23-12-21

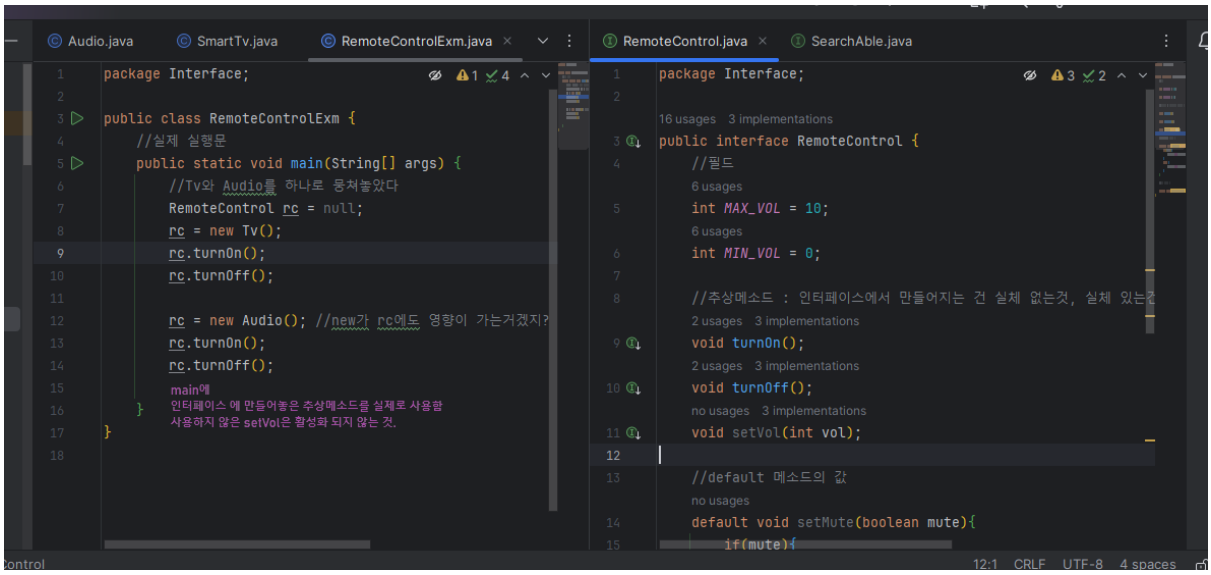
- Java interface

- RemoteControl / Audio / Tv 예제

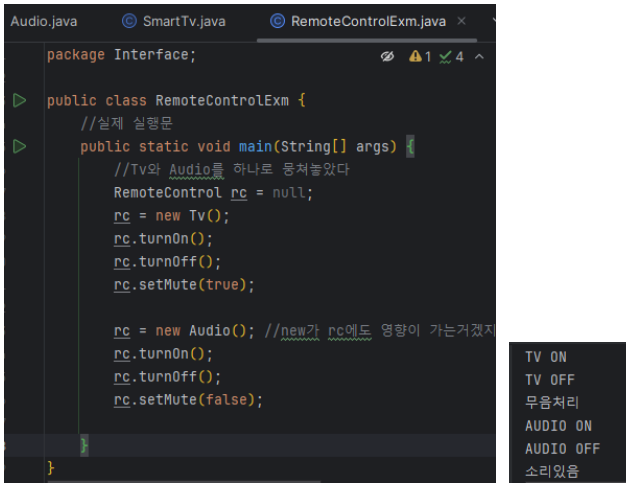
인터페이스에서는 다 공개다.

Class는 다중상속이 불가했는데,

Interface는 다중인터페이스로 가능하다(두개의 인터페이스를 하나의 클래스로 사용할수있다는 것

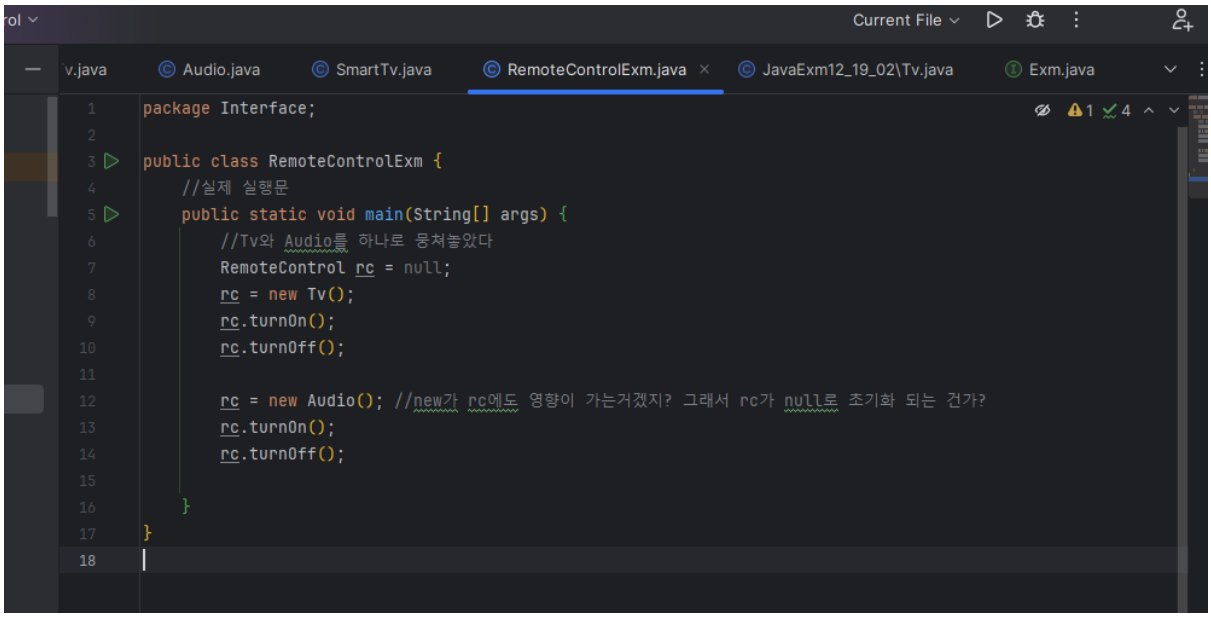


실행은 무조건 메인함수에서!



```
Audio.java   SmartTv.java   RemoteControlExm.java

package Interface;

public class RemoteControlExm {
    //실제 실행문
    public static void main(String[] args) {
        //Tv와 Audio를 하나로 뭉쳐놓았다
        RemoteControl rc = null;
        rc = new Tv();
        rc.turnOn();
        rc.turnOff();
        rc.setMute(true);

        rc = new Audio(); //new가 rc에도 영향이 가는거겠지
        rc.turnOn();
        rc.turnOff();
        rc.setMute(false);

    }
}
```

```
TV ON
TV OFF
무음처리
AUDIO ON
AUDIO OFF
소리있음
```



```
Current File

v.java   Audio.java   SmartTv.java   RemoteControlExm.java   JavaExm12_19_02\Tv.java   Exm.java

1  package Interface;
2
3  public class RemoteControlExm {
4      //실제 실행문
5      public static void main(String[] args) {
6          //Tv와 Audio를 하나로 뭉쳐놓았다
7          RemoteControl rc = null;
8          rc = new Tv();
9          rc.turnOn();
10         rc.turnOff();
11
12         rc = new Audio(); //new가 rc에도 영향이 가는거겠지? 그래서 rc가 null로 초기화 되는 건가?
13         rc.turnOn();
14         rc.turnOff();
15
16     }
17 }
18
```

RemoteControl 이라는 interface를 rc라고 부를 거다 거기에 null을 넣은 것(null넣지 않아도 작동됨)
그리고 rc를 new TV()라는 class로 정의 하면 새롭게 정의된 것이다
그리고 그 안에 메소드를 실행한 것 -> 그러면 그 안에 출력 값 설정해 놓은 것들이 출력됨

```
public class VariableTest2 {

    public static void main(String[] args) {

        VariableTest vt = new VariableTest();
        vt.f();
        vt.f();
        vt.f2();
        vt.f();
        vt = new VariableTest();
        vt.f(); // 출력값 1

    }
```

main에서 전역 변수가

vt = new VariableTest(); 를 만나는 순간 다시 읽는다.

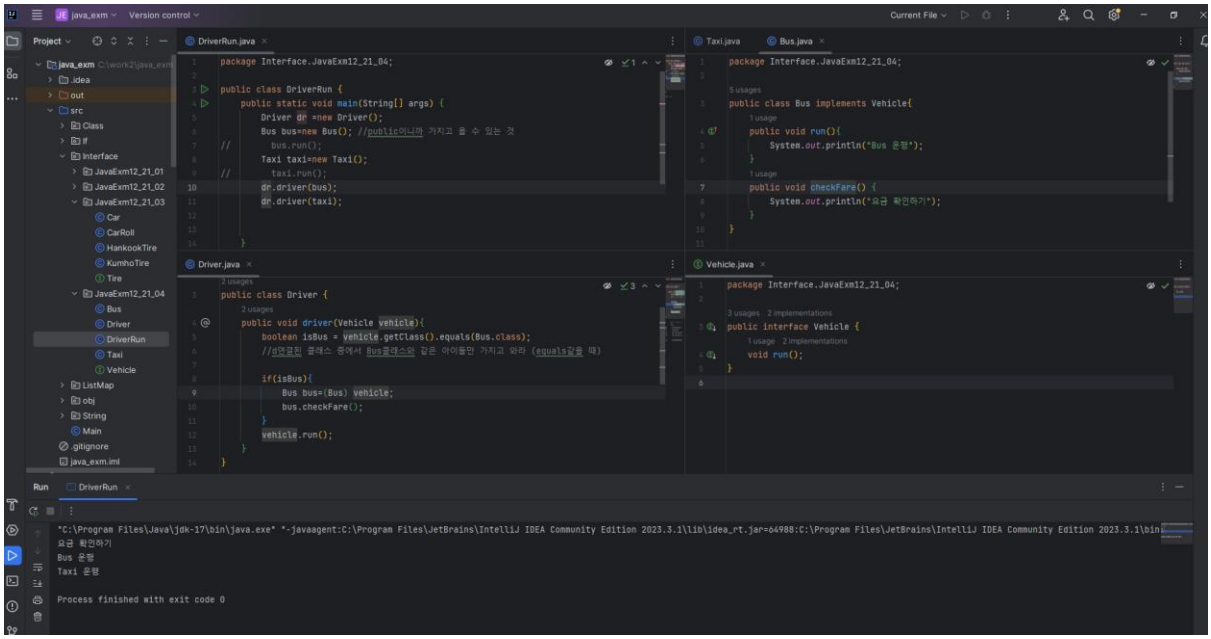
초기화 되어 다음 vt.f(); 값을 출력했을 때 결과는 1이 되어 나온다.

만약 new를 만나도 그대로 그 값을 유지하고 싶다면? -> static

- Car / Tire 예제

//변수의 다형성 인터페이스 상에서 진행

- Driver / Taxi / Bus 예제



//인터페이스 연결해줘야 하고, 드라이버에 불러올 수 있는 상태로 만들어 놓아야 출력이 된다

getValue

getter

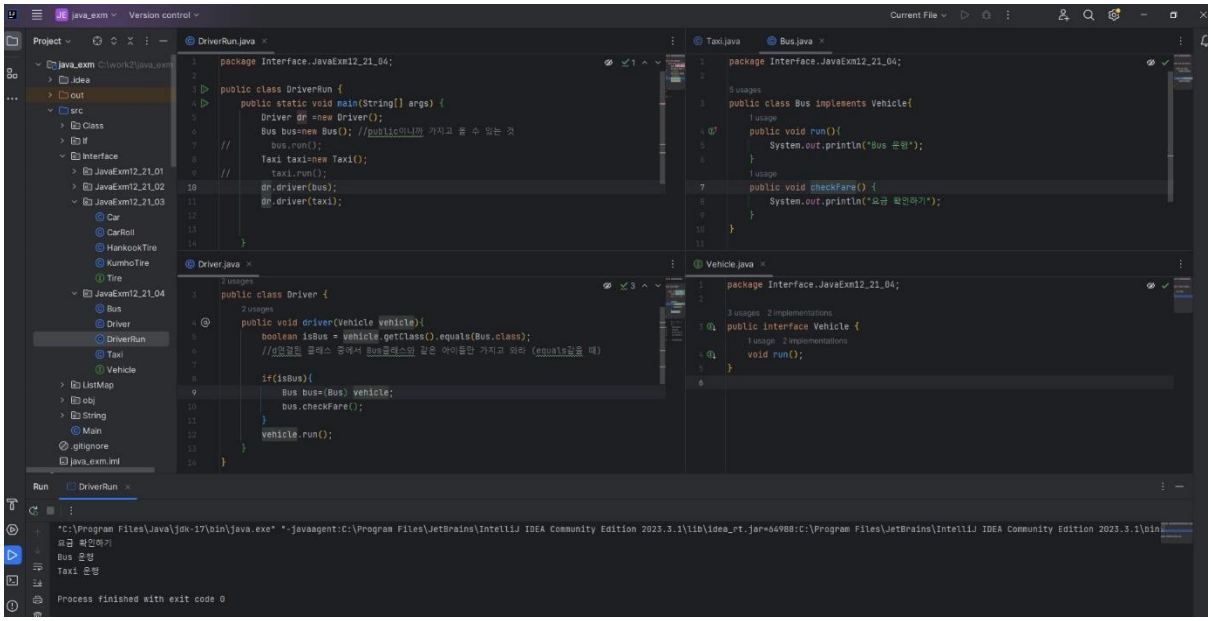
setter

- JavaExm12_21_03 > Car / CarRoll / HankookTire / Tire

tires[i].roll();

```
2 usages
void run(){
    for (int i=0;i<tires.length;i++){

        tires[i].roll();
          new HankookTire().roll();
    }
```

인터페이스를 변수처럼 사용



- JavaExm12_21_05

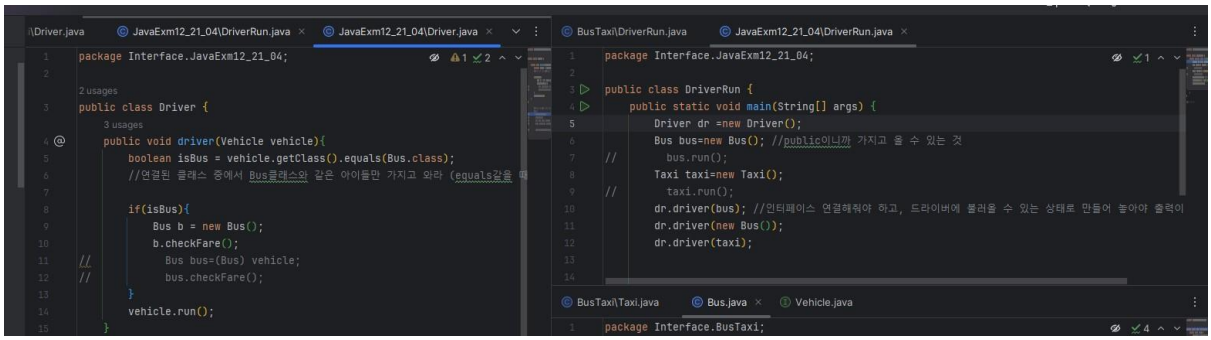
JavaExm12_21_04\Taxi.java Exm.java

```
package FileIo.JavaExm12_21_05;

import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;

public class Exm {
    public static void main(String[] args) throws IOException {
        InputStream in = System.in; //문자열을 입력받을 때 사용
        InputStreamReader reader=new InputStreamReader(in); //문자열 형태
        char[] a =new char[3];
        reader.read(a);

        System.out.println(a);

    }
}
```

```
dkssud
dks
```

//인터페이스 연결해줘야 하고, 드라이버에 불러올 수 있는 상태로 만들어 놓아야 출력이 된다

- JavaExm12_21_04




```
if(isBus){
//      Bus b = new Bus();
//      b.checkFare();
//      Bus bus = (taxi) vehicle;

        Bus bus=(Bus) vehicle;
        bus.checkFare();
    }
    vehicle.run();
```

bus로 형변환 해준 것.