추상 abstract

public abstract int b();

구체적인 로직은없고 형식만 있다

오버라이딩 필수

※추상적으로 정의할 테니(선언해 놓을테니), 객체를 사용할 사용자가 꼭 재정의(overriding) !

. 추상클래스의 문법(※추상메소드에서는 정의만 합니다. 구현하지 않습니다)

① abstract(추상클래스 및 추상 메소드를 선언하는 예약어)이용

public abstract class ClassName {

}

② 추상클래스에는 하나 이상의 추상 메소드가 포함. 추상메소드는 정의만 하고 구현은 하지 않습니다.

③ 추상메소드에는 메소드의 선언부만 있고 실행부(구현부, Body)는 없습니다

abstract 리턴타입 methodName([매개변수])

④ 추상 클래스에서는 메소드 선언만 하고 실제로 구현은 상속받는 클래스에서 한다.

기능는 자식 클래스에게 위임 - 추상클래스에서 정의된 추상적인 기능은 하위 클래스에서 상세 구현

⑤ 클래스의 프레임만 구성. 직접 객체 생성 불가능(abstract는 인스턴스화를 금지하는 키워드)

final(제한자)

※ 추상클래스란 "추상적으로 정의할 테니(선언해 놓을테니), 사용자가 꼭 메소드를 재정의(overriding) 하세요"란 의미입니다.
그와는 정반대되는 개념이 있어 소개합니다.

⑴ 클래스 앞에 붙일 경우 : 상속 금지

```
public final class Test {

}
```

⑵ 멤버 메소드 앞에 붙일 경우 : 오버라이딩 금지

```
public final void print(){

}
```

⑶ 멤버변수 앞에 붙일 경우 : 상수화된다(변경금지).

```
public final int PORT_NUMBER = 80;
```

```java
// 추상 클래스 대충 만들어 놓고 상속받아서 너가 알아서만들어
// HeadQuarterStore hstore = new HeadQuarterStore("본사); 객체 생성 불가
// hstore.kimchi();
public abstract class HeadQuarterStore {
    private String str;// 본사 , 매장 1호점

    public HeadQuarterStore(String str) {
        // super();// 상속받은게 없으니 오브젝트에서 상속

        this.str = str;

    }

    // 메소드 구혀은 없고 선언만 되어 있는 메소드 : 추상 메소드 (상속받은 클래스에서 오버라이드)
    // public void kimchi(); //메소드 구현은 없고 선언만 되어 있는 메소드 : 추상 메소드라고한다
    public abstract void kimchi();

    public abstract void bude();
```