1. **Mô hình là gì? Mô hình hóa là gì? Vì sao cần mô hình hóa? Ngôn ngữ mô hình hóa là gì?**

* Mô hình là sự đơn giản hóa thực tế
* Mô hình hóa là việc dùng mô hình để nhận thức và diễn tả một hệ thống
* Cần phải mô hình hóa vì: (*Mục đích của mô hình hóa*)
  + Giúp chúng ta hình dung một hệ thống như nó vốn có hoặc như chúng ta muốn
  + Mô tả được cấu trúc, hành vi của hệ thống
  + Cung cấp một khuôn mẫu, hướng dẫn để xây dựng một hệ thống
  + Giúp ghi lại (lưu trữ) quyết định mà chúng ta đã đưa ra
* Ngôn ngữ mô hình hóa là ngôn ngữ có từ vựng và quy tắc tập trung vào khái niệm và biểu diễn vật lý của một hệ thống.

1. **Ngôn ngữ UML là gì? Nêu và giải thích các đặc điểm của UML.**

* Ngôn ngữ mô hình hóa là ngôn ngữ có từ vựng và quy tắc tập trung vào khái niệm và biểu diễn vật lý của một hệ thống.
* Đặc điểm:
* Visualizing (trực quan hoá) : sẽ không đủ tốt nếu chỉ nghĩ về nó và mã hoá nó ngay cả khi bạn viết mã nó một cách chính xác.
* Specifying (chỉ rõ) : là xây dựng mô hình chính xác, rõ ràng và hoàn chỉnh.
* Constructing (cấu trúc) : mặc dù nó không phải là một ngôn ngữ lập trình nhưng mã có thể được tạo ra từ nó.
* Documenting (lập tài liệu) : UML giúp lập tài liệu tất cả các phần của vòng đời phần mềm.

1. **Giải thích các nguyên tắc khi mô hình hóa.**

* Việc lựa chọn mô hình ảnh hưởng sâu sắc (nhiều) đến cách giải quyết bài toán, ảnh hưởng đến kết quả
* Mỗi mô hình có thể được thể hiện ở nhiều mức độ chính xác khác nhau
* Các mô hình tốt nhất được kết nối với thực tế
* Một mô hình hoặc một góc nhìn là không đủ. Một hệ thống không tầm thường thì cần được mô hình hóa với nhiều góc nhìn khác nhau.

1. **Vẽ cây mô hình khái niệm của UML.**



1. **Phần tử cấu trúc trong UML cho phép mô hóa gì? Lấy ví dụ minh họa.**

* Khái niệm: Mô tả các khối cơ bản trong lớp hướng đối tượng, mô hình hóa các danh từ
* Lớp (Class): mô tả các đối tượng có cùng thuộc tính, hoạt động các mối quan hệ và ngữ nghĩa

Ví dụ:

|  |
| --- |
| Sinh viên |
| * Họ tên * Ngày sinh |
| * sinhVien() * layNgaySinh() |

* Interface (Giao diện): Là các thao tác làm dịch vụ cho lớp hay thành phần. Giao diện mô tả hành vi quan sát được từ bên ngoài thành phần.

Chỉ khai báo các phương thức xử lý, nhưng không định nghĩa nội dung thực hiện. Nó thường không đứng một mình, mà thường gắn với lớp hay một thành phần.

Ví dụ:

|  |
| --- |
| “interface” Window |
| open()  close() |



* Collaboration (Cộng tác): Xác định (mô tả) sự tương tác trong hệ thống, và là một tập hợp các vai trò, các yếu tố khác phối hợp với nhau để hình thành một số hành vi hợp tác lớn hơn tổng tất cả các yếu tố.

Ví dụ:



* Use Case (Ca sử dụng): mô tả một tập hợp các hành động mà hệ thống sẽ thực hiện (để phục vụ cho các tác nhân ngoài).

Ví dụ:



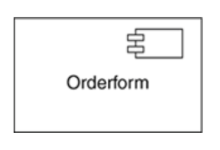
* Active Class (Lớp hoạt động): Là 1 lớp có các đối tượng sở hữu một hoặc nhiều quy trình hoặc luồng.

Ví dụ:



* Component (Thành phần): mô tả thành phần vật lý của hệ thống

Ví dụ:



* Artifact: là 1 phần vật lý, có thể thay thế 1 phần của hệ thống

Ví dụ:

|  |
| --- |
| “artifact”  Window.dll |

* Modes: là 1 phần tử vật lý tồn tại trong thời gian chạy

Ví dụ:



1. **Phần tử hành vi trong UML cho phép mô hình hóa gì? Lấy ví dụ minh họa.**

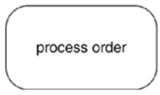
* Khái niệm:
* Là các phần động của mô hình UML.
* Là động từ của một mô hình, đại diện cho hành vi theo thời gian và không gian.
* Interaction: (Tương tác): là hành vi bao gồm một tập các thông điệp trao đổi giữa các đối tượng trong một ngữ cảnh cụ thể để thực hiện một chức năng nào đó.



* State machine (thay đổi trạng thái): là 1 hành vi chỉ ra trật tự thay đổi trạng thái của 1 đối tượng hoặc sự tương tác diễn ra trong suốt thời gian tồn tại của nó để đáp ứng các sự kiện xảy ra.



* Activity: là hành vi xác định trình tự các bước của một quy trình tính toán thực hiện.



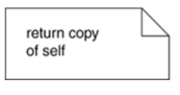
1. **Phần tử gộp nhóm trong UML cho phép mô hình hóa gì? Lấy ví dụ minh họa.**

* Là bộ phận tổ chức của mô hình UML.
* Package (gói): sử dụng để tổ chức các lớp. Dùng để nhóm các phần tử có một ý nghĩa chung nào đó vào thành một nhóm.



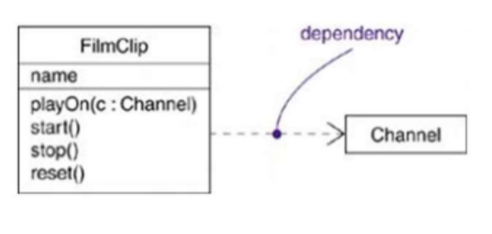
1. **Phần tử chú thích trong UML dùng để làm gì? Lấy ví dụ minh họa.**

* Là phần giải thích của mô hình UML dùng để mô tả nhận xét về bất kỳ phần tử nào trong mô hình.
* Note: là biểu tượng để hiển thị các ràng buộc và nhận xét gắn với 1 phần tử hoặc 1 tập hợp các phần tử.



1. **Trình bày ký hiệu, ý nghĩa, lấy ví dụ minh họa các quan hệ trong UML: quan hệ phụ thuộc, quan hệ kết hợp, quan hệ kết tập, quan hệ hợp thành, quan hệ tổng quát hóa, quan hệ hiện thực hóa.**

* **Quan hệ phụ thuộc** (Dependencies)**:**
* Là quan hệ ngữ nghĩa giữa 2 phần tử, trong đó, sự thay đổi của một phần tử có thể ảnh hưởng đến ngữ nghĩa của phần tử kia
* Kí hiệu: 
* Ví dụ:



* **Quan hệ kết hợp** (Associations)**:**
* Là quan hệ cấu trúc, xác định mối liên kết giữa các lớp đối tượng
* Kí hiệu: 
* Ví dụ: 
* **Quan hệ kết tập** (Aggregation)**:**
* Là một dạng đặc biệt của quan hệ kết hợp, biểu diễn quan hệ cấu trúc giữa toàn thể và bộ phận
* Kí hiệu: 
* Ví dụ: 
* **Quan hệ hợp thành** (Composition)**:**
* Là một dạng đặc biệt của quan hệ kết hợp, trong đó, nếu như đối tượng toàn thể bị hủy bỏ thì các đối tượng bộ phận của nó cũng bị hủy theo
* Kí hiệu: 
* Ví dụ: 
* **Quan hệ tổng quát hóa** (Generalizations)**:**
* Mối quan hệ giữa các lớp trong đó một lớp chia sẻ cấu trúc và/hoặc hành vi với một hoặc nhiều lớp khác
* Kí hiệu: 
* Ví dụ: 
* **Quan hệ hiện thực hóa:**
* Là quan hệ ngữ nghĩa giữa giao diện và lớp
* Kí hiệu: 
* Ví dụ:



1. **Phân biệt giữa quan hệ kết tập và hợp thành.**

* **Quan hệ kết tập** (Aggregation)**:**
* Là một dạng đặc biệt của quan hệ kết hợp, biểu diễn quan hệ cấu trúc giữa toàn thể và bộ phận
* **Quan hệ hợp thành** (Composition)**:**
* Là một dạng đặc biệt của quan hệ kết hợp, trong đó, nếu như đối tượng toàn thể bị hủy bỏ thì các đối tượng bộ phận của nó cũng bị hủy theo

1. **Trình bày các quy tắc trong UML. Lấy ví dụ minh họa.**

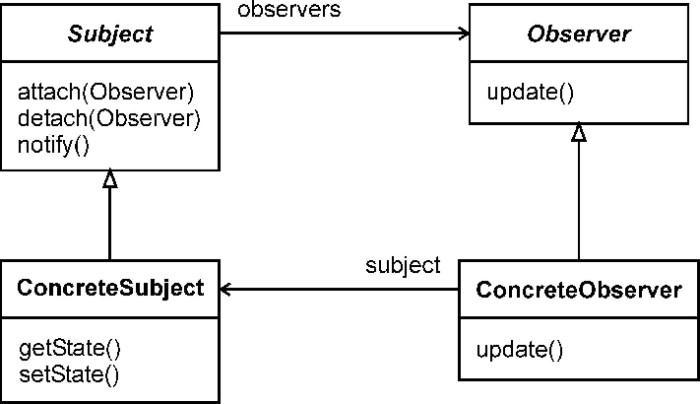
* Name (đặt tên): là đặt tên cho các cái thing.
* Scope (phạm vi): thể hiện ngữ cảnh mà tên nào đó được mô tả.
* Visibility (khả năng hiển thị, nhìn thấy): mô tả cách mà tên đó thấy được bởi một thứ khác.
* Integrity (chính trực): cách mọi thứ liên quan một cách chính xác và nhất quán với nhau
* Execution (chấp hành): ý nghĩa của việc chạy hoặc mô phỏng một hành động
* Elided (giảng lượt): che đi một số phần tử để cho đơn giản
* Incomplete (chưa hoàn thiện): cho phép khuyết một số phần tử, hỗ trợ cho kỹ thuật làm mịn.
* Inconsistent (không nhất quán): chấp hành mô hình không trọn vẹn.

1. **Trình bày các cơ chế chung trong UML. Lấy ví dụ minh họa.**

* Specifications (Thông số kỹ thuật):
* Adornments ( ):
* Common divisions (Bộ phận chung):
* Extensibility mechanism (Cơ chế mở rộng):

1. **Phân loại các biểu đồ trong UML và ý nghĩa từng loại biểu đồ**

* Nhóm biểu đồ về cấu trúc
* Biểu đồ lớp (Class diagram): Cho phép mô tả các cấu trúc tĩnh của hệ thống
* Biểu đồ đối tượng (Object diagram): Một biểu đồ cấu trúc cho thấy một tập hợp các đối tượng và mối quan hệ của chúng
* Biểu đồ thành phần (Component diagram): Biểu đồ cấu trúc cho thấy các giao diện bên ngoài, bao gồm các cổng,và thành phần bên trong của một thành phần
* Biểu đồ gói (Package Diagram): Thiết lập mối quan hệ giữa các pakage.
* Biểu đồ cấu trúc đa hợp (Composite structure): Biểu đồ cấu trúc cho thấy các giao diện bên ngoài và bên trong thành phần của một lớp có cấu trúc.
* Nhóm biểu đồ hành vi
* Biểu đồ ca sử dụng (Use case diagram): Một biểu đồ hành vi cho thấy một tập hợp các trường hợp sử dụng và các tác nhân và các mối quan hệ.
* Biểu đồ trình tự (Sequence diagram): Mô tả sự tương tác của các lớp trong trình tự về thời gian.
* Biểu đồ giao tiếp (Communication diagram): Một biểu đồ hành vi thể hiện sự tương tác, nhấn mạnh tổ chức cấu trúc của các đối tượng gửi và nhận thông điệp
* Biểu đồ máy trạng thái (Statechart Diagram): Mô tả những hành động của các lớp và đối tượng riêng lẻ, mô tả trình tự những trạng thái mà các đối tượng sẽ đi qua.
* Biểu đồ hoạt động (Activity Diagram): Mô tả tiến trình xử lý và trình tự những hành động trong tiến trình xử lý.
* Biểu đồ bao quát tương tác (Interaction overview diagram): Một biểu đồ hành vi kết hợp các khía cạnh của biểu đồ hoạt động và các biểu đồ tuần tự.
* Biểu đồ thời khác (Timing diagram): Biểu đồ hành vi cho thấy sự tương tác với các thông báo của thời gian cụ thể
* Biểu Đồ Triển Khai (Deployment Diagram): Mô tả các tài nguyên vật lý trong hệ thống, bao gồm các nút (node), thành phần và kết nối.

1. Đọc hiểu biểu đồ UML (Sinh viên đọc hiểu các biểu đồ UML cho trước và code đơn giản cho biểu đồ đó).
   1. Đọc hiểu và lập trình Java cho từng biểu đồ lớp sau:

abstract class Observer {

public void update() { … }

}

abstract class Subject {

public Observer observers;

public void attach (Observer) { … }

public void detach (Observer) { … }

public void notify() { … }

}

class ConcreteSubject extends Subject {

public void getState () { … }

public void setState () { … }

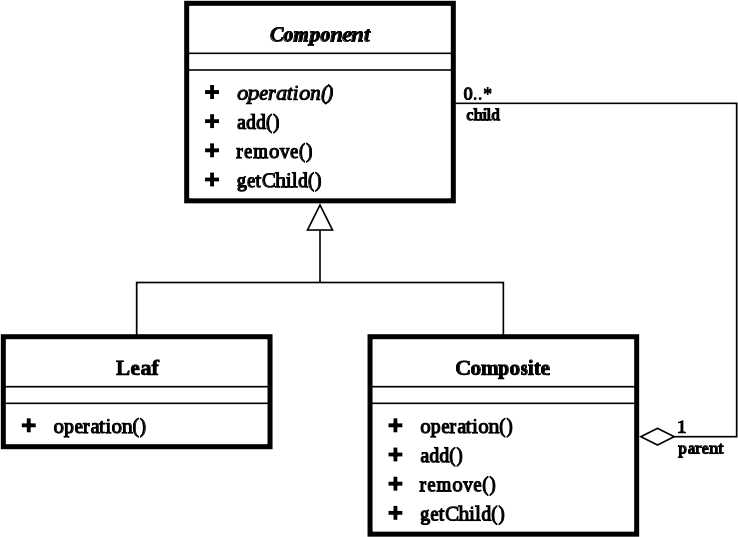
}

class ConcreteObserver extends Observer {

public ConcreteSubject subject;

public void update () { … }

}



abstract class Component{

public Composite composites[];

abstract void operation(){...}

public void add(){...}

public void remove(){...}

public void getChild(){...}

}

class Leaf extends Component{

public void operation(){...}

}

class Composite extends Component{

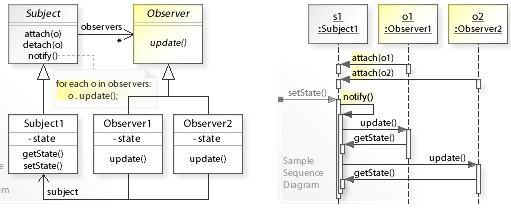
public void operation(){...}

public void add(){...}

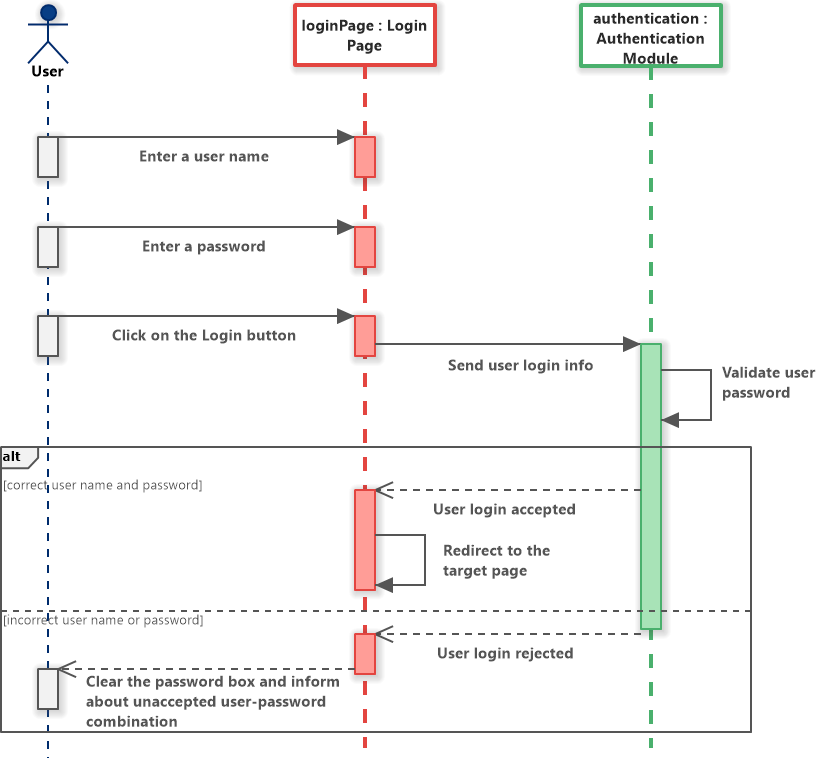
public void remove(){...}

public void getChild(){...}

}

* 1. Đọc hiểu và lập trình Java mô tả kịch bản cho trong từng biểu đồ 

*Lưu ý: Biểu đồ bên phải là biểu đồ lớp cho mẫu Observer, biểu đồ bên phải mô tả tương tác giữa các đối tượng đã mô tả trong mẫu này.*



*Đây là biểu đồ tuần tự mô tả tương tác giữa các đối tượng trong việc đăng nhập một website.*

1. Xây dựng biểu đồ UML theo yêu cầu (Sinh viên xây dựng biểu đồ UML phù hợp để mô tả một bài toán đơn giản cho trước).
   1. Xây dựng biểu đồ lớp tương ứng với mô tả cấu trúc dữ liệu Stack, Queue.
   2. Xây dựng biểu đồ tuần tự tương ứng với mô tả tương tác giữa các thành phần trong giao dịch rút tiền ở máy ATM, tương tác giữa các đối tượng cho việc đăng nhập vào hệ thống mua bán hàng online.

**Biểu đồ tuần tự đăng nhập vào hệ thống mua bán hàng online**

