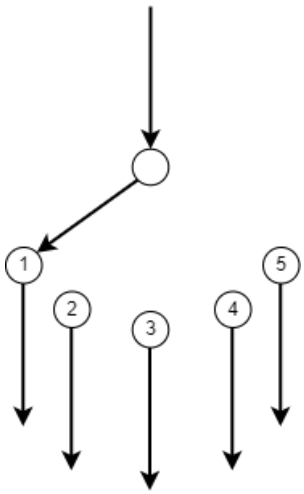
3.4 switch ... case

另一種「多分支」選擇結構

當你遇到像這樣的多分支選擇結構時，可以用 `if...else if...else` 來解決。



例如：一個像這樣的選單功能

```
#include <iostream>

using namespace std;

int main()
{
    cout << "(1) 提款" << endl;
    cout << "(2) 存款" << endl;
    cout << "(3) 查詢帳上餘額" << endl;
    cout << "(0) 結束" << endl;
    cout << "請選擇服務項目(0-3):";

    int i;
    cin >> i;

    if(i==1) {
        cout << "進行提款作業中..." << endl;
        cout << "提款作業完成!" << endl;
    }
    else if(i==2) {
        cout << "進行存款作業中..." << endl;
        cout << "存款作業完成!" << endl;
    }
    else if(i==3) {
        cout << "進行查詢作業中..." << endl;
        cout << "查詢作業完成!" << endl;
    }
    else if(i==0) {
        cout << "結束服務" << endl;
    }
    else {
        cout << "無此項目" << endl;
    }

    return 0;
}
```

除此之外，在 C++ 裡還有另一種 `switch ... case ...` 敘述，非常適合這種選單型的多分支結構。

它的基本語法如下：

```
switch (運算式)
{
    case 常數運算式1:
        ...
        break;
    case 常數運算式2:
        ...
        break;
    ...
    default:
        ...
}
```

- **switch** 後面括號中的運算式，其運算結果必需是 *整數* 或是 *字元*。
- 程式執行到 **switch** 時，會把小括號內運算式的運算結果拿來依序比對 **case** 後面的常數運算式(可能是 *整數* 或是 *字元*)。如果發現符合就跳到那個 **case** 的下一行開始執行，直到遇到 **break** 離開 switch 程式區塊。
- 如果 switch 的程式區塊中有 **default** 關鍵字，則當所有的 case 都不相符時，將由 default 處開始執行。

前面的選單程式，可以改寫成這樣。

```
#include <iostream>

using namespace std;

int main()
{
    cout << "(1) 提款" << endl;
    cout << "(2) 存款" << endl;
    cout << "(3) 查詢帳上餘額" << endl;
    cout << "(0) 結束" << endl;
    cout << "請選擇服務項目(0-3):";

    int i;
    cin >> i;

    // 以下用 switch ... case ... 改寫
    switch(i)
    {
        case 1:
            cout << "進行提款作業中..." << endl;
            cout << "提款作業完成!" << endl;
            break;
        case 2:
            cout << "進行存款作業中..." << endl;
            cout << "存款作業完成!" << endl;
            break;
        case 3:
            cout << "進行查詢作業中..." << endl;
            cout << "查詢作業完成!" << endl;
            break;
        case 0:
            cout << "結束服務" << endl;
            break;
        default:
            cout << "無此項目" << endl;
    }

    return 0;
}
```

break 很重要

使用 `switch ... case ...` 時最常發生的錯誤就是忘記加上 **break**。

case x: 只是一個標籤，程式跳到符合的標籤後開始向下執行，若沒遇到 **break** 會一直向下執行下去，即使遇到另一個 case 也是一樣。

```
    int num = 1;

    // 這不是我們想要的
    switch(num) {
        case 1:
            cout << "this number is 1." << endl;
        case 2:
            cout << "this number is 2." << endl;  // num 為 1 時這行也會執行到
    }
```

程式輸出如下:

```
this number is 1.
this number is 2.
```

```
    int num = 1;
    // 這才是我們想要的
    switch(num) {
        case 1:
            cout << "this number is 1." << endl;
            break;
        case 2:
            cout << "this number is 2." << endl;
            break;
    }
```

程式輸出如下:

```
this number is 1.
```

Fall-through - 利用沒加 break 的副作用

有時候我們會故意不加 **break** 利用它會一直執行下去的副作用來達到特別的目的。

練習：各月份所屬的季節

讀取使用者輸入的一個整數 $m, (1 \leq m \leq 12)$，輸出其所屬的季節。

- 2, 3, 4 月：春
- 5, 6, 7 月：夏
- 8, 9, 10 月：秋
- 11, 12, 1 月：冬

這樣寫看起來很累贅。

```
    int m;
    cin >> m

    switch(month) {
        case 2:
            cout << "Spring" << endl;
            break;
        case 3:
            cout << "Spring" << endl;
            break;
        case 4:
            cout << "Spring" << endl;
            break;
        case 5:
            cout << "Summer" << endl;
            break;
        case 6:
```

```
            cout << "Summer" << endl;
            break;
        case 7:
            cout << "Summer" << endl;
            break;
        case 8:
            cout << "Fall" << endl;
            break;
        case 9:
            cout << "Fall" << endl;
            break;
        case 10:
            cout << "Fall" << endl;
            break;
        default:
            cout << "Winter" << endl;
    }
```

這樣寫就好多了。

```
    int m;
    cin >> m

    switch(month) {
        case 2:
        case 3:
        case 4:
            cout << "Spring" << endl;
            break;
        case 5:
        case 6:
        case 7:
            cout << "Summer" << endl;
            break;
        case 8:
        case 9:
        case 10:
            cout << "Fall" << endl;
            break;
        default:
            cout << "Winter" << endl;
    }
```

Revision #25
★Created 3 May 2023 00:02:39 by huihui
Updated 21 September 2025 05:18:57 by huihui