

객체: 눈에 보이는 것들 (속성과 기능으로만 이루어져 있음)

=> 상호작용을 통해 이벤트(사건) 발생

1. 클래스에 객체가 가져야 하는 **기능과 속성을 정의한다.**

2. new라는 연산자를 통해 클래스에 정의된 내용대로 생성한다.

객체지향 언어의 4가지 특징

1. 캡슐화(Encapsulation) : 아무나 건들이지 못하게 보호 + 그룹화(단위를 나누기)

=> 외부로부터의 접근을 막기 위해서 (정보은닉 효과가 발생)

=> 기능을 이용한 데이터 간접 접근(매개변수, 반환이라는 개념을 적용)

2. 상속(Inheritance) : 부모의 것을 자신이 물려받는 것 => 자식은 부모와 자식의 코드를 둘다 소유하게 됨

3. 다형성(Polymorphism) : 다양한 형태를 지닐 수 있는 형태 (부모가 자식으로, 자식이 부모로 자유롭게 변화가능)

4. 추상화(Astraction) : 추상적(<-> 구체적) 공통적인 기능들을 뽑아서 만드는 것 (ex. 회원가입(이름, 아이디, 비밀번호 ...))

**현실 객체:** 자신의 속성(값, data)과 기능(동작, 행동)이 다른 것들과 구분되어 식별 가능한 것

**객체 지향 언어:** 모든 사건들 안에서의 각 객체간의 상호작용의 개념을 컴퓨터로 옮겨 놓은 것

**자바 객체:** 클래스에 정의된 내용대로 new 연산자를 통해 메모리 영역에 생성된 것

**class:** 객체의 특성(속성, 기능)에 대한 정의를 한 것

**new 연산자:** new 연산자는 객체를 Heap 메모리 영역에 메모리 공간을 할당해주고 메모리 주소를 반환한 후 생성자를 실행시켜준다.

**추상화:** 유연성을 확보하기 위해 구체적인 것은 제거한다는 의미

=> 작성하려는 클래스의 속성과 기능 중 프로그램에서 **필요한 공통점을 추출하고, 불필요한 부분을 제거하는 과정**

**캡슐화:** 추상화를 통해 정리된 데이터들과 기능을 하나로 묶어 관리하는 기법

=> 데이터의 직접 접근 제한을 원칙!!

=> 클래스 외부에서 데이터의 직접적인 접근을 차단(정보은닉의 효과)

=> 대신 간접적으로 데이터에 접근할 수 있는 메소드를 클래스 내부에 작성한다.

**캡슐화 원칙:** 객체의 속성 직접 접근 차단

1) 속성을 나타내는 필드에다가 private

2) 간접 접근 방법

heap 영역 할당 시 변수/요소가 빈칸으로 있을 수 없다.

-> 기본값이 대입이 되어있다.