

```
package MyArrays;
```

MyArrays클래스 만들어 보기

//Arrays를 간단하게 구현하였다.

```
public class MyArrays {
```

```
    public static void sort (Object[] arr) {        //모든 객체타입을 정렬하기 위해서 Object[]
        for( int i=0 ; i< arr.length-1 ; i++) {
            for( int j=i+1 ; j< arr.length ;j++) {

                MyComparable m =(MyComparable)arr[i]; //MyComparable 형변환 시도함
                // MyComparable로 형변환을 해야 compareTo 매서드를 사용할 수 있다.
                if( m.compareTo(arr[j]) >0) {
                    Object tmp;
                    tmp= arr[i];
                    arr[i] = arr[j];
                    arr[j]=tmp;
                }
            }
        }
    }
}
```

//정렬기능 (이것을 먼저 테스트해 본다)

```
public static void sort (Object[] arr , MyComparator c ) {
    // c가 인터페이스라면 구현한 객체를 매개변수로 받겠다는 의미
    for( int i=0 ; i< arr.length-1 ; i++) {
        for( int j=i+1 ; j< arr.length ;j++) {

            // 두 객체를 비교한 값이 양수이면 정렬을 수행하겠다.
            if( c.compare(arr[i], arr[j]) >0) {
                Object tmp;
                tmp= arr[i];
                arr[i] = arr[j];
                arr[j]=tmp;
            }
        }
    }
}

}
```

```
//정렬의 기준을 인터페이스로 만든다.
public interface MyComparable {
    int compareTo(Object obj);
}
```

```
Score0[] arr = new Score0[3];
arr[0]= new Score0("이길동",90,71);
arr[1]= new Score0("김길동",70,99);
arr[2]= new Score0("박길동",90,89);
```

```
class A implements MyComparator{
    @Override
    public int compare(Object o1, Object o2) {
        Score0 s1 =(Score0) o1
        Score0 s2 = (Score0) o2

        return s2.getEng()- s1.getEng();
    }
}
```

사용예시

```
MyArrays.sort( arr , new A() );
```

객체배열

인터페이스를 구현한 객체

라이브러리를 사용하는 쪽에서 정렬의 기준을 작성하여 준다