

Chapter. 11

컬렉션 프레임 워크

| 제네릭 프로그래밍

FAST CAMPUS
ONLINE
자바 기초

강사. 박은종

Chapter. 11

01 제네릭(Generic) 프로그래밍

I 제네릭 프로그래밍이란

변수의 선언이나 메서드의 매개변수를 하나의 참조 자료형이 아닌 여러 자료형을 변환 될 수 있도록 프로그래밍 하는 방식

실제 사용되는 참조 자료형으로의 변환은 컴파일러가 검증하므로 안정적인 프로그래밍 방식

I 자료형 매개 변수 T

여러 참조 자료형으로 대체 될 수 있는 부분을 하나의 문자로 표현

```
public class GenericPrinter<T> {  
    private T material;  
  
    public void setMaterial(T material) {  
        this.material = material;  
    }  
  
    public T getMaterial( ) {  
        return material;  
    }  
}
```

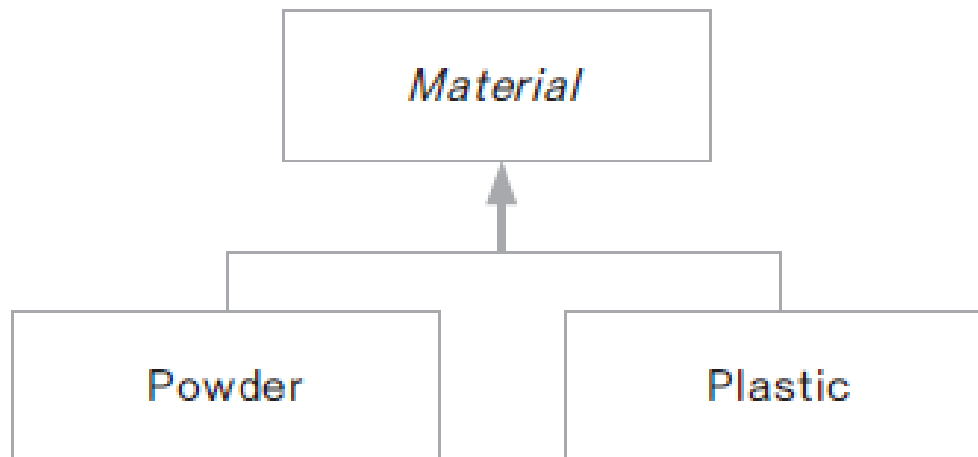
제네릭 클래스

type의 약자. 자료형 매개변수

type의 의미로 T 사용

I <T extends 클래스>

T 대신에 사용될 자료형을 제한하기 위해 사용



Material에 정의 된 메서드를 공유할 수 있음

I 자료형 매개 변수가 두 개 이상일 때

```
public class Point<T, V> {  
    T x;  
    V y;
```

```
    Point(T x, V y) {  
        this.x = x;  
        this.y = y;  
    }
```

```
    public T getX( ) {  
        return x;  
    }
```

```
    public V getY( ) {  
        return y;  
    }
```

```
}
```

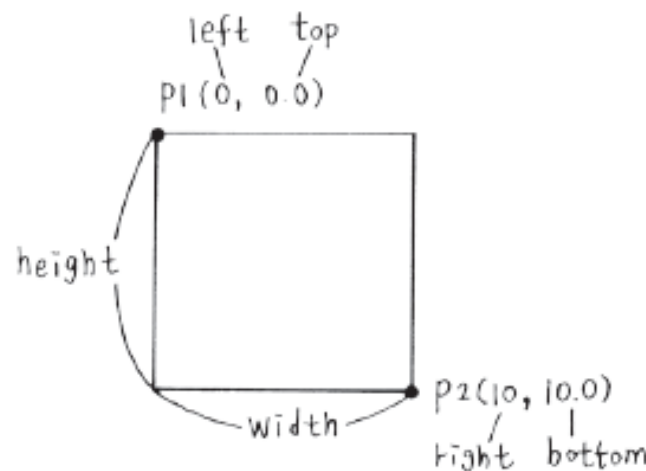
제네릭 메서드

I 제네릭 메서드

메서드의 매개 변수를 자료형 매개 변수로 사용하는 메서드

```
public class GenericMethod {  
    public static <T, V> double makeRectangle(Point<T, V> p1, Point<T, V> p2) {  
        double left = ((Number)p1.getX()).doubleValue( );  
        double right = ((Number)p2.getX()).doubleValue( );  
        double top = ((Number)p1.getY()).doubleValue( );  
        double bottom = ((Number)p2.getY()).doubleValue( );  
  
        double width = right - left;  
        double height = bottom - top;  
  
        return width * height;  
    }  
}
```

제네릭 메서드



I 제네릭 메서드

메서드 내에서의 자료형 매개 변수는 메서드 내에서만 유효 함 (지역 변수와 같은 개념)

```
class Shape<T>{  
    public static <T, V> double makeRectangle(Point<T, V> p1, Point<T, V> p2){  
        .....  
    }  
}
```

Shape 의 T 와 makeRectangle 의 T는 전혀 다른 의미