

```

/*****
*  $MCI Módulo de implementação: Módulo de teste específico
*
*  Arquivo gerado:          TESTVAL.C
*  Letras identificadoras:   TVAL
*
*  Nome da base de software: Exemplo de teste automatizado
*
*  Projeto: Trabalho 2 - Programação Modular
*  Autores: Gustavo Bach, João Lucas Gardenberg e Michelle Valente
*
*  $HA Histórico de evolução:
*       1.00   GB,JG,MV   03/04/2014 Início do desenvolvimento
*
*  $ED Descrição do módulo
*       Este módulo contém as funções específicas para o teste do
*       módulo valor.
*
*  $EIU Interface com o usuário pessoa
*       Comandos de teste específicos para testar o módulo valor:
*
*       "=criar" <Int> <Int>
*           - chama a função VAL_CriarValor( ).
*           Obs. notação: O 1o <Int> é o número de células.
*           Obs. notação: O 2o <Int> é a condição de retorno
esperada.
*       "=altestado" <Int> <Int>
*           - chama a função VAL_AlterarEstado( ).
*           Obs. notação: O 1o <Int> é o estado a ser alterado
para.
*           Obs. notação: O 2o <Int> é a condição de retorno
esperada.
*       "=altnumcel" <Int> <Int>
*           - chama a função VAL_AlterarNumCel( ).
*           Obs. notação: O 1o <Int> é o número de células a ser
alterado para.
*           Obs. notação: O 2o <Int> é a condição de retorno
esperada.
*       "=obterestado" <int> <Int>
*           - chama a função VAL_ObterEstado( ) e compara
*           o valor retornado com o valor <int>.
*           Obs. notação: O 1o <Int> é o estado esperado.
*           Obs. notação: O 2o <Int> é a condição de retorno
esperada.
*       "=obternumcel" <Int> <Int>
*           - chama a função VAL_ObterNumCel( ) e compara
*           o valor retornado com o valor <int>.
*           Obs. notação: O 1o <Int> é o número de células
esperado.

```

```

*                               Obs. notação: O 2o <Int> é a condição de retorno
esperada.
*       "=destruir" <Int>
*                               - chama a função VAL_DestruirValor( ).
*                               Obs. notação: O <Int> é a condição de retorno
esperada.
*
*****/

#include    <string.h>
#include    <stdio.h>
#include    "TST_ESPC.H"

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

#include    "valor.h"

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

#define     CRIAR_VAL_CMD        "=criar"
#define     ALT_EST_CMD          "=altestado"
#define     ALT_NUMCEL_CMD       "=altnumcel"
#define     OBTER_EST_CMD        "=obterestado"
#define     OBTER_NUMCEL_CMD     "=obternumcel"
#define     DESTROI_VAL_CMD      "=destruir"

ptValor Valor ;

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

/*****
*
*   $FC Função: TVAL Efetuar operações de teste específicas para valor
*
*   $ED Descrição da função
*       Efetua os diversos comandos de teste específicos para o módulo
*       valor 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 )
{

    VAL_tpCondRet CondRetObtido    = VAL_CondRetOK ;
    VAL_tpCondRet CondRetEsperada = VAL_CondRetFaltouMemoria ;
                                           /* inicializa para qualquer coisa */

    int ValorEsperado = -1 ;
    int ValorObtido   = 0 ;
    int ValorDado      = -1 ;
        int NumCel = -1 ;
        int Estado = -1 ;

    int  NumLidos = -1 ;

    TST_tpCondRet Ret ;

    /* Testar VAL Criar valor */

    if ( strcmp( ComandoTeste , CRIAR_VAL_CMD ) == 0 )
    {

        NumLidos = LER_LerParametros( "ii" ,
                                       &NumCel, &CondRetEsperada ) ;

        if ( NumLidos != 2 )
        {
            return TST_CondRetParm ;
        } /* if */

        CondRetObtido = VAL_CriarValor( &Valor, NumCel ) ;

        return TST_CompararInt( CondRetEsperada , CondRetObtido ,
                                "Retorno errado ao criar árvore." ) ;

    } /* fim ativa: Testar ARV Criar árvore */

    /* Testar VAL Destruir Valor */

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

        NumLidos = LER_LerParametros( "i" ,
                                       &CondRetEsperada ) ;

        if ( NumLidos != 1 )
        {
            return TST_CondRetParm ;
        } /* if */
    }
}

```

```

        CondRetObtido = VAL_DestruirValor( &Valor ) ;

        return TST_CompararInt( CondRetEsperada , CondRetObtido ,
                                "Retorno errado inserir à direita." ) ;

    } /* fim ativa: Testar VAL Destruir Valor */

/* Testar VAL Alterar Estado */

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

    NumLidos = LER_LerParametros( "ii" ,
                                   &Estado, &CondRetEsperada ) ;

    if ( NumLidos != 2 )
    {
        return TST_CondRetParm ;
    } /* if */

    CondRetObtido = VAL_AlterarEstado( &Valor, Estado ) ;

    return TST_CompararInt( CondRetEsperada , CondRetObtido ,
                            "Retorno errado inserir à direita." ) ;

} /* fim ativa: Testar VAL Alterar Estado */

/* Testar VAL Alterar Número de Células */

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

    NumLidos = LER_LerParametros( "ii" ,
                                   &NumCel, &CondRetEsperada ) ;

    if ( NumLidos != 2 )
    {
        return TST_CondRetParm ;
    } /* if */

    CondRetObtido = VAL_AlterarNumCel( &Valor, NumCel ) ;

    return TST_CompararInt( CondRetEsperada , CondRetObtido ,
                            "Retorno errado inserir à direita." ) ;

} /* fim ativa: Testar VAL Alterar Número de Células */

```

```

/* Testar VAL Obter Estado */

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

    NumLidos = LER_LerParametros( "ii" ,
                                   &ValorEsperado , &CondRetEsperada ) ;

    if ( NumLidos != 2 )
    {
        return TST_CondRetParm ;
    } /* if */

    CondRetObtido = VAL_ObterEstado( &Valor , &ValorObtido ) ;

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

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

    return TST_CompararChar( ValorEsperado , ValorObtido ,
                             "Conteúdo do número de células está
errado." ) ;

} /* fim ativa: Testar VAL Obter Estado */

/* Testar VAL Obter Número de Células */

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

    NumLidos = LER_LerParametros( "ii" ,
                                   &ValorEsperado , &CondRetEsperada ) ;

    if ( NumLidos != 2 )
    {
        return TST_CondRetParm ;
    } /* if */

    CondRetObtido = VAL_ObterNumCel( &Valor , &ValorObtido ) ;

    Ret = TST_CompararInt( CondRetEsperada , CondRetObtido ,

```

```

                                "Retorno errado ao obter valor corrente."
) ;

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

    return TST_CompararChar( ValorEsperado , ValorObtido ,
                                "Conteúdo do número de células está
errado." ) ;

    } /* fim ativa: Testar VAL Obter Número de Células */

    return TST_CondRetNaoConhec ;

} /* Fim função: TVAL Efetuar operações de teste específicas para Valor */

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

```