

### **TRABALHO 3 MODULAR:**

Caio Graça Melo

Lucca Buffara

Mark Ribeiro

### **FUNÇÕES DE ACESSO:**

#### **BAR:**

```
BAR_tpCondRet BAR_CriarBAR( void ) ;
```

```
void BAR_DestruirBAR( void ) ;
```

```
BAR_tpCondRet BAR_Inserir( char c, int n ) ;
```

```
BAR_tpCondRet BAR_Excluir( char c ) ;
```

```
BAR_tpCondRet BAR_NPecas( char c, int *n ) ;
```

#### **DADO:**

```
DAD_tpCondRet DAD_JogaDados(int *d);
```

#### **DADO\_PONTOS:**

```
DP_tpCondRet DP_criaDadosPontos(void);
```

```
DP_tpCondRet DP_dobraValor(char jogador);
```

```
DP_tpCondRet DP_modificaPontos(int valor, char jogador);
```

```
DP_tpCondRet DP_lerPontos(char jogador, int *val);
```

```
DP_tpCondRet DP_lerValorDadosPontos(int *val);
```

```
DP_tpCondRet DP_lerUltimoJogador(char *cor);
```

```
DP_tpCondRet DP_destroiDadosPontos(void);
```

#### **PECA:**

```
PEC_tpCondRet PEC_CriaPeca( Peca *p, char cor );
```

```
PEC_tpCondRet PEC_ObterCorPeca ( Peca *p, char *cor );
```

```
PEC_tpCondRet PEC_DestroiPeca( Peca *p );
```

#### **PFN:**

```
PFN_tpCondRet PFN_CriarPFN( void ) ;
```

```
PFN_tpCondRet PFN_Inserir( char c, int n ) ;
```

```
PFN_tpCondRet PFN_NPecas( char c, int *n ) ;
```

```
void PFN_DestruirPFN( void ) ;
```

**TABULEIRO:**

```
TAB_tpCondRet TAB_CriarTabuleiro( void ) ;

TAB_tpCondRet TAB_ArrumarTabuleiro( void );

void TAB_DestruirTabuleiro( void ) ;

TAB_tpCondRet TAB_MoverPeca( int casaIni, int casaFim, char corJogador ) ;

TAB_tpCondRet TAB_InserirPecasCasa(int n, char cor, int nCasa ) ;

TAB_tpCondRet TAB_RemoverPeca( int nCasa ) ;

TAB_tpCondRet TAB_PrintTabuleiro( void ) ;

int TAB_ChecaCasa (int nCasa, char c);

int TAB_ChecaQuadrante (char c);
```

**LISTA:**

```
LIS_tpplista LIS_CriarLista(void( * ExcluirValor ) ( void * pDado ) ) ;

void LIS_DestruirLista( LIS_tpplista pLista ) ;

void LIS_EsvaziarLista( LIS_tpplista pLista )

LIS_tpCondRet LIS_InserirElementoAntes( LIS_tpplista pLista ,
                                       void * pValor) ;

LIS_tpCondRet LIS_InserirElementoApos( LIS_tpplista pLista ,
                                       void * pValor )

LIS_tpCondRet LIS_ExcluirElemento( LIS_tpplista pLista ) ;

void * LIS_ObterValor( LIS_tpplista pLista ) ;

LIS_tpCondRet pegaNumElementos( LIS_tpplista pLista ,
                               int * num );

void IrInicioLista( LIS_tpplista pLista ) ;

void IrFinalLista( LIS_tpplista pLista ) ;

LIS_tpCondRet LIS_AvancarElementoCorrente( LIS_tpplista pLista ,
                                           int numElem );

LIS_tpCondRet LIS_ProcurarValor( LIS_tpplista pLista ,
                                void * pValor ) ;
```