プログラミング演習 II
2017 年 10 月 3 日

学習ポイント

- スタックを利用する
- スケルトンとドキュメントからコードを定義する

プログラミングでは、同じ種類の複数個のデータの集合を扱う場面がたくさんあります。集合にデータを追加したり、取り出したり、変更したりします。集合では、主に要素に順序が付いたもの、重複がないものを区別します。「列」とは同じ種類のデータが 1 つの列として順番に並んだものであり、「列」に対する基本的な操作は以下の通りです。

- データの挿入：列中の特定の場所に要素を追加する
- データの削除：列中の特定の場所の要素を取り除く
- データの参照：列中の特定の要素のデータを読み出す
- データの更新：列中の特定の要素にデータを書き込む

「スタック」はデータの追加や取り出し（挿入と削除）を一方の端だけで行うという制限を加えたものです。スタック中のデータを直接読み出したり、書き込んだりすることは許されません。また、データの読み出しは必ず端から、そのデータを取り出した上で行います。

挿入と削除を一方の端だけで行うので、スタックは「後入れ先出し(Last In First Out:LIFO)」型のデータ構造であると言われます。すなわち、任意の要素をスタックに１つずつ入れ、１つずつ取り出すことができ、取り出しの順序は最後に入れたものが最初に取り出されるわけです。要素を入れる操作を push、要素を取り出す操作を pop と呼びます。

スタックは例えば、以下のように定義できます。これは１つの実装方法であり、他にも実装方法があります。

```
class Stack{
        private Object[] stack = new Object[3];
        private int top = -1;

        public void push(Object element){
                this.stack[++this.top] = element;
        }
        public Object pop(){
                return this.stack[this.top--];
        }
}
```

スタックはいろいろなアプリケーションでの利用が考えられます。例えば、語「しんぶんし」を反転させる場合に、スタックはつぎのように使えます、

① 語「しんぶんし」を文字に分解し、語の大きさのスタックに格納する　⇒　文字「し」「ん」「ぶ」「ん」「し」が順番にスタックに挿入される。

② スタックから要素を 1 つずつ取り出し、文字を結合して文字列とする　⇒語「しんぶんし」が得られる。

本日の練習

1. この Stack の定義を使って、つぎのことを実行する main メソッドを定義しなさい。main メソッドをもつクラスは Main とする。
   1) Stack のインスタンス　stack を作る。
   2) 文字列"A"を stack に push する。
   3) 文字列"B"を stack に push する。
   4) Stack から要素を pop し、画面に表示する。

※文字列"A"の代わりに、整数 1 を push してみてください。どうですか？

2. 上記の定義では、語を反転させるプログラムにスタックを利用する上で問題があると思われる。どのような問題があるかを他の人と討議しなさい。各自が認識した問題点を、Scomb の「アンケート」に回答しなさい。

3. メソッドの追加：配付のスケルトンコードから javadoc コマンドを使用してドキュメントを生成しなさい。そのドキュメントを読み、ソースコードのスケルトンを埋めて、語を反転させるプログラムを完成しなさい。