1. Encapsulation trong c# là gì ?

Tính đóng gói (Encapsulation) cho phép che giấu thông tin và những tính chất xử lý bên trong của đối tượng. Các đối tượng khác không thể tác động trực tiếp đến dữ liệu bên trong và làm thay đổi trạng thái của đối tượng mà bắt buộc phải thông qua các phương thức công khai (public) do đối tượng đó cung cấp. Làm tăng tính bảo mật cho đối tượng tránh tình trạng dữ liệu bị sửa đổi ngoài ý muốn.

1. Bốn keyword access modifier: public, protected, private, internal ?

Public không có bất kỳ giới hạn nào khi truy cập vào các thành viên công khai.

Protected truy cập bị giới hạn trong phạm vi định nghĩa của class và bất kỳ các class con kế thừa từ class này.

Private truy cập bị giới hạn trong phạm vi định nghĩa của class. Đây là loại phạm vi mặc định.

Internal truy cập trong phạm vi Assembly của dự án hiện tại.

1. Trong project EmployeeManagement đã vận dụng tính chất Encapsulation như thế nào?

Tính đóng gói được sử dụng trong phần abstract class Employee, class Tester, class Developer các thuộc tính và phương thức trong class được bảo mật.

1. Inheritance trong c# là gì ?

Tính kế thừa (Inheritance) là xây dựng một lớp mới (lớp con) kế thừa và tái sử dụng các thuộc tính và phương thức của lớp cũ (lớp cha) đã có trước đó.

1. Trong project EmployeeManagement đã vận dụng tính chất Inheritance như thế nào?

Tính kế thừa được sử dụng ở class Tester và Developer, kế thừa lớp Employee, các interface ISendMail, IExportPdfReport, IexportExcelReport.

1. Abstraction trong c# là gì?

Tính trừu tượng (Abstraction) chỉ tập trung những thứ cốt lõi, quan trọng của đối tượng.

1. Trong project EmployeeManagement đã vận dụng tính chất Abstraction như thế nào?

Tính trừu tượng được sử dụng trong phần tạo abstract class Employee, và các interface ISendMail, IExportPdfReport, IexportExcelReport.

1. Polymorphism trong c# là gì?

Tính đa hình (Polymorphism) cho phép đối tượng khác nhau thực thi chức năng giống nhau theo những cách khác nhau.

1. Trong project EmployeeManagement đã vận dụng tính chất Polymorphism như thế nào?

Tính đa hình được sử dụng trong class Tester và Developer các phương thức GetTask, GetEmployeeSkills, ExportPdfReport, ExportExcelReport, SendRegularMail được sử dụng lại nhưng cách sử dụng in ra là khác nhau.

1. Phân biệt Overriding và Overloading?

* Overloading là kỹ thuật cho phép trong cùng 1 class có thể có nhiều phương thức cùng tên nhưng khác nhau về số lượng tham số hoặc kiểu dữ liệu của tham số. (phương thức SendRegularMail)
* Overriding được sử dụng khi lớp con kế thừa lớp cha và muốn định nghĩa lại lớp cha (phương thức GetTask, GetEmployeeSkills)

1. Khi nào sử dụng abstract class, khi nào sử dụng interface?

Interface sử dụng khi muốn tạo một khung chuẩn gồm những chức năng mà những module và project cẩn phải có.

Abstract class sử dụng khi có một tính năng mà các lớp con kế thừa sử dụng chung (như phương thức GetTask và GetEmployeeSkills thì ở lớp Employee, Developer sử dụng lại 2 phương thức đó được khai báo ở lớp cha, lớp con định nghĩa lại qua Overriding)

1. Khác nhau giữa abstract class, interface

Interface

. Không phải là class.

. Chỉ chứa những phương thức và thuộc tính không thực thi.

. Là khuôn mẫu, một khung để các lớp thực thi (implement) và theo (follow).

. Các lớp có thể implement nhiểu interface.

. Các lớp implement triển khai các phương thức theo interface định nghĩa.

Abstract class

. Giống interface.

. Có 2 loại phương thức là abstract method: phương thức trống không thực thi và method thường: phương thức có thực thi.

. Các lớp chỉ được kế thừa một abstract class.

. Hướng đến những tính năng chung, hàm chung cho các lớp con kế thừa.