

# NỘI DUNG ÔN TẬP MÔN LẬP TRÌNH .NET

## CHƯƠNG 1: TỔNG QUAN VỀ LẬP TRÌNH .NET

Chương 1 này cung cấp một tổng quan về Lập trình .NET, tập trung vào .NET Framework, ngôn ngữ lập trình C#, và môi trường phát triển Visual Studio. Các nội dung được trình bày qua các câu hỏi và trả lời liên quan đến công nghệ, tính năng, và công cụ trong môi trường phát triển .NET.

Lập trình .NET là quá trình phát triển ứng dụng sử dụng .NET Framework, một nền tảng phần mềm được phát triển bởi Microsoft. .NET Framework cung cấp môi trường để phát triển ứng dụng bằng nhiều ngôn ngữ lập trình, nhưng chủ yếu là C#. Visual Studio là một môi trường phát triển tích hợp (IDE) được sử dụng rộng rãi cho việc phát triển ứng dụng trên nền tảng .NET.

### 1. .NET Framework và C#:

- .NET Framework là nền tảng phần mềm do Microsoft phát triển.
- C# là ngôn ngữ lập trình chủ yếu được sử dụng cho việc phát triển ứng dụng trên nền tảng .NET.

### 2. Visual Studio:

- Visual Studio là môi trường phát triển tích hợp (IDE) dành cho việc phát triển ứng dụng trên nền tảng .NET.
- Cung cấp các công cụ hỗ trợ phát triển ứng dụng bằng nhiều ngôn ngữ, bao gồm C# và Java.
- Solution Explorer và Toolbox là các công cụ quan trọng trong Visual Studio giúp quản lý dự án và thực hiện các tác vụ phát triển.

### 3. Các tính năng và công cụ cụ thể:

- Delegate, readonly, Interface, LINQ là các tính năng quan trọng của ngôn ngữ lập trình C#.
- Trong Visual Studio, các công cụ như Debug Inspector, NuGet, Package Manager Console, Code Review, EditorConfig, và Code Map hỗ trợ các quá trình phát triển ứng dụng và kiểm thử.

### 4. Khác biệt giữa .NET Framework và .NET Core:

- .NET Core là phiên bản mới, đa nền tảng của .NET Framework, trong khi .NET Framework chỉ chạy trên Windows.

- .NET Core mang lại tính đa nền tảng và tối ưu hóa hiệu suất hơn.

## 5. Các khái niệm khác:

- Entity Framework trong .NET được sử dụng để tạo và quản lý cơ sở dữ liệu.
- Khái niệm "Debug" và "Release" trong .NET Framework phân biệt giữa quá trình phát triển và triển khai ứng dụng.
- Solution Platforms và EditorConfig là các công cụ hỗ trợ quản lý và cấu hình dự án trong Visual Studio

## 6. Một số câu hỏi gợi ý:

1. NET Framework là một nền tảng phần mềm do công ty nào phát triển?
2. .NET Framework cung cấp môi trường để phát triển ứng dụng dựa trên ngôn ngữ lập trình nào?
3. C# là ngôn ngữ lập trình được thiết kế chủ yếu cho việc phát triển ứng dụng trên nền tảng nào?
4. Đặc điểm nổi bật của C# bao gồm?
5. Visual Studio là môi trường phát triển tích hợp (IDE) dành cho việc phát triển ứng dụng trên?
6. Visual Studio cung cấp các công cụ hỗ trợ phát triển ứng dụng bằng nhiều ngôn ngữ, trong đó bao gồm cả?
7. Trong Visual Studio, Solution Explorer được sử dụng để?
8. Chức năng chính của Toolbox trong Visual Studio là?
9. "Solution" trong Visual Studio là?
10. Công cụ nào trong Visual Studio được sử dụng để kiểm soát việc debug và theo dõi giá trị của biến?
11. .NET Framework là một phần mềm có chức năng gì trong quá trình phát triển ứng dụng?
12. Thư viện lớn và tái sử dụng của .NET Framework được gọi là gì?
13. C# là ngôn ngữ lập trình thuộc họ ngôn ngữ nào?
14. Tính năng "Garbage Collection" trong C# liên quan đến việc gì?
15. "IntelliSense" trong Visual Studio là gì?
16. .NET Framework có thể chạy trên các hệ điều hành nào sau đây?
17. C# hỗ trợ tính năng "Delegate" để làm gì?
18. Trong C#, từ khóa "readonly" được sử dụng để làm gì?
19. "NuGet" trong Visual Studio được sử dụng để làm gì?
20. Trong Visual Studio, "Solution Configurations" được sử dụng để làm gì?
21. Trong Visual Studio, "Package Manager Console" được sử dụng để làm gì?
22. .NET Core và .NET Framework khác nhau như thế nào?
23. Trong C#, "Interface" được sử dụng để làm gì?

24. Trong Visual Studio, "Code Review" có ý nghĩa gì?
25. Trong Visual Studio, "Code Map" được sử dụng để làm gì?
26. Trong .NET Framework, sự khác biệt giữa "Debug" và "Release" là gì?
27. Trong Visual Studio, "Solution Platforms" có liên quan đến việc gì?
28. Trong C#, "LINQ" là viết tắt của cụm từ nào?
29. Trong Visual Studio, "EditorConfig" được sử dụng để làm gì?
30. Trong .NET, "Entity Framework" được sử dụng để làm gì?
31. Trong C#, "Lambda Expression" là gì và vai trò của nó trong lập trình là gì?
32. Trong Visual Studio, "Code Lens" là gì và vai trò của nó trong quá trình phát triển phần mềm là gì?
33. Trong .NET, "Nullable Types" là gì và vai trò của chúng trong lập trình là gì?
34. Trong C#, "Access Modifiers" là gì và vai trò của chúng trong lập trình là gì?
35. Trong Visual Studio, "Dependency Injection" là gì và vai trò của nó trong quá trình phát triển phần mềm là gì?
36. Trong .NET, "Parallel Programming" là gì và vai trò của nó trong lập trình là gì?
37. Trong C#, "Nullable Reference Types" là gì và vai trò của chúng trong lập trình là gì?
38. Trong Visual Studio, "Task List" là gì và vai trò của nó trong quá trình phát triển phần mềm là gì?
39. Trong .NET, "Asynchronous Programming" là gì và vai trò của nó trong lập trình là gì?
40. Trong C#, "Events" là gì và vai trò của chúng trong lập trình là gì?

## CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH C#

### 1. Một số nội dung của chương:

- **Kiểu dữ liệu và khai báo biến:** C# sử dụng các kiểu dữ liệu như int, double, char, bool để lưu trữ dữ liệu. Biến được khai báo bằng từ khóa var hoặc kiểu dữ liệu cụ thể.
- **Từ khóa và toán tử:** C# có các từ khóa như if, else, for, switch-case và toán tử như ++, --, &&, ||, ! để điều khiển luồng chương trình và thực hiện các phép toán.
- **Xử lý ngoại lệ và thao tác với mảng:** Sử dụng cấu trúc try-catch để xử lý ngoại lệ. Có các phương thức như Array.Copy() để sao chép mảng và Convert.ToInt32() để chuyển đổi chuỗi thành số nguyên.
- **Câu lệnh và cấu trúc điều kiện:** Sử dụng câu lệnh if-else và switch-case để thực hiện các hành động dựa trên điều kiện.
- **Cấu trúc lặp và toán tử nhảy:** Sử dụng vòng lặp for, while và các toán tử nhảy như break, continue để điều khiển luồng chương trình.
- **Xử lý chuỗi và ép kiểu:** Sử dụng các phương thức như Console.ReadLine() để nhập chuỗi từ bàn phím và Convert.ToString() để chuyển đổi kiểu dữ liệu.
- **Mảng đa chiều và phương thức của mảng:** Có thể tạo mảng đa chiều bằng cách sử dụng cú pháp int[,] array = new int[m, n] và sử dụng các phương thức như Array.Reverse() để đảo ngược mảng.
- **Cấu trúc điều khiển và xử lý dữ liệu:** Sử dụng các cấu trúc điều khiển để kiểm tra điều kiện và thực hiện các hành động tương ứng. Xử lý dữ liệu bằng cách sử dụng các phép toán và câu lệnh trong C#.

### 2. Chi tiết các nội dung cần ôn tập

- Kiểu dữ liệu và khai báo biến trong C#.
- Từ khóa và toán tử trong C#.
- Xử lý ngoại lệ và thao tác với mảng.
- Câu lệnh và cấu trúc điều kiện trong C#.
- Cấu trúc lặp và toán tử nhảy trong C#.
- Xử lý chuỗi và ép kiểu trong C#.
- Mảng đa chiều và phương thức của mảng trong C#.
- Cấu trúc điều khiển và xử lý dữ liệu trong C#.
- Kiểu dữ liệu và xử lý số trong C#.
- Xử lý chuỗi và thao tác với mảng trong C#.
- Cấu trúc điều khiển và vòng lặp trong C#.

- Kiểm tra điều kiện và thực hiện hành động tương ứng trong C#.
- Cấu trúc điều khiển và xử lý vòng lặp trong C#.
- Xử lý số và chuỗi trong C#.
- Xử lý mảng và kiểu dữ liệu đặc biệt trong C#.

### 3. Một số câu hỏi gợi ý:

1. C# sử dụng kiểu dữ liệu nào để lưu trữ số nguyên?
2. Kiểu dữ liệu nào trong C# được sử dụng để lưu trữ số thực?
3. Chú thích trong mã nguồn C# bắt đầu bằng ký tự nào?
4. Từ khóa nào được sử dụng để khai báo một biến trong C#?
5. Hằng số trong C# được khai báo bằng từ khóa nào?
6. Từ khóa nào không phải là một từ khóa trong C#?
7. Ký tự đặc biệt nào được sử dụng để kết thúc một câu lệnh trong C#?
8. Hàm nào được sử dụng để nhập dữ liệu từ bàn phím trong C#?
9. Để hiển thị giá trị của một biến, hàm nào được sử dụng trong C#?
10. Câu lệnh nào được sử dụng để thực hiện kiểm tra điều kiện trong C#?
11. Biểu thức nào sau đây là biểu thức logic trong C#?
12. Toán tử nào được sử dụng để tăng giá trị của một biến lên 1 trong C#?
13. Trong C#, câu lệnh nào được sử dụng để xử lý ngoại lệ?
14. Phương thức nào được sử dụng để sao chép một mảng trong C#?
15. Phương thức nào được sử dụng để chuyển đổi một chuỗi thành một số nguyên trong C#?
16. Cấu trúc điều kiện switch-case trong C# được sử dụng để làm gì?
17. Trong cấu trúc if-else, khối lệnh trong else sẽ được thực hiện khi nào?
18. Câu lệnh nào được sử dụng để thực hiện một vòng lặp for trong C#?
19. Vòng lặp do-while sẽ thực hiện ít nhất bao nhiêu lần trong C#?
20. Câu lệnh jump nào được sử dụng để thoát khỏi một vòng lặp trong C#?
21. Từ khóa "return" trong C# được sử dụng để làm gì?
22. Mảng trong C# được định nghĩa bằng cách nào?
23. Đối với mảng có kích thước 5, chỉ số cao nhất của mảng là bao nhiêu?
24. Mảng 2 chiều trong C# được gọi là gì?
25. Mảng động trong C# được triển khai bằng cách sử dụng kiểu dữ liệu nào?
26. Kiểu dữ liệu nào trong C# được sử dụng để lưu trữ giá trị boolean?
27. Trong C#, kiểu dữ liệu nào sử dụng để lưu trữ một ký tự?
28. Trong C#, hàm nào được sử dụng để đọc một số nguyên từ bàn phím và lưu vào biến?
29. Câu lệnh "continue" trong C# được sử dụng để làm gì?
30. Hàm nào được sử dụng để đưa ra một dòng mới khi xuất dữ liệu trong C#?
31. Để đọc một chuỗi từ bàn phím trong C#, hàm nào được sử dụng?
32. Trong C#, câu lệnh "else if" được sử dụng để làm gì?
33. Biểu thức nào tương đương với biểu thức "a = a + 1" trong C#?

34. Toán tử "&&" trong C# là toán tử gì?
35. Toán tử "!=" trong C# là toán tử gì?
36. Cấu trúc lặp "for-each" trong C# được sử dụng để làm gì?
37. Trong C#, cấu trúc lựa chọn "switch" được sử dụng với kiểu dữ liệu nào?
38. Trong cấu trúc if-else trong C#, nếu có nhiều điều kiện, nên sử dụng cấu trúc nào?
39. Phương thức nào được sử dụng để kiểm tra xem một phần tử có tồn tại trong mảng không?
40. Trong C#, cách khai báo một mảng động sử dụng List có thể thực hiện như thế nào?
41. Trong C#, kiểu dữ liệu "long" có bao nhiêu byte?
42. Kiểu dữ liệu "decimal" trong C# được sử dụng để làm gì?
43. Trong C#, để đọc một số thực từ bàn phím và lưu vào biến, hàm nào sẽ được sử dụng?
44. Biểu thức điều kiện ba ngôi (ternary) trong C# có cú pháp như thế nào?
45. Trong C#, làm thế nào để chuyển đổi một số nguyên thành một chuỗi có định dạng?
46. Trong C#, nếu có nhiều điều kiện phải kiểm tra và thực hiện các hành động khác nhau cho mỗi điều kiện, nên sử dụng cấu trúc nào?
47. Trong C#, cấu trúc lặp "for-each" có thể được sử dụng với loại dữ liệu nào?
48. Trong C#, làm thế nào để lấy giá trị của phần tử cuối cùng trong một mảng?
49. Trong C#, làm thế nào để tạo một mảng đa chiều có kích thước m x n?
50. Trong C#, làm thế nào để đảo ngược thứ tự các phần tử trong một mảng?

## CHƯƠNG 3: LỚP VÀ ĐỐI TƯỢNG TRONG .NET

Chương này tập trung vào các khái niệm cơ bản về lập trình hướng đối tượng (OOP) trong ngôn ngữ lập trình C#. Nội dung bao gồm các khái niệm về lớp và đối tượng, kế thừa, đóng gói, trừu tượng, đa hình, interface và các từ khóa đặc biệt trong C# như **abstract**, **virtual**, **override**, **sealed**.

### 1. Khái niệm cơ bản về lớp và đối tượng trong lập trình hướng đối tượng:

- Định nghĩa lớp và đối tượng.
- Quan hệ giữa lớp và đối tượng.
- Sử dụng từ khóa **new** để khởi tạo đối tượng.

### 2. Kế thừa và đa hình:

- Giới thiệu về kế thừa và đa hình.
- Sử dụng từ khóa **extends** để thể hiện kế thừa trong C#.
- Sử dụng từ khóa **virtual**, **override** để thể hiện đa hình.

### 3. Đóng gói và trừu tượng:

- Giải thích ý nghĩa của đóng gói và trừu tượng.
- Sử dụng từ khóa **abstract** để khai báo lớp trừu tượng.
- Sử dụng từ khóa **sealed** để ngăn chặn việc kế thừa hoặc ghi đè.

### 4. Interface:

- Định nghĩa và ứng dụng của interface.
- Sự khác biệt giữa lớp trừu tượng và interface.
- Sử dụng từ khóa **interface** để định nghĩa interface.

### 5. Các từ khóa đặc biệt trong C#:

- Giải thích ý nghĩa và cách sử dụng của các từ khóa như **abstract**, **virtual**, **override**, **sealed**.

### 6. Các khái niệm liên quan:

- Sự sử dụng của từ khóa **base**.
- **Static** class và **static** method.

- Từ khóa **this** trong C#.

## 7. Định nghĩa một số khái niệm:

- **Lớp và đối tượng:** Lớp là một khuôn mẫu để tạo ra các đối tượng, trong khi đối tượng là một thể hiện cụ thể của lớp.
- **Kế thừa:** Khái niệm cho phép một lớp con kế thừa tất cả thuộc tính và phương thức của lớp cha.
- **Đóng gói:** Quy trình che giấu thông tin và chi tiết triển khai của một lớp, chỉ tiết lộ các phương thức công khai.
- **Trừu tượng:** Một lớp trừu tượng không thể khởi tạo mà chỉ dùng để định nghĩa giao diện cho các lớp con.
- **Đa hình:** Khả năng của một phương thức để thực hiện các hành vi khác nhau tùy thuộc vào đối tượng gọi nó.
- **Interface:** Một tập hợp các phương thức và thuộc tính mà một lớp có thể triển khai.
- Các từ khóa đặc biệt: **abstract, virtual, override, sealed, this, base.**

## 8. Một số câu hỏi gợi ý:

1. Khái niệm nào miêu tả việc gom nhóm dữ liệu và phương thức vào một đối tượng?
2. Trong lập trình hướng đối tượng, từ khóa nào được sử dụng để tạo một đối tượng mới của một lớp?
3. Trong C#, để một lớp con kế thừa từ một lớp cha, ta sử dụng từ khóa nào sau tên lớp con?
4. Tính chất nào của lập trình hướng đối tượng giúp che giấu thông tin trong một lớp?
5. Làm thế nào để một lớp trừu tượng có thể có các phương thức trừu tượng?
6. Trong lập trình hướng đối tượng, tính chất nào cho phép một lớp có thể kế thừa từ nhiều lớp cha?
7. Từ khóa nào được sử dụng để ngăn chặn việc kế thừa từ một lớp trong C#?
8. Interface trong C# có thể chứa các thành phần nào?
9. Khi một lớp cài đặt triển khai một interface, lớp đó phải cung cấp định nghĩa cho những gì?
10. Trong lập trình hướng đối tượng, tính chất nào cho phép một lớp thay đổi hành vi của phương thức được định nghĩa trong lớp cha?
11. Từ khóa 'base' trong C# được sử dụng để làm gì?



12. Tính chất nào cho phép một lớp có thể có nhiều phương thức cùng tên nhưng có số lượng và kiểu tham số khác nhau?
13. Trong lập trình hướng đối tượng, từ khóa 'this' được sử dụng để tham chiếu đến cái gì?
14. Phương thức nào trong C# được gọi tự động khi một đối tượng được tạo?
15. Lớp con có thể ghi đè một phương thức của lớp cha bằng cách sử dụng từ khóa nào?
16. Từ khóa 'protected' trong C# được sử dụng để làm gì?
17. Từ khóa 'internal' trong C# được sử dụng để làm gì?
18. Làm thế nào để một lớp không thể được kế thừa trong C#?
19. Tính chất nào trong lập trình hướng đối tượng cho phép một lớp con có thể có các thành phần mới mà không làm thay đổi lớp cha?
20. Trong lập trình hướng đối tượng, làm thế nào để một lớp cha có thể ngăn chặn lớp con ghi đè một phương thức?
21. Lớp trừu tượng là gì trong lập trình hướng đối tượng?
22. Interface trong C# có những đặc điểm gì?
23. Tại sao C# không hỗ trợ đa kế thừa của lớp mà lại sử dụng interface?
24. Trong lập trình hướng đối tượng, encapsulation (tính đóng gói) là gì?
25. Từ khóa 'base' trong C# được sử dụng để làm gì?
26. Trong lập trình hướng đối tượng, static class có đặc điểm gì?
27. Polymorphism (tính đa hình) trong OOP có thể thể hiện như thế nào?
28. Từ khóa "sealed" trong C# được sử dụng để làm gì?
29. Làm thế nào để một lớp chỉ chứa các phương thức trừu tượng?
30. Làm thế nào để một lớp chỉ chứa các thuộc tính trừu tượng?
31. Trong OOP, encapsulation (tính đóng gói) giúp làm gì?
32. Trong OOP, polymorphism (tính đa hình) giúp làm gì?
33. Trong OOP, abstraction (tính trừu tượng) giúp làm gì?
34. Trong OOP, inheritance (tính kế thừa) giúp làm gì?
35. Trong OOP, interface giúp làm gì?
36. Từ khóa 'abstract' được sử dụng trong lập trình hướng đối tượng để làm gì?
37. Từ khóa 'virtual' được sử dụng trong lập trình hướng đối tượng để làm gì?
38. Từ khóa 'override' được sử dụng trong lập trình hướng đối tượng để làm gì?
39. Trong C#, từ khóa 'this' được sử dụng để làm gì?
40. Trong C#, từ khóa 'new' được sử dụng để làm gì?

## CHƯƠNG 4: CÁC ĐIỀU KHIỂN NGƯỜI DÙNG

### Phần 1: Tạo ứng dụng Windows Form và điều khiển cơ bản

1. **Tạo ứng dụng:** Windows Forms Application - Một ứng dụng desktop dùng để tạo giao diện người dùng trên hệ điều hành Windows.
2. **Biểu mẫu (Form):** Là cửa sổ chứa các điều khiển (controls) và nội dung của ứng dụng.
3. **Hiển thị văn bản:** Label - Điều khiển để hiển thị văn bản tĩnh trên Form.
4. **Lựa chọn giữa các tùy chọn:** CheckBox, RadioButton - Điều khiển cho phép người dùng chọn một hoặc nhiều tùy chọn từ một danh sách.
5. **Chọn giá trị từ danh sách:** ComboBox - Điều khiển cho phép người dùng chọn một giá trị từ một danh sách thả xuống.
6. **Menu và Toolbar:** Cung cấp các lựa chọn chức năng cho người dùng thông qua menu hoặc thanh công cụ.
7. **Hộp thoại Common Dialog:** Hiển thị các hộp thoại tiêu chuẩn như hộp thoại mở tệp, lưu tệp, hoặc chọn màu sắc.
8. **Hiển thị dữ liệu:** ListView, TreeView - Các điều khiển để hiển thị dữ liệu dưới dạng danh sách hoặc cây thư mục.
9. **Hiển thị hình ảnh:** PictureBox - Điều khiển để hiển thị hình ảnh trong ứng dụng.
10. **Gợi ý khi di chuột:** ToolTip - Hiển thị các gợi ý hoặc thông tin khi di chuột qua một điều khiển.

### Phần 2: Xử lý sự kiện và điều khiển nâng cao

11. **Xử lý sự kiện:** Button - Điều khiển để xử lý sự kiện khi người dùng nhấn vào.
12. **Chọn nhiều tùy chọn:** CheckedListBox - Tương tự như ListBox nhưng cho phép chọn nhiều mục.
13. **Menu ngữ cảnh:** ContextMenu - Hiển thị menu tương tác khi người dùng nhấp chuột phải vào một điều khiển.
14. **Tiến trình hoàn thành:** ProgressBar - Hiển thị tiến độ của một quá trình.
15. **Mở và lưu tệp tin:** OpenFileDialog, SaveFileDialog - Hộp thoại cho phép người dùng mở hoặc lưu tệp tin.

16. **Sự kiện bàn phím:** KeyDown, KeyPress - Xử lý sự kiện khi người dùng nhấn phím trên bàn phím.
17. **Sự kiện chuột:** MouseClick, MouseDoubleClick - Xử lý sự kiện khi người dùng tương tác với chuột.
18. **Chọn ngày tháng:** DateTimePicker - Cho phép người dùng chọn ngày và giờ từ một lịch hoặc nhập thủ công.
19. **Thông báo lỗi:** ErrorProvider - Hiển thị thông báo lỗi hoặc cảnh báo khi dữ liệu không hợp lệ.
20. **Điều chỉnh giá trị:** TrackBar - Điều khiển cho phép người dùng điều chỉnh giá trị trong một phạm vi nhất định.

### **Phần 3: các câu hỏi gợi ý**

1. Để tạo một ứng dụng Windows Form mới, công cụ nào thường được sử dụng trong Visual Studio?
2. Biểu mẫu (Form) trong Windows Form được sử dụng để mục đích nào sau đây?
3. Để hiển thị một dòng văn bản tĩnh trên Form, điều khiển nào thường được sử dụng?
4. CheckBox và RadioButton thường được sử dụng để làm gì?
5. Để chọn một giá trị từ danh sách, điều khiển nào thường được sử dụng?
6. Menu và Toolbar có chung mục đích gì trong một ứng dụng Windows Form?
7. Hộp thoại Common Dialog thường được sử dụng cho mục đích nào sau đây?
8. ListView và TreeView thường được sử dụng để hiển thị dữ liệu dạng gì?
9. PictureBox thường được sử dụng để hiển thị nội dung gì trong ứng dụng?
10. ToolTip được sử dụng để làm gì trong ứng dụng Windows Form?
11. TextBox được sử dụng để nhập liệu từ người dùng, trong khi Button thường được sử dụng để làm gì?
12. Trong các điều khiển sau, điều khiển nào cho phép người dùng chọn nhiều tùy chọn?
13. Khi nào thì ContextMenu thường xuất hiện trong ứng dụng Windows Form?
14. ProgressBar thường được sử dụng để theo dõi tiến độ của quá trình nào?
15. Hộp thoại OpenFileDialog và SaveFileDialog thường được sử dụng cho mục đích gì?

16. Sự kiện nào được kích hoạt khi người dùng nhấn một phím trên bàn phím?
17. Khi người dùng nhấn chuột một lần, sự kiện nào được kích hoạt?
18. DateTimePicker thường được sử dụng để làm gì trong ứng dụng Windows Form?
19. ErrorProvider thường được sử dụng để làm gì trong ứng dụng Windows Form?
20. TrackBar là điều khiển được sử dụng để điều chỉnh gì trong ứng dụng?
21. Thuộc tính AutoSize của PictureBox được sử dụng để làm gì?
22. Làm thế nào để ẩn một điều khiển trên Form trong lúc chương trình đang chạy?
23. ErrorProvider thường được sử dụng để làm gì trong ứng dụng Windows Form?
24. ToolStripMenuItem là một phần của điều khiển nào trong Windows Form?
25. Trong ListBox, thuộc tính nào cho phép người dùng chọn nhiều mục cùng một lúc?
26. Sự kiện CheckedChanged thường liên quan đến điều khiển nào?
27. Điều khiển nào được sử dụng để tạo menu ngữ cảnh trong ứng dụng Windows Form?
28. Thuộc tính TextAlign của TextBox được sử dụng để làm gì?
29. Khi người dùng nhấn đúp chuột, sự kiện nào được kích hoạt?
30. ProgressBar thường được sử dụng để theo dõi tiến độ của quá trình nào?
31. Thuộc tính DropDownStyle của ComboBox quyết định điều gì?
32. Sự kiện SelectedIndexChanged thường liên quan đến điều khiển nào?
33. Sự khác biệt giữa MenuStrip và ContextMenuStrip là gì?
34. Làm thế nào để xử lý sự kiện khi người dùng thay đổi giá trị của một ComboBox?
35. Trong ListView, cách để thêm các cột và dữ liệu vào từng cột là gì?
36. Sự kiện KeyPress và KeyDown có điểm khác biệt như thế nào?
37. Trong ComboBox, sự kiện DropDown được kích hoạt khi nào?
38. Trong CheckBox, thuộc tính ThreeState có ý nghĩa gì?
39. Trong TreeView, phương thức nào được sử dụng để thêm một nút mới?

40. Để hiển thị một hộp thoại FontDialog và cho phép người dùng chọn font, ta sử dụng điều khiển nào?
41. Trong Windows Form, điều khiển nào được sử dụng để hiển thị danh sách các mục có thể chọn?
42. Điều khiển nào được sử dụng để cho phép người dùng chọn một hoặc nhiều mục từ một danh sách?
43. Thuộc tính nào của điều khiển ListBox quy định chế độ lựa chọn của người dùng?
44. Sự kiện nào được kích hoạt khi giá trị của một điều khiển ComboBox được thay đổi?
45. Trong Windows Form, điều khiển nào được sử dụng để hiển thị hộp thoại mở và lưu tệp tin?
46. Điều khiển nào được sử dụng để hiển thị một dải giá trị mà người dùng có thể chọn bằng cách kéo thanh trượt?
47. Sự kiện nào được kích hoạt khi giá trị của một điều khiển TextBox thay đổi?
48. Trong Windows Form, điều khiển nào được sử dụng để hiển thị một danh sách các mục có thể chọn bằng cách nhấp vào?
49. Điều khiển nào được sử dụng để hiển thị một biểu tượng hoặc hình ảnh trong ứng dụng?
50. Thuộc tính nào của điều khiển ProgressBar quy định giá trị tiến trình hiện tại của nó?
51. Trong Windows Form, điều khiển nào được sử dụng để hiển thị một lịch hoặc hộp thoại lịch?
52. Điều khiển nào được sử dụng để hiển thị một hộp thoại lỗi khi người dùng nhập liệu không hợp lệ?
53. Trong Windows Form, điều khiển nào được sử dụng để hiển thị một danh sách các mục dưới dạng cây?
54. Sự kiện nào được kích hoạt khi người dùng nhấn phím trên bàn phím khi điều khiển đang được chọn?
55. Trong Windows Form, điều khiển nào được sử dụng để hiển thị một danh sách các mục có thể chọn bằng cách nhấp vào nút mũi tên?

56. Điều khiển nào được sử dụng để hiển thị một danh sách các mục có thể chọn bằng cách kiểm tra hoặc bỏ chọn các hộp kiểm?
57. Thuộc tính nào của điều khiển DateTimePicker quy định ngày tháng được hiển thị mặc định khi điều khiển được tạo?
58. Trong Windows Form, điều khiển nào được sử dụng để hiển thị thông tin trong một danh sách có thể cuộn?
59. Sự kiện nào được kích hoạt khi người dùng nhấn chuột trái lên một điều khiển?
60. Trong Windows Form, điều khiển nào được sử dụng để hiển thị một hộp thoại để chọn font?

## CHƯƠNG 5: PHÁT TRIỂN ỨNG DỤNG CƠ SỞ DỮ LIỆU

### 1. Cơ bản về ADO.NET

- Tập hợp các lớp trong .NET Framework để làm việc với cơ sở dữ liệu.
- Cung cấp các đối tượng như Connection, Command, DataAdapter, DataSet để kết nối, truy vấn và làm việc với cơ sở dữ liệu.

### 2. Các đối tượng chính trong ADO.NET

1. **Connection:** Đại diện cho kết nối đến cơ sở dữ liệu.
2. **Command:** Thực hiện các truy vấn hoặc thao tác với cơ sở dữ liệu.
3. **DataAdapter:** Cầu nối giữa DataSet và cơ sở dữ liệu.
4. **DataSet:** Bộ nhớ cache dữ liệu không gian, chứa các bảng dữ liệu.
5. **DataReader:** Bộ đọc dữ liệu hàng loạt chỉ đọc dữ liệu một cách tuần tự từ cơ sở dữ liệu.
6. **DataRow:** DataRow đại diện cho một dòng trong DataTable và chứa dữ liệu của dòng đó.
7. **DataTable:** Bảng dữ liệu trong DataSet, chứa các hàng và cột dữ liệu.
8. **DataView:** Bảng nhìn (view) của DataTable, cho phép lọc, sắp xếp và tìm kiếm dữ liệu.
9. **CommandBuilder:** Lớp giúp tự động tạo các lệnh SQL (Insert, Update, Delete) dựa trên schema của DataTable.
10. **DataSet:** Bộ nhớ cache dữ liệu không gian, cho phép kiểm soát các thay đổi trên dữ liệu.

### 3. Các câu hỏi gợi ý:

1. ADO.NET là gì và vai trò của nó trong phát triển ứng dụng cơ sở dữ liệu là gì?
2. Trong ADO.NET, hãy nêu khái niệm "Disconnected Data Architecture"?
3. Mô tả các thành phần chính của ADO.NET và chức năng của mỗi thành phần.
4. CommandBuilder trong ADO.NET được sử dụng để làm gì và cách sử dụng nó như thế nào?
5. Sự khác biệt giữa DataReader và DataSet trong ADO.NET là gì?

6. DataRow và DataTable trong ADO.NET có vai trò gì và cách sử dụng chúng?
7. DataAdapter trong ADO.NET là gì và cách sử dụng để điều phối dữ liệu giữa DataSet và cơ sở dữ liệu là gì?
8. Command.ExecuteScalar() trong ADO.NET được sử dụng để thực hiện công việc gì?
9. Command.CommandType trong ADO.NET có những giá trị nào và mỗi giá trị đại diện cho công việc gì?
10. DataView trong ADO.NET được sử dụng để làm gì và cách sử dụng nó như thế nào?
11. Sự khác biệt giữa ExecuteNonQuery(), ExecuteReader(), và ExecuteScalar() trong ADO.NET là gì?
12. DataSet là gì và cách sử dụng nó để lưu trữ và làm việc với dữ liệu?
13. DataAdapter.Fill() trong ADO.NET được sử dụng để làm gì và cách sử dụng nó như thế nào?
14. ConnectionString trong ADO.NET chứa thông tin gì và cách sử dụng nó như thế nào?
15. Sự khác biệt giữa Command và Connection trong ADO.NET là gì và khi nào cần sử dụng mỗi đối tượng?
16. Mô tả quy trình thực hiện truy vấn dữ liệu từ cơ sở dữ liệu và hiển thị kết quả trên giao diện người dùng sử dụng ADO.NET.
17. Sự khác biệt giữa việc thực hiện truy vấn dữ liệu từ cơ sở dữ liệu bằng SQL trực tiếp và sử dụng ADO.NET là gì?
18. Command.Parameters trong ADO.NET được sử dụng để làm gì và cách sử dụng nó như thế nào?
19. CommandType.StoredProcedure trong ADO.NET được sử dụng để thực hiện công việc gì và cách sử dụng nó như thế nào?
20. Connection.State trong ADO.NET cho biết thông tin gì về trạng thái của kết nối và tại sao nó quan trọng?
21. DataAdapter.Update() trong ADO.NET được sử dụng để làm gì và cách sử dụng nó như thế nào?



22. `DataSet.AcceptChanges()` trong ADO.NET được sử dụng để làm gì và khi nào cần sử dụng?
23. `DataRow.RowState` trong ADO.NET có ý nghĩa gì và cách sử dụng nó để kiểm tra trạng thái của một dòng dữ liệu?
24. Điều gì xảy ra khi sử dụng `Command.ExecuteReader()` để thực hiện truy vấn dữ liệu trong ADO.NET?
25. `DataTable.PrimaryKey` trong ADO.NET có tác dụng gì và cách sử dụng nó để xác định khóa chính của bảng dữ liệu?
26. `DataView.Sort` trong ADO.NET được sử dụng để làm gì và cách sử dụng nó để sắp xếp dữ liệu trong một `DataView`?
27. `Command.Transaction` trong ADO.NET được sử dụng để làm gì và cách sử dụng nó để thực hiện giao dịch trong cơ sở dữ liệu?
28. `DataSet.Merge()` trong ADO.NET được sử dụng để làm gì và khi nào cần sử dụng để hợp nhất dữ liệu từ một `DataSet` vào một `DataSet` khác?
29. `Connection.Open()` và `Connection.Close()` trong ADO.NET có ý nghĩa gì và khi nào cần sử dụng mỗi phương thức?
30. Sự khác biệt giữa `DataAdapter.Fill()` và `DataAdapter.Update()` trong ADO.NET là gì và khi nào cần sử dụng mỗi phương thức?
31. `DataSet.HasChanges()` trong ADO.NET được sử dụng để làm gì và cách sử dụng nó để kiểm tra xem `DataSet` có thay đổi không?
32. Trong ADO.NET, `DataView.RowFilter` được sử dụng để làm gì và cách sử dụng nó để lọc dữ liệu trong một `DataView`?
33. `DataAdapter.FillSchema()` trong ADO.NET được sử dụng để làm gì và khi nào cần sử dụng nó để tạo schema cho `DataTable` trước khi điền dữ liệu?
34. `DataSet.GetChanges()` trong ADO.NET được sử dụng để làm gì và cách sử dụng nó để lấy các thay đổi trong `DataSet`?
35. Đối tượng `CommandBuilder` trong ADO.NET có tác dụng gì và cách sử dụng nó để tự động tạo các lệnh SQL cập nhật?
36. `DataView.Find()` trong ADO.NET được sử dụng để làm gì và cách sử dụng nó để tìm một dòng trong `DataView`?
37. `DataSet.Clone()` trong ADO.NET được sử dụng để làm gì và khi nào cần sử dụng nó để tạo một bản sao của `DataSet`?

38. `DataRow.BeginEdit()` và `DataRow.EndEdit()` trong ADO.NET được sử dụng để làm gì và cách sử dụng chúng để chỉnh sửa dữ liệu trong một dòng?
39. `Command.ExecuteScalar()` trong ADO.NET được sử dụng để làm gì và cách sử dụng nó để trả về một giá trị duy nhất từ một truy vấn?
40. `Connection.ConnectionString` trong ADO.NET chứa thông tin gì và khi nào cần sử dụng nó để thiết lập kết nối đến cơ sở dữ liệu?
41. `DataAdapter.SelectCommand.CommandType` trong ADO.NET được sử dụng để làm gì và khi nào cần sử dụng nó để xác định loại lệnh của Command được sử dụng cho DataAdapter?
42. `DataSet.Tables` trong ADO.NET chứa thông tin gì và khi nào cần sử dụng nó để truy cập các bảng dữ liệu trong DataSet?
43. `ConnectionTimeout` và `CommandTimeout` trong ADO.NET có ý nghĩa gì và khi nào cần điều chỉnh chúng khi làm việc với cơ sở dữ liệu?
44. Sự khác biệt giữa `SqlCommand` và `OleDbCommand` trong ADO.NET là gì và khi nào nên sử dụng mỗi loại?
45. `DataSet.Clear()` trong ADO.NET được sử dụng để làm gì và khi nào cần sử dụng nó để xóa tất cả dữ liệu trong DataSet?
46. `DataAdapter.UpdateBatchSize` trong ADO.NET được sử dụng để làm gì và khi nào cần điều chỉnh kích thước lô cập nhật khi làm việc với cơ sở dữ liệu?
47. Trong ADO.NET, `DataSet.Merge()` được sử dụng để làm gì và cách sử dụng nó để hợp nhất dữ liệu từ một DataSet khác vào DataSet hiện tại?
48. `Command.Parameters` trong ADO.NET được sử dụng để làm gì và cách sử dụng chúng để truyền tham số cho một lệnh SQL?
49. Trong ADO.NET, `DataAdapter.Update()` được sử dụng để làm gì và cách sử dụng nó để cập nhật các thay đổi từ DataSet vào cơ sở dữ liệu?
50. `DataSet.HasChanges()` trong ADO.NET được sử dụng để làm gì và cách sử dụng nó để kiểm tra xem DataSet có chứa các thay đổi hay không?
51. `Command.Transaction` trong ADO.NET được sử dụng để làm gì và cách sử dụng nó để thực hiện giao dịch (transaction) trong cơ sở dữ liệu?
52. `DataRow.RowStateFilter` trong ADO.NET được sử dụng để làm gì và cách sử dụng nó để lọc các dòng dựa trên trạng thái của chúng trong DataView?

53. `Connection.Open()` và `Connection.Close()` trong ADO.NET được sử dụng để làm gì và cách sử dụng chúng để mở và đóng kết nối đến cơ sở dữ liệu?
54. `DataRow.Delete()` trong ADO.NET được sử dụng để làm gì và cách sử dụng nó để đánh dấu một dòng để xóa khỏi `DataSet`?
55. `Command.Prepare()` trong ADO.NET được sử dụng để làm gì và cách sử dụng nó để chuẩn bị một lệnh SQL trước khi thực hiện nó?
56. `DataAdapter.RowUpdating` và `DataAdapter.RowUpdated` trong ADO.NET được sử dụng để làm gì và cách sử dụng chúng để xử lý sự kiện trước và sau khi cập nhật dữ liệu?
57. `DataSet.RejectChanges()` trong ADO.NET được sử dụng để làm gì và khi nào cần sử dụng nó để hủy bỏ tất cả các thay đổi chưa được chấp nhận trong `DataSet`?
58. Trong ADO.NET, điều gì xảy ra khi một lệnh SQL không hợp lệ được thực thi bằng `Command.ExecuteNonQuery()`?
59. `DataAdapter.Fill()` và `DataAdapter.FillSchema()` trong ADO.NET có điểm khác biệt gì và khi nào nên sử dụng mỗi phương thức?
60. `ConnectionStringBuilder` trong ADO.NET được sử dụng để làm gì và cách sử dụng nó để xây dựng chuỗi kết nối một cách linh hoạt và an toàn?
61. Trong ADO.NET, `Command.ExecuteReader()` và `DataReader.Read()` làm gì và cách sử dụng chúng để thực hiện đọc dữ liệu từ cơ sở dữ liệu một cách hiệu quả và linh hoạt?
62. Sử dụng phương thức `GetAutoIncrementValue()` của `DataColumn` để làm gì?