항목 37. 어떤 함수에 대해서도 상속받은 기본 매개변수 값은 절대로 재정의하지 말자.

C++에서 상속받을 수 있는 함수의 종류는 **가상 함수와 비가상 함수**뿐이다. 이들 중 비 가상 함수는 언제라도 재정의해서는 안 되는 함수 이므로(항목 36), ‘**기본 매개변수 값을 가진 가상 함수를 상속하는 경우**’ 재정의 해선 안된 다로 의미를 좁힐 수 있다.

이유는 가상 함수는 동적으로 바인딩 되지만, 기본 매개변수 값은 정적으로 바인딩 된다는 것이다.

[정적 바인딩은 선행 바인딩(early binding), 동적 바인딩은 지연 바인딩(late binding)으로 알려져 있다.]

객체의 정적 타입(static type)은 프로그램 소스 안에 여러분이 놓은 선언문을 통해 그 객체가 갖는 타입이다.

|  |
| --- |
| class Shape{  public:  enum ShapeColor { red , green , blue };  virtual void draw(ShapeColor color = red) = 0;  }  class Rectangle : public Shape{  public:  virtual void draw(ShapeColor color = green);  }  class Circle : public Shape{  public:  virtual void draw(ShapeColor color);  }  Shape \*ps;  Shape \*pc = new Circle;  Shape \*pr = new Rectangle; |

여기서 ps, pc 및 pr은 모두 ‘Shape에 대한 포인터’로 선언 되었기 때문에, 각각의 정적 타입도 모두 이 타입이다. 객체의 동적 타입은 현재 그 객체가 진짜로 무엇이냐에 따라 결정되는 타입이다. 다시 말해, ‘이 객체가 어떻게 동작할 것이냐’ 를 가리키는 타입이 동적 타입이다.

따라서, pc의 동적 타입은 Circle\*, pr의 동적 타입은 Rectable\*, ps는 동적 타입이 없다.

문제는 pr->draw()와 같은 기본 매개 변수가 재 정의된 가상 함수를 호출했을 때 발생한다. 파생 클래스에 정의된 가상 함수를 호출하면서 기본 클래스에 정의된 기본 매개변수 값을 사용해 버릴 수 있다는 것이다.

Pr의 동적 타입은 Rectangle\*이기 때문에 함수는 Ractangle::draw()가 호출되지만 기본 매개변수 값은 기본 클래스의 것인 사용한다. Rectangle::draw(Shape::Red) 이렇게 말이다.

이러한 동작 방식은 런타임 효율이라는 요소가 숨어 있다. 프로그램 실행 중에 가상 함수의 기본 매개변수 값을 결정할 방법을 컴파일러 쪽에서 마련해 주어야 하는데, 이 방법은 컴파일 과정에서 결정하는 현재의 메커니즘 보다 느리고 복잡하다. 때문에 속도와 구현 간편성에 무게를 더 두어야 했고, 그 덕에 효율 좋은 실행 동작을 누릴 수 있게 되었다.

그렇다면, 안전하고 원하는 동작을 하는 가상 함수를 만들기 위해서 어떻게 해야 할 것인가. 가상 함수 대신에 쓸 수 있는 방법들 (항목 35)에 소개 되어 있다. 이들 중 **비가상 인터페이스 관용구(NVI)**를 쓰는 것이다. 이 가상 함수를 호출하는 public 비가상 함수를 기본 클래스에 만들어 두는 것이다. 여기서 이 방법을 응용한다면, 비가상 함수가 기본 매개 변수를 지정 하도록 할 수 있겠다.

|  |
| --- |
| class Shape{  public:  enum Color { red , green , blue };  void draw ( Color color = red ) const  {  **doDraw(color);**  }  private:  **virtual void doDraw(Color color) const = 0;**  }  class Circle : public Shape{  public:  ...  private:  **virtual void doDraw(Color color) const;**  }; |

비 가상 함수는 파생 클래스에서 오버라이드 되면 안 되기 때문에(항목 36), 위와 같이 설계하면 draw 함수의 color 매개변수에 대한기본값을 깔끔하게 red로 고정 시킬 수 있다.

**요약**

- 상속받은 기본 매개변수 값은 정대로 재정의해서는 안됩니다. 왜냐하면 기본 매개변수 값은 정적으로 바인딩되는 반면, 가상 함수(여러분이 오버라이드할 수 있는 유일한 함수이죠)는 동적으로 바인딩되기 때문입니다.