**PROXY**

**Nhóm 8: Giải thích khi nào sử dụng proxy ?**

**Nhóm 11: cho một ví dụ cụ thể về nhược điểm thứ hai của proxy pattern? (Nhược điểm “Phản hồi từ Service có thể bị trì hoãn”)**

**Nhóm 12: Tại sao không dùng Proxy trong các dự án nhỏ?**

**Nhóm ?: Cách xử lý hiệu suất do việc tạo ra quá nhiều đối tượng proxy?**

**Nhóm 6: Vì sao không khởi tạo và gọi trực tiếp class mà phải thông qua proxy?**

**Nhóm 4: Trong nhược điểm phản hồi từ service có thể bị trì hoãn vậy nếu như thế sẽ ảnh hưởng như thế nào tới dữ liệu?**

**Nhóm 13,7: Những ứng dụng thực tế sử dụng proxy ?**

**FACADE**

**Nhóm 8: Có phải Facade là thành phần cung cấp các dịch vụ đến 1 nhóm thành phần con để tránh gọi các dịch vụ đó nhiều lần không? Giải thích lại.**

**Nhóm 3:**

* **Khi sử dụng mẫu thiết kế Facade thì trong lớp phải khởi tạo rất nhiều đối tượng của các thư viện, lớp con. Vậy việc này có ảnh hưởng đến hiệu suất khi chạy ứng dụng lần đầu không?**
* **Trong quá trình xử lý chẳng may 1 thư viện hoặc 1 chức năng con nào đó gặp vấn đề thì cả lớp facade này có còn hoạt động được không?**

**Nhóm 10: Facade và Proxy đều có ưu điểm và nhược điểm, vậy nên sử dụng hai pattern này trong mô hình thực tế nào?**

**Nhóm 11: Giải thích thêm về vai trò của việc kết hợp facade với proxy?**

**Nhóm 2: Facade có thêm bất kỳ chức năng nào không? hay chỉ đưa ra yêu cầu đến hệ thống con**

**Nhóm 11,13,7: khi nào thì cần sử dụng facade**

**Nhóm 4: nếu giao diện phức tạp thì có thể dùng facade pattern đc k? vì sao**

**Nhóm 5: Những bất lợi tiềm tàng nào của việc sử dụng Facade trong phát triển phần mềm?**

**Nhóm 9: Bạn có thể giải thích thêm về mối quan hệ của 2 pattern facade và proxy?**