PRESENTATION

9주차 주제 객체지향

객체 지향 프로그래밍 ~캡슐화~

by mun

객체 지향 프로그래밍

Object-Oriented Programming, OOP

컴퓨터 프로그래밍의 패러다임 중 하나
컴퓨터 프로그램을 명령어의 목록으로 보는 시각에서 벗어나
여러 개의 독립된 단위, 즉 "객체"들의 모임으로 파악하고자 하는 것

-위키피디아

장점

프로그램을 유연하고 변경이 쉽게 만들기 때문에 대규모 소프트웨어 개발에 많이 사용
소프트웨어 개발과 보수를 간편하게 하며, 보다 직관적인 코드 분석을 가능

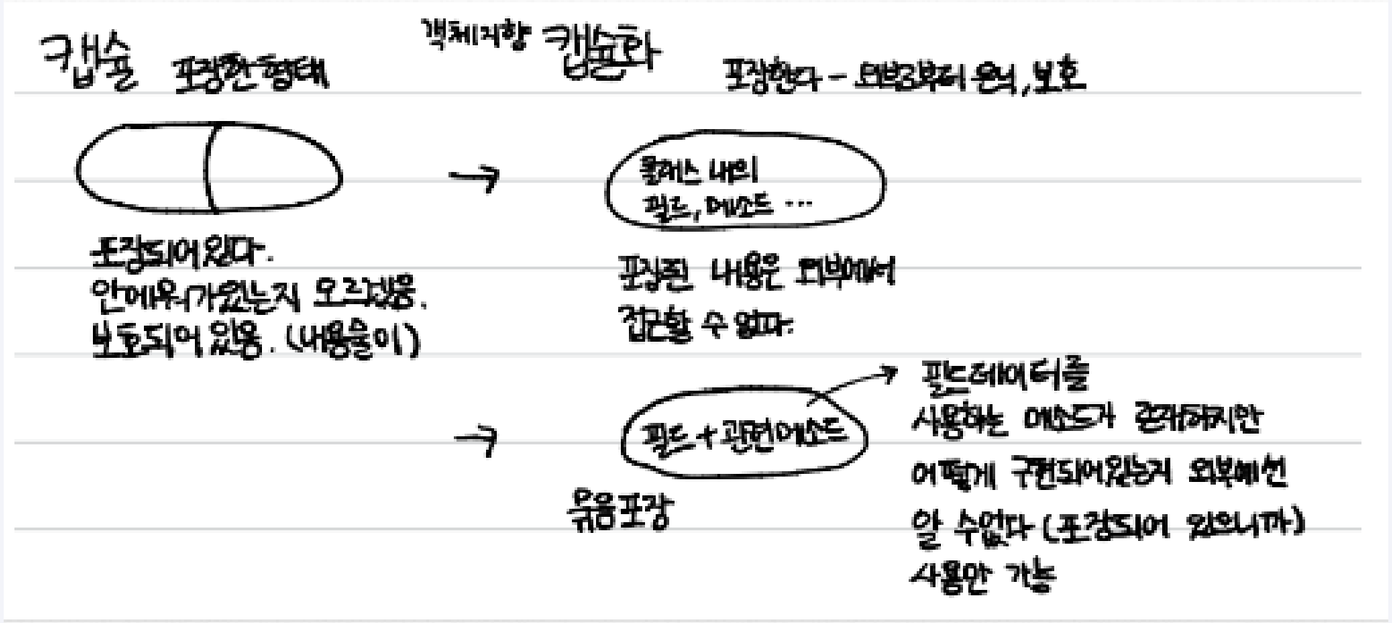
특징

추상화/상속/다형성/캡슐화

캡슐화

캡슐화(영어: encapsulation)는 객체 지향 프로그래밍에서 다음 2가지 측면이 있다:
1. 객체의 속성(data fields)과 행위(메서드, methods)를 하나로 묶고,
2. 실제 구현 내용 일부를 내부에 감추어 은닉한다.

-위키피디아



속성과 행위를 하나로 묶는다

```
class Chef {
    private String name;
    public Chef(String name) {
        this.name = name;
    }
    public void cook() {
        System.out.println(name+" 요리사가 요리를 한다.");
    }
}
```

요리사의 속성

요리사의 행위

실제 내부를 감추어 은닉한다

샌드위치 만들기

```
class Sandwich {
   public void bake() {
        System.out.println("빵을 굽는다.");
    }
    public void spread() {
        System.out.println("잼을 바른다.");
    }
    public void cut() {
        System.out.println("칼로 자른다");
    }
 }
```

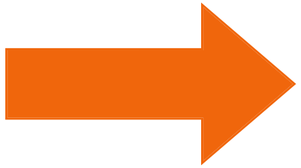
```
public class Main {
   public static void main(String[] args) {
        Chef chef = new Chef("김셰프");
        chef.cook_sand();
   }
}
```

```
class Chef {
    private String name;
    public Chef(String name) {
        this.name = name;
    }
    private Sandwich sand = new Sandwich();
    public void cook_sand() {
        System.out.println(name+" 요리사의 샌드위치 만들기");
        sand.bake();
        sand.spread();
        sand.cut();
    }
}
```

실제 내부를 감추어 은닉한다

만드는 과정이 추가됐다면?

```
class Sandwich {
   public void bake() {
        System.out.println("빵을 굽는다.");
    }
    public void spread() {
        System.out.println("잼을 바른다.");
    }
    public void cut() {
        System.out.println("칼로 자른다");
    }
}
```



```
class Sandwich {
   public void bake() {
        System.out.println("빵을 굽는다.");
    }
    public void spread() {
        System.out.println("잼을 바른다.");
    }
    public void put() {
        System.out.println("토핑을 올린다.");
    }
    public void cut() {
        System.out.println("칼로 자른다");
    }
}
```

실제 내부를 감추어 은닉한다

```
class Sandwich {
   public void bake() {
        System.out.println("빵을 굽는다.");
    }
    public void spread() {
        System.out.println("잼을 바른다.");
    }
    public void cut() {
        System.out.println("칼로 자른다");
    }
}
```

```
class Sandwich {
   public void bake() {
        System.out.println("빵을 굽는다.");
    }
    public void spread() {
        System.out.println("잼을 바른다.");
    }
    public void put() {
        System.out.println("토핑을 올린다.");
    }
    public void cut() {
        System.out.println("칼로 자른다");
    }
}
```

과정을 세세하게 알아야 한다

```
    private Sandwich sand = new Sandwich();
    public void cook_sand() {
        System.out.println(name+" 요리사의 샌드위치 만들기");
        sand.bake();
        sand.spread();
        sand.cut();
    }
}
```

```
public void cook_sand() {
    System.out.println(name+" 요리사의 샌드위치 만들기"
    sand.bake();
    sand.spread();
    sand.put();
    sand.cut();
}
```

실제 내부를 감추어 은닉한다

```
class Sandwich {
    public void bake() {
        System.out.println("빵을 굽는다.");
    }
    public void spread() {
        System.out.println("잼을 바른다.");
    }
    public void put() {
        System.out.println("토핑을 올린다.");
    }
    public void cut() {
        System.out.println("칼로 자른다");
    }
}
```

```
class Sandwich {
    public void bake() {
        System.out.println("빵을 굽는다.");
    }
    public void spread() {
        System.out.println("잼을 바른다.");
    }
    public void put() {
        System.out.println("토핑을 올린다.");
    }
    public void cut() {
        System.out.println("칼로 자른다");
    }
    public void make() {
        bake();
        spread();
        put();
        cut();
    }
}
```

실제 내부를 감추어 은닉한다

```
class Sandwich {
   public void bake() {
        System.out.println("빵을 굽는다.");
    }
    public void spread() {
        System.out.println("잼을 바른다.");
    }
    public void put() {
        System.out.println("토핑을 올린다.");
    }
    public void cut() {
        System.out.println("칼로 자른다");
    }
}
```

```
class Sandwich {
   public void bake() {
        System.out.println("빵을 굽는다.");
    }
    public void spread() {
        System.out.println("잼을 바른다.");
    }
    public void put() {
        System.out.println("토핑을 올린다.");
    }
    public void cut() {
        System.out.println("칼로 자른다");
    }
    public void make() {
        bake();
        spread();
        put();
        cut();
    }
}
```

내부 과정은 알 필요 없다

```
public void cook_sand() {
    System.out.println(name+" 요리사의 샌드위치 만들기");
    sand.bake();
    sand.spread();
    sand.put();
    sand.cut();
}
```

```
public void cook_sand() {
    System.out.println(name+" 요리사의 샌드위치 만들기");
    sand.make();
}
```

실제 내부를 감추어 은닉한다

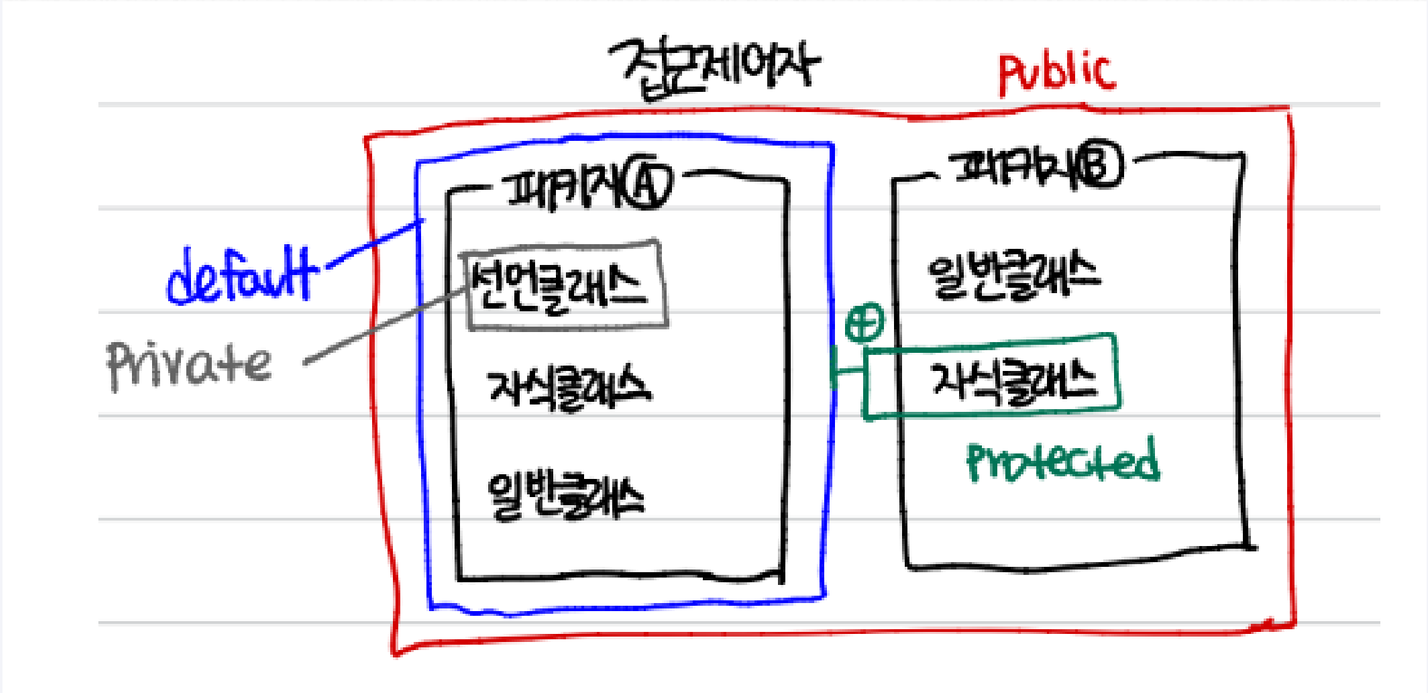
```java
class Sandwich {
    public void bake() {
        System.out.println("빵을 굽는다.");
    }
    public void spread() {
        System.out.println("잼을 바른다.");
    }
    public void put() {
        System.out.println("토핑을 올린다.");
    }
    public void cut() {
        System.out.println("칼로 자른다");
    }
    public void make() {
        bake();
        spread();
        put();
        cut();
    }
}
```

public -> private

```java
class Sandwich {
    private void bake() {
        System.out.println("빵을 굽는다.");
    }
    private void spread() {
        System.out.println("잼을 바른다.");
    }
    private void put() {
        System.out.println("토핑을 올린다.");
    }
    private void cut() {
        System.out.println("칼로 자른다");
    }
    public void make() {
        bake();
        spread();
        put();
        cut();
    }
}
```

접근하지 못하도록 은닉화

접근 제한자



public: 모든 접근
protected: 같은 패키지 또는 상속관계
default: 같은 패키지
private: 현재 객체

접근 제한자를 사용해 적절하게 은닉화

결합도를 낮추기 위해
캡슐화 중요!

~캡슐화를 의식하며 설계해요~

감사합니다

참고 : 위키백과, https://dalmun.tistory.com/7