

IOS 스쿨 양대성

객체지향 프로그래밍

객체지향 프로그래밍에서 가장 많이 나오고 이해를 해야하는 추상화는 객체에서 공통된 속성과 행위를 추출하는 것을 추상화라고합니다.

추상화 과정은 실세계의 상황을 컴퓨터에게 변환시켜주는 것으로 실세계의 상황을 간결하고 명확하게 모델링을 하지 않는다면 프로젝트가 실패할 수도 있기 때문에 분석의 중요성을 강조하는 것이다.

자료 추상화를 이용하여 소프트웨어 객체를 정의하기 위해서는 네가지 요소가 언급이 되는데 자료의 정의, 연산자의 정의, 공리, 예외 처리입니다.

자료의 정의란 대상이 되는 소프트웨어 객체를 컴퓨터 내부에서 표현하고자 하는 자료 구조로 정의하는 것을 말한다. 연산자의 정의는 소프트웨어 객체가 제공하는 함수의 명칭과 시그니처를 밝히는 부분입니다. 그리고 공리란 연산자의 정의에서 나열된 함수들이 본체의 구현시에 알고리즘에 반영 되는 것을 말합니다. 마지막으로 예외 처리란 함수의 실행시에 발생할 수 있는 예외의 종류를 나열하는 부분입니다.

이와같이 추상화는 자료들을 상세한 정보는 무시하고 필요성에 의해 꼭 필요한 자료들로만 구성하는 것입니다.

은닉화란 캡슐화와 뎀레야 뎀 수 없는 사이라고 볼수 있습니다.

두개의 의미는 비슷합니다. 자료를 숨기는 것에는 의미가 똑같지만 은닉화는 주요사항이 아예 드러내지 않는 것을 말하는 것이고 캡슐화는 중요한 자료는 남겨두고 외부에서 그것을 사용할 수 있겠끔 해주는 것입니다.

은닉화는 말그대로 private 을 강조하는 것으로 외부에서는 내부에서 데이터에 어떤 값이 있는지 그리고 수정하는 변화가 보이지가 않습니다..

다형성이란 하나의 가치를 가지고 여러형태로 그 가치가 가능하다는것을 보여주는 것입니다. 예를 들어 만원 짜리 라는 가치가 있으면 만원의 이하나의 가치를 만원짜리 한장으로도 할 수가 있고 5 천원짜리 2 장이 될 수가 있고 천원짜리가 10 장이 될 수가 있습니다.

그래서 실제 동작은 다르지만 개념적으로는 동일한 작업을 하는 멤버 함수들에 똑같은 이름을 부여 할 수있습니다.

그리고 이다형성은 상속과 아주 깊은 관계에 속하고 있어서 캡슐화와 은닉화 처럼 객체지향의 가장 중요한 특징중의 하나입니다.

상속성이란 기존의 코드를 재활용 하는 기법으로 이미 작성된 클래스를 이어 받아서 새로운 클래스를 만드는 방법입니다.

새로운 종류의 서브클래스들을 작성하면 기존의 클래스들의 내장된 기능들을 그대로 받아서 쓸 수가 있습니다.