C++ 스터디

9. 상속(Inheritance)-2

D.Com

상속은 왜 하는 걸까?

- 상속은 이미 존재하는 클래스를 이용하거나 새로운 클래스를 생산하여 다른 동작을 하게 하거나 기능을 추가할 때에 있어 아주 유용한 도구이다.
- 프로그램의 기능 추가나 변경을 할 때에는 존재하는 코드를 변경해서 만들지 않는다. 즉 base class의 소스코드를 건들어서는 안된다.

바인딩 (개념)

- 바인딩이란 프로그램 소스에 쓰인 각종 내부 요소, 이름, 식별자들에 대해 값 또는 속성을 확정하는 과정을 바인딩이라고 한다.
- 예를 들자면 C++에는 여러 컨테이너 객체(list, vector, array 등)이 있는데 얘네 중 어느 종류의 데이터가 들어오든 똑같은 동작을 하기 위해서는 바인딩이 고정되어서는 안된다.
- 바인딩이 빌드(컴파일) 중에 이루어져 실행하는 동안 변경이 불가능하면 정적 바인딩(static binding)이라고 하고 실행 중에 바인딩이 이루어지면 동적 바인딩(dynamic binding)이라고 한다.
- 동적 바인딩과 상속을 이용해 다형성(Polymorphism, 같은 형태로 여러가지 기능을 할 수 있게 프로그래밍 해야함)을 실현 가능하다.

D.Com

정적 바인딩 (예시)

- int a = 0;과 같이 소스상에 명시적으로 타입과 그 타입의 변수명을 선언하는 것.
- 장점은 컴파일시 타입에 대한 정보가 결정되기 때문에 속도가 빠른 것과 타입 에러로 인한 문제를 조기에 발전할 수 있어서 안정적이라는 점이 있다.
- 단점으로는 컴파일시 결정이 되고 그 이후 변경이 불가능하다는 점이 있다.
- 동적 바인딩을 클래스로 설명할 것이기에 정적인 클래스에 대한 예시이다.

```cpp
#include <iostream>
using namespace std;

class Base {
public:
    void f(void) {
        cout << "in function 'Base::f()'\n";
    }
    virtual void vf(void) {
        cout << "in function 'Base::vf()'\n";
    }
};
class Derived : public Base {
public:
    void f(void) {
        cout << "in function 'Derived::f()'\n";
    }
    void vf(void) override {
        cout << "in function 'Derived::vf()'\n";
    }
};
int main()
{
    Base myB;
    Derived myD;
    myB.f();
    myB.vf();
    myD.f();
    myD.vf();
}
```

```
Microsoft Visual Studio 디버그 콘솔
in function 'Base::f()'
in function 'Base::vf()'
in function 'Derived::f()'
in function 'Derived::vf()'
```

D.Com

동적 바인딩 (예시)

- 상속 받은 클래스를 동적 바인딩해 사용할 때 virtual 키워드를 적어주지 않으면 base 클래스의 함수가 사용됨을 알 수 있다.
- new 키워드를 사용해 동적 바인딩을 해줄 경우 꼭 해제를 해줘야 한다.

```cpp
#include <iostream>
using namespace std;

class Base {
public:
    void f(void) {
        cout << "in function 'Base::f()'\n";
    }
    virtual void vf(void) {
        cout << "in function 'Base::vf()'\n";
    }
};
class myB : public Base {
public:
    void f(void) {
        cout << "in function 'myB::f()'\n";
    }
    void vf(void) override {
        cout << "in function 'myB::vf()'\n";
    }
};

class myD : public Base {
public:
    void f(void) {
        cout << "in function 'myD::f()'\n";
    }
    void vf(void) override {
        cout << "in function 'myD::vf()'\n";
    }
};
int main()
{
    Base* p;
    p = new myB;
    p->f();
    p->vf();
    p = new myD;
    p->f();
    p->vf();
    delete p;
}
```

```
Microsoft Visual Studio 디버그
in function 'Base::f()'
in function 'myB::vf()'
in function 'Base::f()'
in function 'myD::vf()'

C:\Users\bluej\Desktop\
 종료되었습니다.
이 창을 닫으려면 아무 키
```

D.Com

동적 바인딩 (예시)

- 이런 식으로 & 오퍼레이터를 사용해서도 가능하다.
- 이러한 동적 바인딩을 사용해서 드디어 다형성을 구현 가능하다.

```cpp
#include <iostream>
using namespace std;

class Base {
public:
    void f(void) {
        cout << "in function 'Base::f()'\n";
    }
    virtual void vf(void) {
        cout << "in function 'Base::vf()'\n";
    }
};
class Derived : public Base {
public:
    void f(void) {
        cout << "in function 'Derived::f()'\n";
    }
    void vf(void) override {
        cout << "in function 'Derived::vf()'\n";
    }
};

int main()
{
    Base* p;
    Base myB;
    Derived myD;
    p = &myB;
    p->f();
    p->vf();
    p = &myD;
    p->f();
    p->vf();
}
```

D.Com

다형성 (예시)

```cpp
#include <string>
#include <vector>
#include <iostream>
using namespace std;

class Text {
    string text;
public:
    Text(const string& t) :text(t) {}
    virtual string get() const {
        return text;
    }
};

class FancyText : public Text {
    string left_bracket;
    string right_bracket;
    string connector;
public:
    FancyText(const string& t, const string& left,
        const string& right, const string& conn) :
        Text(t), left_bracket(left), right_bracket(right),
        connector(conn) {}
    string get() const override {
        return left_bracket + Text::get() + right_bracket;
    }
};

class FixedText : public Text {
public:
    FixedText() : Text("Fixed") { }
};
```

```cpp
int main() {
    vector<Text*> texts{
        new Text("WOW"), new FancyText("Wee","[","]","-"),
        new FixedText,
        new FancyText("whoa",":",":",":")
    };

    for (auto t : texts) {
        cout << t->get() << '\n';
        delete t;
    }

}
```

Microsoft Visual Studio 디버그 콘솔

```
WOW
[Wee]
Fixed
:whoa:

C:\Users\bluej\Desktop\방학프로그래밍\c++스터디\tut
 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```

D.Com