

```

1 // 文件: console.c
2 // 内容: 人机界面操作实现版本1——控制台界面
3 //      本文也包括了解释器、测试语法分析的主函数 main()。
4 // 作者: WXQ#2018
5
6 /* 本文代码基于 C 的库函数 实现, 也定义了
7    "console Application" 型应用的主函数 main()
8    */
9
10 #include <stdio.h> // fprintf(), stdout
11
12
13 // #include "parser.h"
14 // 语法分析器接口
15 extern void Parser(const char * file_name); // in parser.c
16
17 #include "errlog.h"
18
19 // ----- 检查被分析的源程序文件是否合法函数
20 int CheckSrcFile(const char* file_name)
21 {
22     FILE * file = NULL;
23
24     if(file_name == NULL || file_name[0] == '\0')
25     {
26         fprintf(stderr, "未指定源程序文件 !");
27         return 0;
28     }
29     if((file = fopen(file_name, "r")) == NULL)
30     {
31         fprintf(stderr, "打开源程序文件[%s]失败!", file_name);
32         return 0;
33     }
34     else fclose(file);
35     return 1;
36 }
37
38 int main(int argc, char *argv[])
39 {
40     if (argc < 2)
41     {
42         printf("Give me a Source-File !\n" );
43         return 1;
44     }
45     if( CheckSrcFile( argv[1] ) == 0 )
46         return 1;
47
48     InitError(); // 初始化错误处理模块
49                 // 若不调用则信息将显示到屏幕上
50     printf( "\n本次执行产生两个文件: \n"
51            "%s 存储 语法分析器的输出文本\n"
52            "%s 存储 输出的错误信息\n\n",
53            LOGFILE_PATH, ERRFILE_PATH
54            );
55
56     Parser(argv[1]);
57     CloseError();
58
59     return 0;
60 }
61
62

```

```
63 // 下面是两个对外提供的公共操作实现, See ui.h
64 #include "ui.h"
65
66 // 显示信息
67 void ShowMessage(int flag, // 常规类信息=0, 错误类信息=1
68                 const char* msg)
69 {
70     if(flag == 0)
71         fprintf(stdout, msg);
72     else
73         fprintf(stderr, msg);
74 }
75
76 // 绘制一个点
77 void DrawPixel( unsigned long x, unsigned long y, // 点在窗口中的坐标
78               unsigned int color // 点的颜色, 0 为默认色, 保留参数
79               )
80 {
81     fprintf(stdout, ".....DrawPixel(%ld, %ld)\n", x, y);
82 }
83
```