

Truyền dữ liệu giữa hai màn hình

# Nội dung chính

- ❑ Sử dụng thành phần tĩnh toàn cục
- ❑ Đối số hàm tạo
- ❑ Sử dụng delegate & event

Thành phần tính toàn cục

# Cách dùng

- ❑ Sử dụng một lớp ở mức toàn cục
- ❑ Lưu dữ liệu vào thành phần tĩnh

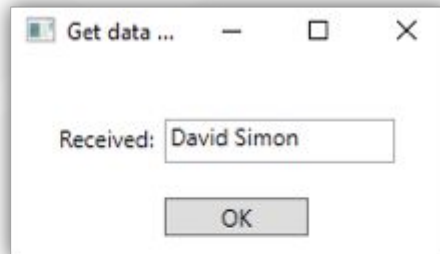
# Ví dụ minh họa



# Cải tiến

- ❑ Sao phải phí sức dùng thêm lớp?
- ❑ Dùng chính lớp màn hình đích được không?

# Thêm thành phần tĩnh cho màn hình đích



```
public static string Data;
```

```
private void btnPassClick(object sender, RoutedEventArgs e)
{
    // Store data before passing
    Data = txtName.Text;

    var screen = new GetGlobalWindow();
    screen.Show();
}
```

```
public partial class GetGlobalWindow : Window
{
    public GetGlobalWindow()
    {
        InitializeComponent();

        // Get data source screen
        txtData.Text = MainWindow.Data;
    }
}
```

Đổi số hàm tạo



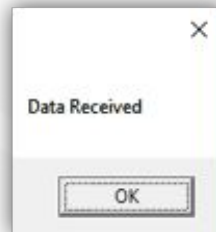
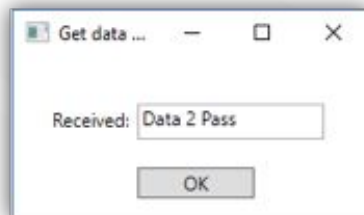
# Truyền đi bằng đối số hàm tạo



```
private void btnPassClick(object sender, RoutedEventArgs e)
{
    var screen = new GetGlobalWindow(txtName.Text);
    screen.Show();
}
```

```
public partial class GetGlobalWindow : Window
{
    public GetGlobalWindow(string data)
    {
        InitializeComponent();
        txtData.Text = data;
    }
}
```

# Nhận dữ liệu về bằng public property



```
public partial class MainWindow : Window
{
    public static string Data;

    private void btnPassClick(object sender, RoutedEventArgs e)
    {
        var screen = new GetGlobalWindow(txtName.Text);
        if (screen.ShowDialog().Value == true)
        {
            MessageBox.Show(screen.ReturnData);
        }
    }
}
```

```
public partial class GetGlobalWindow : Window
{
    public string ReturnData;

    public GetGlobalWindow(string data)
    {
        InitializeComponent();
        txtData.Text = data;
    }

    private void btnOK_Click(object sender, RoutedEventArgs e)
    {
        ReturnData = txtData.Text;
        DialogResult = true;
    }
}
```

Delegate & Event

# Delegate là gì?

- ❑ **Kiểu** con trỏ hàm
- ❑ Chỉ đơn thuần là **kiểu dữ liệu**

```
delegate int Calculator(int a, int b);
```

# Ví dụ về sử dụng delegate

```
delegate int Calculator(int a, int b);

static int Sum(int a, int b)
{
    return a + b;
}

static int Multiply(int a, int b)
{
    return a * b;
}

static void Main(string[] args)
{
    Calculator func = Sum;
    Console.WriteLine(func(5, 7));

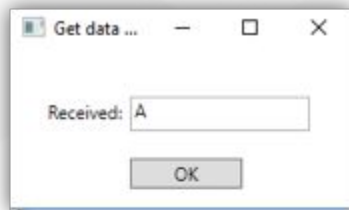
    func = Multiply;
    Console.WriteLine(func(3, 4));
}
```

# Event là gì?

Con trỏ hàm có kiểu định nghĩa từ delegate

```
delegate void ClickFunctionType(object sender, EventArgs e);  
event ClickFunctionType Click;  
  
public Form1()  
{  
    InitializeComponent();  
    Click += button1_Click;  
}  
  
private void button1_Click(object sender, EventArgs e)  
{  
}
```

# Sử dụng delegate & event để truyền nhận dữ liệu



```
private void btnPassClick(object sender, RoutedEventArgs e)
{
    var screen = new GetGlobalWindow(txtName.Text);
    screen.Handler += Screen_Handler;
    screen.ShowDialog();
}
private void Screen_Handler(string data)
{
    txtName.Text = data;
}
```

```
public delegate void MyDelegateType(string data);
public event MyDelegateType Handler;
```

```
private void txtData_TextChanged(object sender, TextChangedEventArgs e)
{
    if (Handler != null)
        Handler(txtData.Text);
}
```

# Ví dụ khác về truyền nhận dữ liệu

Link [demo](#)



# Bài luyện tập

- ❑ Tạo ra một màn hình chứa task list TODO với một nút bấm
- ❑ Bấm vào sẽ hiện ra màn hình khác để nhập tên công việc
- ❑ Bấm Add ở màn hình mới này sẽ quay lại màn hình chính
- ❑ Màn hình chính tự cập nhật danh sách công việc đã thay đổi