

```

/*****
*  $MCI Módulo de implementação: LIS  Lista duplamente encadeada
*
*  Arquivo gerado:           DESENHO.c
*  Letras identificadoras:   DEF
*
*  Nome da base de software:  Arcabouço para a automação de testes de
programas redigidos em C
*
*  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  12/abr/2014  início desenvolvimento
*
*****/

```

```

#include <stdio.h>
#include <malloc.h>

```

```

#define DESENHO_OWN
#include "DESENHO.h"
#undef DESENHO_OWN
/*****
*  $MCI Módulo de implementação: LIS  Lista duplamente encadeada
*
*  Arquivo gerado:           DESENHO.c
*  Letras identificadoras:   DEF
*
*  Nome da base de software:  Arcabouço para a automação de testes de
programas redigidos em C
*
*  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  12/abr/2014  início desenvolvimento
*
*****/

```

```

#include <stdio.h>

```

```

#include <malloc.h>

#define DESENHO_OWN
#include "DESENHO.h"
#undef DESENHO_OWN

#include "CELULA.h"
#include "VALOR.h"
#include "LISTA.h"
#include "MATRIZ.h"

/*****
*
*   $TC Tipo de dados: DES Desenho
*
*****/
typedef struct tgDesenho {

    ptMatriz pMatriz ;
        /* Ponteiro para a matriz */

    ptMatriz pListaColuna ;
        /* Lista de listas para os valores das colunas da matriz */

    ptMatriz pListaLinha ;
        /* Lista de listas para os valores das linhas da matriz */

} tpDesenho ;

ptCelula pCelula = NULL ;
ptValor pValor = NULL ;
void * pVoid = NULL ;
extern int AlturaMatriz ;
extern int LarguraMatriz ;
extern int LinhaMatrizValores ;
extern int ColunaMatrizValores ;
VAL_tpCondRet CondRetVal ;
MAT_tpCondRet CondRetMat ;
CEL_tpCondRet CondRetCel ;

/*****
*
*   Função: DES  Gerar Matriz
*   *****/

    DES_tpCondRet DES_GerarMatriz( ptDesenho * pDesenho, int Altura, int
Largura )

```

```

{

int AlturaMatriz = Altura ;
int LarguraMatriz = Largura ;
int IteradorAltura, IteradorLargura ;
*pDesenho = ( tpDesenho * ) malloc ( sizeof( tpDesenho ) ) ;

CondRetCel = CEL_CriarCelula( &pCelula ) ;

if( CondRetCel == CEL_CondRetFaltouMemoria )
{
    return DES_CondRetFaltouMemoria ;
} /* if */

CondRetMat = MAT_CriarMatriz( &(*pDesenho)->pMatriz, AlturaMatriz,
LarguraMatriz) ;

if( CondRetMat == MAT_CondRetFaltouMemoria )
{
    return DES_CondRetFaltouMemoria ;
} /* if */

for (IteradorAltura = 0 ; IteradorAltura < Altura ; IteradorAltura++ )
{
    for( IteradorLargura = 0 ; IteradorLargura < Largura ;
IteradorLargura++ )
    {
        CondRetMat = MAT_InserirValor ( (*pDesenho)->pMatriz,
pCelula, IteradorAltura + 1, IteradorLargura + 1 ) ;

        if( CondRetMat == MAT_CondRetFimLinhas )
        {

            return DES_CondRetFimLinhas ;
        } /* if */

        if( CondRetMat == MAT_CondRetFimColunas )
        {

            return DES_CondRetFimColunas ;
        } /* if */
    } /* for */
} /* for */

return DES_CondRetOK ;

}

```

```

/*****
*
*   Função: DES  Alterar Estado Correto
*   *****/

DES_tpCondRet DES_AlterarEstadoCorreto( ptDesenho * pDesenho, int
Linha, int Coluna )
{

    if (pDesenho == NULL)
    {
        return DES_CondRetDesenhoNaoExiste ;
    }

    CondRetMat = MAT_ObterValor( (*pDesenho)->pMatriz, Linha, Coluna,
&pVoid ) ;

    pCelula = (ptCelula) pVoid ;

    CondRetCel = CEL_AlterarEstadoCorreto( pCelula ) ;

    return DES_CondRetOK ;

}

/*****
*
*   Função: DES  Alterar Estado Atual
*   *****/

DES_tpCondRet DES_AlterarEstadoAtual( ptDesenho * pDesenho, int Linha,
int Coluna )
{

    pCelula = ( ptCelula ) malloc ( sizeof( ptCelula ) ) ;

    if ( pDesenho == NULL )
    {
        return DES_CondRetDesenhoNaoExiste ;
    }

    CondRetMat = MAT_ObterValor( (*pDesenho)->pMatriz, Linha, Coluna,
&pVoid ) ;

    pCelula = ( ptCelula ) pVoid ;

    CondRetCel = CEL_AlterarEstadoAtual( pCelula );

```

```

        return DES_CondRetOK ;

    }

/*****
*
*   Função: DES   Gerar Listas
*   ****/

DES_tpCondRet DES_GerarListas( ptDesenho * pDesenho )
{

    int AlturaMatrizValores = (AlturaMatriz + 1) / 2 ;
    int LarguraMatrizValores = (LarguraMatriz + 1) / 2 ;
    int Somador = 0 ;
    int IteradorLinha, IteradorColuna = 0 ;
    int EstadoCorreto ;
    ptValor pValor ;

    printf("Altura %d", AlturaMatriz);
    printf("Largura %d", LarguraMatriz);

    CondRetMat = MAT_CriarMatriz( &(*pDesenho)->pListaColuna, AlturaMatriz,
LarguraMatriz) ;
    CondRetMat = MAT_CriarMatriz( &(*pDesenho)->pListaLinha, AlturaMatriz,
LarguraMatriz) ;

    for(IteradorColuna = 0 ; IteradorColuna < LarguraMatriz ;
IteradorColuna++ )
    {
        for( IteradorLinha = 0 ; IteradorLinha < AlturaMatriz ;
IteradorLinha++ )
        {
            CondRetMat = MAT_ObterValor( (*pDesenho)->pMatriz ,
IteradorLinha , IteradorColuna, &pVoid ) ;
            pCelula = ( ptCelula ) pVoid ;
            CondRetCel = CEL_ObterEstadoCorreto( pCelula,
&EstadoCorreto );

            if( EstadoCorreto == 1 )
            {
                Somador++ ;
            } else if( EstadoCorreto == 0)
            {
                if (Somador != 0)
                {
                    CondRetVal = VAL_CriarValor ( &pValor, Somador ) ;

```

```

        CondRetMat = MAT_InserirValor( ( ptMatriz )
(*pDesenho)->pListaColuna , pValor, LinhaMatrizValores + 1, IteradorColuna +
1 ) ;

        LinhaMatrizValores++ ;
        Somador = 0 ;
    } /* if */
    } /* if */

    }

    }

    for(IteradorLinha = 0 ; IteradorLinha < AlturaMatriz ; IteradorLinha++ )
    {
        for( IteradorColuna = 0 ; IteradorColuna < LarguraMatriz ;
IteradorColuna++ )
        {
            CondRetMat = MAT_ObterValor( (*pDesenho)->pMatriz ,
IteradorLinha , IteradorColuna, &pVoid ) ;
            pCelula = ( ptCelula ) pVoid ;
            CondRetCel = CEL_ObterEstadoCorreto( pCelula,
&EstadoCorreto );

            if( EstadoCorreto == 1 )
            {
                Somador++ ;
            } else if( EstadoCorreto == 0)
            {
                if (Somador != 0)
                {
                    CondRetVal = VAL_CriarValor ( &pValor, Somador ) ;
                    CondRetMat = MAT_InserirValor( ( ptMatriz )
(*pDesenho)->pListaLinha , pValor, ColunaMatrizValores , IteradorLinha ) ;
                    ColunaMatrizValores++ ;
                    Somador = 0 ;
                } /* if */
            } /* if */

        }

    }

    return DES_CondRetOK ;
}

/*****
*
* Função: DES  Mostrar Desenho
* *****/

```

```

DES_tpCondRet DES_MostrarDesenho( ptDesenho * pDesenho )
{
    int IteradorAltura, IteradorLargura ;
    int EstadoAtual = 0;
    int NumCel = 0 ;

    printf(" => ");

    for( IteradorAltura = 0 ; IteradorAltura < ColunaMatrizValores ;
IteradorAltura++ )
    {
        for( IteradorLargura = 0 ; IteradorLargura <
LinhaMatrizValores ; IteradorLargura++ )
        {
            printf( " " ) ;
        } /* for */

        printf( " " );

        for( IteradorLargura = 0 ; IteradorLargura < LarguraMatriz ;
IteradorLargura++ )
        {
            CondRetMat = MAT_ObterValor( (*pDesenho)->pListaColuna,
IteradorAltura, IteradorLargura, &pVoid ) ;
            pValor = ( ptValor ) pVoid ;
            CondRetVal = VAL_ObterNumCel( pValor, &NumCel ) ;
            if( NumCel )
            {
                printf( "%d", NumCel );
            } else {
                printf( " " );
            } /* if */
        } /* for */

        printf( "\n" );
    } /* for */

    for( IteradorLargura = 0 ; IteradorLargura < LarguraMatriz +
LinhaMatrizValores ; IteradorLargura++ )
    {
        printf( " " );
    } /*if */

    printf( "\n" );

    for( IteradorAltura = 0 ; IteradorAltura < AlturaMatriz ;
IteradorAltura++ )

```

```

{
    for( IteradorLargura = 0 ; IteradorLargura <
LinhaMatrizValores ; IteradorLargura++ )
    {
        CondRetMat = MAT_ObterValor( (*pDesenho)->pListaLinha,
IteradorAltura, IteradorLargura, &pVoid ) ;
        pValor = ( ptValor ) pVoid ;
        CondRetVal = VAL_ObterNumCel( pValor, &NumCel ) ;
        if( NumCel )
        {
            printf( "%d", NumCel );
        } else {
            printf( " " );
        } /* if */
    } /* for */

    for( IteradorLargura = 0 ; IteradorLargura < LarguraMatriz ;
IteradorLargura++ )
    {
        CondRetMat = MAT_ObterValor( (*pDesenho)->pMatriz,
IteradorAltura, IteradorLargura, &pVoid ) ;
        pCelula = ( ptCelula ) pVoid ;
        CondRetCel = CEL_ObterEstadoAtual( pCelula, &EstadoAtual )
        if( EstadoAtual == 1 )
        {
            printf("X");
        } else {
            printf(" ");
        } /* if */
    } /* for */

    printf( "\n" );
} /* for */

return DES_CondRetOK ;

} /* Fim função: DES  Mostrar Desenho */

```