객체지향프로그래밍 10주차 정리노트

202204145 최대철
202204156 한승헌

1. 클래스 상속

수업자료 7페이지에 보면 class Student, class StudentWorker, class Reasearcher, class Professor 이렇게 네 개의 클래스가 있었다. 네 개의 클래스를 모두 사용하면서 간결하게 클래스를 만들기 위해선, 속성이 비슷한 Student와 StudentWorker 클래스끼리 상속을 이용하고, Reasearcher와 Professor 클래스끼리 상속을 사용하면 된다고 생각했다. Student 클래스를 먼저 선언하고, StudentWorker 클래스에 상속을 시키고 일하기 속성을 추가하면 된다고 생각했고, Researcher 클래스를 먼저 선언하고 Professor 클래스에 상속을 시켜 가르치기 속성을 추가하면 클래스를 간략하게 만들 수 있다고 생각했다. 간략하게 선언을 할 수 있는 상속이라는 키워드에 대해서 잘 알 수 있는 계기였다.

2. 예제 8-1

앞서 살펴본 클래스 상속에 대하여 이야기하고 나서 서로 코딩을 해보았다. 먼저 Point 클래스를 선언하고 ColorPoint 클래스에 상속을 시켜 코딩을 이어나갔다. 코딩 전체를 짜면서 서로 아무 문제는 없었지만 Point 클래스에 선언한 함수들을 ColorPoint 클래스에 상속했다는 이유로 ColorPoint 타입의 변수도 Point 클래스의 함수에 접근할 수 있다는 것에 신기했었다.

3. 업 캐스팅 & 다운 캐스팅

자바에서는 다른 방식이 있었지만 포인터로 객체를 치환한다는게 신기했다. 업 캐스팅은 큰 틀에서 작은틀을 상속시켜 자연스러운거지만, 다운 캐스팅 같은 경우는 작은 틀에서 큰 틀을 소환시키는 것과 같기 때문에 사용하고 싶은 객체의 타입도 큰 틀의 클래스로 변환을 시켜야 한다는 점이 또 신기했다. 위의 상황같은 경우는 pDer 객체만 상속 받은 클래스에 접근 가능하게 하는 것이므로 필요한 상황이 있을 수 있겠구나 싶었다.

4. Protected 멤버 (예제 8-2)

Protected가 Point 클래스에 선언이 되었고, ColorPoint 클래스에 상속이 되었으므로 Point 와 Colorpoint 클래스 내에서만 int x와 y가 선언이 가능하다고 생각해 main에서 x,y를 선언할 수 없다고 생각했다.