

```
package Twelfth;
```

```
// 인터페이스 작성법
```

```
// 1. public 을 적고 interface를 적는다.
```

```
// 2. 인터페이스로 만들 이름을 적는다.
```

```
// 3. 내부에 다형성을 가지게 만들 매서드 프로토타입을 작성한다.
```

```
// 다형성?
```

```
// 누가 이것을 구현하냐에 따라 결과가 달라진다.
```

```
↓ public interface Comparable {  
    // 현재 이부분은 단순히 프로토타이핑만 한 것이다.  
    // 프로토타이핑이 무엇이냐?  
    // 매서드의 리턴타입, 매서드이름, 매서드 인자의 입력만 정의한 형태  
    ↓ public int compareTo(Object other , int order);  
}
```

1.인터페이스 만들기.

```
//클래스 MyStrcmp에 만든 인터페이스 Comparable를 implement
public class MyStrcmp implements Comparable {

    // 변수 할당
    private double area;

    //생성자를 만든다.
    public MyStrcmp(double a){
        area = a;
    }
}
```

MyStrcmp에 변수, 생성자 만듦.

```
// String compare with Interface
public class StrcmpInterfaceTest {
    public static void main(String[] args) {
        MyStrcmp ms1 = new MyStrcmp( a: 105.3);
        MyStrcmp ms2 = new MyStrcmp( a: 87.3);

        StrcmpTest st = new StrcmpTest( a: 105.3);

        // 뭐야 왜 MyStrcmp 를 굳이 Object 로 바꾸고
        // 다시 이것을 형변환해서 MyStrcmp 로 만들었지?

        if(ms1.compareTo(ms2, MYSTRCMP)>0){
            System.out.println("ms1이 ms2보다 더 크다.");
        } else if(ms1.compareTo(ms2, MYSTRCMP)<0){
            System.out.printf("ms1은 ms2보다 더 작다.");
        } else {
            System.out.printf("ms1과 ms2는 같다.");
        }
    }
}
```

생성자를 통해
ms1= 105.3,
ms2= 87.3값을 할당

유틸리티 클래스는
왜 따로
생성해야되나??

m1. '.'의 역할이 정확히 뭔가?
. 이하인
compareTo(ms2,MYSTRCMP)>0 해라?

```
// Object ?? 이건 뭐야?  
// Object 는 모든 정보를 아우르고 있는 가장 거대한 집합이다. int도 될수있고 double도 될수있고 모든..  
// 여기서 모든 정보란? 우리가 사용하는 모든 클래스 등등을 의미한다.  
// 그러므로 타입 캐스팅(형 변환)을 통해  
// 어떤 데이터든 object 형태로 받을 수 있다.  
// 만약 내용을 들은 사람이 C 개발자라면 void * 생각하면 됩니다.
```

```
public int compareTo(Object otherObj , int ORDER){
```

```
    int res = 0;
```

```
    switch (ORDER){
```

```
        case MYSTRCMP:
```

```
            res = processMyStrcmp(otherObj);
```

```
            break;
```

```
        case STRCMPTEST:
```

```
            res =processStrcmpTest(otherObj);
```

```
            break;
```

```
        default:
```

```
            break;
```

```
    }
```

위의 . 이하였던

CompareTo(ms2,MYSTRCMP)를
수행하면

order=> mystrcmp case로

res = processMyStrcmp(otherObj)

여기서 otherObj는 ms2가 맞나???

processMyStrcmp(otherObj) 매서드를
진행.

```
public int processMyStrcmp(Object otherObj) {
    MyStrcmp other = (MyStrcmp) otherObj;

    return processCompare(other);
}
```

res = processMyStrcmp(otherObj)
 otherObj가 위에서 받은 ms2가 맞다면
 MyStrcmp 'other' = (MyStrcmp) ms2;
 인가?
 그렇다면 리턴받는것은
 ProcessCmpare((MyStrcmp)ms2);가
 되는건가?
 ()는 왜 하는건가요?



```
public int processCompare(MyStrcmp other) {
    if(this.area < other.area) {
        return -1;
    } else if(this.area > other.area) {
        return 1;
    } else {
        return 0;
    }
}
```

위에서 받은
 other = (MyStrcmp)ms2 이값이 87.3이고
 this.area 는 ms1 . 이였던 값으로 105.3이다.
 따라서 else if로 들어가서 1값으로 리턴.

```
MyStrcmp ms1 = new MyStrcmp( a: 105.3);
MyStrcmp ms2 = new MyStrcmp( a: 87.3);
```

this.area 가 ms1= 105.3
 이라는 근거 추측...

```
// 변수 할당
private double area;
//생성자를 만든다.
public MyStrcmp(double a){
    area = a;
}

// 뭐야 왜 MyStrcmp 를 굳이
// 다시 이것을 형변환해서 My
if(ms1.compareTo(ms2, M)
```

```
if(ms1.compareTo(ms2, MYSTRCMP)>0){  
    System.out.println("ms1이 ms2보다 더 크다.");  
} else if(ms1.compareTo(ms2, MYSTRCMP)<0){  
    System.out.printf("ms1은 ms2보다 더 작다.");  
} else {  
    System.out.printf("ms1과 ms2는 같다.");  
}
```

compareTo(ms,MYSTRCMP) 값은 1로
받아
ms1 는 1>0 참이 되서
ms1이 ms2보다 더 크다고 출력!?

‘스킬’이라는
인터페이스 생성.

클래스 ‘직업’ 생성 -
‘스킬’ 인터페이스
연결

2차 워로드
skill1,2
전사 스킬2개를
상속

전사,skill
1,2

2차 어쌔신 **skill1,2**
시프 스킬2개를
상속

시프,skill
1,2

2차 위자드 **skill1,2**
위치 스킬2개를
상속

위치,skill
1,2

?? 아무리 생각해도
잘모르겠습니다 ...;