

BTTH#01: Tổng quan Design Pattern

Sinh viên tạo project riêng cho từng bài. Tạo thư mục MSSV và hai thư mục con:

src: chứa source code của project

report: chứa các sơ đồ class diagram cho từng bài ở dạng hình jpg hoặc png.

Nén tất cả vào trong 1 file MSSV.zip.

Chỉ nộp file MSSV.zip

Sinh viên chỉ được sử dụng 1 trong ba ngôn ngữ sau để cài đặt: C++/C#/Java.

1. Bài tập 1:

Trong hệ thống ẩm thực Pháp, một buổi ăn trong nhà hàng sẽ tuân thủ theo: Diner 3 plats (Three courses dinner). Ẩm thực Pháp rất chặt chẽ trong việc phục vụ một bữa ăn tại nhà hàng, các bước sau cần phải được phục vụ theo đúng trình tự:

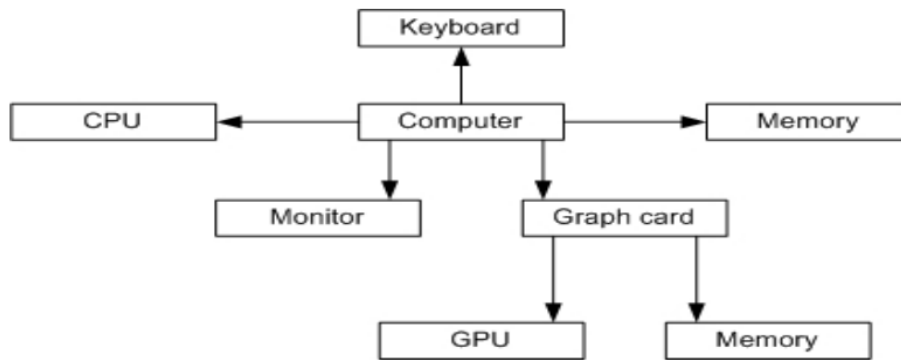
- avantDeCommencer (beforeStart): phục vụ trước bữa ăn.
- servirDesBoisson (serveBeverage): cho order nước và phục vụ nước
 - faireDesBoissons (makeBeverage): chỉ thực hiện việc này khi có yêu cầu phụ vụ nước.
- faire l'entrée: phục vụ món khai vị.
- faire le plat principal: phục vụ món chính.
- faire le dessert: phục vụ món tráng miệng.
- après la fin: kết thúc buổi ăn.

Có hai loại thực đơn chính để phục vụ cho khách: NormalDiner cho người bình thường và VegetarianDiner cho người ăn chay.

Vận dụng mẫu design pattern đã học để implement hệ thống trên. Gợi ý: có thể tạo 1 attribute lưu thông tin có phục vụ nước hay không và được có thể được set trong hàm dựng Constructor. Mô phỏng các step phục vụ bằng việc in ra màn hình món ăn bạn sẽ phục vụ hoặc để thông báo đang thực hiện một step nào đó. Món ăn các bạn tự chọn sao cho phù hợp với 2 loại thực đơn. Demo chương trình trong hàm main().

2. Bài tập 2:

Phân tích hệ thống kiến trúc máy tính sau:



Cụ thể: *CPU*, *GPU*, *memory*, *monitor* và *keyboard* là các đối tượng atomic (đối tượng độc lập đơn giản nhất). Trong khi đó, *graph card* và *computer* là các đối tượng composite. Mỗi đối tượng atomic để có một thuộc tính giá *price* tương ứng. Giá tiền *price* của một đối tượng composite bằng tổng tiền của các phần tử mà nó chứa đựng. Ví dụ, giá tiền *price* của *computer* bằng tổng tiền *price* của *monitor*, *CPU*, *keyboard*, *memory* và *graph card*, giá tiền của *graph card* bằng tổng tiền của *GPU* và *memory*.

Yêu cầu:

2.1: Sử dụng mẫu composite để mô hình hóa cho hệ thống kiến trúc máy tính trên.

2.2: Tìm hiểu thêm mẫu Visitor. Sử dụng mẫu visitor để tính tổng tiền của tất cả các đối tượng *memory* (giá tiền *price* của các bộ phận khác được bỏ qua) trong hệ thống trên.

LƯU Ý: Tạo 2 project riêng biệt cho câu 3.1 và 3.2. Mỗi câu tạo dữ liệu giả và demo trong hàm `main()`.