

김건희 과제 2016.09.19

1. 객체지향 프로그래밍에서 추상화, 은닉화, 캡슐화, 상속, 다형성이라는 특징에 대해 알아보세요.

가. 추상화

인터페이스와 구현을 분리하는 것을 말한다. 추상화를 통해 객체가 가진 특성 중 필수 속성만으로 객체를 묘사하고 유사성만을 표현하며 세부적인 상세사항은 각 객체에 따라 다르게 구현되도록 할 수 있다. 추상화란 중요하지 않거나, 주 관심 대상이 아닌 자세한 부분은 감추거나 무시하고, 가장 중요하고, 근간이 되고, 다른 대상들과 구분될 수 있는 면만을 포함하고 있는 모델이며, 공통점을 강조하기 위해 차이점을 제거한 결과물이라고 정의한다. 실제로 객체의 속성과 행위 중에서 관심대상이 아닌 부분은 드러낼 필요가 없는 것이 추상화이다

추상화는 다른 객체들과 구분되는 핵심적인 특징들에만 집중함으로써, 복잡도를 관리할 수 있도록 한다. 주의할 점은 추상화는 문제영역과 관점에 의존적이라는 것이다. 그래서 어떤 영역에서 중요한 것이 다른 영역에서는 그렇지 않을 수도 있다. 하나의 대상에 목적이거나 원하는 기능에 따라 여러 추상화 모델이 생성될 수 있다.

나. 은닉화

캡슐화의 장점중 하나이자, 사용자에게는 잘못된 데이터 입력을 방지하고 개발자로부터 필요 없는 정보를 숨겨준다.

내부를 숨김으로써 내부를 좀더 자유롭게 변경할 수 있게 되었다. 이전의 함수 중심적인 구조적 프로그래밍 언어에서는 프로그램 내부에서 데이터가 어디서 어떻게 변경되는지 파악하기 어려웠고, 그로 인해 유지 보수가 힘들었기 때문에 자료를 중심으로 함수가 종속되는 구조가 되기도 하였다. 객체 지향에서는 클래스 내부의 데이터를 외부에서 참조하지 못하도록 차단하여 이러한 폐단을 없앨수있다. 이렇게 내부의 데이터나 함수를 외부에서 참조하지 못하도록 차단하는 개념을 정보은닉이라고 하며 이것이 바로 캡슐화라는 개념이다.

다. 캡슐화

객체지향에서 캡슐화라는 개념은 클래스 내부에 여러 속성과 여러 오퍼레이션을 함께 묶음을 의미한다. 그리고 캡슐화는 클래스 내부의 속성이나 오퍼레이션을 외부에 노출하지 않고 보호하는 것을 의미한다. 이렇게 캡슐화는 묶는 것과 보호하는 것을 생각할 수있다. 좀 더 상세하게 생각해 보면, 여러 속성과 여러 오퍼레이션을 함께 묶어 클래스로 취급하는 것과 클래스 내부를 외부에서 접근하지 못하도록 보호하는 것이 바로 캡슐화이다. 묶음으로 인해 프로그램을 바라보는 단위가 커진다. 이전의 프로그래밍 언어인 C 언어는 프로그램을 함수 단위로 구조화할 수 있으나, 프로그램 소스가 커지면 이해하기 어렵고 관리가 힘들어 질수있다. 그러나 객체지향 프로그램에서는 프로그램 소스를 클래스 단위로 바라보게 됨으로써 좀 더 복잡하고 커다란 소스코드도 쉽게 이해하게되었다. 왜냐하면 클래스 내부에 여러 함수를 내포할 수 있기 때문에 프로그램 소스코드를 바라보는 단위가 커졌으며, 그로 인해 프로그램 관리가 좀 더 수월해진 것이다.

라. 상속화

상속의 사전적 의미는 자신이 가지고 있는 것을 하위에게 물려주거나, 하위에서 물려받는 것이다. 객체 지향에 상속도 마찬가지로 의미이며, 클래스의 속성과 오퍼레이션을 하위 클래스에 물려주거나, 상위클래스에서 물려받는것을 지칭한다. 객체지향에 상속도 마찬가지로 의미이며, 클래스의 속성이나 오퍼레이션을 다시 정의하지 않고 상속받아서 사용함으로써 코드가 줄어든다. 그리고 좀 더 범용성있게 사용할수있다. 참고로 하위 클래스는 상위클래스가 가지고 있는 모든 자료와 메소드를 물려받아 자유롭게 사용할수 있지만, 또한 자신만의 자료와 메소드를 추가적으로 덧붙임으로써 새로운 형태의 클래스로 발전하게 된다는 것도 잊지말자. 생성자라는 것은 일종의 오퍼레이션이며, 클래스가 객체화될때 실행된다. 그런데 하위 클래스의 생성자를 호출할 경우, 자동적으로 상위의 생성자가 호출됨을 반드시 기억해야 한다. 하위의 생성자가 호출될 때 묵시적으로 상위의 생성자를 호출하는 것이다.

마. 다형성

오버로딩, 오버라이딩, 상속 등을 복합적으로 사용함으로써 하나로 여러가지 처리를 하는 것을 말한다. 예를들어 오버로딩은 하나의 메서드가 여러가지 기능을 포함하고 있는 것을 다형성이라 한다. 객체지향 개념에서 가장 중요한 것이 바로 다형성이다. 다형성이라는 개념은 객체 지향개념의 꽃이라고 할수있을 정도로 중요하고 근사한 개념이다. 웹사이트에 따르면, 다형성의 일반적인 의미는 다양한 형태로 나타날수 있는능력이라고 한다. 여러 클래스들이 동일한 이름의 오퍼레이션을 서비스하도록 하는것이다. 상위클래스에 있고 상속받았으나 하위 클래스에서 다시 정의하는 것을 메소드 오버라이딩이라고 하며, 클래스 내부에 동일한 이름의 오퍼레이션을 여럿 정의하는 것이 메소드 오버로딩이다.