

## Verificando o Tipo de Caractere

Mário Leite

...

Existem situações em que um cálculo, simples ou complexo, depende da corretude do dado entrado, e sua validação é fundamental para que o resultado seja válido. Em ambientes gráficos com linguagens orientadas a eventos, tais como Visual Basic e C# no ambiente do Visual Studio em aplicações com formulários. Nestes casos, esta verificação pode ser feita simplesmente explorando as propriedades e métodos das *caixas de texto*, sem maiores dificuldades. Por exemplo, um valor digitado numa *textbox* nomeada como **TextBox1** numa aplicação *Windows Forms* em C# pode ser verificado se seu conteúdo é numérico analisando o retorno do método **IsNumeric(TextBox1.Text)**: se for *True* é numérico, caso contrário, se for *False* não é numérico.

Entretanto, para sistemas cujas aplicações não sejam em ambiente gráfico pode ser necessário fazer uma verificação “manual” do valor entrado para fazer os cálculos. Nestes casos existe uma solução muito simples, baseada na tabela ASCII (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange). Sabendo que os códigos desta tabela definem todos os caracteres do teclado, podemos criar uma pequena função para isto: basta verificar se o código do caractere analisado está na faixa **48-57**. Fora desta faixa não é um dígito numérico. A função **VerifNumero(x)**, em pseudocódigo, retorna um valor lógico: **V** se **x** é um dígito (0 a 9) ou **F** se não é.

Por outro lado, também existem situações em que o valor digitado deva ser uma letra (a-z, A-Z); por exemplo, no caso do nome de uma pessoa supondo que cada caractere desse nome seja uma letra. Neste caso, podemos criar uma função semelhante à função VerifNumero(x); por exemplo, **VerifLetra(x)**, onde **x** é um caractere recebido pela função, também baseado na tabela ASCII. Neste caso, são duas faixas de códigos a pesquisar: **65-90** (letras maiúsculas) e **97-122** (letras minúsculas). São duas funções bem simples apresentadas aqui em pseudocódigo, porém, extremamente práticas e que podem ser implementadas em linguagens que não possuem recursos diretos para fazer este tipo de checagem.

```
Função VerifNumero(x:caractere): lógico
//Função que verifica se um caractere lido é um número.
//-----
  Declare Resp: lógico
Início
  Se ( (Asc(x) >= 48) e (Asc(x) <= 57) )
    Resp ← V
  Senão
    Resp ← F
  FimSe
  Retorne Resp
FimFunção
```

```
Função VerifLetra(x:caractere): lógico
//Função que verifica se um caractere lido é uma letra.
//-----
  Declare Resp: lógico
Início
  Se(( (Asc(x) >= 65) e (Asc(x) <= 90) ) ou ( (Asc(x) >= 97) e (Asc(x) <= 122) ) ) Então
    Resp ← V
  Senão
    Resp ← F
  FimSe
  Retorne Resp
FimFunção
```