

# Inheritance(extends) I - Access Control, Static Member

## Access Control Specifier - Protected

```
package default;

class AAA {
    int num1;
    Protected int num2;
}

class BBB extends AAA {
    BBB() {
        num1 = 10; // 동일 패키지로 묶였기 때문에 default 접근 가능!
        num2 = 10; // 상속받았으므로 protected 접근 가능! (다른 Package이더라도 접근가능)
    }
}
```

---

Private Member도 상속이 된다. 다만 접근은 불가능하다!

```
class Accumulator {
    private int val;

    Accumulator (int init) { val = init; }

    Protected void accumulate(int num){
        if(num<0)
            return;
        Val += num;
    }
    protected int getAccVal() { return val; }
}

class SavingAccount extend Accumulator {

    상속을 받았더라도 하위 Class는 val의 접근은 불가!
    그래서 protected Method를 통해 간접 접근을 하고 있다.

    private int val; // 다만 instance를 생성하면 존재한다.

    public SavingAccount(int initDep){
        super(initDef);
    }

    public void saveMoney(int money){
        accumulate(money);
    }
    Public showSaveMoney(){
        System.out.print("지금까지의 누적 금액 : ");
        System.out.println(getAccVal());
    }
}
```

---

## Static Member의 상속

Static Method도 상속이 된다.

```
class Adder {
    public static int val = 0;
    public void add(int num) { val += num; }
}

class AdderFiend extend Adder {
    public void friendAdd(int num) { val += num; }
    public void showVal() { System.out.println(val); }
}
```

---