数据类型、变量与常量

Ref As Pseudocode

六种数据类型

每个值都有一种**数据类型**（data type）。类型说明这个值是什么，以及可以对它做什么。9618 伪代码有六种基本类型。

类型	存放的内容	示例值
`INTEGER`	**整数**（whole number）	`5`、`-3`、`0`
`REAL`	可以带**小数部分**（fractional part）的数	`4.7`、`0.3`、`-4.0`
`CHAR`	恰好一个**字符**（character）	`'x'`、`'C'`、`'@'`
`STRING`	零个或多个字符	`"Hello"`、`""`
`BOOLEAN`	`TRUE` 或 `FALSE`	`TRUE`
`DATE`	一个日历日期	`02/01/2005`

■ 各类值的写法

直接写在代码里的值叫作**字面量**（literal）。每种类型都有自己的写法：

- **INTEGER**：普通数字，如 `25` 或 `-3`。
- **REAL**：小数点**两边**至少各有一位数字。写 `0.5`，不写 `.5`；写 `4.0`，不写 `4.`。
- **CHAR**：一个字符，用**单**引号括起来：`'A'`。
- **STRING**：一串字符，用**双**引号括起来：`"A"`。**空字符串**（empty string）`""` 不含任何字符。
- **BOOLEAN**：直接写 `TRUE` 和 `FALSE`，不加引号。
- **DATE**：通常写成 `dd/mm/yyyy`。世界各地的日期写法不同，所以最好说明你用的是哪种格式。

```
DECLARE Price : REAL
DECLARE Initial : CHAR
DECLARE Name : STRING
DECLARE Paid : BOOLEAN
Price ← 12.50
Initial ← 'M'
Name ← "Mei Lin"
Paid ← TRUE
OUTPUT Name, " (", Initial, ") paid ", Price, ": ", Paid
```

■ 该选哪种类型?

- 用来**计数**的东西（人数、进球数、物品数）：`INTEGER`。
- 需要**测量**、可能带小数的东西（钱、身高、平均值）：`REAL`。
- 是/否的答案，或者记录某件事是否发生过的**标志**（flag）：`BOOLEAN`。
- 电话号码或 `"007"` 这样的编号：`STRING`。你从不对它做算术，而且用 `INTEGER` 会丢掉开头的 0。

常见错误

- 用单引号写 `'Hello'`。单引号里只能放**一个**字符；字符串要用双引号。
- 给 `"TRUE"` 加引号。那是一个由四个字母组成的字符串，不是布尔值。
- 把平均值存进 `INTEGER`。平均值通常不是整数，应该用 `REAL`。

DECLARE 声明与赋值

■ DECLARE

变量（variable）是内存中一个有名字的位置，存放一个值，这个值在程序运行时可以改变。使用变量之前，你要先**声明**（declare）它：写出它的名字和数据类型。

```
DECLARE <identifier> : <data type>
```

```
DECLARE Counter : INTEGER
DECLARE TotalToPay : REAL
DECLARE GameOver : BOOLEAN
Counter ← 0
TotalToPay ← 0.0
GameOver ← FALSE
OUTPUT Counter, " ", TotalToPay, " ", GameOver
```

把变量声明写在答案的**开头**，每行一个。指南认为声明变量是好习惯，很多评分标准也会为此给分。如果题目要求“ 声明所有变量 ”，就必须声明。

■ 赋值

赋值（assignment）是把一个值放进变量。赋值运算符是**向左的箭头** `←`。

```
<identifier> ← <value>
```

`Counter ← 0` 读作"Counter **变为** 0" 或" 把 0 赋给 Counter"。先计算右边，再把结果存到左边。所以下面这一行是讲得通的：

```
DECLARE Counter : INTEGER
Counter ← 5
Counter ← Counter + 1
OUTPUT Counter
```

右边的 `Counter + 1` 用的是**旧**值 5，算出的结果 6 再替换掉旧值。

- 左边必须是**一个变量**（或数组的一个元素，或记录的一个字段）。
- 右边可以是任何**表达式**（expression），只要它算出的值与变量类型相同。
- 用键盘输入时，本站也接受 `<-`，并会把它自动换成 `←`。

常见错误

- 用 `Counter = 0` 来赋值。在伪代码中 `=` 用来**比较**两个值，从不用来存值。
- 写成 `0 ← Counter`，左右颠倒。被改变的变量永远在左边。
- 变量还没有值就使用它，例如在 `Total ← 0` 之前就往 `Total` 上加数。一定要先给出初始值，这叫作 **初始化**（initialising）变量。

CONSTANT 常量

常量（constant）是一个有名字的值，在程序运行期间**永远不变**。

```
CONSTANT <identifier> = <value>
```

```
CONSTANT HourlyRate = 6.50
CONSTANT DefaultText = "N/A"
DECLARE Hours : INTEGER
Hours ← 8
OUTPUT "Pay: ", Hours * HourlyRate
OUTPUT "Comment: ", DefaultText
```

注意这里用的是**等号** `=`，而不是箭头。常量只定义一次，不是被赋值的。

■ 为什么要用常量?

- `HourlyRate` 告诉读者 `6.50` 代表什么；代码中间孤零零的一个数字做不到这一点。
- 如果费率变了，你只需改**一行**，而不用改每一个出现这个数字的地方。

■ 规则

- 把常量写在程序的**开头**。
- 值必须是**字面量**，例如 `6.50` 或 `"N/A"`。不能是变量、另一个常量或一个计算式。`CONSTANT Total = Price * 2` 是不允许的。
- 程序中绝不能给常量赋值。`HourlyRate ← 7.00` 是错误的。

■ 练习

1. 为以下数据写出声明：一个人的年龄、以米为单位的身高、是否是会员、名字的首字母。

2. 某商店的税率固定为 0.08。写出这个常量，然后输出价格为 50 时的税额。

<details> <summary> 参考答案 </summary>

```
DECLARE Age : INTEGER
DECLARE Height : REAL
DECLARE IsMember : BOOLEAN
DECLARE FirstInitial : CHAR
Age ← 17
Height ← 1.72
IsMember ← TRUE
FirstInitial ← 'J'
OUTPUT Age, " ", Height, " ", IsMember, " ", FirstInitial
```

```
CONSTANT TaxRate = 0.08
DECLARE Price : REAL
Price ← 50.0
OUTPUT "Tax: ", Price * TaxRate
```

第二个程序输出 `Tax: 4.0`。

</details>

常见错误

- `CONSTANT TaxRate ← 0.08`。这是 **IGCSE** 的写法。在 9618 中，常量用 `=`。
- 在常量前面写 `DECLARE`。常量只用关键字 `CONSTANT`。