

```

/*****
*   $MCI Módulo de implementação: Módulo de teste específico
*
*   Arquivo gerado:           TESTMAT.C
*   Letras identificadoras:   TNAT
*
*   Nome da base de software:   Exemplo de teste automatizado
*
*   Projeto: Trabalho 2 - Programação Modular
*   Autores: GB - Gustavo Bach
*           JG - João Lucas Gardenberg
*           MV - Michelle Valente
*
*   $HA Histórico de evolução:
*       Versão  Autor      Data          Observações
*       1.00    GB,JG,MV   11/abr/2014   Início do desenvolvimento
*
*   $ED Descrição do módulo
*       Este módulo contém as funções específicas para o teste do
*       módulo matriz. Ilustra como redigir um interpretador de comandos
*       de teste específicos utilizando o arcabouço de teste para C.
*
*   $EIU Interface com o usuário pessoa
*       Comandos de teste específicos para testar o módulo árvore:
*
*       "=criar" <Int> <Int> <Int>
*           - chama a função MAT_CriarMatriz( ).
*           Obs. notação: O 1o <Int> é a altura da matriz.
*           Obs. notação: O 2o <Int> é a largura da matriz.
*           Obs. notação: O 3o <Int> é a condição de retorno
esperada.
*       "=destruir" <Int>
*           - chama a função MAT_DestruirMatriz( ).
*           Obs. notação: <Int> é a condição de retorno esperada.
*       "=inserir" <Int> <Int> <Int> <Int>
*           - chama a função MAT_InserirValor( ).
*           Obs. notação: O 1o <Int> é o valor a ser inserido.
*           Obs. notação: O 2o <Int> é a linha onde o valor será
inserido.
*           Obs. notação: O 3o <Int> é a coluna onde o valor será
inserido.
*           Obs. notação: O 4o <Int> é a condição de retorno
esperada.
*       "=obter" <Int> <Int> <Int> <Int>
*           - chama a função MAT_ObterValor( ).
*           Obs. notação: O 1o <Int> é o valor esperado que será
obtido.
*           Obs. notação: O 2o <Int> é a linha onde o valor está.
*           Obs. notação: O 3o <Int> é a coluna onde o valor está.

```

```

*                               Obs. notação: O 4o <Int> é a condição de retorno
esperada.
*
*****/

#include    <string.h>
#include    <stdio.h>

#include    "TST_ESPC.H"

#include    "generico.h"
#include    "lerparm.h"

#include    "MATRIZ.H"

/* Tabela dos nomes dos comandos de teste específicos */

#define     CRIAR_MAT_CMD       "=criar"
#define     DESTRUIR_MAT_CMD    "=destruir"
#define     INSERIR_MAT_CMD     "=inserir"
#define     OBTERVAL_MAT_CMD    "=obter"

/*****  Declarações de variáveis  *****/

    ptMatriz Matriz ;

/*****  Código das funções exportadas pelo módulo  *****/

/*****
*
*  $FC Função: TMAT  Efetuar operações de teste específicas para matriz
*
*  $ED Descrição da função
*      Efetua os diversos comandos de teste específicos para o módulo
*      matriz sendo testado.
*
*  $EP Parâmetros
*      $P ComandoTeste - String contendo o comando
*
*  $FV Valor retornado
*      Ver TST_tpCondRet definido em TST_ESPC.H
*
*****/

TST_tpCondRet TST_EfetuarComando( char * ComandoTeste )
{
    MAT_tpCondRet CondRetObtido    = MAT_CondRetOK ;

```

```

MAT_tpCondRet CondRetEsperada = MAT_CondRetFaltouMemoria ;

char ValorEsperado = '?' ;
char ValorObtido   = '!' ;
char ValorDado     = '\\0' ;

int  NumLidos = -1 ;

TST_tpCondRet Ret;

/* Testar Criar Matriz */

    if ( strcmp( ComandoTeste , CRIAR_MAT_CMD ) == 0 )
    {
        int Altura = -1, Largura = -1 ;
        NumLidos = LER_LerParametros( "iii" ,
                                     &Altura, &Largura, &CondRetEsperada
) ;

        if ( NumLidos != 3 )
        {
            return TST_CondRetParm ;
        } /* if */

        CondRetObtido = MAT_CriarMatriz( &Matriz, Altura,
Largura ) ;

        return TST_CompararInt( CondRetEsperada ,
CondRetObtido ,
                                     "Retorno errado ao criar
matriz." ) ;
    } /* Fim Ativa: Testar CriarMatriz */

/* Testar Destruir Matriz */

    else if( strcmp( ComandoTeste, DESTRUIR_MAT_CMD ) == 0 )
    {
        NumLidos = LER_LerParametros( "i", &CondRetEsperada );

        CondRetObtido = MAT_DestruirMatriz(&Matriz);

        return TST_CompararInt( CondRetEsperada , CondRetObtido ,
                                     "Retorno errado ao destruir
matriz." ) ;
    } /* Fim Ativa: Testar Destruir Matriz */

/* Testar Inserir Valor no Elemento */

```

```

else if( strcmp( ComandoTeste, INSERIR_MAT_CMD ) == 0 )
{
    int Linha = -1, Coluna = -1, Valor = -1 ;

    NumLidos = LER_LerParametros( "iiii", &Valor,
                                   &Linha, &Coluna,
&CondRetEsperada ) ;

    CondRetObtido = MAT_InserirValor( &Matriz, (int *) Valor,
Linha, Coluna ) ;

    return TST_CompararInt( CondRetEsperada , CondRetObtido ,
                           "Retorno errado ao inserir
elemento em matriz." ) ;
} /* Fim Ativa: Testar Inserir Valor */

/* Testar Obter Valor do Elemento */

else if( strcmp( ComandoTeste, OBTERVAL_MAT_CMD ) == 0 )
{

    int Linha = -1, Coluna = -1, ValorObtido = -1,
ValorEsperado = -1 ;
    void * ValObtido ;

    NumLidos = LER_LerParametros( "iiii", &ValorEsperado,
&Linha, &Coluna, &CondRetEsperada ) ;

    CondRetObtido = MAT_ObterValor( &Matriz, Linha, Coluna,
&ValObtido ) ;

    ValorObtido = (int) ValObtido ;

    Ret = TST_CompararInt( CondRetEsperada, CondRetObtido,
                           "Retorno errado ao obter valor." ) ;

    if ( Ret != TST_CondRetOK )
    {
        return Ret;
    } /* if */

    return TST_CompararInt( ValorEsperado, ValorObtido,
                           "Valor errado do elemento." ) ;

} /* Fim Ativa: Testar Obter Elemento */

return TST_CondRetNaoConhec ;

```

```
    } /* Fim função: TMAT  Efetuar operações de teste específicas para  
matriz */
```

```
/***** Fim do módulo de implementação: Módulo de teste específico  
*****/
```