```
/*=+1================================================================*/
/*   MODULE                    ERROR.H                                */
/*====================================================================*/
/*   FUNCTION      Header file for module ERROR.C (and user modules).
 *
 *   SYSTEM        Standard (ANSI/ISO) C.
 *                 Tested on PC/MS DOS 5.0 (MSC 600A).
 *
 *   SEE ALSO      Modules : ERROR.C
 *
 *   PROGRAMMER    Allan Dystrup.
 *
 *   COPYRIGHT     (c) Allan Dystrup, 1991
 *
 *   VERSION       $Header: d:/cwk/kf/error/RCS/error.h 1.1 92/10/25 17:13:10
 *                 Allan_Dystrup Exp Locker: Allan_Dystrup $
 *                 ------------------------------------------------------
 *                 $Log:error.h $
 *                 Revision 1.1  92/10/25  17:13:10  Allan_Dystrup
 *                 Initial revision
 *
 *====================================================================*/

#ifndef _ERROR_H             /* Make sure this header is included only once */
#define _ERROR_H             /* Matching #endif is at end-of-file           */


/* Include header with general definitions */
#ifndef _GENERAL_H
#include "general.h"
#endif



/**********************************************************************/
/************************** DATA STRUCTURES ***************************/
/**********************************************************************/

/* Set up enumerated error-type, ie. unique ErrorNumbers coding for
 *    ErrorDescription, type :
 *       Inp : Input error in user or application program data
 *       Pgm : Program error in module data structure (size or format)
 *       Int : Internal error in "virtual machine" (stack, heap)
 *    ErrorLocation, module & function :
 *       MODULE[typeFunction]
 */
typedef enum errors { /* Unique E# : Error description : Error location  */
                      /* ------------------------------------------- */
    EARG000,         /* E[000]ARG : Inp.arg.err. : BOOL[pzParse]        */
    EARG001,         /* E[001]ARG : Inp.arg.err. : BOOL[fInterpret]     */
    EARG002,         /* E[002]ARG : Inp.arg.err. : BM[vBuildBM]         */
    EARG003,         /* E[003]ARG : Inp.arg.err. : BM[vRunBM]           */
    EARG004,         /* E[004]ARG : Inp.arg.err. : BM[vDelBM]           */
    ELEX000,         /* E[005]LEX : Inp.lex.err. : BOOL[bScan]          */
    ESYN000,         /* E[006]SYN : Inp.syn.err. : BOOL[pzParse]        */
    ETAB000,         /* E[007]TAB : Pgm.sym.err. : BOOL[vEmit]          */
    ETAB001,         /* E[008]SYM : Pgm.lex.err. : BOOL[iSymInsert]     */
    ETAB002,         /* E[009]SYM : Pgm.lex.err. : BOOL[iSymInsert]     */
    ETOK000,         /* E[010]TOK : Pgm.tok.err. : BOOL[vEmit]          */
    ETOK001,         /* E[011]TOK : Pgm.tok.err. : BOOL[fInterpret]     */
    EMEM000,         /* E[012]MEM : Int.all.err. : AC[vBuildKeyword]    */
    EMEM001,         /* E[013]MEM : Int.all.err. : AC[vBuildFailMove]   */
    EMEM002,         /* E[014]MEM : Int.all.err. : AC[psAllocState]     */
    EMEM003,         /* E[015]MEM : Int.all.err. : AC[psAllocTrans]     */
    EMEM004,         /* E[016]MEM : Int.all.err. : AC[vAllocQElem]      */
```

```
    EMEM005,          /* E[017]MEM : Int.all.err. : BM[vBuildBM]          */
    EMEM006,          /* E[018]MEM : Int.all.err. : BM[vBuildBM]          */
    EMEM007,          /* E[019]MEM : Int.all.err. : BM[iShCompMS]         */
    EMEM008           /* E[020]MEM : Int.all.err. : BM[iShCompOM]         */

}   ERRNUM;


/* Set max ErrorNumber E[xxx] in ERRNUM enumeration */
#define ERRMAX 20



/*****************************************************************************/
/************************** FUNCTION PROTOTYPES ******************************/
/*****************************************************************************/

/* Report error 'type' on stderr & return EXIT_FAILURE */
extern void
   vError(ERRNUM type, char *param);



#endif   /* #define _ERROR_H */

/* END module error.h                                                        */
/*=-1=========================================================================*/
```