제6장
객체와 클래스 - 1

1. 객체지향 언어란?

- 원래, 객체지향언어는 과학, 군사적 모의실험을 위해 컴퓨터를 이용한 가상세계를 구현하려는 노력으로부터 객체지향이론이 시작됨.
- 1960년대 최초 객체지향언어인 시뮬라(Simula)가 나왔지만, H/W의 불충분으로 사장되어짐.
- 1980년대 절차방식의 프로그래밍의 한계를 극복하기 위해 객체지향언어가 등장함. (C++, Smalltalk와 같은 보다 진보된 객체지향언어가 등장함)
- 1995년 말 Java가 출시됨. 이후 인터넷의 보급으로 인하여 프로그래밍 언어의 선두주자가 됨.

2. 객체지향언어의 특성

- 코드의 재사용성이 매우 높다.
  새로운 코드를 작성할 때, 이미 만들어 놓은 코드를 이용해서 쉽게 작성이 가능함.
  (상속, 클래스 등등)
- 코드의 관리가 쉽다.
  코드간의 직,간접적인 관계를 맺어줌으로써 보다 적은 노력으로 코드 변경이 가능함.
- 보안과 신뢰성이 높은 프로그램 개발을 가능하게 함.
  제어자(접근제어자)와 메서드를 이용하여, 데이터의 캡슐화가 가능하고, 코드의 중복을 제거하여 코드의 불일치로 인한 예외를 방지할 수 있다.
- 기존의 절차지향적 언어와 별반 다르지 않다
  약간의 규칙만 추가했을 뿐이다.

3. 클래스와 객체

1) 클래스의 정의

▶ 클래스란 객체를 정의해 놓은 코드이다.

2) 클래스의 용도

▶ 클래스는 객체를 생성하는데 사용된다.

3) 객체의 정의

▶ 실존하는 사물이나 도구 등

4) 객체의 용도

▶ 객체의 속성과 기능에 따라 다름

클래스	객체(인스턴스)
상품 설계도	상품
모니터 설계도	Monitor
붕어빵 기계	붕어빵

* 다시 말해, 객체는 클래스에서 정의된 내용대로 메모리에 생성된 것을 의미한다.

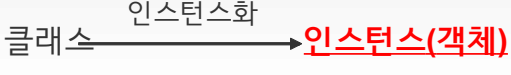
4. 객체와 인스턴스의 관계

- 일반적으로, 객체나 인스턴스나 같은 말이라고 생각하자.
- 객체(Object)는 인스턴스(instance)를 포함하는 형태이다.
- 엄밀히 말하자면, 인스턴스는 new연산자에 의해 메모리에 할당된 것을 말함.

TV는 인스턴스다.	TV는 TV클래스의 객체다.
TV는 객체다.	TV는 TV클래스의 인스턴스이다.

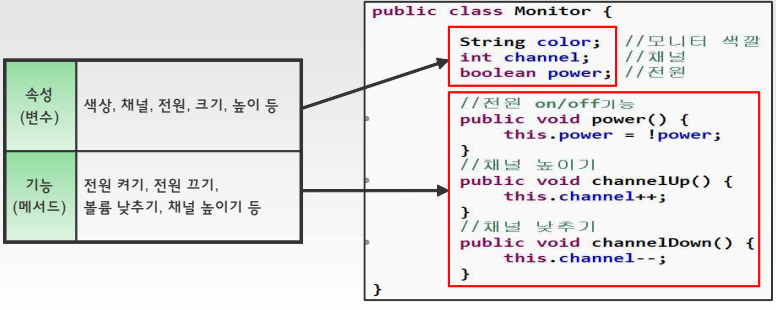
- **인스턴스화(instantiate(예시하다), 인스턴스化)**

  class로부터 instance를 생성하는 것.



5. 객체의 구성요소 – 속성과 기능

- 객체는 통상 **속성(멤버변수)와 기능(멤버 메서드)**으로 되어 있다.
  (물론, 속성과 기능 없이도 객체를 만들 수는 있다.하지만 의미가 없지 않은가?)
- 하여, 속성은 멤버변수로, 기능은 멤버 메서드로 코드를 작성한다.

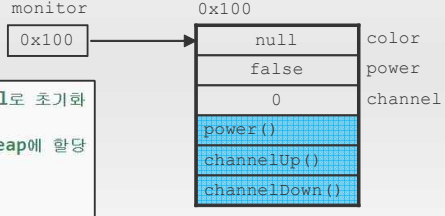


6. 인스턴스의 생성

- 인스턴스 생성 코드

  클래스명 참조변수명;

  참조변수명 = new 클래스명();

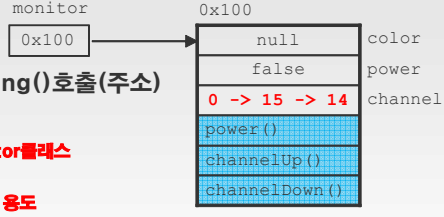



```
//monitor참조변수의 타입이 Monitor이다.null로 초기화
Monitor monitor = null;
//new연산자로 Monitor의 인스턴스를 만들어 heap에 할당
monitor = new Monitor();
//참조변수(리모컨)로 인스턴스를 조작
monitor.power();
```

monitor라는 참조변수는 Monitor를 조작하는 리모컨과 같은 개념

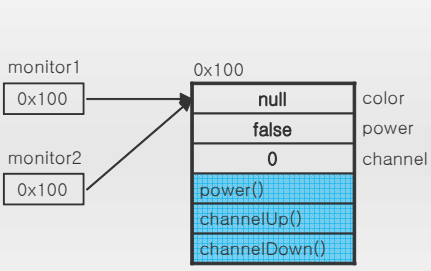
7. 인스턴스의 사용-1

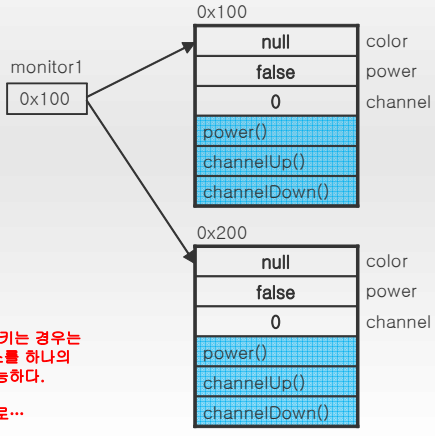
```
Monitor m = new Monitor();
m.channel = 15;
m.channelDown();
System.out.println(m.channel);
System.out.println(m);  //  자동 toString()호출(주소)
```



Object클래스의 toString()호출됨.
단, Monitor클래스에서 toString()오버라이딩시 Monitor클래스의 toString()호출됨.
* toString()의 용도 : 멤버변수의 값들을 수시로 확인할 용도

7. 인스턴스의 사용-2

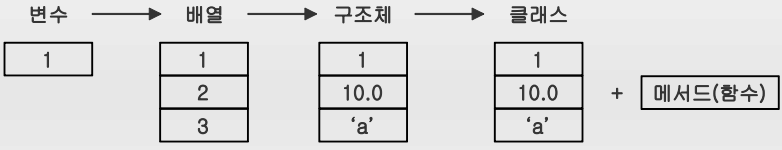




하나의 인스턴스는 여러 개의 참조변수를 가리키는 경우는 얼마든지 가능하다. 하지만 여러 개의 인스턴스를 하나의 참조변수로 가리키는 것은 짐작하겠지만 불가능하다.

이유 : 하나의 변수에는 하나의 값만 들어가므로…

8. 데이터 저장개념의 발전 과정



- 변수 : 하나의 데이터 타입의 데이터를 저장할 수 있는 공간
- 배열 : 같은 데이터 타입의 여러 데이터를 저장할 수 있는 공간
- 구조체 : 데이터 타입에 관계없이 서로 관련된 데이터를 저장할 수 있는 공간
  (자바에는 구조체란 자료구조는 없다.C언어에 존재함)
- 클래스 : 데이터와 메서드(함수)의 결합 구조

9. 사용자 정의 클래스(User Define Class)-1

- 프로그래머가 직접 새로운 타입의 클래스를 얼마든지 생성할 수가 있다.(제한이 없다)
- 서로 의미가 있으며, 관련된 데이터라면 묶어서 하나의 타입으로 정의할 수 있다.

```
class Time {
        int hour;
        int minute;
        int second;
}
```

```
int hour;
int minute;
int second;
```

→

```
Time t = new Time();
```

```
int hour1, hour2, hour3 ;
int minute1, minute2. minute3;
int second1, secound2, second3;
```

→

```
Time t1 = new Time();
Time t2 = new Time();
Time t3 = new Time();
```

```
int[] hour = new int[3];
int[] minute = new int[3];
int[] second = new int[3];
```

→

```
Time[] t = new Time[3];
t[0] = new Time();
t[1] = new Time();
t[2] = new Time();
```

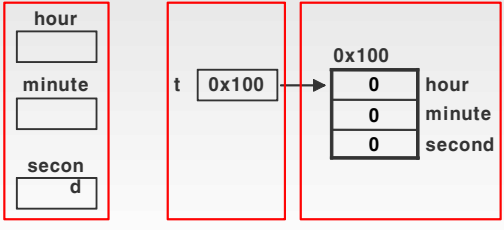
9. 사용자 정의 클래스(User Define Class)-2

```
class Time {
        int hour;
        int minute;
        int second;
}
```

클래스 설계(시각)

```
int hour;
int minute;
int second;
```

→ `Time t = new Time();`

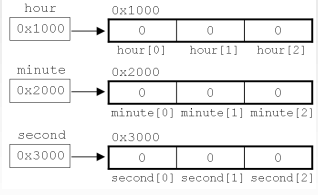


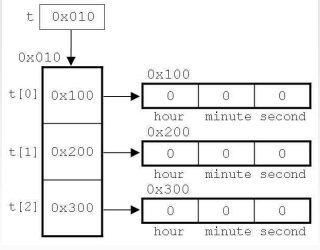
9. 사용자 정의 클래스(User Define Class)-3

```
int[] hour = new int[3];
int[] minute = new int[3];
int[] second = new int[3];
```

```
Time[] t = new Time[3];
t[0] = new Time();
t[1] = new Time();
t[2] = new Time();
```

```
class Time {
        int hour;
        int minute;
        int second;
}
```





좌측 두 개의 그림은 9개의 동일한 저장 공간을 할당하지만,데이터의 의미가 다르다. 시간은 시,분,초가 한 곳에 있어야 의미가 있으므로 Class로 작성하는 것이 적절하다.

감사합니다.