



## Visual Basic .NET

### Procedimentos, Escopo e Tratamento de Exceções

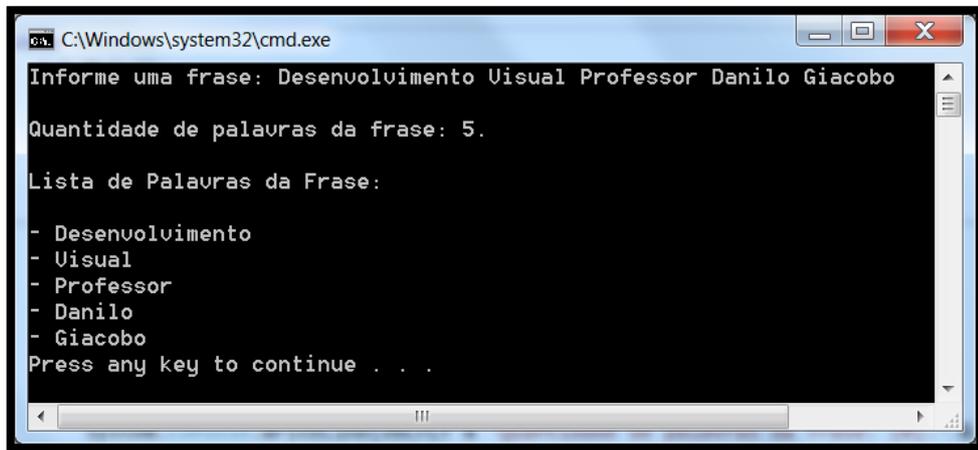
### Lista de Exercícios

A lista de questões a seguir aborda a criação de programas usando os conceitos de procedimentos (sub procedures e functions), escopo e tratamento de exceções (estruturadas e não estruturadas) da linguagem VB .NET. Para desenvolvimento eficaz da mesma é necessária a utilização da ferramenta Microsoft Visual Studio Express 2012 for Windows Desktop. Todos os exemplos devem ser realizados como projetos do tipo "Console". Para cada programa crie uma solução em separada. Em cada exemplo uma tela de como o aplicativo deve funcionar é fornecida, mas nada impede que você crie a sua própria interface.

1. Escreva um programa usando sub procedures que apresente a tabela ASCII completa (códigos de 0 a 255). Dica: Use a função Chr. Exemplo (apenas parte da mesma foi exibida):

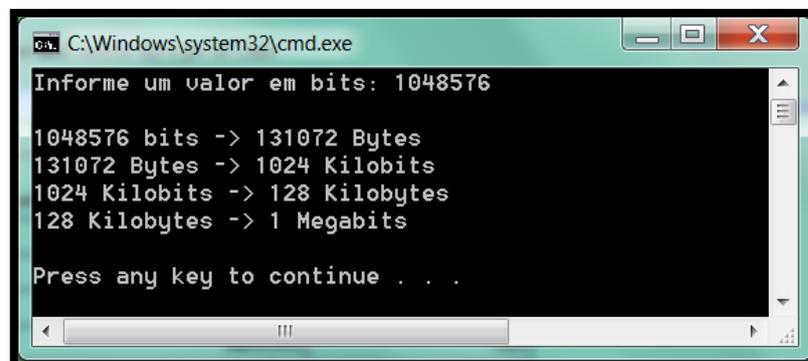
```
C:\Windows\system32\cmd.exe
41 = )
42 = *
43 = +
44 = ,
45 = -
46 = .
47 = /
48 = 0
49 = 1
50 = 2
51 = 3
52 = 4
53 = 5
54 = 6
55 = 7
56 = 8
57 = 9
58 = :
59 = ;
60 = <
61 = =
62 = >
63 = ?
64 = @
65 = A
```

2. Escreva um programa usando sub procedures que receba como entrada do usuário uma frase qualquer e exiba as palavras da mesma e a quantidade destas. Para simplificar considere um espaço em branco como sendo o separador de palavras. Exemplo:



```
C:\Windows\system32\cmd.exe
Informe uma frase: Desenvolvimento Visual Professor Danilo Giacomo
Quantidade de palavras da frase: 5.
Lista de Palavras da Frase:
- Desenvolvimento
- Visual
- Professor
- Danilo
- Giacomo
Press any key to continue . . .
```

3. Escreva um programa que receba um número representando um valor em bits e mostre a conversão deste para bytes, kilobits, kilobytes e megabits. Crie uma função para cada conversão de dados. Exemplo:



```
C:\Windows\system32\cmd.exe
Informe um valor em bits: 1048576
1048576 bits -> 131072 Bytes
131072 Bytes -> 1024 Kilobits
1024 Kilobits -> 128 Kilobytes
128 Kilobytes -> 1 Megabits
Press any key to continue . . .
```

4. Escreva e execute o programa abaixo:

```
1  Module Module1
2      Public mintVariavel As Integer
3      Friend fintVariaval As Integer
4      Private pintVariavel As Integer
5
6      Sub Main()
7          Dim mintVariavel As Integer
8          Dim fintVariaval As Integer
9          Dim pintVariavel As Integer
10
11         mintVariavel = 2
12         fintVariaval = 3
13         pintVariavel = 4
14
15         System.Console.WriteLine(mintVariavel)
16         System.Console.WriteLine(fintVariaval)
17         System.Console.WriteLine(pintVariavel)
18
19         Escopo()
20     End Sub
21
22     Sub Escopo()
23         System.Console.WriteLine(mintVariavel)
24         System.Console.WriteLine(fintVariaval)
25         System.Console.WriteLine(pintVariavel)
26     End Sub
27 End Module
```

- Quais são as variáveis locais e globais deste programa?
- Eu consigo usar as palavras reservadas de escopo (Private, Friend e Public) dentro de sub procedures e funções?
- É uma boa prática usar variáveis de escopo de módulo e procedimento com o mesmo nome?
- Porque eu não consigo usar a declaração Protected dentro de um código do tipo Módulo?

5. O código abaixo quando executado gera um erro fatal. Reescreva o mesmo colocando o código de tratamento de exceções não estruturado.

```
1 Module Module1
2     Sub Main()
3         Dim arrVetor() As String = {"a", "b", "c"}
4         System.Console.WriteLine(arrVetor(4))
5     End Sub
6 End Module
```

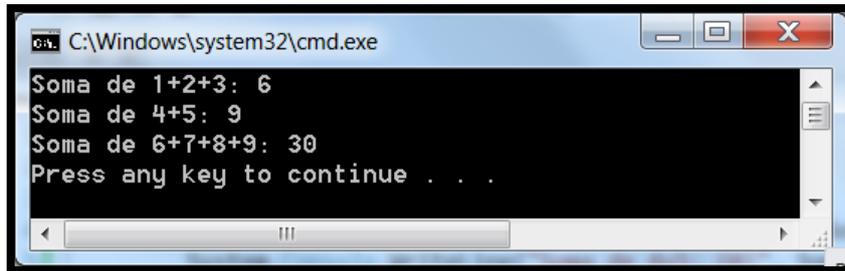
6. O código abaixo quando executado gera um erro fatal. Reescreva o mesmo colocando o código de tratamento de exceções estruturado.

```
1 Module Module1
2     Sub Main()
3         Dim arrVetor() As Integer = {1, 2, 3}
4         arrVetor(2) = "Danilo"
5     End Sub
6 End Module
```

7. Reescreva o código abaixo para ele conter o comentário padrão de um procedimento em linguagem VB .NET para a função QuantidadeCaracteres.

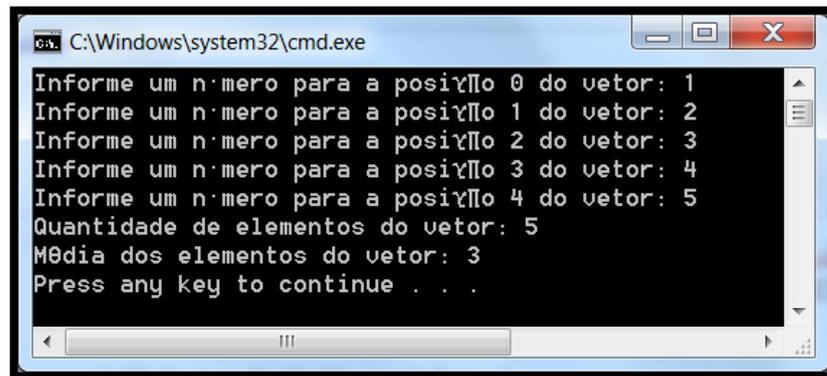
```
1 Module Module1
2     Sub Main()
3         System.Console.WriteLine(QuantidadeCaracteres("Danilo"))
4     End Sub
5
6     Function QuantidadeCaracteres(ByVal strPalavra As String) As Integer
7         Return strPalavra.Length()
8     End Function
9 End Module
```

8. Crie um programa que contenha uma função que receba argumentos variáveis para realizar a soma de um conjunto de números fornecidos por você. Exemplo:



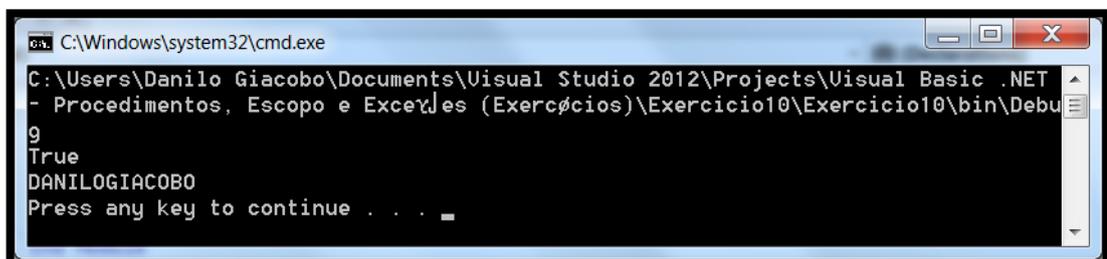
```
C:\Windows\system32\cmd.exe
Soma de 1+2+3: 6
Soma de 4+5: 9
Soma de 6+7+8+9: 30
Press any key to continue . . .
```

9. Crie um programa que contenha um procedimento com dois parâmetros opcionais. O primeiro é um número inteiro que recebe 1 ou 2 e o segundo é um array com 5 valores do tipo inteiro. Se a pessoa informar a opção 2 o programa deve exibir a média dos elementos além da contagem dos mesmos. Exemplo:



```
C:\Windows\system32\cmd.exe
Informe um número para a posição 0 do vetor: 1
Informe um número para a posição 1 do vetor: 2
Informe um número para a posição 2 do vetor: 3
Informe um número para a posição 3 do vetor: 4
Informe um número para a posição 4 do vetor: 5
Quantidade de elementos do vetor: 5
Média dos elementos do vetor: 3
Press any key to continue . . .
```

10. Crie um programa com dois módulos. Um deles com 3 propriedades apenas em modo leitura que retornam o caminho completo do programa no computador, se o computador é 64 bits e o nome do computador. O outro módulo deve exibir as informações na tela. Dica: Use a classe Environment do namespace System. Exemplo:



```
C:\Windows\system32\cmd.exe
C:\Users\Danilo Giacobbo\Documents\Visual Studio 2012\Projects\Visual Basic .NET
- Procedimentos, Escopo e Exceções (Exercícios)\Exercicio10\Exercicio10\bin\Debug
9
True
DANILOGIACOBO
Press any key to continue . . .
```