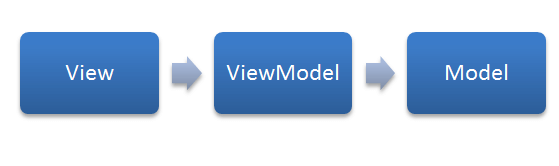
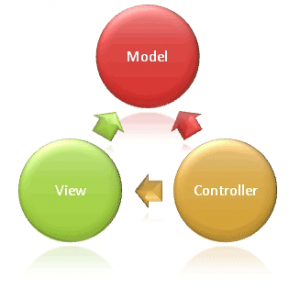
Câu 1: Tính chất về OOP:

* Tính đóng gói (Encapsulation):
  + - Thể hiện cách che dấu dữ liệu đặc trưng của đối tượng
    - Truy cập ở nhiều mức độ khác nhau, ta có thể dùng các phạm vi để tương tác đến như private, protected, internal, public hoặc get set để tương tác
* Tính kế thừa (Inheritance):
  + - Là kỹ thuật cho phép kế thừa lại những tính năng mà một đối tượng cha đã có, giúp tránh việc code lặp dư thừa mà chỉ xử lý công việc tương tự.
* Tính đa hình (Polymorphism ):
  + - Thể hiện khả năng đa dạng hóa của đối tượng, một đối tượng có thể thể hiện nhiều dạng thức khác nhau
    - có 2 loại là đa hình run-time và đa hình complie-time:
      * Run-time: dùng override để thực hiện, khi chạy thì thì phương thức được ghi đè lên phương thức mà nó override
      * Complie-time: dùng overload: một lớp có nhiều phương thức trùng tên, miễn sao khác nhau tham số truyền hoặc kiểu trả về
* Tính trừu tượng (Abstraction):
  + - Là cách thức thể hiện cụ thể một đối tượng thông qua các đặc trưng của nó, như là thông qua interface
    - Sử dụng tính trừu tượng để tránh việc phải implement cụ thể vấn đề, ta có thể dùng qua các method abstract hoặc interface. Còn cụ thể hóa của việc implement là phần việc của tính đóng gói

Những điểm khác nhau giữa Interface và Abstract Class

|  | **Interface** | **Abstract Class** |
| --- | --- | --- |
| Constructors, Fields and Constants | Không | Có |
| Multiple inheritance | Một class có thể hiện thực nhiều interface.(tạm coi là thừa kế) | Không hỗ trợ đa thừa kế |
| Default implementation | Không thể định nghĩa code xử lý, chỉ có thể khai báo. | Có thể định nghĩa thân phương thức, property. |
| Access Modifiers | Mọi phương thức, property đều mặc định là public. | Có thể xác định modifier. |
| Adding functionality | Mọi phương thức, property của interface cần được hiện thực trong class. | Không cần thiết. |

Câu 2: Khi muốn thao tác hiệu quả với chuỗi nên sử dụng string và stringbulder.Tại sao?

Câu 3: Thành phần và luồng xử lý trong mô hình MVC,MVVM, so sánh viewmodel vs controller.****

Câu 4:

|  |  |  |  |
| --- | --- | --- | --- |
| * Tăng performance của website MVC.NET bằng những cách optimize nào? | | | |
|  | | Cache, SQL Optimize, Bundle, minimize resource  Câu 5:   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | Làm sao để copy dữ liệu lớn giữa các bảng trong sql.  (Cách thường để insert nhiều dòng dữ liệu vào SQL thì chúng ta sẽ quét từng dòng rồi Insert lên nhưng như vậy rất mất thời gian nếu như dữ liệu cần Insert của chúng ta là hàng nghìn dòng hoặc nhiều hơn nữa) | | |  |  |  |  |  | |  | * Dùng SqlBulkCopy: có thể truyền DataTable, DataSet hoặc dùng DataReader là nhanh nhất | | | | | | | |  | * Nên để chạy background |  |  |  |  |  |  | |  | * Nhớ sử dụng SqlTransaction để Commit hoặc Rollback | | | |  |  |  |   Câu 6:   * So sánh Union và union all. * Sự khác biệt giữa các lệnh TRUNCATE, DELETE và DROP trong sql server?   + **DELETE** xóa một hoặc tất cả các hàng từ một bảng dựa trên điều kiện và có thể được phục hồi lại.   + **TRUNCATE** xóa tất cả các hàng từ một bảng bằng cách phân bổ các trang bộ nhớ và không thể phục hồi lại   + **DROP** xóa hoàn toàn một bảng từ cơ sở dữ liệu * Làm sao để lập lịch tự động thực hiện 1 công việc cụ thể trong sql server? **Jobs**. |  |
| * Tối ưu câu truy vấn thế nào? | | | | |
|  | Dùng tool check performance : SQL Query Analyzer để dùng Execution Plan | | | |
|  | Sau đó optimize | | | |

* Làm cách nào có thể chọn **dữ liệu** trong cùng **một truy vấn từ** hai cơ sở **dữ liệu khác** nhau trên hai máy chủ **khác** nhau trong SQL **Server**? **Linked Server**

Quản lý source code:

Git: git flow, thiết kế git flow cho team.

Giải quyết khi sảy ra conflict.