过程与函数

Ref Pseudocode

PROCEDURE and CALL

一个 **过程**过程是一组命名语句块。编写一次并从多个地方调用它，这就是 **分解**分解 — 整个纸笔 2 考试所依赖的理念。

```
PROCEDURE <identifier>
    <statements>
ENDPROCEDURE
```

通过名称调用过程，使用关键字 CALL：

```
PROCEDURE DefaultLine
    OUTPUT "------------------------------"
ENDPROCEDURE

CALL DefaultLine
OUTPUT "Report"
CALL DefaultLine
```

- 定义定义本身不会执行任何操作。只有 CALL 才能运行它。
- 当过程结束时，控制流返回到 CALL 之后的那一行。
- 过程 **不返回值**。如果需要返回值，你需要使用函数。

常见错误

- 单独编写 DefaultLine 来运行它。0478 需要 CALL。
- 遗漏 ENDPROCEDURE。
- 期望过程自身的变量在其外部存在。

FUNCTION and RETURN

函数函数是一个返回值的命名例程。它在需要该值的地方书写即位于 OUTPUT 内部、赋值号右侧或条件语句中。

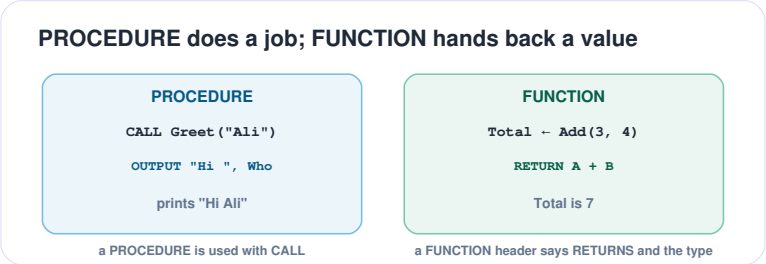
```
FUNCTION <identifier>(<parameters>) RETURNS <data type>
    <statements>
    RETURN <value>
ENDFUNCTION
```


exercises.lol exercises.com.cn

```
FUNCTION SumSquare(Number1 : INTEGER, Number2 : INTEGER) RETURNS INTEGER
    RETURN Number1 * Number1 + Number2 * Number2
ENDFUNCTION

OUTPUT "Sum of squares = ", SumSquare(10, 20)
```

- RETURNS <data type> 在头部声明了返回值类型，且不可省略。
- RETURN 会**立即**结束函数执行 — 其后的代码均不会运行。
- 调用函数时从不使用 CALL；只需在需要使用其值的位置直接使用即可。



程序通过 *CALL* 调用执行任务；函数则返回一个值供后续使用

■ 条件语句中的函数

```
FUNCTION IsEven(Number : INTEGER) RETURNS BOOLEAN
    RETURN MOD(Number, 2) = 0
ENDFUNCTION

DECLARE Index : INTEGER
FOR Index ← 1 TO 6
    IF IsEven(Index)
      THEN
        OUTPUT Index, " is even"
    ENDIF
NEXT Index
```

常见错误

- CALL SumSquare(10, 20) – CALL 调用的是过程而非函数。请在表达式中使用函数，例如 Answer ← SumSquare(10, 20)。
- 函数在某些路径上缺少 RETURN。所有出口路径都必须返回一个值。
- 头部遗漏了 RETURNS INTEGER。

Parameters

一个**参数**参数是调用过程或函数时传入的值。每个参数都有其独立的数据类型。

```
PROCEDURE Line(Size : INTEGER)
    DECLARE Index : INTEGER
    FOR Index ← 1 TO Size
        OUTPUT "-"
    NEXT Index
ENDPROCEDURE

CALL Line(5)
CALL Line(20)
```

多个参数之间用逗号分隔，调用时的顺序必须与定义中的顺序一致：在定义中：

```
PROCEDURE Greet(Name : STRING, Times : INTEGER)
    DECLARE Index : INTEGER
    FOR Index ← 1 TO Times
        OUTPUT "Hello ", Name
    NEXT Index
ENDPROCEDURE

CALL Greet("Mei", 2)
CALL Greet("Sam", 1)
```

△ 在 **0478 传值传递**模式下，每个参数都是按值传递的：过程操作的是副本，因此在内部修改它不会改变外部值。若要获取返回值，请使用 `FUNCTION` 并 `RETURN` 该变量。(A-Level 9618 增加了 `BYREF` 以支持另一种行为 — 参见主题 11。)

```
FUNCTION Doubled(Number : INTEGER) RETURNS INTEGER
    RETURN Number * 2
ENDFUNCTION

DECLARE Value : INTEGER
Value ← 7
Value ← Doubled(Value)
OUTPUT Value
```

常见错误

- `PROCEDURE Line(Size)` 没有类型。
- 参数顺序错误 — 这属于 `Greet(2, "Mei")` 类型错误。
- 期望过程修改传入的变量。在 0478 中无法做到这一点。

exercises.lol exercises.com.cn