ウィンドウフォーム

GUI アプリケーション

多くの場合、.NET の基礎を System.Console 名前空間を用いて学習したでしょう
特に、.NET 独自の言語である C# を学習する時には、これを用いたはずです

しかし、.NET を実践で用いるには、当然 GUI アプリケーションの建築が必要となります
Microsoft .NET はこの要求にも高度に応えてくれるでしょう

Microsoft .NET の GUI アプリケーションは
System.Windows.Forms 名前空間を用います
この名前空間は、Windows の基本的なコントロールやウィンドウ、イベントなど
GUI アプリケーションに必要不可欠な機能をカプセル化しています

筆者の講座は、その全てが総合開発環境を用いることなく
本質を理解するために、コマンドラインから Framework SDK を用います
Microsoft が提供するグラフィカルな Visual シリーズは用いないので
テキストによるコードで、細部までプログラムできるように講座を進めていきます

C 言語による Win32 API を直接呼び出すプログラムは大変難しいものです
強力なプログラムを作れる反面、理解が難しく開発にはそれなりの技術力が必要なため
利用者も少数派で書籍も少ないというデメリットが存在しました

.NET における GUI プログラミングは、オブジェクト指向の恩恵を受けることになります
MFC もオブジェクト指向ではありましたが、C++ 言語は完全なオブジェクト指向ではありません
しかし、C# 言語と .NET クラスライブラリは統一された完成度の高いオブジェクト指向であり
Java の AWT や Swing 等と比較しても、統一された高い完成度を感じさせます

GUI プログラムの作成には **System.Windows.Forms.Form** クラスを継承します
このクラスには、ウィンドウの基本的な機能と処理がパッケージ化されています
私たちは、必要に応じてこれをオーバーライドして機能を書きかえることができるのです

```
Object
   MarshalByRefObject
      Component
         Control
            ScrollableControl
               ContainerControl
                  Form

public class Form : ContainerControl
```

見てのように、ずいぶんと深い階層に存在するクラスなのですが
これは、ウィンドウも基本的なコンポーネントから構成され、その上に成り立っているからです

ウィンドウを生成する場合、Main() メソッドの処理が問題となります
Main() メソッドがすぐに終了してしまえば、プログラムも勝手に終了してしまいます
GUI プログラムの場合、Main() メソッドは**メッセージループ**の処理を行います

メッセージループとは、マウスやキーボードなどのイベントを監視するループ処理を指します
これを行うには **System.Windows.Forms.Application** クラスを用います
このクラスは Windows アプリケーションの基本動作を制御する役割を持っています

public sealed class Application

このクラスは、メッセージループを起動する **Application.Run()** 静的メソッドを持ちます
このメソッドを用いれば、フォームを表示して基本的なイベントに対応させることができます

public static void Run();
public static void Run(ApplicationContext context);
public static void Run(Form mainForm);

パラメータを指定しない場合、フォームの無い状態でメッセージループを起動させます
フォームが無い場合は、Form クラスのようなものがアプリケーションを終了してくれないので
どこかでアプリケーションの終了を明示しなければ、無限ループに陥ってしまいます

context は ApplicationContext でメッセージループを走らせます
このクラスは、アプリケーション・スレッドに関連した情報を表すものです
この場では説明できないため、後ほど機会があれば詳しく解説します

mainForm には、メッセージループを用いる Form オブジェクトを指定します
この場合、このフォームがメインウィンドウとなります
メッセージループが作られると同時に、ウィンドウが自動的に表示されます

Form オブジェクトを Run() に渡せば、メインウィンドウが表示され
このウィンドウは、最小化、最大化、アプリケーションの終了などの基本機能を持ちます
たったこれだけの作業で Windows プログラムができてしまうため
Win32 API を直接制御する方法から比べると、極めて簡単で美しい設計だと感じるでしょう

```
using System.Windows.Forms;

class WinMain : Form {
 public static void Main(string[] args) {
```

```
    Application.Run(new WinMain());
  }
}
```

どうでしょう、わずか7行足らずで基本的なウィンドウプログラムが完成しました
ウィンドウの破棄や終了処理は、Formクラスが行ってくれています

私たちは、このような基本機能が完成されているオブジェクトを最初に定義し
これに、必要な機能を付け足したり、すでにある機能を書き換えたりすることができるのです
まさに、オブジェクト指向の賜物といえるでしょう

因みに、コマンドプロンプトから起動すると問題ないように思えますが
エクスプローラーから起動した場合、プロンプトが表示されるという問題点があります
これを解決するには、コンパイラオプションで **/t:winexe** を指定します
こうすれば、ウィンドウだけが美しく表示されることでしょう

戻る　　次のページへ