

C 語言實作獨立的類別



高等 C 語言

簡介

字串

指標與陣列

函數

結構

物件導向

記憶體

檔案

錯誤處理

巨集處理

C 與組合語言

工具鏈

Make

開發環境

Cygwin

MinGW

DevC++

wxDevC++

編譯器

gcc 編譯器

TinyCC 編譯器

LCC 編譯器

應用主題

CGI 程式

GNU 程式

訊息

相關網站

文章

評論

授權

在前述的範例當中，我們直接將資料與函數封裝在結構當中，以形成物件，這種實作方式並沒有為類別定義獨立的結構，於是每個物件當中都會有一份所有成員函數的指標，當物件的數量很多時，這可能會浪費不少記憶體。

前述的這種實作方式，比較像是一種變形後的物件導向實作法，這種方法稱為原型 (Prototype) 導向的實作法，像是 JavaScript 就採用了類似的實作方式。如果我們要實作出像 Java 或 C# 一樣的物件導向作法，應該將類別的結構獨立出來，這樣會比較能夠規模化，而且通常可以節省記憶體。

在以下的程式中，我們將再度用 C 語言實作出這種方式，將物件與類別獨立成兩個不同結構。如此，不管我們建立幾份物件，類別物件永遠都只會有一個，請看下列程式碼。

程式實作：將類別獨立出來

檔案：polyClass.c

```
#include <stdio.h>

#define ShapeClassMembers(OBJ) float (*area)(struct OBJ*)

#define ShapeMembers(OBJ)

struct _Shape;

typedef struct _ShapeClass { // Shape 物件，沒有欄位
    ShapeClassMembers(_Shape);
} ShapeClass;

typedef struct _Shape {
    ShapeClass *class;
```

```
float ShapeArea(Shape *obj) { return 0; }

ShapeClass shapeClass = { .area = ShapeArea };

struct _Circle;

typedef struct _CircleClass {
    ShapeClassMembers(_Circle);
    float r;
} CircleClass;

typedef struct _Circle {
    CircleClass *class;
    ShapeMembers(_Circle);
    float r;
} Circle;

float CircleArea(Circle *obj) { return 3.14 * obj->r * obj->r; }

CircleClass circleClass = { .area = CircleArea };

int main() {
    Shape s = { .class = &shapeClass };
    Circle c = { .class = &circleClass };
    c.r = 3.0;
    Shape *list[] = { &s, (Shape*) &c };
    int i;
    for (i=0; i<2; i++) {
        Shape *o = list[i];
        printf("s.area()=%G\n", o->class->area(o));
    }
}
```

## 執行結果

```
D:\>gcc polyClass.c -o polyClass
```

```
D:\>polyClass
```

```
s.area()=0
```



page revision: 1, last edited: 2 Sep 2010, 11:58 GMT+0800 (510 days ago)

[Edit](#) [Rate \(0\)](#) [Tags](#) [Discuss \(0\)](#) [History](#) [Files](#) [Print](#) [Site tools](#) [+ Options](#)Powered by [Wikidot.com](#)[Help](#) | [Terms of Service](#) | [Privacy](#) | [Report a bug](#) | [Flag as objectionable](#)

Unless otherwise stated, the content of this page is licensed under [Creative Commons Attribution-NonCommercial-ShareAlike 3.0 License](#)

## Other interesting sites



jQuery EasyUI  
easyui help you build your  
web page easily!



Wiki Benfica  
Tudo sobre o Sport Lisboa e  
Benfica



History in the New Media  
Digital Project Design in  
Archives and Public History



Biblioteca Digital Curt  
Nimuendaju  
livros e artigos sobre línguas  
sul-americanas