

디지털컨버전스 기반 UI/UX Front 전문 개발자양성 과정(비트캠프)

강사: 이승훈
수강생: 오진욱

Interface

인터페이스

- 개념: 클래스를 작성할때 기본이되는 틀을 제공하면서 다른 클래스 사이의 중간 매개 역할을 하는 다소 추상적 개념, 어떤 객체가 있고 그 객체가 특정한 인터페이스를 사용한다면 그 객체는 반드시 인터페이스의 메소드들을 구현해야한다. 즉, 인터 페이스 내에는 프로토타입의 메소드만을 입력하고 Implements를 하는 클래스에서는 반드시 메소드 세부사항들을 작성해야한다.
- 왜 써야 할까? 프로젝트는 팀작업 많다 -> 인터페이스를 통해 시간과 돈을 절약할 수 있다
- 미리 상의 후 인터페이스를 만들어 프로토타입을 통해 맞춰 개발이 가능함
- 작성법 : `public interface 이름{ 프로토타입의 메소드 }`

Example

- Interface를 통해 String을 비교하기
- (이미지 첨부확인)

```
1  package Twelveth;
2
3  // 인터페이스 작성법
4  // 1. public 을 적고 interface를 적는다.
5  // 2. 인터페이스로 만들 이름을 적는다.
6  // 3. 내부에 다형성을 가지게 만들 매서드 프로토타입을 작성한다.
7
8  // 다형성 ?
9  // 누가 이것을 구현하냐에 따라 결과가 달라진다.
10 public interface Comparable { interface 작성하는 방법,
11     // 현재 이부분은 단순히 프로토타이핑만 한 것이다.
12     // 프로토타이핑이 무엇이나 ?
13     // 매서드의 리턴타입, 매서드이름, 매서드 인자의 입력만 정의한 형태
14     public int compareTo(Object other, int order);
15 }
```

여러 형태의 매서드를 만들 수 있다

Example

```
1 package Twelveth;
2
3 import static Twelveth.Utility.MYSTRCMP;
4 import static Twelveth.Utility.STRCMPTEST;
5
6 // 실질적으로 interface 에 작성한 프로토타이핑한 매서드는
7 // 이것을 implements 하는 클래스에서 작성해줘야 한다.
8 public class MyStrcmp implements Comparable {
9     private double area;
10
11     public MyStrcmp(double a) {
12         area = a; // 만약 105.3 이면
13     }
14     // area a = a = 105.3
15     public int processCompare(MyStrcmp other) {
16         if (this.area < other.area) {
17             return -1; // 만약 a = 105.3
18         } else if (this.area > other.area) {
19             return 1;
20         } else {
21             return 0;
22         }
23     }
24     // getter 이용해서
25     // area를 받는다?
26     public int processCompare(StrcmpTest other) {
27         if (this.area < other.getArea()) {
28             return -1;
29         } else if (this.area > other.getArea()) {
30             return 1;
31         } else {
32             return 0;
33         }
34     }
35     public int processMyStrcmp(Object otherObj) {
36         MyStrcmp other = (MyStrcmp) otherObj;
37
38         return processCompare(other);
39     }
40 }
```

```
53     public int compareTo(Object otherObj, int ORDER) {
54         int res = 0;
55         switch (ORDER) {
56             case MYSTRCMP:
57                 res = processMyStrcmp(otherObj);
58                 break;
59             case STRCMPTEST:
60                 res = processStrcmpTest(otherObj);
61                 break;
62             default:
63                 break;
64         }
65         return res;
66     }
67 }
```

① 만약 ms 2 , MYSTRCMP 이면
이름으로
해석하는 것이
아니야?
②
③
④
전파이커로 이동

Example

```
1 package Twelveth;
2
3 public class StrcmpTest {
4     private double area;
5
6     public StrcmpTest(double a) {
7         area = a;
8     }
9
10    public double getArea() {
11        return area;
12    }
13 }
```

여기는 STR 값은 해당
반가워한 클래스도?

```
1 package Twelveth;
2
3 public class Utility { // Switch Case 4번 Finish
4     public static final int MYSTRCMP = 1;
5     public static final int STRCMPTEST = 2;
6 }
```

```
1 package Twelveth;
2
3 import static Twelveth.Utility.MYSTRCMP;
4 import static Twelveth.Utility.STRCMPTEST;
5
6 // String compare with Interface
7 public class StrcmpInterfaceTest {
8     public static void main(String[] args) {
9         MyStrcmp ms1 = new MyStrcmp(105.3);
10        MyStrcmp ms2 = new MyStrcmp(105.3);
11
12        StrcmpTest st = new StrcmpTest(105.3);
13
14        // 뭐야 ? 왜 MyStrcmp 를 구지 Object 로 바꾸고
15        // 다시 이것을 형변환해서 MyStrcmp 로 만들었지 ?
16        if(ms1.compareTo(ms2, MYSTRCMP) > 0) {
17            System.out.println("ms1이 ms2 보다 더 크다.");
18        } else if(ms1.compareTo(ms2, MYSTRCMP) < 0) {
19            System.out.println("ms1은 ms2 보다 더 작다.");
20        } else {
21            System.out.println("ms1과 ms2는 같다.");
22        }
23
24        if(ms1.compareTo(st, STRCMPTEST) == 0) {
25            System.out.println("이게 되네 ?");
26        }
27    }
28 }
```

105.3.
MyStrcmp Class
Objective order
int order.

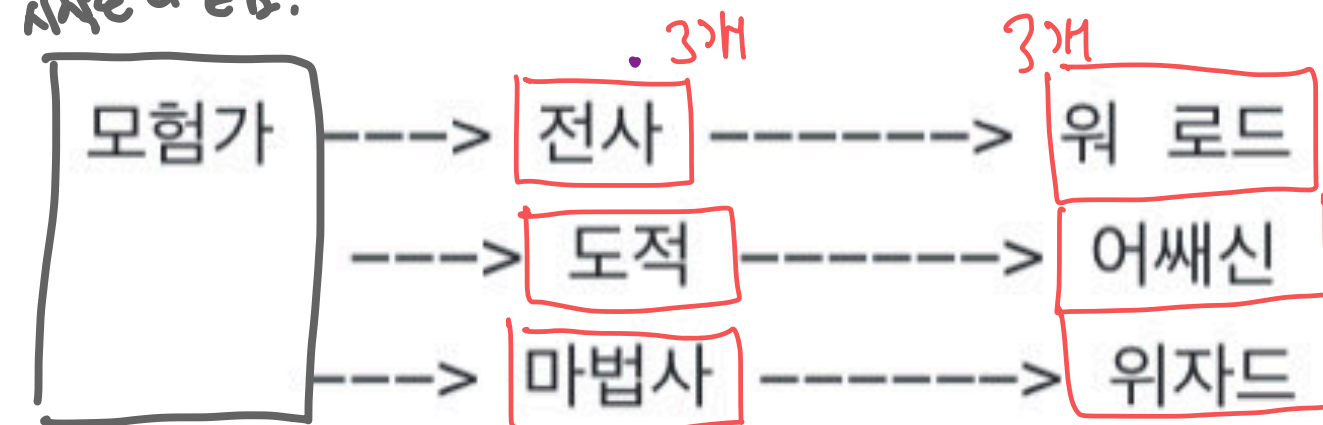
Quiz

1) 질문 노트 작성하기 전체적인 저나 해석 확인부터 드립니다.

2) interface를 사용하기 가장 좋은 예는 게임이다.
아래와 같은 상황을 구현해보도록 한다.

시작 ----> 1차 전직 ----> 2차 전직

시작은 다같은.



① 전직이있다. interface.

② 모험가에서 시작한다.

⇒ 전직 (직명, 레벨) 필요.

Object other int 1 주겠다.

1) 레벨에 따라 1차 / 2차 전직준역

2) 직명에 따라 1차 / 2차 다른기출역

★ 각각의 캐릭터들은 공격 스킬을 두 개씩 가지고 있다.

전직을 할 때마다 추가적인 전직 스킬 두 개를 얻게 된다.

⇒ 모험가: 스킬1, 스킬2
전사: 스킬1, 스킬2, 스킬3, 스킬4
2차전직: 스킬1, 스킬2, 스킬3, 스킬4, 스킬5, 스킬6

스킬을 사용할 때 모두가 일관되게

skill1(), skill2(), skill3() 형식으로

데미지가 계산되도록 만들어보자!