

제 16 강 : 객체지향 - 지정어

※ 학습목표

- ✓ static 지정어를 설명할 수 있다.
- ✓ final 지정의 용도를 설명할 수 있다.
- ✓ 클래스를 활용한 프로그래밍을 수행할 수 있다.

1. static

- ✓ 메서드나 멤버변수에 정의할 수 있으며 지역변수나 클래스에는 정의 불가
- ✓ static 키워드를 사용하면 static 변수(클래스 변수), static 메서드라 지칭함
- ✓ 멤버변수나 멤버 메서드는 해당 객체가 생성될 때 heap 영역에 존재함
- ✓ static으로 선언된 필드, 메서드는 static 영역에 유일하게 만들어지면서 모든 객체들이 사용할 수 있는 공유개념을 가지기 때문
- ✓ 객체를 생성하지 않더라도 사용할 수 있음
- ✓ 클래스명.변수명 or 클래스명.메서드명 으로 접근 가능

static 메서드 선언법

[접근제한자] static 반환형 메서드명([자료형 인자들...]) {}

static 필드 선언법

[접근제한자] static 자료형 변수명;

[실습]

```
1 package tommy.java.exam01;
2
3 class StaticEx {
4     int x;
5     static int y;
6 }
7
8 public class StaticExTest {
9     public static void main(String[] ar) {
10         System.out.println(StaticEx.x); // Error (객체를 발생시켜야 함)
11         System.out.println("StaticEx.y = " + StaticEx.y); // 0 출력
```

12	StaticEx.y = 100;
13	System.out.println("StaticEx.y = " + StaticEx.y); // 100 출력
14	StaticEx se = new StaticEx(); // x = 0 , y = 0
15	System.out.println("se.x = " + se.x); // 0 출력
16	System.out.println("se.y = " + se.y); // 100 출력
17	}
18	}

2. final

- ✓ final은 더 이상의 확장이 불가능함을 알리는 종단(상수)과 같은 것을 의미한다
- ✓ 변수에 final을 적용 시 상수를 의미한다.
- ✓ 메소드에 final을 적용 시 오버라이딩으로의 확장이 불가능하다
- ✓ 클래스에 final을 적용 시 더 이상의 상속 확장이 불가능하다.

3. 종합예제 : 계좌관리를 수행하는 예제

- ✓ 아래의 클래스를 같은 패키지에 작성하고 실행하여 보자.

① Account.java 작성

1	package tommy.java.exam02;
2	
3	public class Account {
4	private String name;
5	private long balance;
6	Account() {
7	}
8	Account(String name) {
9	this.name = name;
10	}
11	public String getName() {
12	return name;
13	}
14	public long getBalance() {
15	return balance;
16	}
17	public void deposit(long amount) {
18	balance += amount;
19	}
20	public void withdraw(long amount) {
21	if (balance < amount)
22	System.out.println("잔고가 부족합니다");

23	else
24	balance -= amount;
25	}
26	}

② Banking.java 작성

1	package tommy.java.exam02;
2	
3	import java.io.BufferedReader;
4	import java.io.IOException;
5	import java.io.InputStreamReader;
6	
7	public class Banking {
8	public static void main(String[] args) throws IOException {
9	Account na = new Account("홍길동");
10	BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
11	String strWork = null;
12	do {
13	System.out.println("\n \n 작업을 선택하세요");
14	System.out.println("=====");
15	System.out.println("입 금 =====> 1");
16	System.out.println("출 금 =====> 2");
17	System.out.println("잔 고 확 인 =====> 3");
18	System.out.println("중 료 =====> 0");
19	System.out.println("=====");
20	System.out.print("작업내용을 선택하세요 : ");
21	strWork = br.readLine();
22	int switchInt = 0;
23	if (strWork.length() != 0)
24	switchInt = Integer.parseInt(strWork);
25	else {
26	System.out.println("작업내용을 입력하지 않았습니다.");
27	System.exit(0);
28	}
29	switch (switchInt) {
30	case 0:
31	break;
32	case 1:
33	System.out.println("\n =====");
34	System.out.print("금액을 입력하세요 : ");
35	String strdepositIn = br.readLine();
36	long depositLong = Long.parseLong(strdepositIn);
37	na.deposit(depositLong);
38	break;
39	case 2:

40	System.out.println("\n =====");
41	System.out.print("금액을 입력하세요 : ");
42	String strwithdrawln = br.readLine();
43	long withdrawLong = Long.parseLong(strwithdrawln);
44	na.withdraw(withdrawLong);
45	break;
46	case 3:
47	System.out.println(na.getName() + "님의 잔고는 " + na.getBalance() + " 원입니다");
48	break;
49	default:
50	System.out.println("0~3 사이 숫자 입력\n");
51	}
52	} while (!strWork.equals("0"));
53	}
54	}

4. 돌발퀴즈

✓ 아래의 조건과 그림을 보고 클래스를 정의하고 구현해 보자

Animal
live : boolean age : int name : String
setName() getName() getLive()

[조건1]

- 클래스 이름은 Animal로 한다.
- 클래스 내부에는 생존여부를 가리키는 속성을 가진다.
- 나이와 이름 정보를 기억하는 속성을 가진다.
- 이름 정보를 바꾸는 기능이 있어야 한다.
- 이름 정보를 반환하는 기능이 있어야 한다.
- 생존 여부를 반환하는 기능이 있어야 한다.

AnimalTest
main()

[조건2]

- 클래스 이름은 AnimalTest 라고 한다.
- AnimalTest 클래스에는 main() 메서드를 가진다.
- [조건1]의 Animal 클래스를 이용하여 나이가 3살 이름이 “강아지” 이며 살아있는 객체를 생성해 보자. 그리고 이름과 생존 여부를 출력해 보자.
- 위의 “강아지” 객체의 이름을 “고양이” 로 변경한 후 이름과 생존여부를 출력해 보자.