1. **Trong phỏng vấn khi mà chúng ta muốn khai thác được những thông tin sâu và có giá trị hơn thì nên đặt câu hỏi loại gì? Tại sao?**

"Các câu hỏi mở (open-ended questions) là lựa chọn tốt khi muốn khai thác thông tin sâu và có giá trị trong một cuộc phỏng vấn. Lý do chính là vì câu hỏi mở không giới hạn câu trả lời và khuyến khích người được phỏng vấn nói thêm về kinh nghiệm, kiến thức, và quan điểm của họ. Thay vì hỏi các câu hỏi có thể trả lời bằng 'đúng' hoặc 'sai', câu hỏi mở tạo điều kiện cho người được phỏng vấn chia sẻ các thông tin chi tiết, ví dụ cụ thể, và kịch bản trong công việc hoặc học tập của họ. Điều này giúp tôi có cái nhìn tổng thể và sâu hơn về kỹ năng, sự hiểu biết, và cách tiếp cận vấn đề của người được phỏng vấn, làm cho quá trình đánh giá trở nên chính xác và thông tin đầy đủ."

1. **Quan hệ “là-một loại” (is-a-kind-of) thể hiện quan hệ gì giữa các lớp?**

Quan hệ "là-một loại" (is-a-kind-of) thể hiện quan hệ kế thừa hoặc quan hệ thuộc về trong lập trình hướng đối tượng (OOP) giữa các lớp. Quan hệ này liên quan đến việc một lớp cụ thể (lớp con) là một dạng cụ thể của một lớp tổng quát hơn (lớp cha). Lớp con được xem như là một phần của lớp cha và chia sẻ một số tính chất hoặc phương thức của lớp cha.

Ví dụ, nếu bạn có một lớp cha có tên là "Động vật" và một lớp con có tên là "Chó," quan hệ "là-một-loại" sẽ ám chỉ rằng "Chó là một loại Động vật." Điều này có nghĩa rằng lớp con "Chó" kế thừa các thuộc tính và phương thức của lớp cha "Động vật" và có thể bổ sung hoặc mở rộng chúng.

1. **Xây dựng mô hình thư viện nhỏ cần sử dụng ca sử dụng nào và tác nhân nào?**
2. **Ca sử dụng: Tìm kiếm sách trong thư viện**

* **Tác nhân: Người đọc**

1. **Ca sử dụng: Mượn sách**

* **Tác nhận: Người đọc**

1. **Ca sử dụng: Trả sách**

* **Tác nhân: Người đọc**

1. **Ca sử dụng: Quản lý sách trong thư viện**

* **Tác nhân: Thủ thư**

1. **Ca sử dụng: Đăng ký thẻ thư viện mới**

* **Tác nhân: Người đọc mới**

1. **Ca sử dụng: Quản lý tài khoản người đọc**

* **Tác nhận: Người đọc**

1. **Tính cấu kết lớp là gì?**

Tính cấu kết lớp (class cohesion) là một khái niệm trong lập trình hướng đối tượng (OOP) mô tả mức độ liên quan và tương tác giữa các thành phần (hoặc phương thức) của một lớp. Nó đo lường sự tập trung của các phương thức trong lớp để thực hiện một nhiệm vụ cụ thể và đạt được một mức độ tương tự về mục tiêu hoặc chức năng.

1. **Biểu đồ trạng thái để làm gì?**

Biểu đồ trạng thái (state diagram) được sử dụng để mô hình hóa các trạng thái và chuyển đổi giữa chúng trong một hệ thống, quá trình hoặc đối tượng cụ thể.

1. **Kể tên 2 biểu đồ tương tác?**

a. Biểu đồ trình tự (Sequence Diagram):

* Tác dụng: Biểu đồ trình tự được sử dụng để mô tả tương互 tác giữa các đối tượng trong một hệ thống hoặc một phần của hệ thống theo thứ tự thời gian. Nó cho phép hiển thị các thông điệp và sự tương tác giữa các đối tượng trong quá trình thực thi.
* Đặc điểm:
  + Biểu đồ trình tự giúp hiểu cách các đối tượng tương tác và gửi thông điệp cho nhau trong một chuỗi thời gian cụ thể.
  + Các đối tượng được biểu diễn dưới dạng thẻ hình chữ nhật được xếp theo thứ tự từ trên xuống dưới theo dòng thời gian.
  + Thông điệp được biểu diễn bằng các mũi tên chỉ ra hướng tương tác giữa các đối tượng, và thời gian được biểu diễn bằng số thứ tự hoặc thẻ thời gian.
  + Biểu đồ trình tự có thể chứa các điều kiện, vòng lặp, và lựa chọn để mô phỏng các luồng điều khiển trong tương tác giữa các đối tượng.

b. Biểu đồ giao tiếp (Communication Diagram):

* Tác dụng: Biểu đồ giao tiếp được sử dụng để mô tả tương tác giữa các đối tượng trong một hệ thống hoặc một phần của hệ thống. Nó tập trung vào các đối tượng và các thông điệp được gửi giữa chúng.
* Đặc điểm:
  + Biểu đồ giao tiếp biểu diễn các đối tượng dưới dạng hình tròn hoặc hình chữ nhật và các thông điệp được biểu diễn bằng các mũi tên.
  + Các đối tượng được xếp ngang và các mũi tên chỉ ra thông điệp được gửi từ một đối tượng đến đối tượng khác.
  + Biểu đồ giao tiếp không biểu diễn thứ tự thời gian như biểu đồ trình tự, mà tập trung vào cách các đối tượng tương tác với nhau.
  + Ngoài ra, biểu đồ giao tiếp cũng có thể hiển thị các tương tác song song, các đối tượng tham gia và các quan hệ giữa chúng.

 biểu đồ trình tự tập trung vào thứ tự thời gian và điều khiển, trong khi biểu đồ giao tiếp tập trung vào tương tác và thông điệp giữa các đối tượng.

1. **Trình bày các lớp trong quan hệ?**

Tên Lớp: Đầu tiên và quan trọng nhất, bạn cần chỉ định tên của lớp. Tên này nên được chọn sao cho nó phản ánh được mục đích chính của lớp.

Thuộc Tính (Attributes): Liệt kê các thuộc tính hoặc trạng thái mà lớp chứa. Đây có thể là các biến hoặc các đối tượng khác. Ví dụ: trong lớp "SinhVien", các thuộc tính có thể bao gồm "mã sinh viên", "tên", "điểm số", v.v.

Phương Thức (Methods): Liệt kê các phương thức hoặc hành vi mà lớp có thể thực hiện. Phương thức mô tả các hành động được thực hiện trên các thuộc tính của lớp hoặc trên các đối tượng khác. Ví dụ: trong lớp "SinhVien", các phương thức có thể bao gồm "nhập điểm", "hiển thị thông tin", v.v.

Mối Quan Hệ (Relationships): Mô tả các mối quan hệ giữa lớp hiện tại với các lớp khác trong hệ thống. Mối quan hệ có thể là kế thừa, liên kết, hoặc bất kỳ loại quan hệ nào mà lớp hiện tại có với các lớp khác.

Ghi Chú (Comments): Thêm ghi chú nếu cần thiết để giải thích về ý nghĩa hoặc công dụng của lớp.

1. **Uml là gì?**

UML (Unified Modeling Language): Là một ngôn ngữ mô hình hóa sử dụng để biểu diễn, thiết kế và trình bày các phần mềm hướng đối tượng. UML cung cấp các biểu đồ (biểu đồ lớp, biểu đồ use case, biểu đồ hoạt động, biểu đồ tuần tự và nhiều loại biểu đồ khác) và ký hiệu để mô tả các khía cạnh khác nhau của hệ thống.

1. **Phân biệt biểu đồ lớp với biểu đồ hướng đối tượng?**
   1. **Biểu đồ lớp (Class Diagram):**

Biểu đồ lớp tập trung vào hiển thị cấu trúc của các lớp (hoặc các đối tượng) trong hệ thống hoặc ứng dụng.

Nó thể hiện các lớp, các thuộc tính của lớp, các phương thức của lớp và mối quan hệ giữa các lớp, chẳng hạn như kế thừa, sự liên quan và sự phụ thuộc.

Biểu đồ lớp là một biểu đồ tĩnh, không biểu thị sự tương tác giữa các đối tượng trong thời gian chạy.

* 1. **Biểu đồ hướng đối tượng (Object Diagram):**

Biểu đồ hướng đối tượng tập trung vào hiển thị các đối tượng cụ thể và mối quan hệ giữa chúng tại một thời điểm cụ thể trong ứng dụng hoặc hệ thống.

Nó giúp mô phỏng trạng thái của các đối tượng và cách chúng tương tác tại một thời điểm cụ thể.

Biểu đồ hướng đối tượng thường được sử dụng để minh họa cách các đối tượng cụ thể tương tác trong quá trình thực hiện một chức năng hoặc tình huống cụ thể.

* 1. **Tổng kết**

Tóm lại, biểu đồ lớp tập trung vào cấu trúc và thiết kế của hệ thống, trong khi biểu đồ hướng đối tượng tập trung vào mối quan hệ giữa các đối tượng cụ thể trong thời gian chạy. Cả hai loại biểu đồ đều quan trọng trong quá trình phát triển phần mềm hướng đối tượng và thường được sử dụng cùng nhau để biểu thị cả khía cạnh tĩnh và động của hệ thống.

1. **Quan hệ “là-một-phần-của” (is-a-part-of hoặc has-a) thể hiện quan hệ gì giữa các lớp**

Quan hệ "là-một-phần-của" (is-a-part-of hoặc has-a) thể hiện một quan hệ sự thuộc giữa các lớp trong biểu đồ lớp hoặc hướng đối tượng. Đây là một quan hệ quan trọng để mô tả cách một lớp hoặc đối tượng chứa hoặc bao gồm một lớp hoặc đối tượng khác. Quan hệ này thường ám chỉ sự kết hợp hoặc gắn kết giữa các phần của một hệ thống hoặc đối tượng.

Ví dụ, nếu bạn có một lớp "Car" và trong lớp "Car" có một thuộc tính "Engine" hoặc "Wheel", thì quan hệ "là-một-phần-của" (has-a) được sử dụng để biểu thị rằng một xe hơi chứa một động cơ hoặc bánh xe. Tương tự, một quan hệ "là-một-phần-của" có thể được sử dụng để mô tả rằng một "Person" có một thuộc tính "Address" hoặc "Phone Number".

1. **Hãy kể tên loại biểu đồ UML dùng để mô hình hóa (modelling) các ứng xử (trạng thái) của một đối tượng? Biểu đồ gồm các thành phần nào?**

Biểu đồ trạng thái

Nó bao gồm

Trạng thái (State): Đại diện cho các điều kiện hoặc tình huống mà đối tượng có thể ở trong một khoảng thời gian cụ thể.

Chuyển đổi (Transition): Đại diện cho các sự kiện hoặc điều kiện khiến đối tượng chuyển từ một trạng thái sang trạng thái khác.

Hành động (Action): Đại diện cho các hành động được thực hiện khi chuyển đổi từ một trạng thái sang trạng thái khác.

1. **Mô hình Ca sử dụng (Use case model) là gì? Nó gồm có những thành phần nào?**

Mô hình Ca sử dụng được sử dụng để mô tả các chức năng của hệ thống từ góc độ người dùng. Nó tập trung vào các tác vụ mà người dùng cuối có thể thực hiện trên hệ thống

A white background with black text

Description automatically generated

1. **Trình bày về các quan hệ <<include>> và <<extend>> cùng mục đích sử dụng?**

<<include>>: Quan hệ này được sử dụng khi một ca sử dụng (use case) bao gồm các chức năng của một ca sử dụng khác. Điều này có nghĩa là ca sử dụng được bao gồm (include) các hành động từ một ca sử dụng khác. Nó thường được sử dụng để tái sử dụng các chức năng chung.

<<extend>>: Quan hệ này được sử dụng khi một ca sử dụng có thể mở rộng (extend) các chức năng của một ca sử dụng khác. Nó thường được sử dụng khi có các tình huống đặc biệt mà chỉ xảy ra trong một số trường hợp.

1. **Thể CRC là gì và dùng để làm gì? Cho ví dụ?**

CRC là một phương pháp phổ biến trong phân tích thiết kế hướng đối tượng để thu thập và xác định các lớp (class) trong hệ thống, cũng như các trách nhiệm (responsibility) và các mối quan hệ (collaboration) giữa các lớp.

Lớp (Class): Đại diện cho một tập hợp các đối tượng có các thuộc tính và phương thức chung.

Trách nhiệm (Responsibility): Mô tả các hành động mà lớp cần thực hiện hoặc thông tin mà lớp cần biết.

Mối quan hệ (Collaboration): Mô tả cách các lớp hợp tác với nhau để thực hiện các chức năng hoặc nhiệm vụ trong hệ thống.

Ví dụ: Xét một hệ thống quản lý thư viện. Một lớp có thể là "Sách" (Book) với các trách nhiệm như "Lưu thông tin sách", "Tìm kiếm sách", và hợp tác với lớp "Thẻ thành viên" (Member Card) để kiểm tra việc mượn sách. Trách nhiệm của lớp "Thẻ thành viên" có thể bao gồm "Kiểm tra số lần mượn sách" và "Xác nhận hạn trả sách". Các lớp này có thể hợp tác với nhau để thực hiện chức năng quản lý mượn sách trong thư viện.

1. **Hãy kể tên loại biểu đồ UML dùng để mô hình hóa tương tác giữa các đối tượng? biều đồ gồm các thành phần nào**

Biểu đồ Tuần tự (Sequence Diagram): Biểu đồ tuần tự trong UML được sử dụng để mô tả tương tác giữa các đối tượng theo thứ tự thời gian. Nó thường được sử dụng để mô hình hóa các luồng công việc và tương tác giữa các đối tượng trong một khoảng thời gian cụ thể. Trong biểu đồ tuần tự, các đối tượng được hiển thị dọc theo trục thời gian, và các thông điệp giữa chúng được biểu thị bằng các mũi tên.

Các thành phần trong biểu đồ tuần tự bao gồm:

Đối tượng (Object): Các thực thể hoặc đối tượng tham gia vào tương tác.

Thông điệp (Message): Các tương tác giữa các đối tượng được biểu thị bằng các thông điệp.

Thời gian (Lifeline): Thời gian mà một đối tượng tồn tại trong tương tác được biểu thị bằng các đường thẳng dọc theo trục thời gian.

Biểu đồ Hợp tác (Collaboration Diagram hoặc Communication Diagram): Biểu đồ hợp tác được sử dụng để mô hình hóa các tương tác giữa các đối tượng trong hệ thống, không theo thứ tự thời gian như biểu đồ tuần tự. Thay vào đó, nó tập trung vào các tương tác giữa các đối tượng một cách tổng quát. Trong biểu đồ hợp tác, các đối tượng và các thông điệp được hiển thị một cách tự nhiên hơn.

Các thành phần trong biểu đồ hợp tác bao gồm:

Đối tượng (Object): Các thực thể hoặc đối tượng tham gia vào tương tác.

Thông điệp (Message): Các tương tác giữa các đối tượng được biểu thị bằng các thông điệp.