

9 Windows GUI

1. 历史
2. RAD
3. 消息循环
4. 添加组件
5. 组件的事件
6. 事件的参数
7. 事件的发送者
8. 多个处理器的组合
9. 应用类的事件
10. 创建多个窗口
11. 控件的使用
12. 自定义Windows 控件
13. GDI + 编程

1 历史

- GUI 事件驱动机制。GUI程序结构。
 - 1.注册窗口类, 大小,形状, 窗口过程.
 2. 创建窗口, 显示窗口
 - 3.进入消息循环
 - 4.其他对象向消息队列中发送消息(硬件,系统,或其他应用等)
 - 5.消息循环从消息队列中取出消息送给窗口过程处理.
 - 6.窗口过程针对不同的消息采取不同的处理方式.
- Xwindow/Motif
- Windows SDK.
- Visual Basic
- MFC
- VCL Delphi
- RAD 快速开发环境。
- C# GUI 脱胎于Delphi。 Anders Hejlsberg。

2 . RAD

- 使用Visual 开发一个C# Windows GUI。使用向导生成代码框架。
- 面向对象的封装。
- 使用折叠编辑器查看。

```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;
namespace WindowsApplication3
{public class Form1 :
    System.Windows.Forms.Form.....
} //只定义了一个类。
```

3 消息循环

```
public class Form1 : System.Windows.Forms.Form
{private System.ComponentModel.Container
    components = null;
```

```
public Form1(){ InitializeComponent();} //构造函数
```

```
protected override void Dispose( bool disposing )
```

```
    //清理函数
```

```
private void InitializeComponent()
```

```
static void Main() { Application.Run(new Form1()); }
```

//Application类提供静态方法和属性以管理应用程序，例如启动和停止应用程序、处理 Windows 消息的方法和获取应用程序信息的属性。不能继承此类不能实例化

//静态的主函数: 创建出窗体实例，在当前线程上开始运行标准应用程序消息循环。

```
while (GetMessage(&msg, NULL, 0, 0))
```

```
{ TranslateMessage(&msg);
```

```
    DispatchMessage(&msg); } }
```

4. 添加组件

- 从工具箱中拖放一个按钮到窗体上，并双击该按钮。
IDE为我们添加了若干代码：

```
public class Form1 : System.Windows.Forms.Form
{private System.Windows.Forms.Button button1;
//窗体类添加了一个button1作为其成员。
public Form1(){ InitializeComponent();}//构造函数
protected override void Dispose( bool disposing )//清理
    函数
private void InitializeComponent()
static void Main()
}
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/018012062003006121>