

8. abstract 추상메소드

-예를들어 본사 가격이 존재하고, 각 지점마다 다르게 가격을 측정할수도 있음 하지만 본사가 실수로 빼먹을수 있는 요소들이 있기 때문에 HeadQuater에서 강제로 오버라이드 할수있게끔 수정(강요넣어 놓기)

```
Public void kimchid(){}
```

Public abstract void kimchi(); 이런식으로 abstract를 붙여주기 그럼 오버라이드를 반드시 해야 오류가 안남

*추상 메소드가 하나 이상이면 클래스도 추상이니, 클래스명 앞에도 abstract 써줘야됨

상속받는 애들 override 다 해주어야함 (이 과정이 귀찮으면 오류난 부분에 마우스 가져다 댄 후 한번에 override 해주기

Maintest 에서 추상클래스는 객체 못만드니 new OO 할 수 없고, 생성함수만 가져다 쓸 수 있음

변수가 만들어지긴 함. 객체도 만들려면 메인클래스에서 메소드 만들어주면 됨

*추상클래스 : 상수, 변수, 생성자, 추상메소드, 일반메소드

*인터페이스 : 상수, 추상메소드

*Final 쓸 때 static 쓸때와 안쓸 때 구분하기

9. 인터페이스

인터페이스는 class가 아님 추상메소드와 상수만 이용 가능

클래스가 클래스에게 물려주는 것 '상속' extends

인터페이스가 인터페이스에게 물려주는 것 '상속' extends

인터페이스가 클래스에게 물려주는 것 implement

상속은 받는 사람이 구현할 필요가 없지만, 인터페이스를 implement받을 경우 받은 아이가 구현해줘야함

인터페이스에서 static final 생략가능 . (class에선 생략불가능)

인터페이스 1개만들어서 여러 클래스에 뿌릴 수 있음

하나의 기준메뉴가(인터페이스) 식당마다(클래스여러개) 다른 즉 부품화 시키는 것

인터페이스 여러 개 만들어서 하나의 클래스가 받을수도 있음 (다형성을 뜻하게 됨)

```
public interface HeadQuarterStore {  
  
    public void kimch();  
    public void bude();  
    public void bibib();  
    public void sunde();  
    public void gonggibab();  
    public String getStr();  
}
```

// 추상메소드와 변수 설정만 가능

// String 데이터값의 getStr메소드 만들어주기

//각각의 스토어에 getter라는 메소드를 만들어 놓아야 함. 예를 공통으로 묶어 쓰려면 인터페이스에다가 메소드 만들어 놓으면 됨

```
public class Store1 implements HeadQuarterStore {  
  
    private String str;  
    public Store1(String str) {  
        this.str = str;  
    },  
}
```

headQuarterStore 이름을 가진 인터페이스로부터 Store1은 implement됨

```
@Override  
public void gonggibab() {  
    System.out.println("공기밥 - 1000원");  
}  
  
public String getStr() {  
    return str;  
}
```

@override 인터페이스에 만들어 놓은 함수를 오버라이드 한 후 store1에서 구현 해 낸 것

String getStr() = 인터페이스에 만들어 놓은 String형태를 return값으로 메소드 만들어준 것

```
public static void main(String[] args) {  
    HeadQuarterStore [] store = {new Store1("===1호점==="),  
                                   new Store2("===2호점==="),  
                                   new Store3("===3호점===")};  
    //HeadQuarterStore로 객체는 만들지 못하지만 대표 이름으로 오는 건 가능함
```

인터페이스인 headQuarterStore은 객체생성을 못함

즉 HeadQuarterStore store = new HeadQuarterStore(); 이건 못하지만

대표이름으로 앞에 쓴 후 각각의 store 객체를 만드는 건 가능

For문 혹은 확장for문을 통해 위 store들 출력해주기