

# けん太の プログラミング 修行記



第12回

「けん太、例外処理問題进行处理する」の巻 - その1 -

構造化例外処理について考える

碗仔 けん太 (Pochi Company)  
WANKO, Kenta

Visual Basic.NET (以下VB.NET) やVisual C#.NET (以下C#)では“構造化例外処理”がサポートされています。例外処理は最近の本格的なプログラミング言語には不可欠な“エラーリカバリ”の機能であるといわれています。しかし、例外処理になじみのない初心者の中にはプログラムの実行中に「ハンドルされていない例外が発生しました」というようなメッセージが出ると、どうしたらよいかわからない人が

いるようです。

また、プログラミングの経験がかなりあっても、事前に問題を検出して修正するようなエラー対処のためのコードをプログラムの中を書くことと例外処理との違いや、Visual Basic 6.0 (以下VB6.0)のときのエラー処理とVB.NETやC#の例外処理がどのように違うのか、よくわからない人もいます。これまで例外処理に真正面から取り組んだことはありません。そこで

今回と次回は、.NETプログラムの例外処理について考えてみます。



## 開発現場の驚き

プログラムを開発していると、途方もないと思われるユーザーの要求に遭遇することがよくあります。そして、その原因の多くは、プログラマがプログラミングのときにはまったく予期していなかった使い方を、ユーザーはまるで当然であるかのようにすることがある、という点にあるように思えます。

たとえば、フロッピーディスクやMOのような取り外し可能な記憶媒体からデータを読み込んで編集しているときに、その記憶媒体をユーザーが装置から取り出してしまうことがあります。通常、そのようなことを行なうとデータが消えてなくなってしまいますが、たとえそんな場合でも、編集中的数据が失われてしまったのは、自分の責任ではなくプログラムが不親切だからだ、あるいはプログラムのバグ

### 本稿で前提となるもの

OS Windows 2000 Professional (SP2)

開発環境 Visual Studio.NET  
Microsoft Development Environment 2002 Version 7.0.9486  
.NET Framework 1.0 Version 1.0.3705 (SP2)

初級

中級

上級

この記事で解説したサンプルプログラムは、付録CD-ROMの¥DMAG¥KENTAフォルダ以下に収録しています。

¥HEXDECCS : 数値アップダウン (NumericUpDown) コントロールを使った10進数16進数変換プログラム (C#版)

¥HEXDECVB : 数値アップダウン (NumericUpDown) コントロールを使った10進数16進数変換プログラム (VB.NET版)

¥DEC2HEX : 例外処理を使った全角対応の10進数16進数変換プログラム (VB.NET版)

であると考えられるユーザーは少なくありません。

また、数値や英数文字を入力する場所に、全角文字をユーザーが入力することもよくあります。プログラマは、数値や英数文字は半角で入力されるのが暗黙の了解であると想定することがありますが、ユーザーの中には全角文字と半角文字の区別さえしない、あるいは、全角文字と半角文字の違いを知らずに「パソコン」(すなわちソフトウェア)を使っている人も数多くいます。

そのようなケースに対して、「使用中の記憶媒体を装置から取り出すなんてもってのほか」とか「数値やアルファベットは半角で入力するのが当然じゃないか」という態度をとり、プログラマにとって都合の悪いことはすべて「仕様」にしてしまうのもひとつの考え方でしょう。実際問題として、現実的な開発期間の中でユーザーのさまざまな要求にすべて答えることは無理です。また、一部のユーザーだけに役立つ、多くのユーザーにとっては無意味な機能や一部のユーザーのために過剰なエラーリカバリのコードを組み込むために多大な時間とコストを割くことは、ユーザー全体の利益には必ずしもならないでしょう。

しかし、それでも「仕様」で片付ける、というか、逃げているワケにはほかない場合があります。そのひとつの例を考えてみましょう。

「英数字やカタカナには“全角文字”と“半角文字”がある」ということがあります。

Unicodeには全角の英数字も定義されているのですから、もっぱら英語だけを使うシステムであっても全角の英数字を使うことはできます(図1)。つまり、全角の英数字の使用に関しては、日本語環境のシステムであるか、そうでないかという点でみると技術的には変わりはなく、他の言語環境のシステムでも全角半角問題は起こり得るのです。しかし、実際問題としては、日本語入力システム(以下IME)を常時使う日本語環境のシステムでは、全角文字と半角文字が問題になることがよくあります。

たとえば、インターネットのホームページ(以下Webページ)のフォーム

で、住所や電話番号、Eメールアドレスを入力するフィールドがあるとします。その入力欄は、英数字を半角で入力するようになっている場合と、全角で入力するようになっている場合、そして、どちらでもよい場合があります。

Webページに何も指示がないので、自分で“正しい”と判断した方法(あるいは正しいと信じて疑わない方法)で各欄に情報を入力したら、「入力データが間違っています」というメッセージが表示されて、どこがどう間違っているのか容易にはわからない、というサイトがあるのは困ったものです。

いちばんガッカリするのは、大量の情報をを入力することを要求するWebページで、要求された情報をすべて入力して[送信]ボタンをクリックしたら、全角文字と半角文字の違いで情報の登

図1: 全角の英数文字の例



FF2A

文字コードに注目。通常の(半角の)「J」の文字コードは「004A」、全角「J」の文字コードは「FF2A」



## 全角半角問題

日本語環境のシステム特有の問題に、