**Câu hỏi phỏng vấn**

(TienHM3, DuongNQ9, TuyenDQ, HoanNS1, HieuTT26)

1. Em hãy giới thiệu bản thân

2. 4 tính chất của OOP và nêu ví dụ

- Tính đóng gói (Encapsulation):

+ Che giấu thông tin không cho phép truy cập, thay đổi từ bên ngoài một cách trực tiếp thong qua Access modified.

+ Nhóm các đặc tính lien quan lại thành một lớp, một class thì chỉ gồm các thông tin của đối tượng đó.

- Tính kế thừa (Inheritance):

+ Cho phép xây dựng một lớp mới dựa trên các định nghĩa của một lớp đã có. Lớp con kế thừa lại các thuộc tính phương thức của lớp cha, có thể mở rộng them mà không cần định nghĩa lại.

- Tính đa hình (Polymorphism):

+ Một hành động được thực hiện theo nhiều các khác nhau. Thể hiện rõ nhất qua việc gọi phương thức của đối tượng. Các phương thức có thể giống nhau, nhưng việc xử lý lại khác nhau. Cho phép người lập trình gọi trước một phương thức của đối tượng chưa xác định mà đến khi runtime thì chương trình sẽ xác định đối tượng để gọi ra phương thức của đối tượng đó. Thể hiện qua Overloading và overriding.

- Tính trừu tượng (Abstraction):

+ Loại bỏ đi tính chất phức tạp của đối tượng bằng các chỉ đưa ra thuộc tính và phương thức cần thiết của đối tượng lập trình, ẩn đi chi tiết triển khai cụ thể, giúp tập trung vào cốt lõi cần thiết thay vì quan tâm đến cách nó thực hiện. Thể hiện qua Abstract và interface.

+ Interface là một bản thiết kế gồm các thuộc tính và phương thức chung cho một loại đối tượng, các class kế thừa phải implement tất cả thành phần bên trong interface. Access modified của interface có thể khai báo là internal, public.

3. So sánh abstract và interface? Khi nào dùng interface?

- So sánh Abstract và interface

|  |  |
| --- | --- |
| Interface | Abstract |
| Hỗ trợ đa kế thừa | Không hỗ trợ đa kế thừa |
| Không chứa Constructor | Có thể có Constructor |
| Các phương thức không thể viết Access modified, mặc định public | Có thể có Access modified. |
| Không thể chứa Static | Có thể chứa static |
| Không thể chứa phương thức có than hàm | Có thể chưa phương thức có than hàm |
| Không thể chứa biến (int a) nhưng có thể chứa property(int a{set; get;}) | Có thể chứa biến. |
| Bắt buộc phải kế thừa phương thức, thuộc tính | Chỉ phải kế thừa các thuộc tính, phương thức abstract |

- Sử dụng interface khi ta cần đa kế thừa.

4. Abstract có thể dùng làm phạm vi truy cập không?

5. ưu điểm của việc dùng store procedure so với dùng câu lệnh truy vấn Command

Store proceduce: là các câu lệnh được nhóm lại để sử dụng nhiều lần thay vì phải viết từng câu lệnh và thực hiện từng câu lệnh đơn lẻ.

Ưu điểm:

+ Nhanh ( được biên dịch một lần khi tạo ra, lưu lại trong database và sau đó ta sử dụng lại không cần biên dịch lại trừ khi có thay đối)

+ Bảo mật( Có thể phân quyền truy cập)

+ Tái sử dụng: Có thể gọi lại nhiều lần thông qua tên, có thể chuyển sang bất kì ứng dụng nào muốn sử dụng chúng, không cần viết lại.

+ Giảm băng thông: Thay vì gửi những câu lệnh dài dòng chưa được biên dịch, thì chỉ cần gọi tên phương thức đã biên dịch rồi.

- Nhược điểm:

+ Nếu tạo quá nhiều Proceduce thì hệ quản trị sẽ sử dụng nhiều bộ nhớ để lưu. Nếu thực hiện nhiều câu lệnh trong một thủ tục sẽ gây cho CPU phải làm việc nặng hơn

+ Nếu sử dụng thủ tục thì sẽ rất khó phát triển ứng dụng, gây khó khan ở mức logic business trên các ứng dụng, khó xử lí thao tác.

+ Không thể Debug được Proceduce

+ Có kỹ năng mới có thể maintain được.

6. Hoán vị mà không dùng biến tạm

A = a + b

B = a-b

A = a-b

7. em đã làm nhiều về front end không? Nếu có thì em đã dùng những framework nào?

- Biết một chút.

- Các framework front-end bao gồm:

+ Boostrap

+ Css

+ Ajax, jquery

8. trong những project mà em đã làm, em dùng interface như thế nào? Mục đính của việc dùng interface đấy là để làm gì?

Đã sử dụng trong Irepository, Iservice.

Mục đích sử dụng tạo ra các thiết kế chung cho các lớp cụ thể kế thừa phải chứa những phương thức mà mình muốn chúng cần phải có.

Một số repository đặc biệt có them một số phương thức khác interface chung thì tạo ra them một interface khác kế thừa interface có. Hoặc tạo một interface có them một số phương thức mới ta cần có. Rồi cho class kế thừa cả 2 interface đó.

9. Em có tham gia hoạt động thể dục thể thao không? Bia rượu thế nào?

Em Very TỐT.

10. Người ta dùng những framework nào để xử lý những câu lệnh truy vấn Command đến CSDL? (dùng ADO .NET)

- Dùng Ado.net hoặc dùng Entity,linq2 to sql, dapper, …

11. Các đối tượng khai báo trong interface

- Methods

- Properties

- Indexers

- Events

12. Em hãy nói về từ khóa Static

- Để khai báo 1 thành viên tĩnh như class, method, property

13. Lớp string có phải là static không?

Không.

14. So sánh struct và class?

|  |  |
| --- | --- |
| Struct | Class |
| Kiểu tham trị | Kiểu tham chiếu |
| Không kế thừa. | Có kế thừa. |
| Constructor phải có tham số | Có thể có or không. |
| Không thể khởi tạo giá trị của thuộc tính. | Có thể khởi tạo giá trị của thuộc tính |
| Không hỗ trợ hàm hủy | Hỗ trợ hàm hủy |
| Không thể gán giá trị null | Có thể gán giá trị null |

15. trong 1 solution có 2 project, ở project 1 có 1 interface có phạm vi truy cập là internal thì project 2 có gọi đến được interce trong project 1 không?

Không.

16. Các method trong interface có thể khai báo phần thân được không?

Không

17. Protected internal là gì?

Chỉ truy cập được vào thành viên trong cùng class định nghĩa nó hoặc các class con nằm trong cùng một assembly.

18. Dự định của em trong 2 3 năm tới là như thế nào?

19. Trong MockProject của em, em đóng vai trò gì? Nếu là team leader em hãy mô tả lại 1 ngày làm Mock của em như thế nào?

Suốt ngày ăn chửi, làm bù nhìn ☹ (Cường than thở)

20. String, int là reference type hay là value type

String – References

Int - Value

21. Mục đích của Interface là gì?

Hỗ trợ đa kế thừa, bắt buộc các class implement phải implement đầy đủ.

22. trong Mockproject sử dụng công nghệ nào? Ai là người quyết định? Ai là người thiết kế Database?

23. Mô hình MVC và mô hình 3 lớp có giống nhau không?

Không.

24. Em có biết gì về angular không?

25 Em có dùng Unit test trong project không? Em dùng nó như thế nào?

26. Em có biết gì về Repository không?

Là một mẫu dung để tạo một lớp trung gian giữa lớp data và business. Lớp này chứa đựng phương thức thao tác mà để giao tiếp với lớp data đê rphucj vụ cho businesss từ lớp logic. Mục đích tạo ra để cách ly việc tiếp cận data sao cho những thay đối không ảnh hưởng trực tiếp đến lớp logic business.

27. Em có biết các loại data framework nào không? Entity framework có phải là 1 data framework không?

28. câu lệnh truy vấn Command và entity framework cái nào truy vấn nhanh hơn?

Command nhanh hơn. Vì thao tác trực tiếp còn entity framework phải map với các entity.

29. Mô hình 3 lớp có ghép được với mô hình MVC không? Nếu ghép được thì các tầng nó hoạt động như thế nào?

Có.

30. Em đã làm về MVC nhiều chưa?

31. Đơn vị chị hay phải đi Onside ở trong sài gòn trong khoảng thời gian dài, em có đi được không?

32. Về mặt ăn chơi của em thế nào? Em có chơi môn thể thao điện tử nào không?

33. Em có sử dụng Linq không?