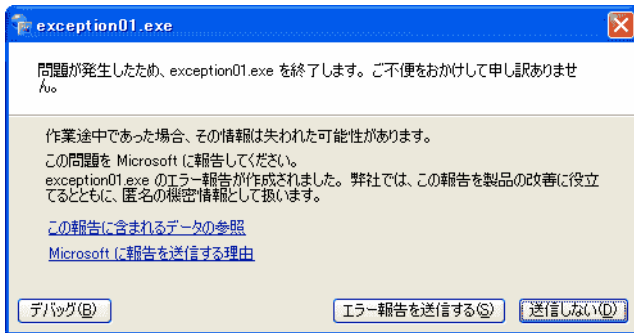


第65章 最も簡単な例外処理



C#のプログラムを作っていると、無事にビルドできたのに、いざ実行すると例外が発生したと言われて、強制的に終了させられることがあります。

プログラム実行中に、しばらく空白の時間があって



これは、いやですね。何としても避けたいものです。

自分の作ったプログラムの欠陥がいちいちマイクロソフトに報告されるのか!? と、思ってしまいますね。

例外とは、一種のエラーです。ある状況でまずいことになるとき、例外が発生します。たとえばint型の変数x,yを用意してx/yの演算をするプログラムを作ったとします。

多くの場合、何ともないのですがyの値が0になってしまうと、まずい状況になりますね。この時、例外が発生します。

例外が、発生したときシステムに任せるのではなく、プログラム内で何とか対処したいものです。

これには、try-catch構文を使います。

最も、単純なtry-catch構文は次のようになります。

```
try
{
    //例外が発生するかも知れない処理
}
catch
{
    //例外が発生したときの処理
}
```

例外が発生するとcatchブロック内の処理が実行され、それが終了すると何事もなかったかのように、次の処理に移ることになります。

```
// exception01.cs

using System;

class exception01
{
    public static void Main()
    {
        int x, y, z;

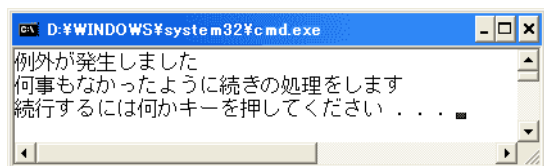
        try
        {
            x = 100;
            y = 0;
            z = x / y;
        }
        catch
        {
            Console.WriteLine("例外が発生しました");
        }

        Console.WriteLine("何事もなかったように続きの処理をします");
    }
}
```

tryブロックでyに0が代入され、x/yのところで例外が発生します。

すると、catchブロックの処理(「例外が発生しました」と表示)が行われ、これが、終了するとブロックを抜けて、次の処理に取りかかれます。

実行結果は次のようになります。



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

Update 10/Oct/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。