



소프트웨어프로젝트 02분반

project4 report

소프트웨어학부

20181269 박현우

I. 소스 프로그램 및 설명

<GenericStack.java>

```
public class GenericStack {
    int stackSize;
    int top;
    Object[] stack;

    public GenericStack(int i) {
        this.stackSize = i;
        this.top = -1;
        this.stack = new Object[stackSize];
    }

    public boolean isEmpty() {
        if (this.top == -1)
            return true;
        else return false;
    }

    public void push(Object value) {
        this.stack[++top] = value;
    }

    public Object pop() {
        return this.stack[top--];
    }
}
```

스택의 용량을 나타내는 int형 변수 stackSize와 스택에 담겨있는 데이터 상태를 나타내는 변수 top을 선언하고 Object 배열 stack을 선언한다. 스택의 용량을 입력값으로 하여 모든 타입의 데이터를 수용할 수 있는 스택을 생성하는 GenericStack 생성자를 만들고 초기 top 값을 -1로 하도록 한다. 스택이 비어있으면 true를 반환하고, 스택에 데이터가 존재하면 false를 반환하는 함수 isEmpty를 선언한다. 스택에 데이터를 집어넣는 함수 push를 선언한다. 이 때 top을 1 증가시키고 그 위치에 입력받은 값 value를 저장시킨다. 스택에서 데이터를 빼내는 함수 pop을 선언한다. 현재 top 위치에 있는 데이터를 리턴해주고 top을 1 감소시킨다.

<ParaStack.java>

```
public class ParaStack<E> {
    int stackSize;
    int top;
    E[] stack;

    public ParaStack(int i) {
        this.stackSize = i;
        this.top = -1;
        this.stack = (E[])new Object[stackSize];
    }

    public boolean isEmpty() {
        if (this.top == -1)
            return true;
        else return false;
    }

    public void push(E value) {
        this.stack[++top] = value;
    }

    public E pop() {
        return this.stack[top--];
    }
}
```

다른 모든 기능은 GenericStack과 같도록 하되 특정한 타입만을 취급하도록 <E>를 사용하여 구현한다.

<ParaStackNoLimit.java>

```
public class ParaStackNoLimit<E extends Object> {
    int stackSize;
    int top;
    E[] stack;

    ParaStackNoLimit(int i) {
        this.stackSize = i;
        this.top = -1;
        this.stack = (E[])new Object[stackSize];
    }

    public void push(E value) {
        if (stackSize >= stack.length) {
            E[] temp = (E[])new Object[stack.length * 2];
            System.arraycopy(stack, 0, temp, 0, stack.length);
            stack = temp;
        }
        stack[stackSize++] = value;
    }
}
```

ParaStack을 extend하여 데이터 수용 개수에 제한이 없도록 한다. 배열의 크기를 모두 소진한 경우 배열의 크기를 두 배로 늘린다.

II. 실행 화면

stack의 push, pop 테스트

```
GenericStack.java  ParaStack.java  ParaStackNoLimit.java  StackTest.java
5      GenericStack stack1 = new GenericStack(5);
6
7      System.out.println(stack1.isEmpty());
8      stack1.push("apple");
9      stack1.push(1.0);
10     System.out.println(stack1.isEmpty());
11     stack1.push(2);
12     stack1.push("4");
13     System.out.println(stack1.pop());
14     System.out.println(stack1.pop());
15     System.out.println(stack1.pop());
16     System.out.println(stack1.pop());
17
18     ParaStack<String> stack2 = new ParaStack<String>(5);
19
20     stack2.push("hi");
21     stack2.push("hello");
22     stack2.push("good");
23     System.out.println(stack2.pop());
24     System.out.println(stack2.pop());
25     System.out.println(stack2.pop());
26     stack2.push("xyz");
27     stack2.push("100");
28     System.out.println(stack2.pop());
29     System.out.println(stack2.pop());
30     System.out.println(stack2.isEmpty());
31
32     ParaStackNoLimit<String> stack3 = new ParaStackNoLimit<String>(5);
33     stack3.push("A");
```

< Problems Javadoc Declaration Console

<terminated> StackTest [Java Application] C:\Program Files\Java\jre1.8.0_211\bin\javaw.exe (2019. 5. 8. 오전 2:46:32)

```
true
false
4
2
1.0
apple
good
hello
hi
100
xyz
true
```