

11. 생성자(constructor)

- ① 클래스 이름과 같은 특별한 메서드를 생성자라 한다.
- ② 생성자는 객체가 생성될 때, 자동 호출되어 객체 초기화를 담당한다.
- ③ 생성자는 리턴값을 가질 수 없다.
즉, 생성자는 반환형이 없으므로 메서드 형을 기술할 수 없다.
- ④ 하나의 클래스는 여러 개의 생성자를 가질 수 있다. 즉, 생성자도 Overloading된다.
- ⑤ 만약, 생성자를 하나도 정의하지 않으면 기본 생성자가 자동으로 만들어진다.
기본 생성자(default constructor)는 인수가 없는 생성자이다.
- ⑥ 만약, 하나 이상의 생성자를 정의하면 기본 생성자가 자동으로 만들어지지 않는다.
이러한 경우는 사용자가 기본 생성자를 정의해야 호출할 수 있다.(주의할 것!)
- ⑦ 생성자는
 - 지정자 final, abstract, synchronized, native는 사용할 수 없고,
 - 지정자 public, protected, private은 사용할 수 있다.
 - 만약, 생성자를 private으로 지정하면 외부에서 객체를 생성하는 것이 불가능하다.
 - 이는 정적 멤버만을 포함하는 클래스에 적용(java.lang.Math 같은 수치리 클래스에 이용)

```
public class Test{
    int a;
    float b;
    public Test(){ } //기본 생성자(다른 생성자가 있으므로 반드시 정의해야 호출 가능)
    public Test(int x){ a = x; System.out.println(a); }
    public Test(int x, float y){ a = x; b = y; System.out.println(a+b); }

    public static void main(String args[]){
        Test t1 = new Test(); //기본 생성자 호출(정의되어 있으므로 가능)
        Test t2 = new Test(1);
        Test t3 = new Test(2, 3.0f);

        System.out.println("end");
    }
}
```

[실행결과]

```
1
5.0
end
```

// 생성자 중첩(overloading)

하나의 클래스는 여러 개의 생성자를 가질 수 있다. 즉, 생성자는 중첩될 수 있다.
단, 생성자가 중첩되려면 생성자가 가지는 **매개변수 개수나 타입이 달라야** 한다.

```
class Date{
    int year, month, day;           //속성
    public Date(){
        year = 2030;
        month = 12;
        day = 25;
    }
    public Date(int y, int m){
        year = y;
        month = m;
        day = 26;
    }
    public Date(int year, int month, int day){
        this.year = year;           //매개변수와 속성 이름이 같으면 this로 구분
        this.month = month;
        this.day = day;
    }
}
class DateDemo{
    public static void main(String args[])
    {
        Date d1 = new Date();
        System.out.println(d1.year+"년 "+d1.month+"월 "+d1.day+"일");

        Date d2 = new Date(2040, 12);
        System.out.println(d2.year+"년 "+d2.month+"월 "+d2.day+"일");

        Date d3 = new Date(2050, 12, 27);
        System.out.println(d3.year+"년 "+d3.month+"월 "+d3.day+"일");
    }
}
```

[실행 결과]

2030년 12월 25일
2040년 12월 26일
2050년 12월 27일