

제 32 강

Wrapper ,Random 클래스

교재:p183~187

목차

1. 기본 API 클래스

1. Wrapper 클래스
2. Random 클래스

1. Wrapper 클래스

Wrapper 클래스

: 기본 자료형들을 객체로 다루는데 사용
자바는 객체 지향 이므로, 객체 값을 다룰 일이 많기 때문에,
일반자료형의 데이터를 객체로 변환하는 작업이 필요!

기본 자료형	Wrapper 클래스
boolean	Boolean 클래스
chare78u	Character 클래스
byte	Byte 클래스
short	Short 클래스
int	Integer클래스
long	Long 클래스
double	Double 클래스
float	Float 클래스

1. Wrapper 클래스

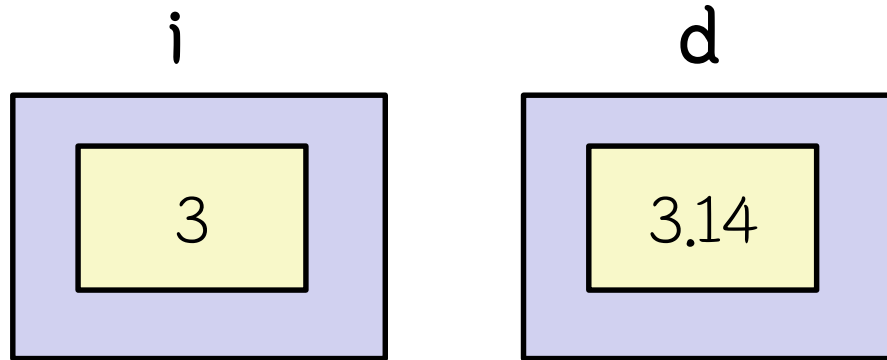
<생성방법>

```
Wrapper클래스명 객체명 = new Wrapper클래스명(값);
```

ex)

```
Integer i = new Integer(3);
```

```
Double d = new Double(3.14);
```



1. Wrapper 클래스

<실습> WrapperEx1.java

```
public class WrapperEx1 {  
    public static void main(String[] args) {  
        Integer i = new Integer(3);  
        System.out.println(i);  
    }  
}
```

<실행 화면>

3

jdk 9 버전 이후부터는 Wrapper 클래스의 생성자 사용을 권장 하지 않는다.
이후에 배울 auto boxing 과 auto unboxing에서 자세히 보도록 하며,

```
Integer i = 3; //autoBoxing
```

Wrapper클래스 객체 = 일반자료형데이터; 로 사용할 수 있다.

1. Wrapper 클래스

<실습> IntegerWrapper.java

```
public class IntegerWrapper {  
    public static void main(String[] args) {  
        Integer i = 100;  
        //intValue():Integer ->int  
        System.out.println(i.intValue()+3);  
  
        //doubleValue():Integer->double  
        System.out.println(i.doubleValue());  
  
        //floatValue(): Integer->float  
        System.out.println(i.floatValue());  
  
        //toString(): Integer->String  
        System.out.println(i.toString()+3);  
    }  
}
```

<실행 화면>

```
103  
100.0  
100.0  
1003
```

1. Wrapper 클래스

<실습> IntegerWrapper.java

```
public class IntegerWrapper {  
    public static void main(String[] args) {  
        Integer i = 100;  
        //intValue():Integer ->int  
        System.out.println(i.intValue()+3);  
  
        //doubleValue():Integer->double  
        System.out.println(i.doubleValue());  
  
        //floatValue(): Integer->float  
        System.out.println(i.floatValue());  
  
        //toString(): Integer->String  
        System.out.println(i.toString()+3);  
  
        //Integer.parseInt():String->int  
        System.out.println(Integer.parseInt("100"));  
    }  
}
```

<실행 화면>

```
103  
100.0  
100.0  
1003  
100
```

1. Wrapper 클래스

1) 오토박싱(Auto Boxing)

: 기본 자료형 값을 자동으로 Wrapper 클래스의 객체로 변환시켜 주는 것

2) 오토 언박싱 (Auton UnBoxing)

: Wrapper 클래스의 객체 값을 자동으로 기본 자료형 값으로 변환시켜 주는 것

```
public class AutoBoxingEx {  
    public static void main(String[] args) {  
        //Autoboxing  
        Integer i=3;  
        Double d=3.14;  
        Float f= 3.14f;  
        Character c='A';  
  
        //Autounboxing  
        Integer i2= i;  
        Double d2 =d;  
        Float f2=f;  
        Character c2=c;  
    }  
}
```


2. Random 클래스

Random 클래스

: 무작위의 값을 얻고 싶을 때 사용하는 클래스

- Random 클래스는 내부적으로 무작위의 값을 추출하기 위한 내부 알고리즘이 존재
- 생성방법?

```
Random 객체명 = new Random();
```

- 생성자 종류?
 - > Random(): 현재 시간을 seed 값으로 인스턴스 생성
 - > Random(int seed): 받아오는 seed 값으로 인스턴스 생성

2. Random 클래스

Random 클래스

: 무작위의 값을 얻고 싶을 때 사용하는 클래스

- 관련 메서드

메서드	설명
<code>int nextInt()</code>	무작위의 int 값을 반환
<code>int nextInt(int n)</code>	0보다 크고 n보다 작은 int 값 반환
<code>boolean nextBoolean()</code>	무작위의 boolean 값 반환
<code>long nextLong()</code>	무작위의 Long 값 반환
<code>double nextDouble()</code>	무작위의 double 값 반환
<code>float nextFloat()</code>	무작위의 float 값 반환

2. Random 클래스

<실습> Random1.java

```
public class Random1 {  
    public static void main(String[] args) {  
        Random random = new Random();  
        Random random2 = new Random(2);  
        Random random3 = new Random(2);  
        for(int i=0;i<5;i++) {  
            System.out.println("기본 생성자:"+random.nextInt());  
        }  
        for(int i=0;i<5;i++) {  
            System.out.println("random2:"+i+"번째 값:"+random2.nextInt());  
        }  
        for(int i=0;i<5;i++) {  
            System.out.println("random3:"+i+"번째 값:"+random3.nextInt());  
        }  
    }  
}
```

<실행 화면>

```
기본 생성자:1118940529  
기본 생성자:-330757168  
기본 생성자:1881856119  
기본 생성자:-1349521880  
기본 생성자:-2084591649  
random2:0번째 값:-1154715079  
random2:1번째 값:1260042744  
random2:2번째 값:-423279216  
random2:3번째 값:17850135  
random2:4번째 값:2133836778  
random3:0번째 값:-1154715079  
random3:1번째 값:1260042744  
random3:2번째 값:-423279216  
random3:3번째 값:17850135  
random3:4번째 값:2133836778
```

seed 값이 동일하므로

결과 값이 같다