



進階 C 語言實務

Homework 3

老師：蔣政諺

班級：電機碩一

學號：111318133

姓名：魏千竣

日期：112/05/16

1. Program Description

1. 匯入所需的標頭檔，及定義定義了 PI 來表示圓周率，和使用命名空間 std。

```
#include <iostream>
#include <cmath>

#define PI 3.14159265

using namespace std;
```

2. 定義一個基礎類別 Shape，該類別包含一個純虛函數 Area() $=0$ ，用於計算形狀的面積。Shape 類別的建構函數和解構函數都輸出一條消息，以便我們在程序運行時，能看到它們被調用的情況。

```
class Shape {
public:
    Shape() {
        cout << "Shape ctor" << endl;
    }

    virtual ~Shape() {
        cout << "Shape dtor" << endl;
    }

    virtual float Area() const = 0;
};
```

3. Circle 類別繼承 Shape 類別。且具有一個額外的數據成員 radius，用於存儲圓的半徑。該類別的建構函數和解構函數也會輸出相應的消息。Circle 類別重寫了 Shape 類別的 Area() 函數，用於計算圓的面積。

```

class Circle : public Shape {
protected:
    float radius;

public:
    Circle(float r) : radius(r) {
        cout << "Circle ctor" << endl;
    }

    ~Circle() {
        cout << "Circle dtor" << endl;
    }

    float Area() const override {
        return PI * radius * radius;
    }
};

```

4. Cylinder 類別繼承 Circle 類別。且具有一個額外的數據成員 height，用於圓柱體的高度。該類別的建構函數和解構函數也會輸出相應的消息。Cylinder 類別重寫了 Circle 類別的 Area() 函數，用於計算圓柱的面積。

```

class Cylinder : public Circle {
protected:
    float height;

public:
    Cylinder(float r, float h) : Circle(r), height(h) {
        cout << "Cylinder ctor" << endl;
    }

    ~Cylinder() {
        cout << "Cylinder dtor" << endl;
    }

    float Area() const override {
        return 2 * Circle::Area() + 2 * PI * radius * height;
    }
};

```

5. Sphere 類別繼承 Circle 類別。該類別的建造函數初始化 Circle 類別的半徑，並輸出一條消息以顯示它已被調用。解構函數同樣輸出調用消息。Sphere 類別重寫了 Circle 類別的 Area() 函數，用於計算球體的面積。

```

class Sphere : public Circle {
public:
    Sphere(float r) : Circle(r) {
        cout << "Sphere ctor" << endl;
    }

    ~Sphere() {
        cout << "Sphere dtor" << endl;
    }

    float Area() const override {
        return 4 * Circle::Area();
    }
};

```

6. 定義了一個 FindArea 的函數，該函數接受一個 Shape 類型的引用作為參數，並返回該形狀對象的面積。

```

float FindArea(Shape& shape) {
    return shape.Area();
}

```

7. 主程式：

- 創建了兩個指向 Shape 的指針，分別為 pS1 和 pS2。pS1 指向一個 Cylinder 物件，其半徑為 2，高為 4；pS2 指向一個 Sphere 物件，其半徑為 3。由於 Cylinder 和 Sphere 類別都繼承 Shape 類別，因此可以將指向它們的指針賦值給 Shape 類型的指針。

```

int main() {
    Shape* pS1 = new Cylinder(2, 4);
    Shape* pS2 = new Sphere(3);
}

```

- 輸出 Cylinder、Sphere 物件的面積。它們使用 FindArea 函數來獲取面積。FindArea 函數接受一個 Shape 類型的引用，因此將指針使用*運算符以傳遞對象的引用。

```

cout << "Cylinder area: " << FindArea(*pS1) << endl;
cout << "Sphere area: " << FindArea(*pS2) << endl;

```

- 釋放使用 new 分配的兩個 Shape 類型物件（Cylinder 和 Sphere）的內存。使用 delete 運算符調用相應物件的解構函數並釋放內存。

```

delete pS1;
delete pS2;

return 0;
}

```

2. Result Display

```
PS C:\Users\503503\Desktop\AdvC> ./hw3.exe
Shape ctor
Circle ctor
Cylinder ctor
Shape ctor
Circle ctor
Sphere ctor
Cylinder area: 75.3982
Sphere area: 113.097
Cylinder dtor
Circle dtor
Shape dtor
Sphere dtor
Circle dtor
Shape dtor
```