

03

03-7. *inner, enum class*

Java 객체지향

Inner class

- 클래스 안에 필드나 메서드를 선언하듯 이 클래스도 선언할 수 있습니다
- 클래스 안에 선언된 클래스를 "내부 클래스(Inner Class)"라고 부르며, 내부 클래스를 멤버로 포함하는 클래스는 "외부 클래스(Outer Class)"라고 합니다.

【내부 클래스 선언】

```
제어자 class 클래스명 {           ← 외부 클래스
    필드 선언;
    생성자 선언
    메서드 선언
    클래스 선언                   ← 내부 클래스
}
```

Inner class

```
package com.ruby.java.ch08.innerClass;

public class MyLinkedList {
    private Node head = null;

    private class Node {
        private String data;
        private Node link;

        public Node(String data) {
            this.data = data;
        }
    }
}
```

외부 클래스 { 내부 클래스 {



Inner class

- Inner 클래스 종류

```
public class OuterClass {  
    class InstanceClass { ... }           ← 인스턴스 멤버 클래스  
    static class StaticClass { ... }      ← 정적 멤버 클래스  
    public void method1() {  
        class LocalClass { ... }         ← 로컬 클래스  
    }  
}
```

Inner class

- Instance 멤버 클래스
- 인스턴스 클래스를 어떤 클래스의 멤버로 선언.
- 인스턴스 멤버 클래스는 자신이 속한 외부 클래스의 인스턴스를 통해서만 접근.

```
public class OuterClass {  
    class InstanceClass {  
        int a;  
        void method2() {  
            System.out.println("Instance Class : " + a);  
        }  
    }  
}
```

```
public class OuterClassTest {  
    public static void main(String[] args) {  
        OuterClass outer = new OuterClass();  
        OuterClass.InstanceClass inner01 = outer.new InstanceClass();  
        inner01.a = 123;  
        inner01.method2();  
    }  
}
```

Inner class

- 정적 멤버 inner 클래스
- 내부 클래스 중 static이 선언된 클래스를 '정적 멤버 클래스'라고 합니다.

```
public class OuterClass {  
    static class StaticClass {  
        int b;  
        static int c;  
        void method3() {  
            System.out.println("Static Class : " + b);  
        }  
        static void method4() {  
            System.out.println("Static Class : " + c);  
        }  
    }  
}
```

```
public class OuterClassTest {  
    public static void main(String[] args) {  
        OuterClass.StaticClass inner02 = new OuterClass.StaticClass();  
        inner02.b = 456;  
        inner02.method3();  
        OuterClass.StaticClass.c = 789;  
        OuterClass.StaticClass.method4();  
    }  
}
```

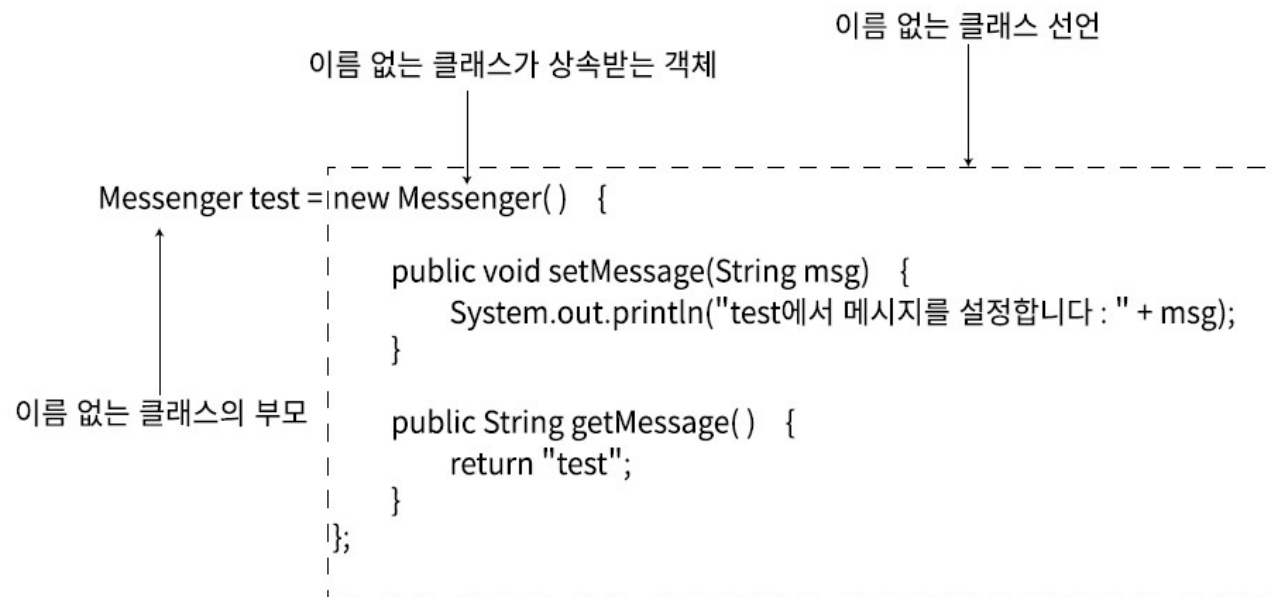
Inner class

- 로컬 클래스
- 메서드 내에 선언하는 내부 클래스로서 메서드 내부에서만 사용.
- 접근 제한자(public, protected, private)를 지정할 수 없습니다.
- 로컬 클래스는 필드와 메서드를 static으로 선언할 수 없습니다.

```
public class OuterClass {  
    public void method1() {  
        class LocalClass {  
            int d;  
            void method5() {  
                System.out.println("LocalClass : " + d);  
            }  
        }  
        LocalClass inner = new LocalClass();  
        inner.d = 1004;  
        inner.method5();  
    }  
}
```

Inner class

- 익명 클래스
- 일회용처럼 딱 한 번만 인스턴스를 생성하여 사용하는 클래스.
- 한 번만 사용하므로 클래스 이름이 필요 없기 때문에 "익명 클래스(Anonymous Class)"라고 합니다.



enum class

- 열거형 클래스를 선언할 때는 class 대신 enum 키워드를 사용합니다.
- 변수 이름으로 사용했던 문자열을 콤마(,)를 구분자로 나열합니다. 이것을 "열거 상수(enumeration constants)"라고 합니다.
- 열거 상수는 자동으로 public static final로 선언되므로 별도로 지정하지 않아도 됩니다.

```
class Status {  
    public static final int READY = 1;  
    public static final int SEND = 2;  
    public static final int COMPLETE = 3;  
    public static final int CLOSE = 4;  
}
```

```
enum Status {  
    READY, SEND, COMPLETE, CLOSE  
}
```

enum class

- java.lang.Enum 클래스는 열거형 클래스가 공통으로 사용하는 기본 클래스입니다.
- Enum 클래스에 선언된 메서드는 모든 열거형 클래스에서 사용할 수 있습니다.

메서드	설명
name()	열거 상수의 이름 반환
toString()	name() 메서드와 동일
ordinal()	열거 상수의 선언된 위치값을 반환
compareTo()	열거 상수의 상대적 위치를 구함
valueOf	인자로 지정한 이름의 열거 상수를 반환
values()	모든 열거 상수 반환

enum class

EnumTest02.java

```
01: package com.ruby.java.ch08;
02:
03: enum Mandarin {
04:     금귤, 한라봉, 레드향, 천혜향, 황금향;
05: }
06:
07: public class EnumTest02 {
08:     public static void main(String[] args) {
09:
10:         Mandarin ma = Mandarin.한라봉;
11:
12:         System.out.println("이름 : " + ma.name());
13:         System.out.println("위치 : " + ma.ordinal());
14:         System.out.println("황금향과의 상대 위치 : " + ma.compareTo(Mandarin.황금향));
15:
16:         Mandarin ma2 = Mandarin.valueOf("레드향");
17:         System.out.println(ma2);
18:
19:         Mandarin list[] = Mandarin.values();
20:         System.out.println("== 귤의 종류 ==");
21:         for(Mandarin m : list)
22:             System.out.println(m);
23:     }
24: }
```

enum class

- 열거형 클래스의 생성자는 열거형 타입의 변수가 선언 될수 있다.
- 생성자에 전달하는 인자값은 열거 상수 다음 괄호 안에 지정한 값입니다

EnumTest02.java

```
01: package com.ruby.java.ch08;
02:
03: enum Mandarin {
04:     금귤(600), 한라봉(500), 레드향(300), 천혜향(400), 황금향(800);
05:
06:     private int price;
07:
08:     Mandarin(int p) {
09:         price = p;
10:     }
```

enum class

```
11:
12: int getPrice() {
13:     return price;
14: }
15: }
16:
17: public class EnumTest02 {
18:     public static void main(String[] args) {
19:
20:         Mandarin ma = Mandarin.한라봉;
21:         if(ma == Mandarin.한라봉)
22:             System.out.println("ma는 한라봉입니다.");
23:         System.out.println(ma + " 가격 : " + ma.getPrice());
24:
25:         Mandarin list[] = Mandarin.values();
26:         System.out.println("= 귤의 종류 =");
27:         for(Mandarin m : list)
28:             System.out.println(m + ":" + m.getPrice());
29:     }
30: }
```



감사합니다

단단히 마음먹고 떠난 사람은
산꼭대기에 도착할 수 있다.
산은 올라가는 사람에게만 정복된다.



윌리엄 셰익스피어
William Shakespeare