

객체지향 프로그래밍



Chapter 21 ~ 22



11주차 과제

문제 1-1

Stack 자료구조를 제네릭 클래스로 선언하고, String과 Integer 스택을 각각 사용하는 예를 MyStack 클래스에서 보여주고 있다. 주어진 MyStack 클래스와 결과 내용을 참고하여 Gstack 제네릭 클래스를 작성하시오.

* Console 결과

```
LA  
busan  
seoul  
5  
3  
1
```

문제 1-2

```
public class MyStack {  
    public static void main(String[] args) {  
        int size=10;  
        GStack<String> stringStack = new GStack<String>(size);  
  
        stringStack.push("seoul");  
        stringStack.push("busan");  
        stringStack.push("LA");  
  
        for(int i=0; i<size; i++) {  
            String str = stringStack.pop();  
            if(str != null) System.out.println(str);  
        }  
  
        GStack<Integer> intStack = new GStack<Integer>(size);  
        intStack.push(1);  
        intStack.push(3);  
        intStack.push(5);  
  
        for(int i=0; i<size; i++) {  
            Integer is = intStack.pop();  
            if(is != null) System.out.println(is);  
        }  
    }  
}
```

* Console 결과

```
LA  
busan  
seoul  
5  
3  
1
```

문제2-1

Key, Value를 쌍으로 데이터를 관리하는 PairGen 클래스는 제네릭 클래스이다. Ex2 클래스에서 보는 바와 같이 <String, Integer> 또는 <String, String> 등으로 다양한 자료를 쌍으로 관리할 수 있다. 그리고 CompareGen은 PairGen에서 관리하는 데이터를 서로 비교하는 제네릭 메소드를 가지고 있다. Ex2 클래스를 통해 PairGen 제네릭 클래스와 compare라는 제네릭 메소드를 멤버 메소드로 가지고 있는 CompairGen 클래스를 작성하시오.

문제2-2

```
public class Ex2 {  
  
    public static void main(String[] args) {  
        PairGen<String, Integer> p1 = new PairGen<>("Apple", 3);  
        PairGen<String, Integer> p2 = new PairGen<>("Apple", 3);  
        System.out.println(CompareGen.compare(p1, p2));  
  
        PairGen<String, String> p3 = new PairGen<>("객체지향", "Java");  
        PairGen<String, String> p4 = new PairGen<>("객체지향", "C++");  
        System.out.println(CompareGen.compare(p3, p4));  
    }  
}
```

* Console 결과

```
true  
false
```

문제3

제네릭 메소드를 Ex3 클래스에 추가하여 다음과 같이 배열의 평균을 구하시오.

```
public class Ex3 {  
    public static void main(String[] args) {  
        Integer[] ArrayOne = { 1, 2, 3, 4, 5 };  
        Double[] ArrayTwo = { 3.0, 4.0, 1.5, 2.5, 1.5 };  
        Long[] ArrayThree = { 7L, 8L, 9L, 10L };  
  
        System.out.println(getAverage(ArrayOne));  
        System.out.println(getAverage(ArrayTwo));  
        System.out.println(getAverage(ArrayThree));  
    }  
  
    ...  
}
```

* Console 결과

```
3.0  
2.5  
8.5
```