10-3 如何建立複雜的類別

一、成員函數可以定義在 class 之外

我們可以只在 class body 中 `宣告` member function，再將它 `定義` 在 class body 之外。

```
class sprite {
private:
    string name;
public:
    sprite(string _name);  // 只有宣告
    string getName();      // 只有宣告
};

// 定義 sprite 類別的 sprite 成員函數
sprite::sprite(string _name)
{
    name = _name;
}

// 定義 sprite 類別的 getName 成員函數
string sprite::getName()
{
    return name;
}
```

以下是一個遊戲中角色的範例

```
#include <iostream>

using namespace std;

class sprite
{
private:
    string name;
    string status;
    int HP;
    int maxHP;

    void setStatus(void);  // 根據 hp 設定健康狀態
    void drawHpBar(void);  // 繪製 hp 長條圖
public:
    sprite(string _name);
```

```
    string getName();      // 取得姓名
    void decHP(int n);     // 減少 HP
    void addHP(int n);     // 增加 HP
    void show(void);       // 顯示狀態
};

void sprite::setStatus(void)
{
    if(HP >= maxHP *0.9)
        status = "健康";
    else if(HP >= maxHP *0.6)
        status = "受傷";
    else if(HP >= maxHP *0.3)
        status = "重傷";
    else if(HP > 0)
        status = "瀕死";
    else
        status = "死亡";
}

sprite::sprite(string _name)
{
    name = _name;
    HP = maxHP = 20;
    status = "健康";
}

string sprite::getName()
{
    return name;
}

void sprite::decHP(int n)
{
    HP = HP - n;
    if(HP <0)
        HP = 0;
    setStatus();
}

void sprite::addHP(int n)
{
    HP = HP + n;
    if(HP > maxHP)
        HP = maxHP;
    setStatus();
}

void sprite::drawHpBar(void)
{
```

```
    int a = 10*HP/maxHP;
    cout << "[";
    for(int i=0; i<a; i++)
        cout << "#";
    for(int i=a; i<10; i++)
        cout << "-";
     cout << "] ";
}

void sprite::show(void)
{
    cout << "[" << name << "] ";
    cout << "狀態：" << status << endl;
    drawHpBar();
    cout << "HP: " << HP  << "/" << maxHP << endl;
    cout << endl;
}

void attack(sprite &s, int n)
{
    cout << s.getName() << " 受到 " << n << " 點的傷害。" << endl;
    cout << endl;
    s.decHP(n);
}


int main()
{
    sprite fighter1("David");
    fighter1.show();
    attack(fighter1, 6);
    fighter1.show();
    attack(fighter1, 6);
    fighter1.show();
    attack(fighter1, 6);
    fighter1.show();
    attack(fighter1, 6);
    fighter1.show();

    return 0;
}
```

David 受到 6 點的傷害。
[David] 狀態：受傷 [########---] HP: 14/20

David 受到 6 點的傷害。

[David] 狀態：重傷 [####------] HP: 8/20

David 受到 6 點的傷害。

[David] 狀態：瀕死 [#---------] HP: 2/20

David 受到 6 點的傷害。

[David] 狀態：死亡 [----------] HP: 0/20

二、將類別放在獨立的檔案中

在前一個例子裡，我們把 sprite 類別和 main 放在同一個檔案裡，在程式碼短的情形下是沒問題的。但在動輒使用到數十乃至上百個類別的程式裡，這麼做就不切實際了，會造成維護和分工上的困擾。

我們可以將sprite類別移到獨立的檔案裡，這樣修改 sprite 時就可以專注在 sprite 的程式碼，main.cpp 裡也不會顯得凌亂，建置專案時的效率也會更好。

sprite.h

```
#ifndef SPRITE_H_INCLUDED
#define SPRITE_H_INCLUDED

#include <iostream>

using namespace std;

class sprite
{
private:
    string name;
    string status;
    int hp;
    int maxHp;

    void setStatus(void);  // 根據 hp 設定健康狀態
    void drawHpBar(void);  // 繪製 hp 長條圖
public:
    sprite(string _name);
    string getName();      // 取得姓名
    void decHP(int n);     // 減少 HP
    void addHP(int n);     // 增加 HP
    void show(void);       // 顯示狀態
};

#endif // SPRITE_H_INCLUDED
```

sprite.cpp

```
#include "sprite.h"

void sprite::setStatus(void)
{
    if(HP >= maxHP*0.9)
        status = "健康";
    else if(HP >= maxHP*0.6)
        status = "受傷";
    else if(HP >= maxHP*0.3)
        status = "重傷";
    else if(HP > 0)
        status = "瀕死";
    else
        status = "死亡";
}

sprite::sprite(string _name)
{
    name = _name;
    HP = maxHP = 20;
    status = "健康";
}

string sprite::getName()
{
    return name;
}

void sprite::decHP(int n)
{
    HP = HP - n;
    if(HP<0)
        HP = 0;
    setStatus();
}

void sprite::addHP(int n)
{
    HP = HP + n;
    if(HP > maxHP)
        HP = maxHP;
    setStatus();
}

void sprite::drawHpBar(void)
{
    int a = 10*HP/maxHP;
```

```
    cout << "[";
    for(int i=0; i<a; i++)
        cout << "#";
    for(int i=a; i<10; i++)
        cout << "-";
     cout << "] ";
}

void sprite::show(void)
{
    cout << "[" << name << "] ";
    cout << "狀態：" << status << endl;
    drawHpBar();
    cout << "HP: " << HP  << "/" << maxHP << endl;
    cout << endl;
}
```

main.cpp

```
#include <iostream>
#include "sprite.h"

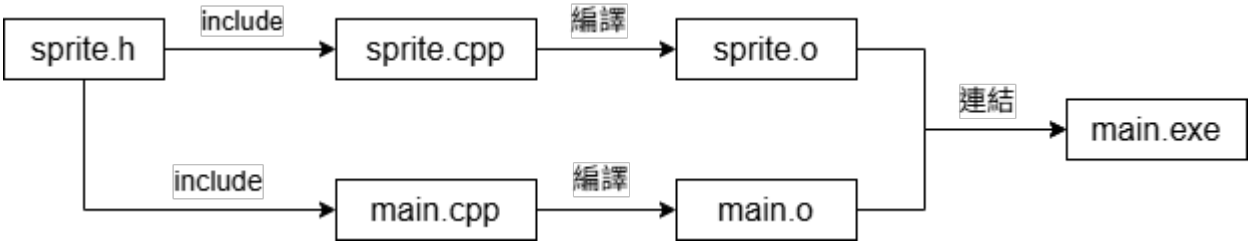
using namespace std;

void attack(sprite &s, int n)
{
    cout << s.getName() << " 受到 " << n << " 點的傷害。" << endl;
    cout << endl;
    s.decHP(n);
}

int main()
{
    sprite fighter1("David");
    fighter1.show();
    attack(fighter1, 6);
    fighter1.show();
    attack(fighter1, 6);
    fighter1.show();
    attack(fighter1, 6);
    fighter1.show();
    attack(fighter1, 6);
    fighter1.show();

    return 0;
}
```

建置時的檔案相依性



Revision #2
★Created 2026-05-31 00:32:57 UTC by huihui
Updated 2026-05-31 00:51:39 UTC by huihui