

제 20 강 : 인터페이스

※ 학습목표

- ✓ 인터페이스의 개념을 설명할 수 있다.
- ✓ 인터페이스를 작성할 수 있다.
- ✓ 인터페이스와 추상 클래스의 차이를 설명할 수 있다.
- ✓ 인터페이스와 다형성을 활용할 수 있다.

1. 인터페이스 개념

- ✓ 인터페이스는 음식점의 메뉴판과 같은 것이다.
- ✓ 메뉴판을 보고 고객이 원하는 음식을 요청하게 되는데 메뉴판 자체가 음식을 주지는 않는다. 실제 음식은 주방이라는 곳에서 나오므로 메뉴판은 고객이 호출할 수 있는 서비스의 목록이라 할 수 있다.

2. 인터페이스 구조

```
[접근제한] interface [인터페이스명] {  
    상수;  
    추상메소드;  
}
```

- ✓ 인터페이스 내에는 상수 또는 추상 메소드들만 정의가 가능하다.
- ✓ 사용하기 위해서는 일반 클래스에서 구현(implements)력을 가져야 한다.
- ✓ 다시 말해서 일반 클래스에서 “implements” 라는 예약어로 특정 인터페이스를 구현하겠다고 명시하는 것이다. 그렇게 되면 명시한 인터페이스가 가지는 추상 메소드는 구현 받은 클래스에서 하나도 빠짐없이 Overriding(재정의)해야 한다.
- ✓ 인터페이스는 두 객체가 서로 통신하는 규칙, 약속 이고 추상 메서드를 사용해서 자식들이 그 메소드를 오버라이드 하여 사용하도록 만드는 일종의 약속을 정한 것이다.
- ✓ 자바의 인터페이스는 모든 메소드가 추상메서드 이고, 모든 변수가 static 상수이다.
- ✓ 인터페이스의 모든 변수는 static이고 final이다. 만약에 붙이지 않으면 컴파일러가 붙여준다.

[실습]

```
1 package tommy.java.exam01;
2
3 interface InterEx {
4     static final int A = 100;
5
6     abstract int getA(); // abstract예약어는 생략 가능!
7 }
8
9 public class InterfaceEx implements InterEx {
10     public int getA() {
11         return A;
12     }
13
14     public static void main(String[] args) {
15         InterfaceEx if1 = new InterfaceEx();
16         System.out.println("getA(): " + if1.getA());
17     }
18 }
```

3. 인터페이스의 상속관계

- ✓ 인터페이스 내에는 상수 또는 동작부분을 구현하지 않은 추상 메소드들이 정의된다.
- ✓ 그러므로 인터페이스를 구현(implements)하겠다고 명시한 일반 클래스에서 원하는 형태로 실제 구현력을 가지게 된다.
- ✓ 실제 구현력이 없는 **인터페이스들 간의 상속에서는 다중 상속이 제공** 된다
- ✓ 인터페이스는 메뉴판과 같이 음식점에서 어떤 음식을 만들 수 있는지를 알려주는 중계자 역할만 할 뿐이다. 음식을 만들어 가져오는 것은 그 메뉴판을 포함(구현)하고 있는 음식점이 반드시 제공해야 할 의무가 있는 것이다.
- ✓ 인터페이스가 다른 인터페이스로부터 상속을 받았다고 하지만 Overriding을 할 수는 없다. 왜냐하면 앞서 공부했듯이 인터페이스는 body를 가지는 일반 메소드를 포함할 수 없다.
- ✓ 그러므로 상속을 받은 자식 인터페이스를 구현(implements)하는 일반 클래스에서 부모 인터페이스와 자식 인터페이스의 추상 메소드들을 모두 Overriding(재정의)해야 한다.

[실습]

```
1 package tommy.java.exam02;
2
3 interface Inter1 {
4     public int getA();
5 }
6
7 interface Inter2 {
8     public int getA();
9 }
10
11 interface Inter3 extends Inter1, Inter2 {
12     public int getData();
13 }
14
15 public class InterEx2 implements Inter3 {
16     int a = 100;
17
18     public int getA() {
19         return a;
20     }
21
22     public int getData() {
23         return a + 10;
24     }
25
26     public static void main(String[] args) {
27         InterEx2 it = new InterEx2();
28         Inter1 it1 = it;
29         Inter2 it2 = it;
30         Inter3 it3 = it;
31         System.out.println(it1.getA());
32         System.out.println(it2.getA());
33         System.out.println(it3.getData());
34     }
35 }
```

4. 추상클래스와 인터페이스 비교

① 차이점

<pre>abstract class class명{ 모든 member 변수들; 모든 member 메소드들; }</pre> <p>- 단일상속을 한다.</p>	<pre>interface interface명{ (public static final) int x = 10; (public abstract) void disp(); }</pre> <p>- 다중상속이 가능하다.</p>
---	--

② 공통점

<ul style="list-style-type: none">✓ 객체를 발생 시킬 수 없다.✓ 상속해서 하위 class 를 통하여 객체를 발생시킨다.✓ 상속 시 하위클래스에서 메서드를 재정의 해야 함

5. instanceof 예약어

- ✓ 예약어를 중심으로 왼쪽의 객체가 오른쪽에 명시한 클래스로 형 변환이 가능한지 알게 하는 예약어
- ✓ 비교객체이름 instanceof 비교클래스명

[실습]

1	package tommy.java.exam03;
2	
3	class AClass {
4	}
5	
6	interface CInter {
7	}
8	
9	class BClass extends AClass implements CInter {
10	}
11	

```
12 public class ABCEX {
13     public static void main(String[] ar) {
14         BClass bp = new BClass();
15         System.out.println(bp instanceof Object);
16         System.out.println(bp instanceof AClass);
17         System.out.println(bp instanceof BClass);
18         System.out.println(bp instanceof CInter);
19     }
20 }
21 }
```