



🏠 / C++프로그래밍과실습 (CB3500572-062) / 실습 101 - IShape interface / 개요

개요

제출

편집

코딩 결과

실습 101 - IShape interface

제출 마감일: 2023-06-09 23:59

업로드 가능한 파일 수: 2

제출 방식: 개인

목적

이 실습은 여러 개의 연관 있는 객체들의 집합으로 하나의 객체를 구성하는 방법을 연습합니다.

설명

파일시스템에서 `directory_iterator` 는 특정 경로의 파일과 디렉토리 정보를 순회하는 기능을 제공합니다.

`recursive_directory_iterator` 는 거기에 더해서 서브 디렉토리까지 순회하는 기능을 제공합니다.

이렇게 여러 종류의 파일과 디렉토리를 하나의 집합으로 구성하여 동일한 방법으로 접근할 수 있는 객체지향적 시크릿은 상속과 다형성에 있습니다.

파일 시스템처럼 여러 개의 객체를 이용하여 하나의 객체로 구성 (composite) 해서 사용하는 방법은 객체 지향 세계에서는 자주 사용되는 설계 기법 (패턴) 입니다.

대표적인 예로는 그래픽 객체들 (사각형, 텍스트, 이미지 등) 을 그룹으로 묶어서 하나의 객체로 이용하는 것입니다.

파워 포인트에서 마우스로 여러 개의 그래픽 객체를 선택하고 오른쪽 버튼을 눌러 그룹 설정하는 상상을 해 보십시오.

코드로 살펴 보겠습니다.

클래스 계층도에서 최상위 클래스를 추상 클래스로 정의합니다.

방법은 pure virtual function을 선언하는 것입니다.

잊지 마십시오!

virtual 소멸자를 선언해 줘야, 파생 클래스의 소멸자가 정상적으로 호출된다는 것어요.

딱 봐도 인터페이스처럼 생겼네요.

```
class IShape {
```

```
public:
```

```
virtual void draw() = 0;

virtual ~IShape() {}

};
```

그래픽 객체 3가지 (버튼, 이미지, 테이블) 을 간단하게 만들어 봅니다.

당연히 IShape 를 상속 받아서 virtual 함수들을 override 해 줘야 합니다.

```
class Button : public IShape {

public:

    ~Button() {}

    void draw() override {

        cout << "Button is drawn!" << endl;

    }

};
```

```
class Image : public IShape {

public:

    ~Image() {}

    void draw() override {

        cout << "Image is drawn!" << endl; }

};
```

```
class Table : public IShape {

public:

    ~Table() {}

    void draw() override {

        cout << "Table is drawn!" << endl;

    }

};
```

여러 개의 객체를 하나의 객체로 구성하는 핵심 클래스 입니다. 예상 하셨듯이 vector 와 같은 컨테이너를 이용해 그래픽 객체들을 저장합니다.

```

class GroupShape : public IShape {
public:
    GroupShape(string name) : _name{name} {}
    ~GroupShape() {}
    void add(IShape* shape) {
        shapes.emplace_back(shape);
    }
    void draw() override {
        cout << "Group Name: " << _name << endl;
        for(const auto& s : shapes)
            (*s).draw();
    }
private:
    string _name;
    vector<IShape*> shapes;
};

```

자 이제 여러 개의 객체를 만들어서 구성을 해보겠습니다.

버튼 하나, 버튼 둘, 이미지 하나, 테이블 하나.

오른쪽 버튼 누르고 그룹 설정..... 뭐 해볼까요?

선 색상을 변경해 볼까요? 폰트를 변경해 볼까요?

객체 지향 텍스트북의 전통에 따라서 draw() 해 보겠습니다.

짜잔! 객체 지향 세계에 오신 것을 환영합니다!

```

int main() {
    GroupShape root("root");
    auto button1 = Button{};
    root.add(&button1);

    auto sub1 = GroupShape("sub1");

```

```
auto button2 = Button{}; sub1.add(&button2);  
auto image1 = Image{}; sub1.add(&image1);  
auto table1 = Table{}; sub1.add(&table1);  
  
root.add(&sub1);  
root.draw();  
  
return 0;  
}
```

문제

위 예제 코드를 동작시켜 제출하시오.

입력

없음

출력

Group Name: root

Button is drawn!

Group Name: sub1

Button is drawn!

Image is drawn!

Table is drawn!

제출파일

main.cpp