

第50章 ちゃんとしたイベント



「ちゃんとしたイベント」とは変な表題です。じゃあ、今までのイベントは ちゃんとしていなかったの？

「ちゃんと」を国語辞典で調べると「完全できちんとしているさま。」とあります。

実は、実際のプログラミング(特にFormプログラミング)で、今までのようにイベントに関するものを何から何まで、自分で用意することはほとんどありません。クラスライブラリの既存のイベントを利用することになります。

C#2.0では、Consoleクラスには、ただ一つ既存のイベントが存在します。

これは、CancelKeyPressイベントです。

コンソール・アプリケーションでは、Ctrl+Cキーが押されるとプログラムが強制的に 終了します。Ctrl+Cキーが押されたときに発生するイベントがCancelKeyPressです。

```
public static event ConsoleCancelEventHandler CancelKeyPress
```

ConsoleCancelEventHandlerは、このイベントに対応したデリゲートです。

```
public delegate void ConsoleCancelEventHandler (
    Object sender,
    ConsoleCancelEventArgs e
)
```

senderは、イベントを送信してきたオブジェクトです。

eは、イベントデータを格納しているSystem.ConsoleCancelEventArgsオブジェクトです。

```
Console.CancelKeyPress += new ConsoleCancelEventHandler(メソッド名);
```

とすると、CancelKeyPressイベントが発生すると指定のメソッドが実行されます。デリゲートからわかるようにこのメソッドには、ObjectとConsoleCancelEventArgs型の 引数を持たなくてはなりません。

通常xxxxArgs型の引数を利用して、イベントが発生したときの特殊な処理ができます。

ConsoleCancelEventArgsクラスには、次の2つのプロパティが定義されています。

```
public bool Cancel { get; set; }
public ConsoleSpecialKey SpecialKey { get; }
```

Cancelは、Ctrl+Cキーが押されたときに、現在のプロセスを終了するかどうかを表す値を取得または、設定します。trueならキャンセルし、falseなら終了します。CancelKeyPress イベントが来たとき、このプロパティを利用してそのままプロセスを終了してしまうのか、どうかをコントロールできますね。

ConsoleSpecialKeyは、現在のプロセスを中断した修飾子キーとコンソールキーの 組み合わせを取得します。

では、簡単なサンプルを見てみましょう。

```
// event04.cs
```

```
using System;
```

```
class event04
```

```
{
    public static void Main()
    {
        DateTime dt;
        int oldSecond = 0;
        Console.CancelKeyPress += new ConsoleCancelEventHandler(MyEventHandler);

        Console.CursorVisible = false;

        while (true)
        {
            dt = DateTime.Now;
            if (dt.Second != oldSecond)
            {
                Console.SetCursorPosition(0, 0);
                Console.WriteLine("{0}年{1:00}月{2:00}日{3:00}時{4:00}分{5:00}秒",
                    dt.Year, dt.Month, dt.Day, dt.Hour, dt.Minute, dt.Second);
                oldSecond = dt.Second;
            }
        }
    }
    protected static void MyEventHandler(object sender, ConsoleCancelEventArgs args)
    {
        Console.SetCursorPosition(0, 2);
    }
}
```

```

        Console.WriteLine("CancelKeyPressイベントが発生しました");
        Console.Write("Ctrl + C を有効にしますか (Y/N) ");
        while (true)
        {
            if (Console.KeyAvailable)
            {
                ConsoleKeyInfo cki = Console.ReadKey(true);
                if (cki.KeyChar == 'Y')
                {
                    args.Cancel = false;
                    Console.Clear();
                    break;
                }
                else
                {
                    args.Cancel = true;
                    Console.Clear();
                    break;
                }
            }
        }
    }
}

```

まずは、Mainメソッドを見てみましょう。

いきなり

```
DateTime dt;
```

という得体の知れないものが出てきましたDateTimeはその名の示すとおり、日付や時刻を表す構造体です。構造体はまだ解説していませんが、取り扱いほとんどクラスと同じです。

Nowプロパティは、現在のコンピュータの日時を取得するプロパティです。

```
public static DateTime Now { get; }
```

読み取り専用で、staticなプロパティです。

現在の日時を表すDateTimeオブジェクトを取得します。

また、この構造体には年、月、日、時、分、秒、ミリ秒、曜日をDateTimeオブジェクトから取得するプロパティがあります。

```

public int Year { get; }
public int Month { get; }
public int Day { get; }
public int Hour { get; }
public int Minute { get; }
public int Second { get; }
public int Millisecond { get; }
public DayOfWeek DayOfWeek { get; }

```

DayOfWeekは、列挙体で日曜日が0で土曜日が6を表します。

```
Console.CancelKeyPress += new ConsoleCancelEventHandler(MyEventHandler);
```

これで、CancelKeyPressイベントが発生したらMyEventHandlerメソッドが呼ばれることになります。

while文で永久ループに突入しています。

```
dt = DateTime.Now;
```

で、現在日時をdtに取得しています。

次がミソです。dtのSecondプロパティとoldSecondを比較して、異なれば(0,0)位置に現在日時を表示し、oldSecondに現在の秒を代入しています。

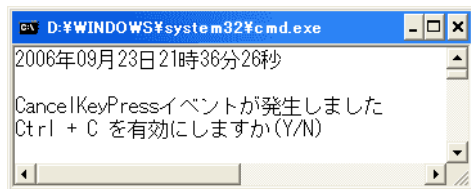
こうすることにより、1秒に1回のみ現在日時を表示することになります。

さて、MyEventHandlerメソッドではCancelKeyPressイベントが発生したことを表示して、Ctrl + C を有効にするかどうか尋ねています。

有効にするなら、args.Cancelをfalseにして無限ループを抜けます。

有効にしないならtrueにして無限ループを抜けます。

では、実行結果を見てみましょう。



Ctrl + Cが押されて、イベントが発生した旨を表示している間も 刻々と時刻が刻まれています。

CapLocksキーが有効になっていないときは、Shiftキーを押しながらYキーを押すとプログラムが終了します。

[\[C# Index\]](#) [\[総合Index\]](#) [\[Previous Chapter\]](#) [\[Next Chapter\]](#)

Update 25/Sep/2006 By Y.Kumei

当ホーム・ページの一部または全部を無断で複写、複製、転載あるいはコンピュータ等のファイルに保存することを禁じます。