Thanh cuộn ScrollBar

(nói)Thanh cuộn (ScrollBar) trong WinForms là một thành phần giao diện người dùng cho phép người dùng chọn giá trị từ một phạm vi cụ thể. Có hai loại thanh cuộn chính trong WinForms: Thanh cuộn ngang (HScrollBar) và thanh cuộn dọc (VScrollBar), được sử dụng để cuộn nội dung ngang và dọc trong các điều khiển khung (Panel, Form, UserControl) hoặc bất kỳ điều khiển nào có nội dung vượt ra khỏi kích thước hiển thị.

(nói)1. Trước hết, hãy tạo một ứng dụng Windows Forms mới.

(nói)2. Thêm thanh cuộn vào mẫu của bạn. Bạn có thể thêm chúng bằng cách kéo và thả từ hộp công cụ (Toolbox) hoặc bằng cách thêm chúng trong code. Ví dụ:

*```csharp*

*// Thêm thanh cuộn ngang*

*HScrollBar hScrollBar = new HScrollBar();*

*hScrollBar.Location = new Point(10, 10);*

*hScrollBar.Width = 200;*

*hScrollBar.Minimum = 0;*

*hScrollBar.Maximum = 100;*

*hScrollBar.SmallChange = 1;*

*hScrollBar.LargeChange = 10;*

*hScrollBar.Scroll += HScrollBar\_Scroll;*

*this.Controls.Add(hScrollBar);*

*// Thêm thanh cuộn dọc*

*VScrollBar vScrollBar = new VScrollBar();*

*vScrollBar.Location = new Point(220, 10);*

*vScrollBar.Height = 200;*

*vScrollBar.Minimum = 0;*

*vScrollBar.Maximum = 100;*

*vScrollBar.SmallChange = 1;*

*vScrollBar.LargeChange = 10;*

*vScrollBar.Scroll += VScrollBar\_Scroll;*

*this.Controls.Add(vScrollBar);*

*```*

(nói)3. Định nghĩa các sự kiện Scroll cho cả thanh cuộn ngang và dọc:

*```csharp*

*private void HScrollBar\_Scroll(object sender, ScrollEventArgs e)*

*{*

*// Xử lý sự kiện khi thanh cuộn ngang được cuộn*

*int value = ((HScrollBar)sender).Value;*

*// Thực hiện xử lý tương ứng với giá trị cuộn*

*}*

*private void VScrollBar\_Scroll(object sender, ScrollEventArgs e)*

*{*

*// Xử lý sự kiện khi thanh cuộn dọc được cuộn*

*int value = ((VScrollBar)sender).Value;*

*// Thực hiện xử lý tương ứng với giá trị cuộn*

*}*

*```*

4. Bạn có thể xử lý sự kiện Scroll để thay đổi nội dung hiển thị hoặc thực hiện các tác vụ tùy ý dựa trên giá trị cuộn của thanh cuộn.

Phương thức ,thuộc tính, sự kiện của thanh HscrollBar

(nói)Trong C#, thanh HScrollBar (Horizontal ScrollBar) là một thành phần giao diện người dùng (UI) được sử dụng để cuộn nội dung ngang trên các điều khiển khung hình (viewport) trong ứng dụng Windows Forms hoặc WPF. Dưới đây là một số phương thức, thuộc tính và sự kiện quan trọng của thanh HScrollBar trong C#:

\*\*Phương thức:\*\*

1. `Scroll`: Sử dụng để cuộn thanh ScrollBar đến một giá trị cụ thể.

2. `PerformScroll`: Sử dụng để thực hiện cuộn tự động của thanh ScrollBar.

3. `Update`: Cập nhật lại thanh ScrollBar sau khi đã thay đổi các thuộc tính hoặc dữ liệu.

\*\*Thuộc tính:\*\*

1. `Value`: Xác định giá trị hiện tại của thanh ScrollBar. Bạn có thể đọc hoặc gán giá trị này.

2. `Minimum` và `Maximum`: Xác định giá trị tối thiểu và tối đa mà thanh ScrollBar có thể đạt được.

3. `SmallChange` và `LargeChange`: Định rõ lượng giá trị mà ScrollBar sẽ cuộn khi người dùng thực hiện cuộn nhỏ hoặc cuộn lớn.

4. `Enabled`: Xác định liệu thanh ScrollBar có được kích hoạt hay không.

\*\*Sự kiện:\*\*

1. `Scroll`: Được kích hoạt khi người dùng thực hiện cuộn trên thanh ScrollBar. Sự kiện này cung cấp thông tin về giá trị mới của thanh ScrollBar.

2. `ValueChanged`: Tương tự như `Scroll`, sự kiện này xảy ra khi giá trị của thanh ScrollBar thay đổi, cho phép bạn xử lý sự kiện khi giá trị thay đổi bất kể nguồn gây ra thay đổi là gì.

*```csharp*

*// Tạo một thanh HScrollBar*

*HScrollBar hScrollBar = new HScrollBar();*

*hScrollBar.Minimum = 0;*

*hScrollBar.Maximum = 100;*

*hScrollBar.SmallChange = 1;*

*hScrollBar.LargeChange = 10;*

*// Gắn sự kiện Scroll*

*hScrollBar.Scroll += (sender, e) =>*

*{*

*// Xử lý khi người dùng cuộn thanh ScrollBar*

*int newValue = hScrollBar.Value;*

*// Thực hiện hành động tương ứng với giá trị mới (newValue)*

*};*

*// Thêm thanh ScrollBar vào giao diện người dùng*

*this.Controls.Add(hScrollBar);*

*```*

phương thức ,thuộc tính, sự kiện của thanh VscrollBar

(nói)Trong C#, thanh VScrollBar (Vertical ScrollBar) là một thành phần giao diện người dùng (UI) tương tự như thanh HScrollBar, nhưng được sử dụng để cuộn nội dung dọc trong các điều khiển khung hình (viewport) trong ứng dụng Windows Forms hoặc WPF.

\*\*Phương thức:\*\*

1. `Scroll`: Sử dụng để cuộn thanh ScrollBar đến một giá trị cụ thể.

2. `PerformScroll`: Sử dụng để thực hiện cuộn tự động của thanh ScrollBar.

3. `Update`: Cập nhật lại thanh ScrollBar sau khi đã thay đổi các thuộc tính hoặc dữ liệu.

\*\*Thuộc tính:\*\*

1. `Value`: Xác định giá trị hiện tại của thanh ScrollBar. Bạn có thể đọc hoặc gán giá trị này.

2. `Minimum` và `Maximum`: Xác định giá trị tối thiểu và tối đa mà thanh ScrollBar có thể đạt được.

3. `SmallChange` và `LargeChange`: Định rõ lượng giá trị mà ScrollBar sẽ cuộn khi người dùng thực hiện cuộn nhỏ hoặc cuộn lớn.

4. `Enabled`: Xác định liệu thanh ScrollBar có được kích hoạt hay không.

\*\*Sự kiện:\*\*

1. `Scroll`: Được kích hoạt khi người dùng thực hiện cuộn trên thanh ScrollBar. Sự kiện này cung cấp thông tin về giá trị mới của thanh ScrollBar.

2. `ValueChanged`: Tương tự như `Scroll`, sự kiện này xảy ra khi giá trị của thanh ScrollBar thay đổi, cho phép bạn xử lý sự kiện khi giá trị thay đổi bất kể nguồn gây ra thay đổi là gì.

```csharp

*// Tạo một thanh VScrollBar*

*VScrollBar vScrollBar = new VScrollBar();*

*vScrollBar.Minimum = 0;*

*vScrollBar.Maximum = 100;*

*vScrollBar.SmallChange = 1;*

*vScrollBar.LargeChange = 10;*

*// Gắn sự kiện Scroll*

*vScrollBar.Scroll += (sender, e) =>*

*{*

*// Xử lý khi người dùng cuộn thanh ScrollBar*

*int newValue = vScrollBar.Value;*

*// Thực hiện hành động tương ứng với giá trị mới (newValue)*

*};*

*// Thêm thanh ScrollBar vào giao diện người dùng*

*this.Controls.Add(vScrollBar);*

*```*

**ToolTip**

(nói)Trong lập trình Windows Forms (WF) bằng C#, bạn có thể sử dụng Tooltip để hiển thị các thông báo hoặc gợi ý khi người dùng di chuột qua hoặc tập trung vào các điều khiển giao diện người dùng. Đây là cách thêm và sử dụng Tooltip trong ứng dụng Windows Forms:

1. \*\*Thêm Tooltip vào Form:\*\*

Trước tiên, bạn cần thêm một Tooltip vào Form của bạn. Bạn có thể làm điều này trong phần thiết kế hoặc cả trong mã.

- \*\*Thiết kế (Design Time):\*\*

- Trong Visual Studio, mở Form của bạn.

- Tìm trong hộp công cụ (Toolbox) và kéo TooltipControl vào Form.

- Tooltip sẽ xuất hiện trong thanh tác vụ phía dưới Form và bạn có thể thay đổi thuộc tính của nó thông qua cửa sổ Properties.

phương thức ,thuộc tính, sự kiện của thanh Tooltip

Trong Windows Forms (WF) của C#, thanh Tooltip được biểu diễn bởi lớp `ToolTip`. Dưới đây là một số phương thức, thuộc tính và sự kiện quan trọng của lớp `ToolTip`:

\*\*Phương thức:\*

1. `SetToolTip`: Sử dụng để gán một Tooltip cho một điều khiển cụ thể. Cần cung cấp điều khiển và nội dung Tooltip.

2. `GetToolTip`: Sử dụng để lấy nội dung Tooltip được gán cho một điều khiển cụ thể.

\*\*Thuộc tính:\*\*

1. `AutoPopDelay`: Xác định khoảng thời gian (theo mili giây) mà Tooltip tự động biến mất sau khi hiển thị.

2. `InitialDelay`: Xác định khoảng thời gian (theo mili giây) trước khi Tooltip xuất hiện khi người dùng di chuột qua điều khiển.

3. `ReshowDelay`: Xác định khoảng thời gian (theo mili giây) giữa việc di chuột ra khỏi điều khiển và khi Tooltip xuất hiện lại khi người dùng quay lại điều khiển.

4. `ShowAlways`: Xác định liệu Tooltip sẽ hiển thị luôn (true) hoặc chỉ khi cần thiết (false).

\*\*Sự kiện:\*\*

1. `Popup`: Sự kiện này xảy ra trước khi Tooltip hiển thị. Bạn có thể sử dụng nó để thay đổi nội dung Tooltip hoặc hủy việc hiển thị Tooltip.

2. `Draw`: Sự kiện này xảy ra khi Tooltip được vẽ. Bạn có thể sử dụng nó để tùy chỉnh cách Tooltip được hiển thị.

Dưới đây là một ví dụ đơn giản về cách sử dụng các phương thức và thuộc tính chính của lớp `ToolTip` trong Windows Forms:

```csharp

using System;

using System.Windows.Forms;

namespace TooltipExample

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

// Tạo đối tượng Tooltip

ToolTip tooltip = new ToolTip();

// Gán Tooltip cho Button

tooltip.SetToolTip(button1, "This is a Button");

// Cài đặt khoảng thời gian xuất hiện Tooltip

tooltip.AutoPopDelay = 2000; // 2 giây

// Cài đặt khoảng thời gian trước khi Tooltip xuất hiện

tooltip.InitialDelay = 500; // 0.5 giây

// Cài đặt khoảng thời gian giữa việc di chuột ra và hiển thị lại Tooltip

tooltip.ReshowDelay = 100; // 0.1 giây

}

}

}

```