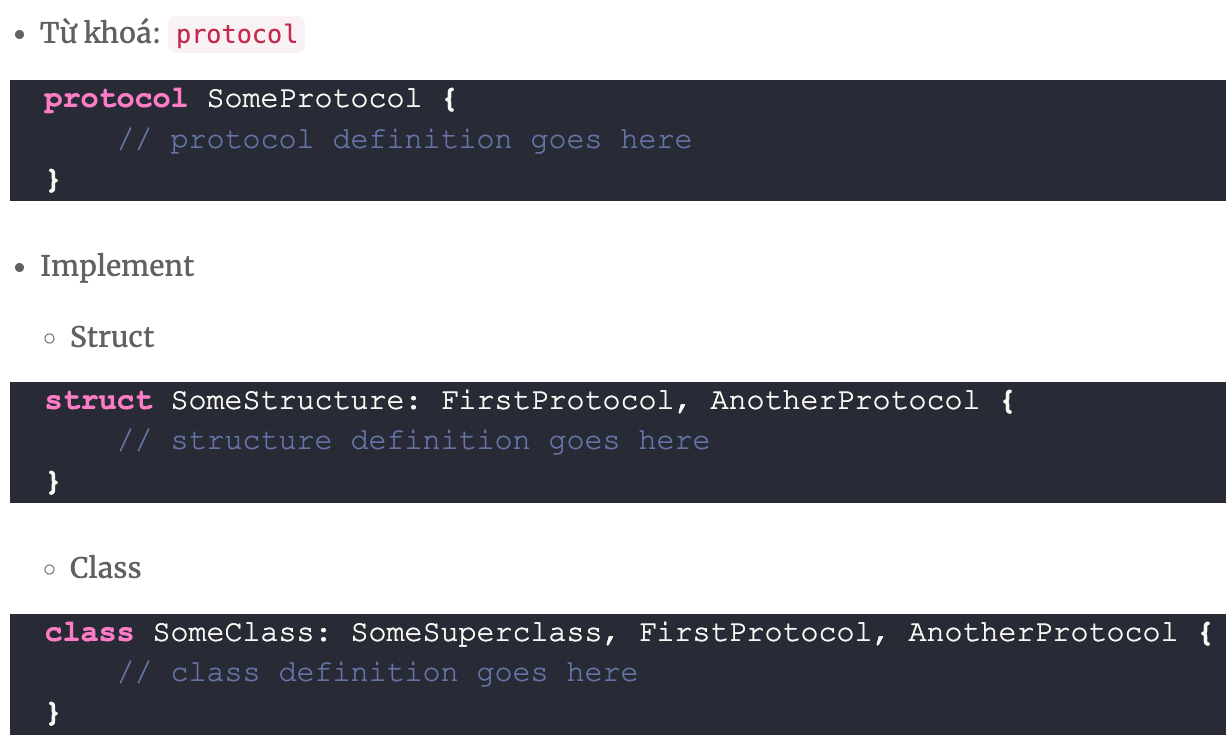
1. **Định nghĩa**

- Bạn có thể hiểu protocol dịch ra đầy đủ và đúng nghĩa là giao thức.

Mình có thể hiểu khai quát về protocol qua một số gạch đầu dòng sau: Protocol

* Là một kiểu interface
* Mang tính chất trừu tượng
* Khai báo các properties và các phương thức
* Không cần định nghĩa chúng
* Implement được vào class/ struct/ enum
* Implement được vào nhiều chứ không phải 1
* Có thể xem như là một kiểu dữ liệu

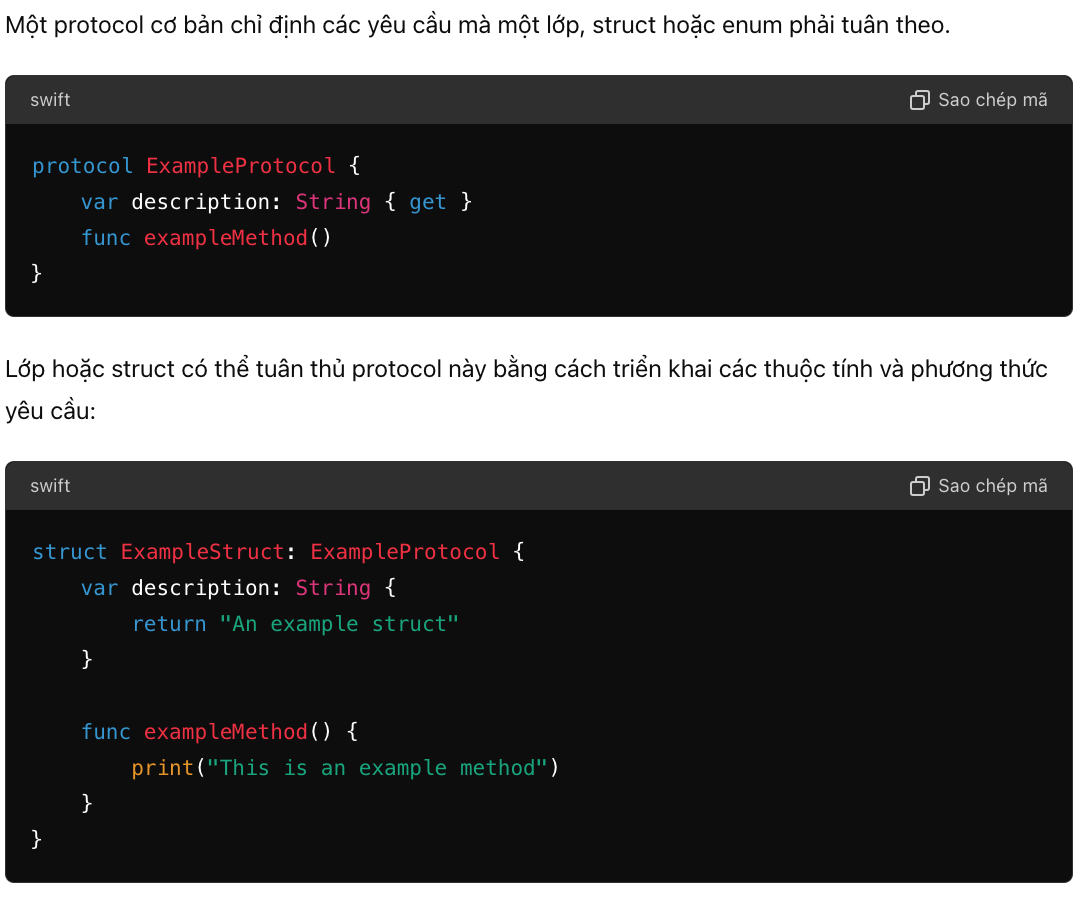
1. **Cú pháp**



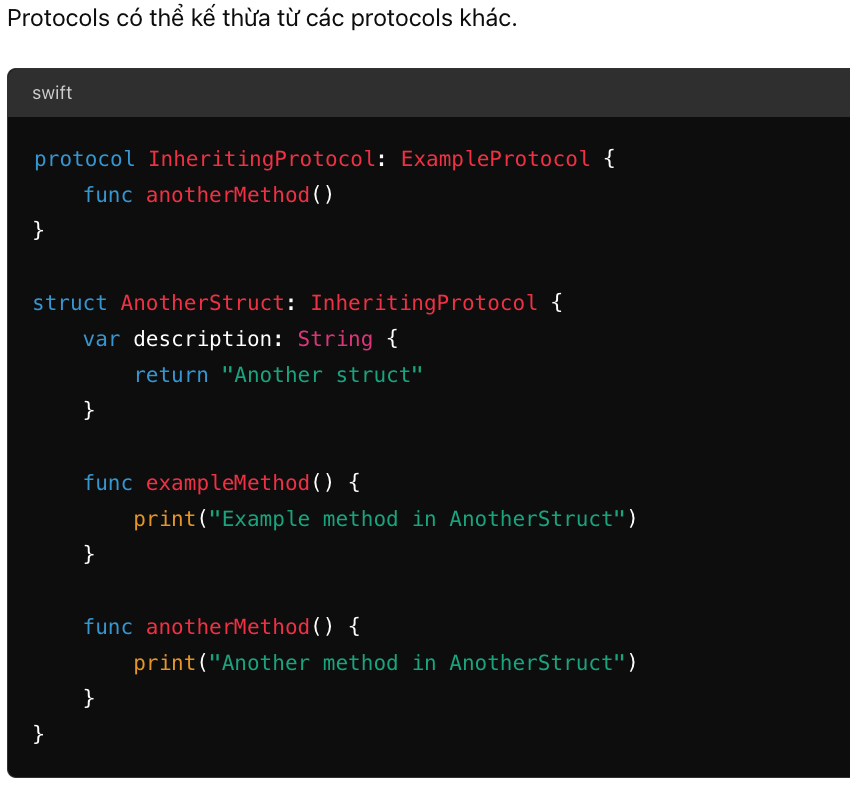
* Việc Implement thì cũng được xem là một hình thức đa kế thừa để tăng cường sức mạnh cho class/struct.
* Khi class/struct implement một protocol nào đó thì các phương thức và thuộc tính của protocol phải được định nghĩa lại ở class/struct đó

1. **Cách thức thao tác**

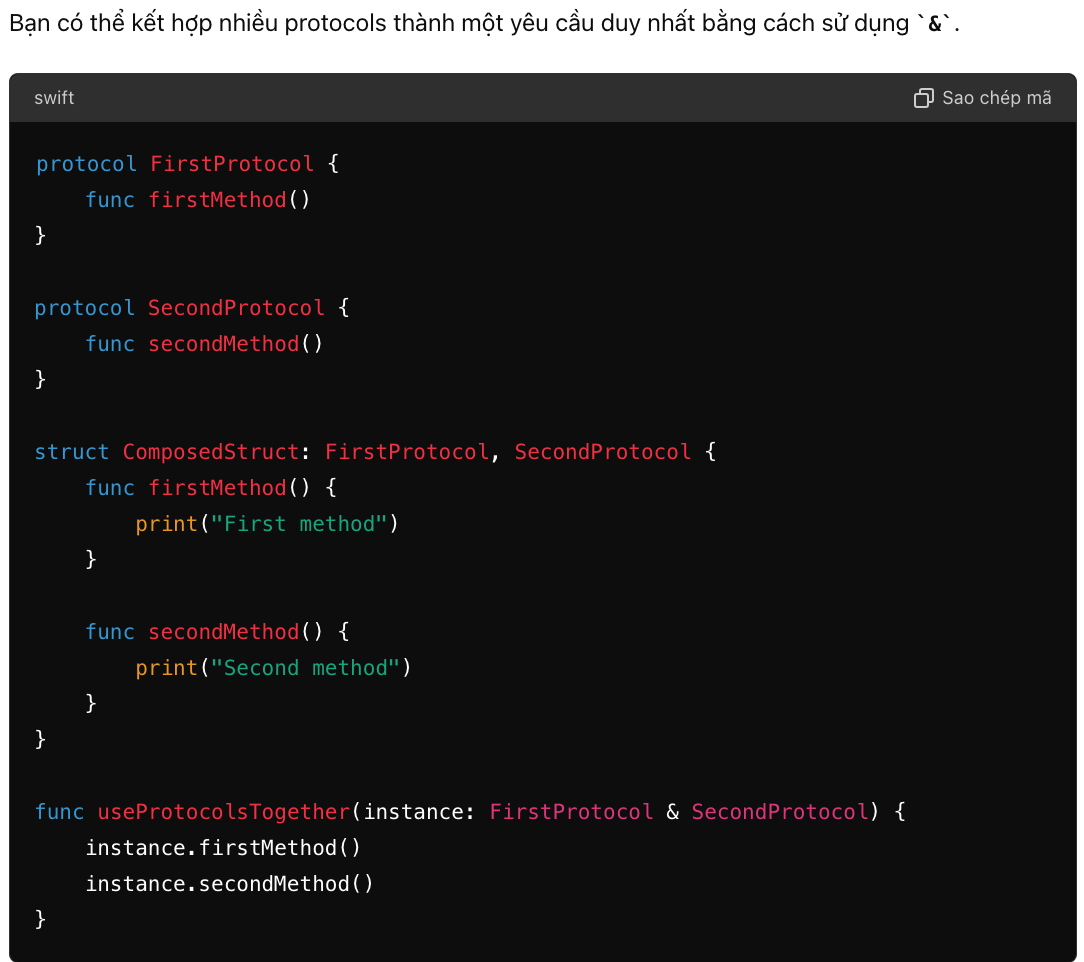
* **Cơ bản**



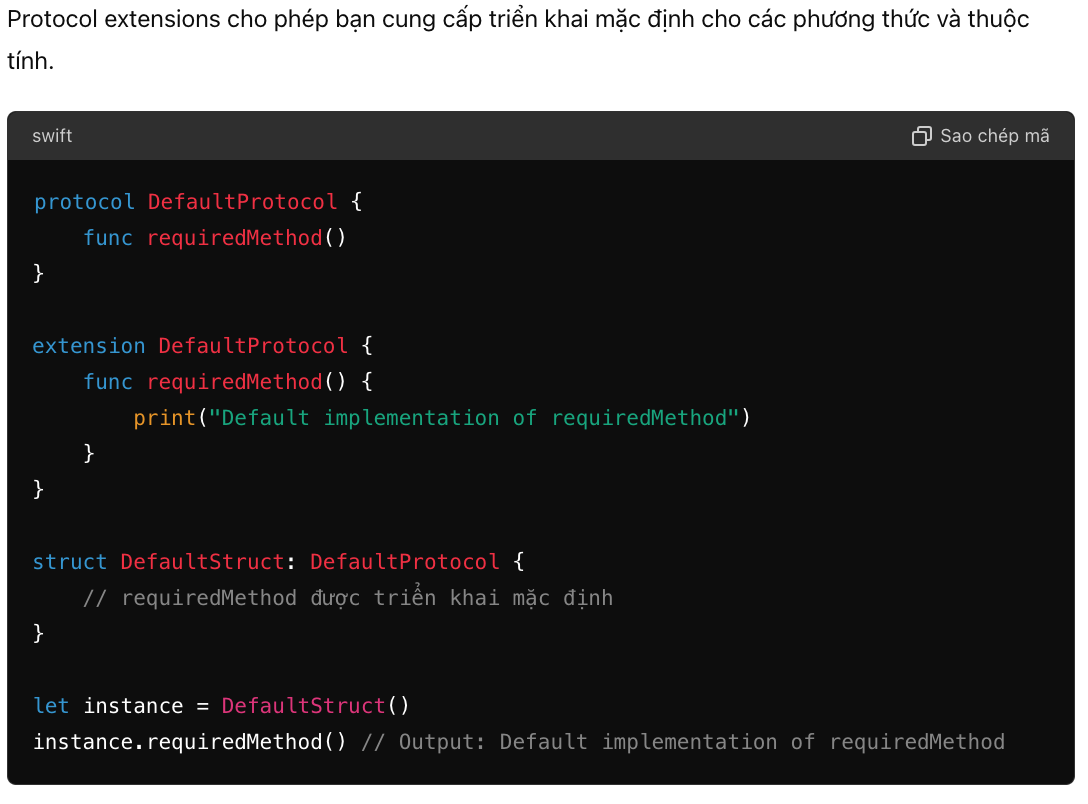
* **Protocol** **Inheritance (Kế thừa)**



* **Protocol Composition**



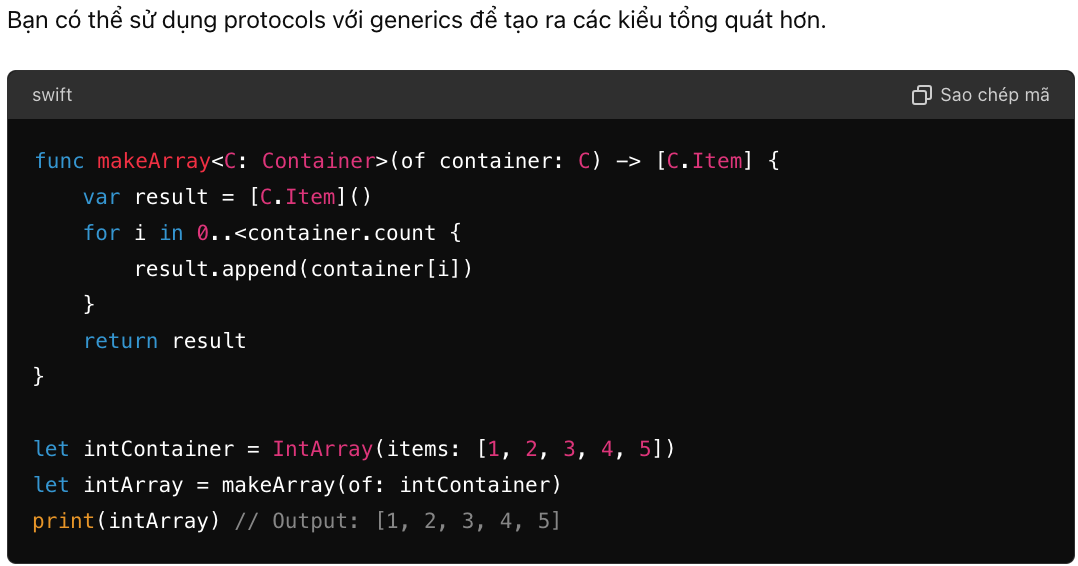
* **Protocol Extensions**



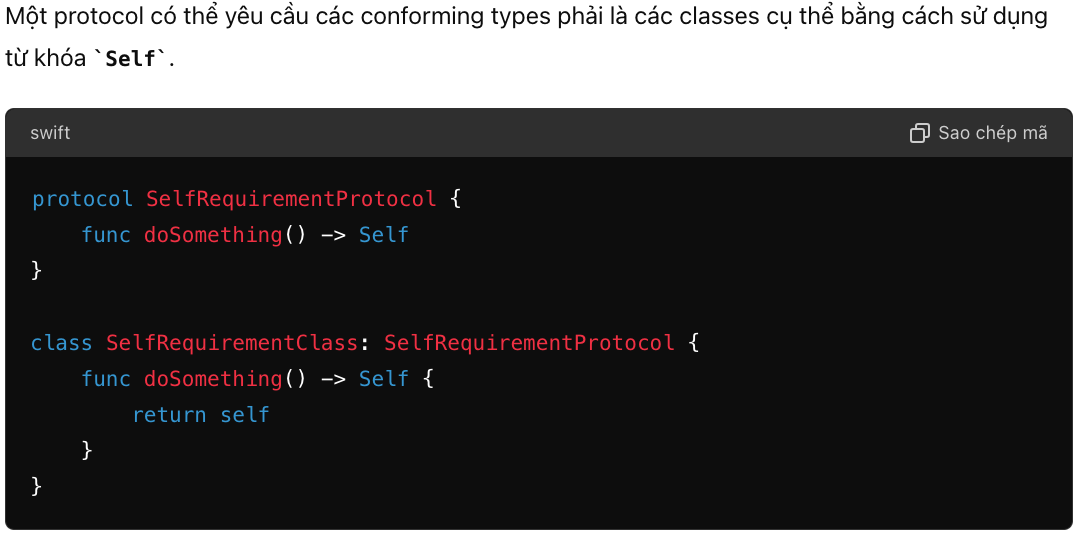
* **Associated Types**



* **Generics với Protocols**



* **Protocols với Self requirements**



Tóm lại: Protocols trong Swift là một công cụ mạnh mẽ để tạo ra các thiết kế linh hoạt và có thể tái sử dụng. Bạn có thể sử dụng chúng để xác định các giao diện cho các lớp, cấu trúc hoặc enums, và kết hợp chúng với các tính năng tiên tiến như kế thừa protocol, generics, associated types và extensions để tạo ra các thiết kế phần mềm linh hoạt và mạnh mẽ.