

* 절차지향언어(Procedure Oriented Language)

- Procedure, 순차적, 계산중심

* 객체지향언어(Object Oriented Language)

- 객체를 모델링하여 객체들의 상호작용으로 서술

* 객체지향언어 VS 절차지향언어

- 절차지향은 데이터를 중심으로 프로시저를 구현한다. 객체 지향은 기능 중심으로 구현 한다.
- 구조체와 클래스

자동차를 만들기 위해서는 엔진, 핸들, 바퀴, 의자, 엑셀, 브레이크 등등...

많은 부품들이 있어야 합니다. 기존 방식 절차지향적 관점으로 본다면 자동차를 만들기 위해서는 어느 한 곳을 기점으로 순서대로 만들어가야 합니다.

엔진 -> 차체 -> 핸들 -> 의자 -> 바퀴 이런 식으로 만들어가며 이들은 서로 분리되면 안되고 순서가 틀려서도 안되며 하나가 고장 나면 전체 기능이 마비되도록 설계되었다고 가정해봅시다.

또한 이들은 처음에 조립될 때의 부품을 다른 종류의 것으로 대체할수도 없다는 전제도 있다고

한다면! 얼마나 비효율적이고 비생산적일까요? 허나 여기에 객체지향 개념의 방식이 도입되면

상황은 달라집니다. 그러면 일단 제작에 있어서 이들은 순서적이지 않아도 됩니다.

핸들을 먼저 만들든, 바퀴를 먼저 만들든, 엔진을 먼저 만들든 상관이 없어집니다.

이들은 각각 따로 독립적으로 개발되어 나중에 한곳에 모여 자신의 기능만 제대로 발휘하면 되니까요.

실제 자동차를 만드는 과정과 똑같은 원리입니다. 부품들이 결합되어 움직이다 어느 하나가 고장이

나더라도 전체 부품들에 영향을 미치지 않습니다. 고장난 부품만 고쳐주면 될 뿐, 다른 부품들은

아무 영향을 받지 않습니다. 그리고 필요하다면 어느 때든 마음에 안드는 부품을 다른 것으로 교체할

수 있고 더 좋은 것으로 바꿀수도 있습니다.

* Struct VS Class

- Struct : Value
- Class : Reference