

3. Object와 Class 이해하기

자바는 OOP(Object Oriented Programming) 언어이다.
 해서 Java를 공부하려면 먼저 object와 class를 이해해야 한다.

실세계에 존재하는 Object(객체, 대상, 물체)를 프로그램에 반영하려면,
 Object를 프로그램에 적용할 수 있는 형태로 변형시켜야 한다.
 이런 작업을 추상화(abstract)라 한다.

추상화는 Object를 표현함에 있어서
 Object의 주된 특징은 부각시키고 나머지는 과감하게 생략하는 것이다.(단순화 작업)
 Object는 '명사와 동사'를 이용해서 특징을 추출할 수 있다. 이를 전산에서 추상화라 한다.

예를 들어 '집'이라는 Object를 명사와 동사로 추출해 보자.

명사 : 주인이름, 층수, 평수, 건축년도, 창문수,
 동사 : 구입하다, 팔다, 청소하다, 고치다, 세를 놓다,

여기서

명사는 그 Object의 속성(property)을 나타내고
 동사는 그 Object와 관련된 행위(method)를 나타내는데 사용되었음을 알 수 있을 것이다.

이렇게 명사와 동사로 구분하면 '집'이라는 Object를 쉽게 프로그램에 반영할 수 있게 된다.

하나 더 예를 들어보자.

'계좌'라는 Object를 추상화하면,

명사 : 계좌명, 계좌번호, 비밀번호, 잔고, 이율, 이체한도,
 동사 : 입금하다, 출금하다, 조회하다, 이체하다,

추상화에 대한 개념을 충분히 이해하였으면,
 다음 단계는 '어떻게 프로그램에 적용하느냐?' 이다.

다음은 추상화한 '계좌'를 자바코드로 구현한 것이다.

```
public class 계좌
{
    private String 계좌번호 = "123-45-567890";    //속성(멤버변수)
    private String 비밀번호 = "1234";
    private long 잔고 = 1000;
    private float 이율 = 6.02F;

    public void 입금하다(long 금액)                //메서드(멤버함수)
    {
        잔고 = 잔고 + 금액;
    }
    public void 출금하다(long 금액) { 잔고 = 잔고 - 금액; }
    public long 조회하다() { return 잔고; }
    public void 이율변경(float 새이율) { 이율 = 새이율; }
}
```

먼저, 클래스는 사용자가 정의하는 자료형이다.

클래스는 동일한 구조의 객체를 여러 개 생성하기 위해 제공되는 틀(template)과 같다. 같은 클래스에서 생성되는 객체는 동일한 구조의 속성과 메서드를 가진다.

```
계좌 홍길동 = new 계좌(계좌번호, 비밀번호, .....)
```

여기서, 계좌 : 클래스 이름
홍길동 : 생성되는 객체 이름(객체참조변수)
new : 객체를 생성하는 연산자

결론적으로, 객체지향언어에서 객체를 생성하여 사용하려면 먼저 클래스를 만들어야 한다.

모든 객체는 클래스로부터 생성되기 때문이다.

객체가 생성되는 과정을 **실체화**(instantiation)라 하고, 생성된 객체를 **인스턴스**라 한다.

객체와 인스턴스(instance)는 같은 뜻으로 사용될 수 있다.

먼저, 자바 응용프로그램을 하나 살펴본다.

```
import java.lang.*;           //java.lang 패키지의 모든 클래스를 사용할 수 있다.
class A{
    protected static int kor;  //국어 점수
    protected static int eng;  //영어 점수
    public void set(int k, int e){ //((국어, 영어) = (70, 90)으로 대입
        kor = k;
        eng = e;
    }
}
class B extends A{           //상위클래스 : A, 하위클래스 : B
    private int sum;
    public int add(){         //합을 구한다.
        sum = kor + eng;
        return sum;
    }
}
class C extends A{
    private float avg;
    public float average(){   //평균을 구한다.
        avg = (kor + eng) / 2f;
        return avg;
    }
}
public class Test{
    public static void main(String args[]){
        A a = new A();
        B b = new B();
        C c = new C();

        a.set(70, 90);        [실행결과]
        System.out.println(b.add());    160
        System.out.println(c.average()); 80.0
    }
}
}
```

- 관례적으로 클래스 이름의 첫 문자는 대문자로, 메서드는 소문자로 정의한다.
- 메서드 main()은 C의 함수 main()과 같은 역할을 한다.
- 매개변수 args[]는 프로그램 실행시 명령어 행에서 String형으로 자료를 전달받는다.
- 사용자가 명령어 행에 입력한 인자의 순서에 따라 배열에 저장된다.