文件处理

Ref Pseudocode

写入文件

文件 (file) 在程序结束后仍保存数据。0478 用四条命令, 每一条都点名它所操作的文件。

```
OPENFILE <file identifier> FOR WRITE
WRITEFILE <file identifier>, <data>
CLOSEFILE <file identifier>
```

```
OPENFILE "names.txt" FOR WRITE
WRITEFILE "names.txt", "Mei"
WRITEFILE "names.txt", "Sam"
CLOSEFILE "names.txt"
OUTPUT "Saved"
```

三种**文件模式** (file mode):

模式	作用
WRITE	新建文件 —— **原有内容全部丢失**
APPEND	追加到已有内容的末尾
READ	从头读取

```
OPENFILE "log.txt" FOR WRITE
WRITEFILE "log.txt", "first"
CLOSEFILE "log.txt"
OPENFILE "log.txt" FOR APPEND
WRITEFILE "log.txt", "second"
CLOSEFILE "log.txt"
OUTPUT "Both lines saved"
```

- 一条 WRITEFILE 写**一行**。
- CLOSEFILE 不是可选的。未关闭的文件可能丢失仍在等待写出的数据, 这是一个常考的答案。

常见错误

- 本想 FOR APPEND 却写了 FOR WRITE, 悄无声息地毁掉旧内容。
- 忘记 CLOSEFILE。
- 打开一个已经打开的文件。

读取文件到末尾

READFILE 把一行读入一个变量。该变量必须先声明。

```
OPENFILE <file identifier> FOR READ
READFILE <file identifier>, <variable>
CLOSEFILE <file identifier>
```

通常你并不知道有多少行, 所以要一直读到**文件末尾** —— 这正是 EOF 函数所报告的。

```
OPENFILE "names.txt" FOR WRITE
WRITEFILE "names.txt", "Mei"
WRITEFILE "names.txt", "Sam"
WRITEFILE "names.txt", "Ana"
CLOSEFILE "names.txt"

DECLARE Line : STRING
OPENFILE "names.txt" FOR READ
WHILE NOT EOF("names.txt") DO
    READFILE "names.txt", Line
    OUTPUT Line
ENDWHILE
CLOSEFILE "names.txt"
```

EOF("names.txt") 在每一行都读完之后变为 TRUE, 所以 WHILE NOT EOF(...) 的意思是还有东西可读时。

边读边计数

```
OPENFILE "marks.txt" FOR WRITE
WRITEFILE "marks.txt", 40
WRITEFILE "marks.txt", 65
WRITEFILE "marks.txt", 88
CLOSEFILE "marks.txt"

DECLARE Mark, Count, Total : INTEGER
Count ← 0
Total ← 0
OPENFILE "marks.txt" FOR READ
WHILE NOT EOF("marks.txt") DO
    READFILE "marks.txt", Mark
    Count ← Count + 1
    Total ← Total + Mark
ENDWHILE
CLOSEFILE "marks.txt"
OUTPUT Count, " marks, mean ", Total / Count
```

常见错误

- 写成 `WHILE EOF(...)`—— 那只在没有东西可读时才运行。你要的是 `NOT`。
- 循环没有 `EOF` 守卫, 读过了文件末尾。
- 对一个从未写入过的文件 `FOR READ`。