

20-7-15

* setter 용도 : 값을 전달하여 멤버변수에 값을 저장하기 위한 용도

(1) [고용] 클래스 만들기

```
고용.java
1 package a20_7_15;
2
3 public class 고용 {
4     private String name;    //멤버변수
5     private int salary;    //멤버변수는 private
6     public void show() {    //멤버메소드는 public
7         System.out.println("이름="+name+", 연봉="+salary);
8     }
9     public void setName(String name) {
10        this.name=name;
11    }
12    public void setSalary(int salary) {
13        this.salary=salary;
14    }
15 }
16 }
```

(2) [메니저] 클래스 만들기

```
*메니저.java  mainTest.java
2
3 public class 메니저 {
4     private String name;    //멤버변수는 private
5     private int salary;
6     private String department;
7     public void show() {    //멤버메소드 public
8         System.out.println("매니저명="+name);
9         System.out.println("연봉="+salary);
10        System.out.println("부서명="+department);
11    }
12    public void show(String man) {
13        System.out.println(man+"명="+name);
14        System.out.println(man+"연봉="+salary);
15        System.out.println(man+"부서="+department);
16    }
17    public void setName(String name) {
18        this.name=name;
19    }
20    public void setSalary(int salary) {
21        this.salary=salary;
22    }
23    public void setDepartment(String department) {
24        this.department=department;
25    }
26 }
```

오버로딩

(3) 메인에서 클래스 객체 생성하기

```

1 package a20_7_15;
2
3 public class mainTest {
4     public static void main(String[] args) {
5         고용 kim=new 고용();
6         kim.setName("홍길동");
7         kim.setSalary(3000);
8         kim.show();
9
10        //메니저 클래스를 생성하여 이름은 이민호, 급여는 5000, 부서는 홍보부
11        //저장하고 출력
12
13        매니저 lee=new 매니저();
14        lee.setName("이민호");
15        lee.setSalary(5000);
16        lee.setDepartment("홍보부");
17        lee.show();
18        lee.show("연예인");
19    }
20
21 }

```

* overloading(오버로딩) : 이름이 동일한 메소드를 2개 이상 가질수 있음
자바언어는 오버로딩이 가능하다

ex) void show() { }
void show(String man) { }

* 생성자(construct) 메소드 용도 : 생성시(new할때) 에 값을 초기화 하기 위한 용도
클래스명과 동일한 메소드명이고 리턴값 자체가 없음(void가 없음)

```

1 package a20_7_15;
2
3 public class 개 {
4     private String dog_name;
5     private int dog_age;
6     public void show() {
7         System.out.println("강아지 이름="+dog_name);
8         System.out.println("강아지 나이="+dog_age);
9     }
10    public void setDog_name(String dog_name) {
11        this.dog_name=dog_name;
12    }
13    public void setDog_age(int dog_age) {
14        this.dog_age=dog_age;
15    }
16    //생성자 메소드 만들기
17    public 개() {
18        dog_name="";
19        dog_age=0;
20    }
21
22 }

```

→ 생성자 메소드

생성자메소드는 오버로딩이 가능하다

```
16 //생성자 메소드 만들기
17 public 개() { //디폴트 생성자메소드
18     dog_name="몽키";
19     dog_age=3;
20 }
21 public 개(String dog_name) { //한개짜리 생성자메소드
22     this.dog_name=dog_name;
23 }
24
25 public 개(String dog_name, int dog_age) { //두개짜리 생성자메소드
26     this.dog_name=dog_name;
27     this.dog_age=dog_age;
28 }
29
30 }
31
```

[실습]

1. Animal 클래스 만들기

멤버변수는 name, age

멤버메소드는 void show() { }

(1) setter만들기

(2) 디폴트생성자 메소드 만들기

(3) 2개짜리 생성자 메소드 만들기

2. mainTest2 클래스 만들기

(내부에 main메소드 가짐)

Animal kim = new Animal()

Animal lee = new Animal("기린",1)

```
1 package a20_7_15;
2
3 public class Animal {
4     private String name;
5     private int age;
6     public void show() {
7         System.out.println(name+","+age);
8     }
9     public void setName(String name) {
10         this.name=name;
11     }
12     public void setAge(int age) {
13         this.age=age;
14     }
15     public Animal() {
16     }
17     public Animal(String name, int age) {
18         this.name=name;
19         this.age=age;
20     }
21 }
22
```

mainTest2.java

```
1 package a20_7_15;
2
3 public class mainTest2 {
4
5     public static void main(String[] args) {
6         Animal kim=new Animal();
7         Animal lee=new Animal("기린", 1);
8
9         kim.setName("호랑이");
10        kim.setAge(2);
11        kim.show();
12
13        lee.show();
14
15    }
16
17 }
18
```