PHẦN 1

NGÔN NGỮ LẬP TRÌNH JAVA

Mục đích: Rèn kỹ năng/Ôn tập ngôn ngữ lập trình JAVA

Bài tập J1:

Viết class NhanVien gồm các thuộc tính:

- Tên
- Tuổi
- Địa chỉ
- Tiền lương (double)
- Tổng số giờ làm (int)

Constructor không tham số. Constructor đầy đủ các tham số. Các hàm get/set

Các phương thức:

- String toString(): Trả về thông tin của nhân viên.
- double tinhThuong(): Tính toán và trả về số tiền thưởng của nhân viên theo công thức:

+ Nếu tổng số giờ làm của nhân viên >=200 thì thưởng = lương * 20%.

+ Nếu tổng số giờ làm của nhân viên <200 và >=100 thì thưởng = lương * 10%.

+ Nếu tổng số giờ làm của nhân viên <100 thì thưởng =0.

Hàm *main* thực hiện nhiệm vụ sau:

- Khởi tạo hai biến nhân viên và in thông tin của nhân viên ra màn hình.

Bài tập J2:

Tiếp tục phát triển bài tập 1, khai báo interface IQuanLy gồm các phương thức:

- void them(NhanVien nv): thêm một nhân viên vào danh sách.
- void inDS() in danh sách nhân viên ra màn hình.

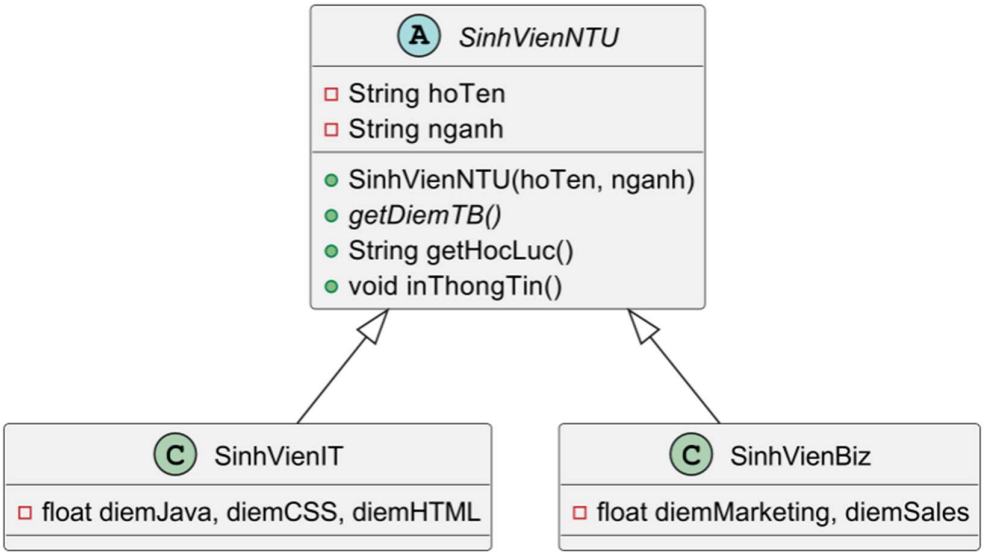
Lớp QuanLyNhanVien chứa danh sách các đối tượng NhanVien và thực thi giao diện IQuanLy.

Hàm *main* thực hiện nhiệm vụ sau đây:

- Khai báo và khởi tạo một đối tượng QuanLyNhanVien.
- Khởi tạo 5 nhân viên và thêm vào danh sách của lớp QuanLyNhanVien.
- In danh sách các nhân viên ra màn hình.

Bài tập J3

Viết các lớp theo sơ đồ sau:



- Phương thức getHocLuc(): trả về chuỗi thể hiện học lực (Yếu, Trung bình, Khá, Giỏi) dựa vào điểm trung bình của sinh viên.

- Phương thức getDiemTB:

+ SinhVienIT: diemJava hệ số 2 các môn khác hệ số 1.

+ SinhVienBiz: diemMarketing hệ số 2, diemSales hệ số 1.

- phương thức inThongTin() in họ tên và ngành học của sinh viên ra màn hình.

Hàm *main* thực hiện công việc sau:

Khởi tạo 3 sv (1 NTU, 1 IT, 1 Biz) và in họ tên, ngành học, điểm, học lực của họ ra màn hình.

Bài tập J4:

Công ty vận tải V quản lý thông tin là các chuyến xe. Thông tin của 2 loại chuyến xe:

- Chuyến xe nội thành: Mã số chuyến, Họ tên tài xế, số xe, số tuyến, số km đi được, doanh thu.

- Chuyến xe ngoại thành: Mã số chuyến, Họ tên tài xế, số xe, nơi đến, số ngày đi, doanh thu.

Thực hiện các yêu cầu sau:

- Vẽ sơ đồ lớp cho bài tập.

- Xây dựng các lớp với chức năng thừa kế.

- Viết lớp QuanLyChuyenXe (bao gồm cả nội thành và ngoại thành) với các chức năng sau:

- Thêm chuyến xe, xuất danh sách các chuyến xe (danh sách có thể dùng cấu trúc mảng), in thông tin từng chuyến xe.

- Tính tổng doanh thu cho từng loại xe, tổng doanh thu cả hai loại xe.

Hàm main thực hiện các công việc:

- Khởi tạo các chuyến xe (3 ngoại thành, 3 nội thành)
- Thêm các chuyến xe đã khởi tạo vào đối tượng thuộc lớp QuanLyChuyenXe.
- In thông tin của từng chuyến xe.
- Tổng doanh thu của xe ngoại thành.
- Tổng doanh thu của xe nội thành.
- Tổng doanh thu của cả 2 loại xe.

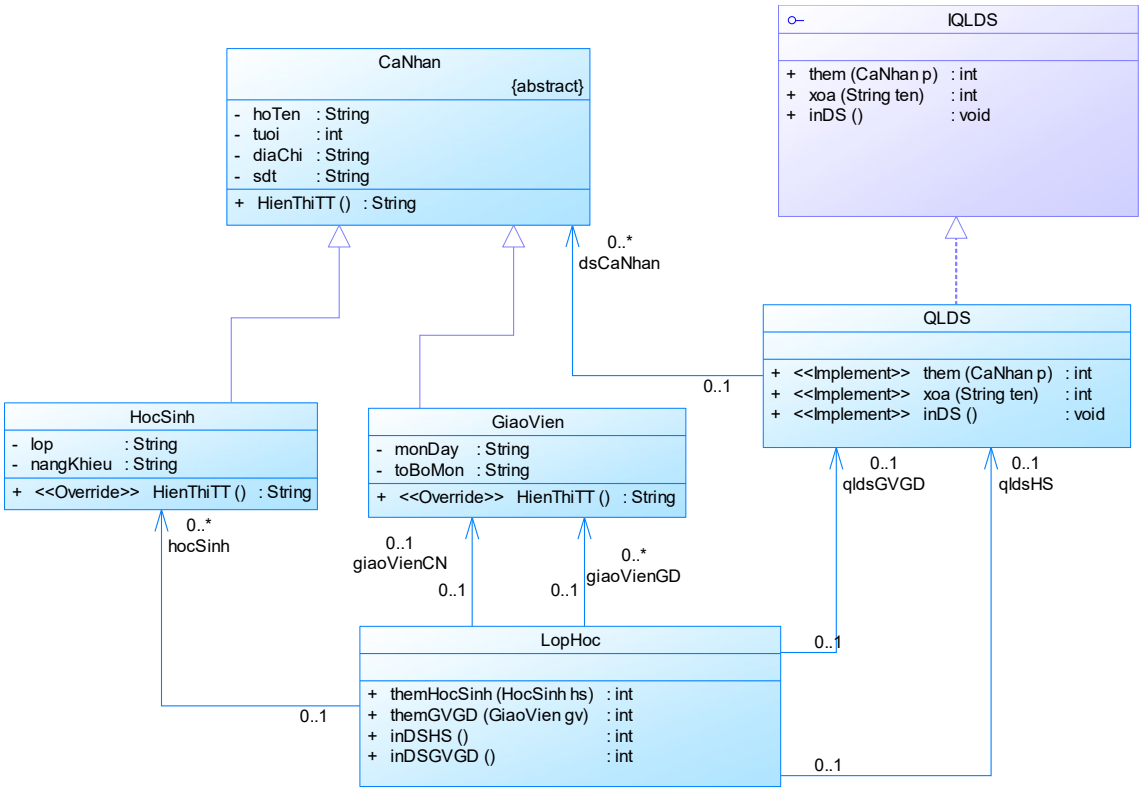
Hướng dẫn:

- Viết lớp ChuyenXe
- Viết hai lớp XeNgoaiThanh, XeNoiThanh thừa kế từ lớp ChuyenXe.
- Lớp QuanLyChuyenXe sử dụng một ArrayList để quản lý các chuyến xe

+ phương thức tinhDoanhThuNoiThanh() duyệt qua từng thành phần trong ArrayList, sử dụng toán tư *instanceof* để kiểm tra thành phần đó có phải là XeNoiThanh hay không.

Bài tập J5:

Sinh viên rèn kỹ năng làm việc với abstract class và interface trong lập trình hướng đối tượng.



- Sinh viên làm bài tập theo sơ đồ lớp như hình.
- Thêm lớp có hàm Main để tạo một lớp học.
- Sinh viên tự thêm các getter, setter, phương thức khởi tạo thích hợp vào các lớp đã cho.

PHẦN 2

CÁC MẪU THIẾT KẾ

A. NHÓM MẪU KHỞI TẠO

A1. (Builder Pattern)

Mục đích: Củng cố kiến thức về mẫu Builder Pattern.

Sử dụng **Builder Pattern** để thực hiện công việc: tạo một đối tượng HoaDon (hóa đơn) bao gồm 2 thành phần:

1. Thành phần thông tin chung bao gồm: mã hóa đơn, ngày bán, tên khách hàng. Các thông tin này được lưu trong đối tượng là thể hiện của lớp HoaDonHeader.

2. Thông tin từng chi tiết bao gồm: sản phẩm, số lượng, đơn giá, chiết khấu. Mỗi chi tiết hóa đơn là một thể hiện của lớp CTHD. Các chi tiết của một hóa đơn được lưu trong một ArrayList.

A2. **(Builder Pattern)** Mục đích củng cố kiến thức về mẫu Builder Pattern

Xây dựng lớp MyStringBuilder tương tự như lớp StringBuilder (Chú ý không được sử dụng lớp String Builder đã được cung cấp sẵn). Các thuộc tính và các phương thức của lớp MyStringBuilder gồm:

- str: Thuộc tính có kiểu String.

- Phương thức addString(String s): Thêm chuỗi s vào sau chuỗi str và trả về kiểu MyStringBuilder.

- Phương thức addFloat(float f): Thêm số s vào sau chuỗi str và trả về kiểu MyStringBuilder.

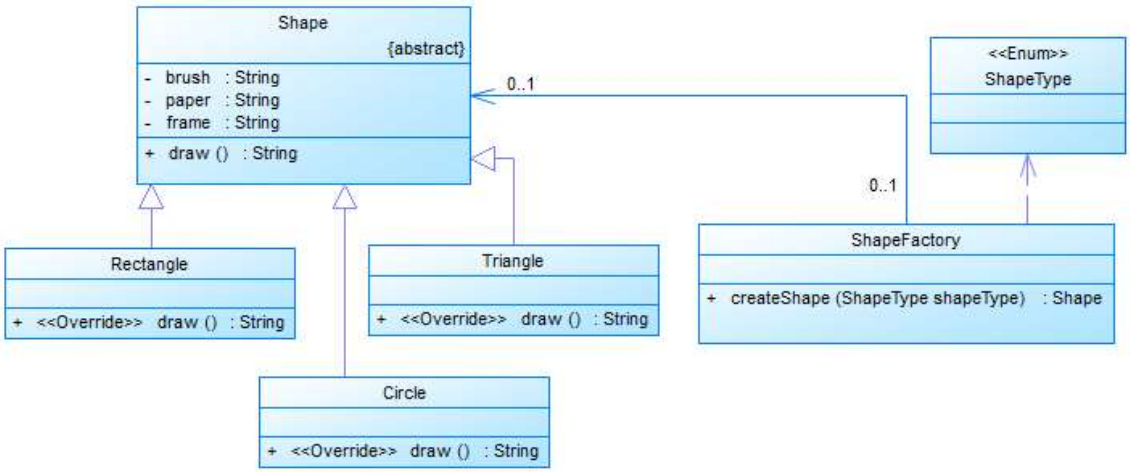
- Phương thức addBool(boolean b): Thêm giá trị b vào sau chuỗi str và trả về kiểu MyStringBuilder.

- Phương thức toString() trả về chuỗi str đã được xây dựng.

Lớp Main: SV sử dụng lớp MyStringBuilder để xây dựng một chuỗi và in chuỗi đó ra màn hình.

A3. Factory Method Singleton

Cài đặt các lớp của ứng dụng trong sơ đồ sau theo mẫu factory :



- Sử dụng mẫu Singleton để đảm bảo chỉ có một đối tượng Rectangle, Triangle, Circle được tạo ra và đảm bảo rằng đối tượng đó được truy cập ở mức toàn cục.

- Tạo lớp có hàm main để demo ứng dụng.

A4. Singleton

Trình bày giải pháp trong Design Pattern để một đối tượng có thể được sử dụng chung cho nhiều lớp. Đối tượng được sử dụng chung được khởi tạo theo kiểu "Lazy", có nghĩa là được khởi tạo khi sử dụng (không được khởi tạo trước).

Áp dụng: Thiết kế một ứng dụng có thể cho phép nhiều người dùng truy cập/tạo một đối tượng (duy nhất) thuộc lớp Election để bầu chọn cho hai ứng viên Donald Trump và Joe Biden. Đối tượng thuộc lớp Election phải ghi lại số lượt bầu chọn cho từng ứng viên bằng phương thức vote, người bầu chọn. Lớp người dùng (User) có phương thức vote để bầu chọn một trong hai ứng viên.

Trình bày giải pháp để mỗi người dùng chỉ có thể bầu chọn tối đa một lần.

Vẽ sơ đồ lớp của ứng dụng.

A5. Builder Pattern

Sử dụng mẫu Builder Pattern để xây dựng một quyển sách bao gồm các thông tin sau:

- Tựa đề: Tựa đề của cuốn sách.
- Số trang.
- Tác giả.
- Danh sách các chương (String) của cuốn sách.
- Phương thức toString dùng để hiển thị thông tin của một cuốn sách.

A6.

Một ứng dụng có 3 đối tượng ui1, ui2, ui3 cùng sử dụng một đối tượng dataAccess để truy cập dữ liệu bao gồm các thao tác thêm, xóa, và cập nhật các đối tượng SanPham (gồm các thông tin: mã sản phẩm, tên sản phẩm, số lượng, đơn giá) vào CSDL (giả sử là một danh sách được lưu trong bộ nhớ RAM).

-Thiết kế ứng dụng để đối tượng dataAccess là một đối tượng được truy cập ở mức toàn cục và duy nhất trong ứng dụng. Đối tượng này chỉ xuất hiện khi nó cần được sử dụng.

- Giả sử ứng dụng có 2 (hay nhiều) loại đối tượng dataAccess, mỗi loại đối tượng truy cập vào một danh sách riêng cũng với các thao tác: thêm, xóa sửa các đối tượng SanPham. Sử dụng mẫu thiết kế để thiết kế mở rộng trong trường hợp này.

A7.

Sử dụng mẫu Builder Pattern để xây dựng cấu hình cho một máy tính bao gồm: CPU, RAM, SCREEN, Hard Disk.

A8. Prototype Pattern

Một hóa đơn bao gồm các trường sau:

+ Tên khách hàng, số điện thoại, ngày bán

+ Chi tiết hóa đơn: Danh sách các sản phẩm. Mỗi sản phẩm bao gồm các trường: tên sp, số lượng, đơn giá.

a. Viết phương thức clone và copy cho các lớp. Phương thức copy sử dụng ByteArrayOutputStream, ObjectOutputStream, ByteArrayInputStream, ObjectOutInputStream

b. Viết phương thức clone và copy cho các lớp. Phương thức copy sử dụng phương thức clone đã viết cho các lớp khác.

A9. Prototype Pattern

Việc copy hay clone một đối tượng thường không phụ thuộc vào cấu trúc của đối tượng đó. Cài đặt Prototype Pattern theo hai cách:

1. Cài đặt lớp Prototype thực thi giao diện IPrototype (gồm hai phương thức copy và clone, giao diện IPrototype thừa kế hai giao diện Serializable và Cloneable có sẵn của java)

```
public interface IPrototype extends Cloneable, Serializable{
    Object copy() ;
    Object clone();
}
```

Lớp demo, ví dụ MonHoc, thừa kế lớp Prototype. Lớp demo này thừa kế lớp Prototype và ghi đè lại hai 2 phương thức copy và clone. Việc cài đặt hai lớp này chỉ đơn giản là gọi các phương thức của lớp cha và ép kiểu kết quả trả về của các phương thức này một cách phù hợp

2. - Cài đặt lớp PrototypeHelper với phương thức tĩnh copy:

```
public static <T extends Serializable> T copy(T t){
        .....
}
```

- Lớp demo, ví dụ MonHoc, implement hai giao diện Cloneable và Serializable

+ Phương thức copy: Sử dụng phương thức copy của lớp PrototypeHelper

+ Phương thức clone: Ghi đè phương thức clone trong giao diện Cloneable

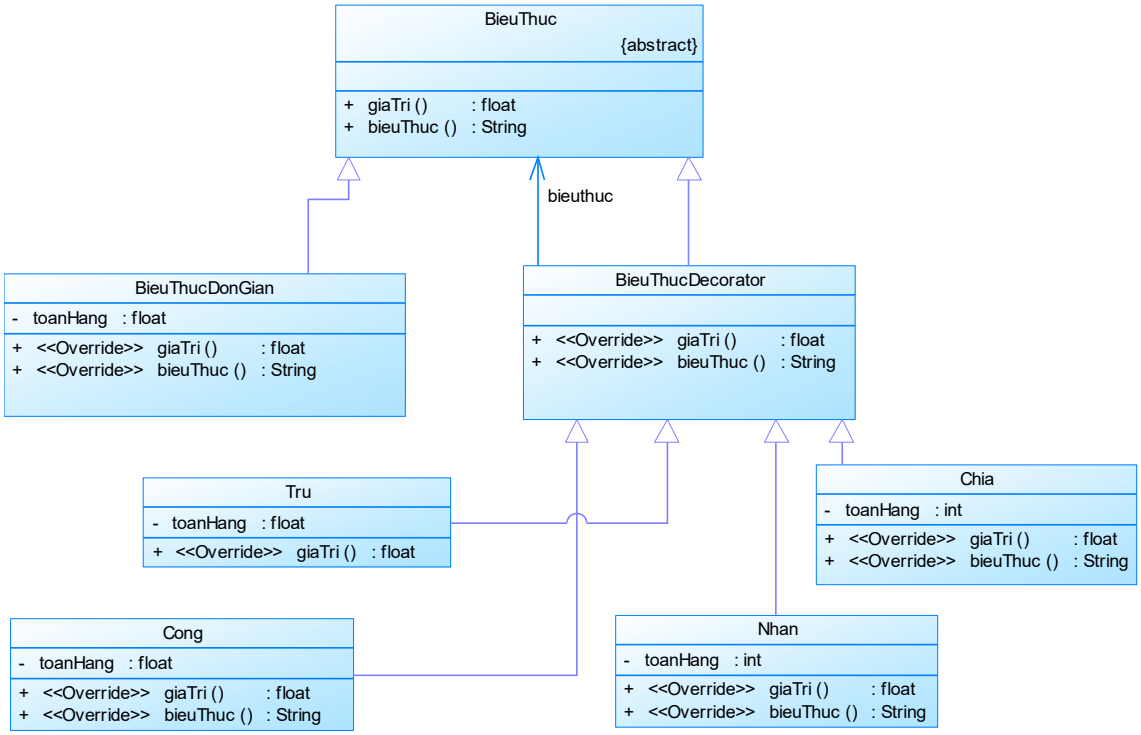
Sinh viên so sánh ưu và nhược điểm của hai cách thực hiện, nêu rõ trường hợp thực hiện của cách.

B. NHÓM MẪU CẤU TRÚC

B1. Decorator Pattern

Viết chương trình tính giá trị của biểu thức dựa vào mẫu Decorator theo sơ đồ dưới đây

Áp dụng: Tính giá trị biểu thức: (6+9)*8 + 5; (8+6)/2*5 +8-4



B2. Decorator Pattern

Tokenize là quá trình tách một chuỗi thành các từ. Sau khi Tokenize, các từ có thể được xử lý ở các bước tiếp theo như loại bỏ các từ dừng (Stop word) như các từ: và, hoặc, thì, mà, là; loại bỏ các dấu câu (dấu chấm phẩy). Hãy sử dụng mẫu Decorator để xây dựng ứng dụng tách từ cho một chuỗi, sau đó có thêm hai lựa chọn là loại bỏ từ dừng và loại bỏ các dấu câu.

B3. Decorator

- Trong Flutter, khi bọc một hay nhiều Widget rõ (ví dụ: Text, Button...) vào một Layout Widget (ví dụ: Center, Container, Row, Column...) các widget rõ sẽ thừa hưởng các thuộc tính của widget Layout.

Hãy sử dụng mẫu Decorator để minh họa cho các widget (rõ và layout) trong Flutter. Mỗi widget có phương thức build để trả về mô tả của nó trong giao diện ứng dụng.

B3. Composite Pattern

Sử dụng Composite Pattern để tổ chức thông tin của một thư mục.
- Thư mục: gồm các thông tin tên thư mục, ngày tạo, danh sách các thư mục cũng như file bên trong.

- File: Gồm các thông tin: Tên file, ngày tạo...
- Phương thức getStringTreeFolder():
    + Thư mục: Trả về chuỗi hiển thị cây thư mục.
    + File: trả về tên file.
- Phương thức: getPath(): Trả về đường dẫn của tập tin hay cây thư mục
- Phương thức addItem() nhận tham số là một đối tượng tập tin hay thư mục và thêm vào thư mục hiện hành.
- Phương thức removeItem() nhận tham số là một đối tượng tập tin hay thư mục và xóa tập tin hay thư mục đó ra khỏi cây thư mục.

Hàm main: Sử dụng cấu trúc trên để lưu trữ cây thư mục

Data (D:)
    TaiLieu
        Design Pattern
            CreationalPattern.pptx
            StructuralPattern.pptx
        Lap Trinh Java
            LapTrinhJavaCoBan.docx
            LapTrinhJavaNangCao.pdf
        NgonNguLapTrinhC.pdf
        LapTrinhThietBiDiDong
            CoBan.pptx
            NangCao.pptx

B4. Composite Pattern

Sử dụng mẫu Composite để tổ chức quản lý các môn học của một khóa học gồm các năm học và các kỳ học. Mỗi kỳ học sv được học nhiều môn học. Ứng dụng phải liệt kê các môn học trong khóa học, năm học, kỳ học; Tổng số tín chỉ, học phí của cả khóa học, từng năm học, từng kỳ học, từng môn học.

B5. Composite bổ sung:

Cấu trúc giao diện của chương trình bao gồm các thành phần:

- Layout: Column, Row, Stack: Dùng để chứa các thành phần trong giao diện.

- Component: Text, Button, Icon … là các thành phần cụ thể của giao diện.

a. Sử dụng mẫu Composite để xây dựng các thành phần của giao diện. Mỗi thành phần có phương thức build dùng để hiển thị nội dung của nó (hoặc con của nó) trên giao diện chương trình.

b. Giả sử có nhiều loại kiểu giao diện: Classic, Mordern, Material với các kiểu component khác nhau hãy xây dựng các bộ công cụ cho mỗi kiểu giao diện.

B6.

a. Sử dụng Composite Pattern để tổ chức cây nhị phân. Phương thức duyetCay của mỗi nút sẽ in ra thông tin của giá trị của nút và các nút con.

b. Giả sử có nhiều cách duyệt cây (LNR, RNL, NLR...). Phương thức duyetCay sẽ tùy vào từng trường hợp mà in ra kết quả là giá trị các nút ứng với mỗi cách duyệt cây.

B7. **Adapter Pattern**

Để quản lý các phần tử của một tập hợp là các sản phẩm, Lớp AppData của một ứng dụng sử dụng một HashMap. Ứng dụng gồm các các lớp:

- AppData: Chứa một HashMap:
    + các phương thức put, remove: Thêm một sản phẩm, xóa một sản phẩm ra khỏi map
    + các getter, settter tương ứng với các thuộc tính.
- Adater: Được lớp Client sử dụng để truy cập dữ liệu từ AppData, gồm các phương thức:
    + add: thêm một sản phẩm
    + remove: Xóa một sản phẩm
    + getListProduct: trả về một danh sách sản phẩm
    + displayProducts: Nhận vào một danh sách sản phẩm và hiển thị lên màn hình.

B8. Bridge Pattern

- Tập hợp có hai cách biểu diễn: Mảng, List có các phương thức:
    + add thêm một phần tử vào danh sách/mảng
    + get(int index): trả về một phần tử tại vị trí index trong danh sách/mảng
    + remove(int index): xóa một phần tử ở vị trí index trong danh sách/mảng
- Dữ liệu truy cập tuần tự: Stack, Queue có các phương thức:
    + push: Thêm một phần tử vào danh sách
    + pop: Đưa một phần ra khỏi danh sách và trả về phần tử này
    + clear: Xóa Stack, xóa Queue

Ứng dụng sử dụng Stack, Queue theo giao diện chung để truy cập dữ liệu; dữ liệu của Stack/Queue có thể được lưu trữ trong Mảng hoặc List. Sử dụng mẫu Bridge Pattern để cài đặt ứng dụng trên.

B9. Bridge Pattern

Một remote TV/Radio có các chức năng cơ bản dựa trên các chức năng của thiết bị:

- Mở TV/Radio
- Tắt TV/Radio
- Mở kênh: Đầu vào là một số nguyên (chỉ số kênh)

Remote có thêm chức năng: thêm/xóa kênh đang mở vào/ra khỏi danh sách yêu thích

Sử dụng Bridge Pattern để cài đặt mô phỏng hoạt động của TV/Radio và remote

C. NHÓM MẪU HÀNH VI

A. OBSERVER PATTERN

Bài tập CA1

a. Stream data là một dạng dữ liệu liên tục trong các kiến trúc lập trình mới, trong đó thành phần client đăng ký lắng nghe các sự kiện từ một nguồn phát để thực hiện data binding. Sử dụng Observer Pattern để viết lớp Stream gồm các phương thức:

- void addEvent(T t): Thêm một sự kiện, dữ liệu T vào Stream, sự kiện này sẽ được thành phần client lắng nghe.

- addListener(Listener l): Dùng để đăng ký một client với Stream.

Viết ứng dụng sử dụng Stream gồm các lớp:

- Client: Hiển thị dữ liệu là một danh sách các môn học lên màn hình.

- Lớp truy cập dữ liệu chứa một stream, thực hiện việc truy cập dữ liệu trên dữ liệu là một danh sách các Môn học. Khi Thêm/Cập nhật/Xóa dữ liệu, danh sách các môn học đều được "tự động" hiển thị trên màn hình.

b. Giả sử có nhiều thành phần client cùng truy cập một đối tượng (là biến toàn cục, duy nhất) là thể hiện của lớp truy cập dữ liệu ở câu 1. Sử dụng mẫu thiết kế để thực hiện điều này.

Bài tập CA2

Sử dụng mẫu Observer Pattern để mô phỏng ứng dụng trên Console gồm 2 lớp gồm: Button và Activity. Lớp Button có sự kiện click để kích hoạt một sự kiện trên lớp Activity. Mỗi Button chỉ có thể nằm trong một Activivity. Giả sử khi click, lớp Activity thực hiện công việc đơn giản là in ra màn hình dòng chữ "bạn click lần thứ n" trong đó n là số lần đã click.

Sử dụng mẫu Observer Pattern để thiết kế ứng dụng trên.

Bài tập CA3

Sử dụng mẫu Observer pattern để thiết kế ứng dụng

- Dịch vụ Tỉ giá cung cấp tỉ giá USD-VND cho các nhà đầu tư A, B, C...

- Các nhà đầu tư A, B, C... sử dụng dịch vụ tỉ giá. Khi tỉ giá thay đổi, các nhà đầu tư sẽ quyết định việc bán ra hay mua vào theo chiến lược của riêng mình.

Bài tập CA4

Sử dụng mẫu Observer Pattern để thiết kế ứng dụng mô phỏng việc các thành viên nhận thông tin một cách tự động từ một Topic khi có thông tin mới. Ứng dụng gồm các lớp sau đây:

- Lớp Topic có các phương thức:

+ Tạo thông tin mới, khi tạo thông tin mới, phương thức này lưu trữ thông tin vào một danh sách tin đồng thời chuyển tiếp tin này cho các thành viên đăng ký nhận tin. Các thành viên tự đăng ký hoặc tự hủy đăng ký việc nhận tin.

+ Cập nhật tin ở lớp Topic, khi tin được cập nhật, thông tin cập nhật được tự động chuyển cho các thành viên đăng ký nhận tin.

- Lớp thành viên A, là lớp nhận tin, chỉ in tin mới nhận ra màn hình khi có thông báo tin mới, hoặc cập nhật tin.

- Lớp thành viên B, là lớp nhận tin, in tất cả các tin đã nhận cùng với tin mới kèm theo số thứ tự của tin. Khi tin được cập nhật, cập nhật tin trong danh sách tin nhận của mình và in tin cập nhật lên màn hình.

1. Khai báo các lớp theo đúng cấu trúc mẫu Observer.

2. Viết mã lệnh cho các lớp, phương thức, hàm main dùng để demo ứng dụng.

Bài tập CA5

Sử dụng mẫu Observer để viết chương trình bao gồm 2 lớp:

- Lớp ATM có phương thức rutTien.

- Lớp TaiKhoan có thuộc tính soDu và thực thi giao diện gồm 2 phương thức kiemTraSoDu, nhanThongBao.

Khi thực hiện rút tiền, trước tiên lớp ATM sẽ kiểm tra số tiền gởi của tài khoản dựa vào phương thức kiemTraSoDu, nếu hợp lệ sẽ tiến hành rút tiền và gởi thông báo cho tài khoản. Nếu không hợp lệ thì gởi thông báo không hợp lệ cho cho tài khoản. Tài khoản tùy vào trường hợp sẽ tính toán lại số dư và hiển thị thông báo.

Bài tập CA6

1. Sử dụng mẫu Observer để hiển thị thông tin của một nhân vật trong một trò chơi. Thông tin về nhân vật được lưu trữ trong các lớp PlayerData gồm các thuộc tính:

- thoiGian: Thời gian còn lại của trò chơi.

- countdown: Số lượt chơi còn lại.

- grade: điểm số đạt được của trò chơi.

Lớp Dashboard là lớp dùng để hiển thị thông tin của nhân vật trên màn hình, lớp này sẽ tự động cập nhật thông tin cho người chơi khi các thuộc tính của trò chơi thay đổi, tức là khi các thuộc tính của trò chơi được thiết lập trong quá trình chơi game.

2. Giả sử khi chuyển sang các level cao hơn sẽ có các thông tin cần bổ sung: như level, mức thưởng bổ sung…., các thông tin này sẽ được bổ sung trong quá trình phát triển của trò chơi, khi một trong các thông tin này thay đổi thì toàn bộ Drashboard sẽ được cập nhật. Giải quyết vấn đề trên bằng một mẫu Design Pattern.

B. CHAIN OF RESPONSIBILITY

Bài tập CB1

a. Một máy ATM sử dụng các loại tiền có mệnh giá 500, 100, 50 , 10, 1 để chi cho khách hàng theo cách sao cho sử dụng ít tờ tiền nhất cho mỗi lần chi. Ứng với một số tiền cần rút cho trước, ứng dụng phải in ra mệnh giá của mỗi tờ tiền và số tờ tiền tương ứng.

Ví dụ ứng với số tiền cần rút là 293 thì in ra kết quả như sau:
- 2 tờ mệnh giá 100
- 1 tờ mệnh giá 50
- 4 tờ mệnh giá 10
- 3 tờ mệnh giá 1

b. Cho biết giải pháp, và viết ứng dụng cho phép người rút tiền lựa chọn mệnh giá tiền lớn nhất mà mình cần rút. Ví dụ với số tiền 293, nhưng người rút tiền chọn mệnh giá lớn nhất là 50 thì kết quả như sau:

- 5 tờ mệnh giá 50
- 4 tờ mệnh giá 10
- 3 tờ mệnh giá 1

Bài tập CB2

Sử dụng mẫu Chain of Resposibility để kết nối chuỗi những người: Phó trưởng phòng, Trưởng phòng, giám đốc, chủ tịch để phê duyệt các khoản vay với các mức vay khác nhau. Chức vụ càng cao thì có khả năng duyệt các khoản vay càng lớn. Nếu khoản vay được duyệt trong khả năng phê duyệt của người nào thì in ra tên người duyệt, thông tin dự án vay, cùng với số tiền vay.

Bài tập CB3

Tính tiền điện bán lẻ cho sinh hoạt với giá điện bậc thang dựa vào mẫu Chain of Responsibility. Giá điện bậc thang cung cấp cho sinh hoạt được cho theo bảng sau đây:

Theo đó, biểu giá bán lẻ điện sinh hoạt mới nhất áp dụng từ ngày 09/11/2023 như sau:

Bậc	Mức sử dụng	Giá cũ	Giá mới
1	0-50 kWh	1.728	1.806
2	51-100 kWh	1.786	1.866
3	101-200 kWh	2.074	2.167
4	201-300 kWh	2.612	2.729
5	301-400 kWh	2.919	3.050
6	401 kWh trở lên	3.015	3.151

Bài tập CB4

1. Dò vé số: Viết ứng dụng dò vé số dựa trên mẫu Chain of Resposibility. Nếu vé trúng nhiều giải thì chỉ được lĩnh thưởng giải cao nhất.
2. Xét trường hợp có trao giải khuyến khích.

Bài tập CB5

Xếp loại học sinh:

[0..5): Yếu

[5...7) Trung bình

[7..8): Khá

[8...9): Giỏi

[9...10]: Xuất sắc

C. STRATERY PATTERN

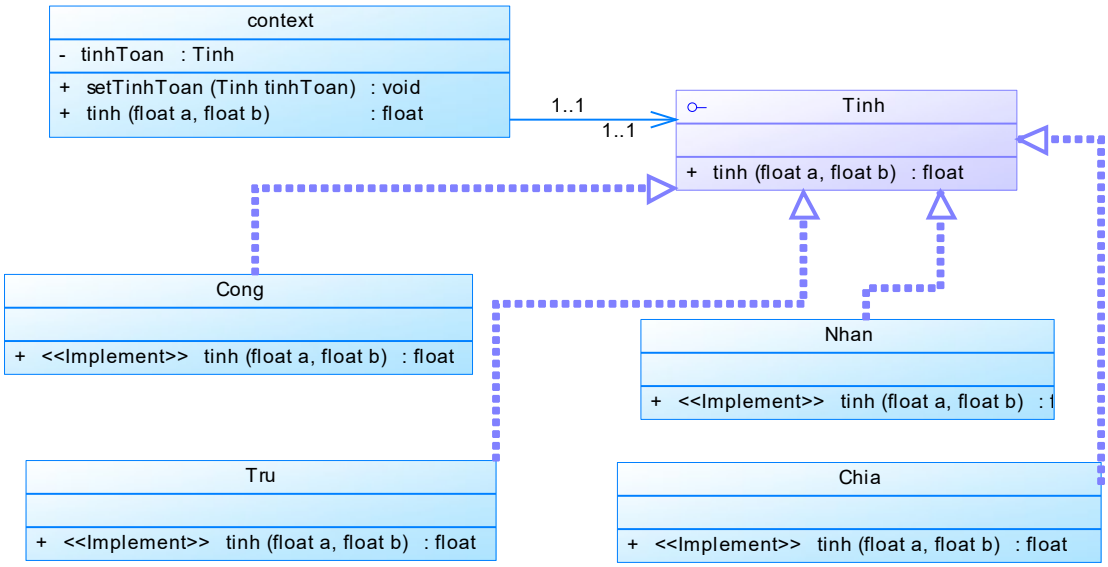
mẫu stratery pattern được sử dụng trong tường hợp muốn tùy biến đối tượng được sử dụng trong 1 lớp, đối tượng khai báo thông qua lớp đại diện

Bài tập CC1

Cài đặt các lớp trong sơ đồ sau đây theo mẫu Stratery Pattern

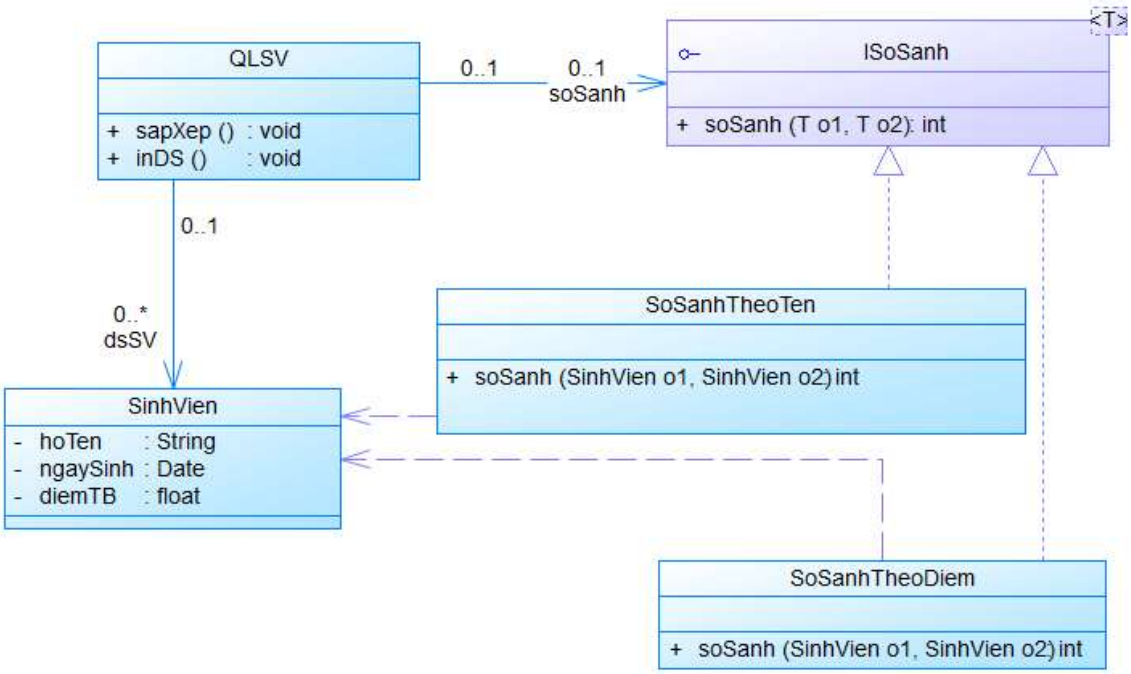
2. Viết mã lệnh cho các lớp trong sơ đồ trên.

3. Thực hiện việc tính toán biểu thức 75 + 12, 54 - 78 trong context.



Bài tập CC2

Cài đặt các lớp sau đây theo mẫu Stratery Pattern



+ Interface ISoSanh khai báo phương thức soSanh:

- Nếu o1>o2: trả về giá trị 1
- Nếu o1 = o2: trả về giá trị 0
- Nếu o1<o2: trả về giá trị -1

+ Hai lớp SoSanhTheoTen và SoSanhTheoDiem thực thi giao diện ISoSanh.

+ Lớp QLSV chứa danh sách các đối tượng SinhVien và một đối tượng ISoSanh.

- Phương thức sapXep() thực hiện công việc sắp xếp danh sách các đối tượng SinhVien.
- Phương thức inDS thực hiên công viên in danh sách sinh viên ra màn hình.

+ Hàm main của lớp thực thực hiện công việc:

- Tạo một đối tượng QLSV và một danh sách các SinhVien của đối tượng QLSV.
- Thực hiện công việc sắp xếp danh sách sinh viên theo tên và in danh sách này ra màn hinh.
- Thực hiện công việc sắp xếp danh sách sinh viên theo điểm và in danh sách này ra màn hình.

Bài tập CC3

Sử dụng mẫu Strategy Pattern để viết ứng dụng khi mua hàng gồm các bước sau:

- Thêm mặt hàng vào giỏ hàng: Thêm mặt hàng (Tên mặt hàng, số lượng, đơn giá) vào danh sách các mặt hàng trong giỏ hàng.

- Chọn hình thức thanh toán: COD, Airpay, MasterCard... (Các hình thức có thể bổ sung mà không ảnh hưởng đến ứng dụng).

+ Thanh toán COD: Được giảm 2% nếu đơn hàng có giá trị từ 2.000.000 trở lên.

+ Thanh toán Airpay: Được giảm 3% nếu đơn hàng có giá trị 1.000.000 trở lên.

- Chọn hình thức khuyến mãi: Giảm x% trên giá trị đơn hàng, giảm tối đa đến y vnd (Các hình thức khuyến mãi có thể bổ sung mà không ảnh hưởng đến ứng dụng), trả về số tiền được khuyến mãi.

Lớp Giỏ hàng có phương thức thanhToan (ngoài ra có thể có các phương thức khác) thực hiện công việc tính toán và hiển thị tổng tiền của giỏ hàng, tiền được giảm ứng với hình thức thanh toán, tiền được khuyến mãi, tiền cần thanh toán.

D. TEMPLATE METHOD

Bài tập CD1

Sử dụng Template Method để xây dựng HoaDon cho 3 loại khách hàng: Thân thiết, Vàng, Kim Cương. Mỗi hóa đơn gồm các thuộc tính và các phương thức sau đây:

- Thuộc tính dsHangHoa: Danh sách các mặt hàng và số lượng, đơn giá của mỗi mặt hàng trong hóa đơn.

- Phương thức tinhTien: Tính tiền hàng của các mặt hàng (không bao gồm chiết khấu).

- Phương thức tinhChietKhau: dựa vào tổng tiền mua hàng trong hóa đơn.

+ Nếu là khách hàng Thân thiết, với hóa đơn 500.000 sẽ được chiết khấu 2%.

+ Nếu là khách hàng Vàng, với hóa đơn từ 500.000 sẽ được chiết khấu 3%, từ 1.000.000 sẽ được chiết khấu 5%.

+ Nếu là khách hàng Kim cương, với hóa đơn từ 500.000 sẽ được chiết khấu 3%, từ 1.000.000 sẽ được chiết khấu 5%, từ 1.500.000 sẽ được chiết khấu 6%.

(Các loại khách hàng có thể được bổ sung, mở rộng mà không làm ảnh hưởng đến các phần khác của ứng dụng)

- Phương thức hiển thị giỏ hàng sẽ hiển thị các thông tin sau đây:

+ Thông tin mỗi mặt hàng trong giỏ hàng gồm: Tên mặt hàng, số lượng, đơn giá, thành tiền.

+ Tổng tiền mua hàng (chưa tính chiết khấu).

+ Tiền chiết khấu.

+ Tiền khách hàng cần thanh toán.

Bài tập CD2

Viết ứng dụng theo sơ đồ dưới đây.

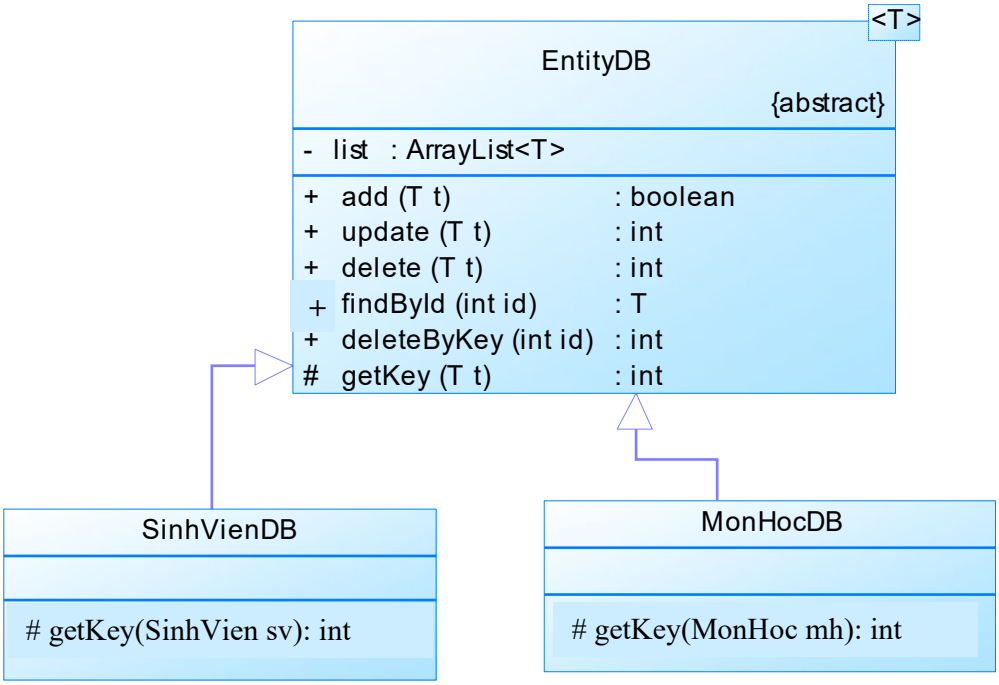
- Phương thức add: Thêm đối tượng t vào danh sách list. Nếu thêm thành công trả về true, ngược lại trả về false.

- Phương thức update: Tìm đối tượng *o* trong list có cùng key với t và cập nhật thông tin của t vào o. Phương thức update trả về số lượng đối tượng được cập nhật.

- Phương thức delete: tìm đối tượng *o* có cùng key với t và xóa đối tượng *o* trong danh sách. Phương thức delete trả về số đối tượng xóa thành công ra khỏi danh sách list.

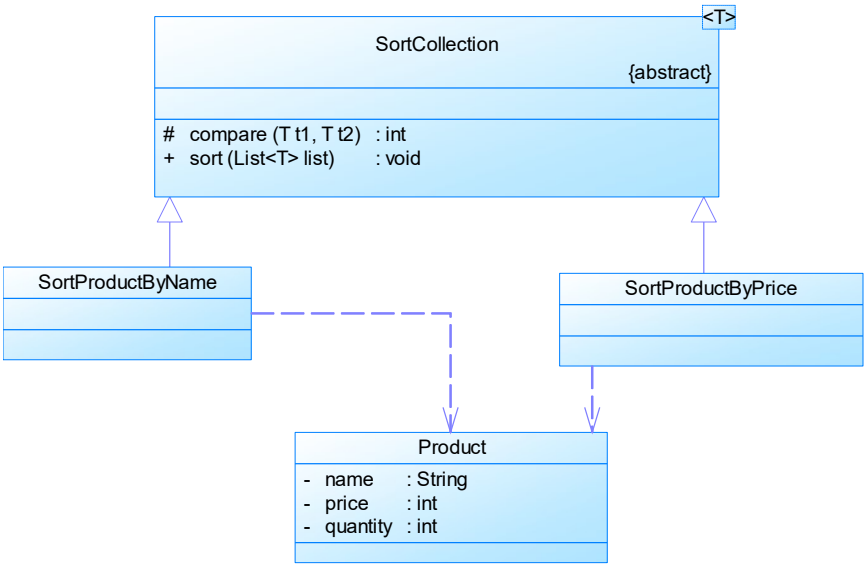
- Lớp SinhVien bao gồm các trường: maSV, tenSV, ngaySinh, queQuan.

- Lớp MonHoc bao gồm các trường: maMH, tenMH, soTC.



Bài tập CD3

Ứng dụng Template Method để sắp xếp danh sách



E. Iterator

Bài tập CE1

Giả sử một lớp tập hợp quản lý một mảng các đối tượng (để đơn giản có thể giả sử các đối tượng này là các số nguyên). Lớp này có các phương thức:

- getItem(int i): Trả về đối tượng thứ i của mảng
- count(): trả về số phần tử của mảng.

Hãy cài đặt mẫu Iterator cho tập hợp trên, chỉ dựa vào các phương thức đã có của tập hợp này.

Bài tập CE2

- Tìm hiểu Iterator pattern được cài đặt sẵn trong Java: Giao diện Iterator, các lớp có sử dụng Iterator.

a. Hãy sử dụng lớp Iterator được cung cấp sẵn trong java để duyệt một List.

b. Cài đặt bài tập CE1 dựa trên hai giao diện Iterable và Iterator có sẵn của Java. (SV cần tìm hiểu mục đích và chức năng của hai giao diện này)

Bài tập CE3

- Sử dụng Iterator có sẵn của Java để viết phương thức nhận vào một Iterator chứa các đối tượng MonHoc và in các môn học lên màn hình.

F. STATE PATTERN

CF1. Một Remote Controll có hai trạng thái On và Off. Khi bấm vào nút power nếu TV đang ở trạng thái on thì TV sẽ bị tắt, nếu TV đang bị tắt thì sẽ được mở. Sử dụng mẫu State để viết ứng dụng remote.

CF2. Một máy ATM (hoạt động bình thường) có ba trạng thái:

1. sẵn sàng: Có tiền để khách hàng rút. Khách hàng có thể rút số tiền tối đa mỗi lần rút.
2. hạn chế: có tiền cho khách hàng rút nhưng lại không đủ số tiền tối đa cho mỗi lần rút.
3. hết tiền: số tiền trong máy bằng 0.

Sử dụng mẫu State để viết ứng dụng cho máy ATM. Ứng với mỗi trạng thái, khi rút tiền máy ATM sẽ hiển thị các thông điệp khác nhau tương ứng.

G. MEMEMTO PATTERN

G1. Ứng dụng soạn thảo văn bản có chức năng Undo và Redo hoạt động tương tự như MS Word. Ngoài ra còn có các chức năng:

- append: Thêm nội dung mới vào nội dung văn bản có sẵn.
- display: Hiện thị nội dung văn bản

Sử dụng Mememto Pattern để viết ứng dụng trên

H. COMMAND PATTERN

H1. Viết ứng dụng bao gồm các lớp:

- Order: Đóng gói một yêu cầu của khách hàng. Ví dụ: 1 ly trà sữa

- Waitress: Tiếp nhận các order từ khách hàng và yêu cầu nhân viên chế biến bằng cách gọi phương thức orderUp. Trong Waitress có phương thức setOrder để nhận một danh sách các order.

- Chef: Tiếp nhận các yêu cầu là các order và thực hiện chế biến

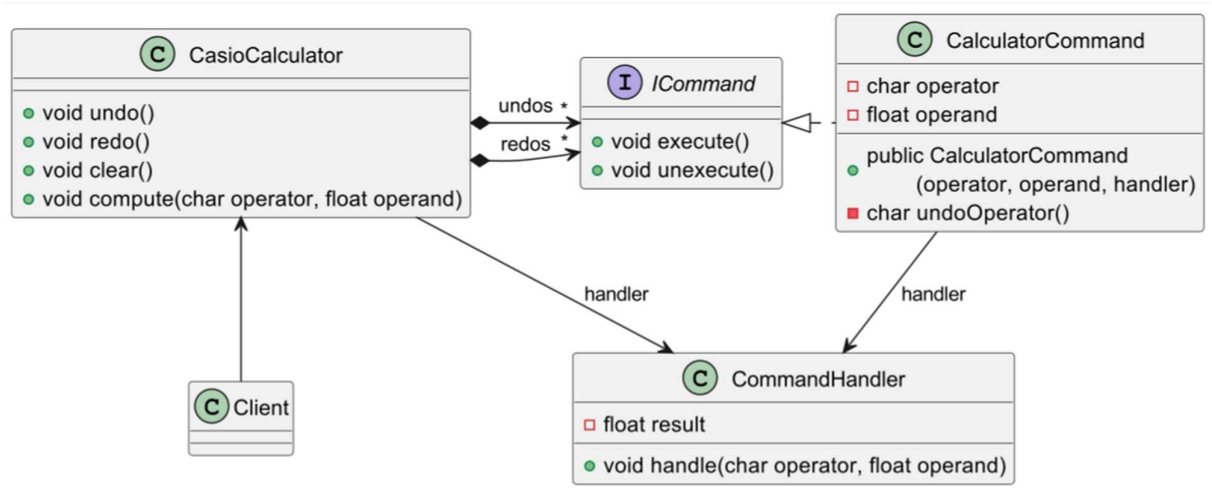
- Client: Lớp có chứa hàm Main để demo ứng dụng.

Một kết quả khi gọi phương thức orderUp của client: (Minh là tên của Chef)

```
Trà sữa was cooked by Minh
Quantity: 2

Mì xào was cooked by Minh
Quantity: 3
```

H2. Viết ứng dụng Calculator cơ bản (bốn phép tính +,-, x, :) có chức năng undo và redo theo sơ đồ sau:

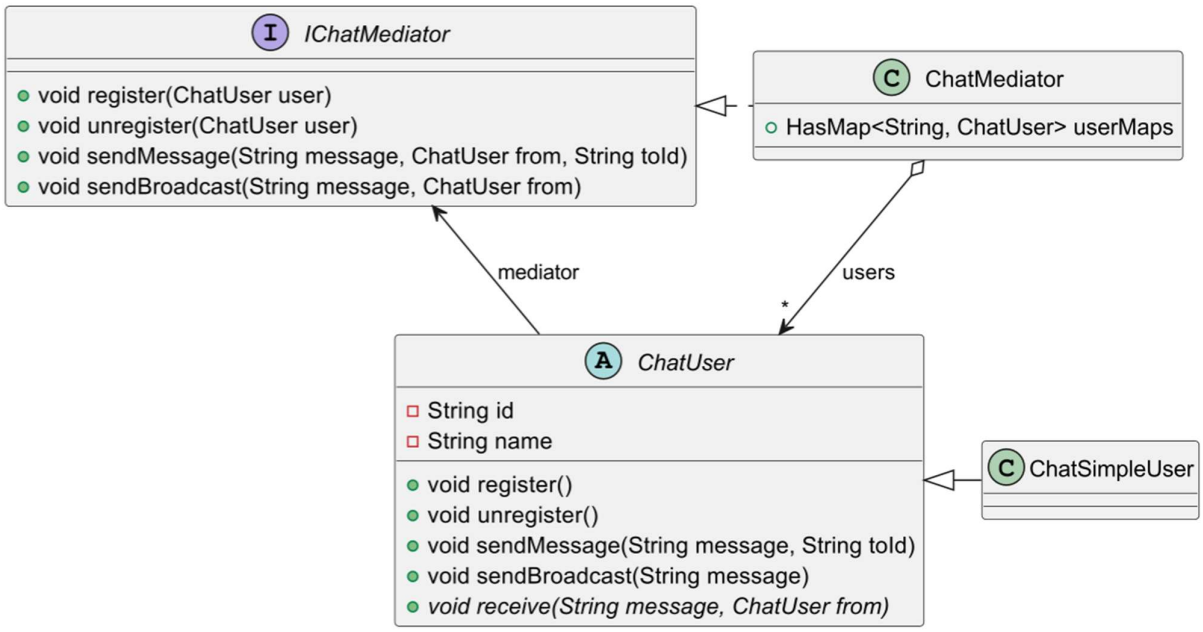


a. Sinh viên chú thích cho các sơ đồ, liên hệ các thành phần của sơ đồ với các thành phần cơ bản trong máy tính.

b. Cài đặt ứng dụng.

K. MEDIATOR PATTERN

K1. Xây dựng ứng dụng chat dựa trên mẫu Mediator theo sơ đồ:



a. Chú thích các thành phần của sơ đồ theo mẫu Mediator

b. Cài đặt ứng dụng theo sơ đồ trên

K2. Một ứng dụng thương mại điện tử bao gồm các đối tượng Customer, Shop, Delivery Service.

- Khi khách hàng mua một mặt hàng từ một shop và chọn đơn vị vận chuyển thì thông tin đặt hàng sẽ được gởi đến Shop.

- Khi shop xác nhận đơn hàng thì thông tin đặt hàng sẽ được gởi đến đơn vị vận chuyển và khách hàng.

- Khi đơn vị vận chuyển thực hiện giao hàng thì thông tin giao hàng sẽ được gởi đến shop, khách hàng.

- Khi đơn vị vận chuyển giao hàng thành công thì gởi thông tin xác nhận đến shop.

Sử dụng mẫu Mediator để viết ứng dụng trên.