

Java プログラミング第 14 回 演習課題

【演習課題 1】 ArrayList を用いて，スタックとスタックへの操作を実装せよ．

- ・スタックの要素は全て String 型とする
- ・push() : スタックの先頭に要素を 1 つ追加し，リストの内容をプリントするメソッド
- ・pop() : スタックの先頭の要素を 1 つ取り出し，それを削除し，リストの内容をプリントするメソッド
- ・top() : スタックの先頭の要素を参照するメソッド

以上のメソッドを宣言したインタフェースを以下にまとめる．

```
interface Top{
    String top(); //スタックの先頭の要素を参照
}

interface Push{
    void push(String element); //要素をスタックの先頭に格納
}

interface Pop{
    void pop(); //スタックの先頭の要素を取り去る
}
```

また，スタックの基本形を以下の Stack.java に示す．

```
import java.util.*;

public abstract class Stack {
    private String name;
    ArrayList<String> stack;

    void setName(String name) {
        this.name = name;
    }

    String getName() {
        return this.name;
    }

    abstract void printStack();
}
```

Stack を継承し，必要なインタフェースを実装することにより，以下に示す StackTest.java を完成させよ．

```
import java.util.ArrayList;

public class StackTest 必要な記述を追加 {

    public static void main(String[] args) {
        StackTest st = new StackTest("StackTest");
    }
}
```

```

        st.push("a");
        st.push("b");
        System.out.println(st.top());
        st.push("c");
        st.pop();
        st.pop();
        st.push("d");

    }

    //ここにコードを追加

}

```

出力例

```

StackTest[a]
StackTest[ba]
b
StackTest[cba]
StackTest[ba]
StackTest[a]
StackTest[da]

```

【演習課題 2】 printStack をラムダ式を用いて実装せよ.

【演習課題 3】

コーヒーを出すロボット(CoffeeServer), ケーキを出すロボット(CakeServer)がある. ロボットがおいたコーヒーとケーキはケーキセットとして運び出される (ケーキセットが作られると台の上のケーキはその分減る). コーヒーを置く台(coffeeBuffer)とケーキを置く台(cakeBuffer)があり, それぞれ, 5つまでしか置くことができない. 従って, どちらかがこれ以上置けなくなると, もう一方の用意ができるまで, 待つ必要がある. CoffeeServer は 1 回に 1 つずつコーヒーを出し, 最大 1 秒でランダムに休止する. CakeServer は 1 回に 2 個ずつケーキを出し, 最大 2 秒でランダムに休止する. 実行結果を参考にしながら, 空欄部分を実装してプログラムを完成させよ.

```

import java.util.*;

class CoffeeServer extends Thread {
    static final Random random= new Random();
    String name;
    Table tb;

    CoffeeServer(String nm, Table t){
        this.name=nm;
        this.tb=t;
    }

    public void run() {

```

```

        try {
            while(true) {
                Thread.sleep(random.nextInt(1000));
                tb.putCoffee();
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

cakeServerクラスの実装

```

class Table {
    private int coffeeBuffer;
    private int cakeBuffer;

    Table(int cob, int cab){
        this.coffeeBuffer=cob;
        this.cakeBuffer=cab;
    }
}

```

putCoffee()の実装
coffeeBufferのコーヒーの数を1つ増やし、ケーキセットを出す。

putCake()の実装
cakeBufferのケーキの数を2つ増やし、ケーキセットを出す。

```

    public void makeCakeSet() {
        if (this.coffeeBuffer==this.cakeBuffer) {
            System.out.println("ケーキセット
                                "+this.coffeeBuffer+"個出ました");
            this.coffeeBuffer=0;
            this.cakeBuffer=0;
        } else if (this.cakeBuffer<this.coffeeBuffer) {
            System.out.println("ケーキセット"+
                                this.cakeBuffer+"個出ました");
            this.coffeeBuffer=this.coffeeBuffer
                                -this.cakeBuffer;
            this.cakeBuffer=0;
        } else if (this.coffeeBuffer<this.cakeBuffer) {
            System.out.println("ケーキセッ"+
                                this.coffeeBuffer+"個出ました");
            this.cakeBuffer=this.cakeBuffer
                                -this.coffeeBuffer;
            this.coffeeBuffer=0;
        } else {
            System.out.println("error");
        }
    }
}

public class CakeSet {
    public static void main(String[] args) {
        Table tb = new Table(0,0);
        CoffeeServer coffee = new CoffeeServer ("coffee1",
tb);
    }
}

```

```

        CakeServer cake = new CakeServer("cake1", tb);
        coffee.start();
        cake.start();
    }
}

```

実行例

```

...
コーヒー出ました:1
ケーキセット1個出ました
ケーキ出ました:2
ケーキセット0個出ました
ケーキ出ました:4
ケーキセット0個出ました
コーヒー出ました:1
ケーキセット1個出ました
Thread-1 これ以上ケーキをおけません
コーヒー出ました:1
ケーキセット1個出ました
ケーキ出ました:4
ケーキセット0個出ました
コーヒー出ました:1
ケーキセット1個出ました
Thread-1 これ以上ケーキをおけません
コーヒー出ました:1
ケーキセット1個出ました
Thread-1 これ以上ケーキをおけません
コーヒー出ました:1
ケーキセット1個出ました
Thread-1 これ以上ケーキをおけません
コーヒー出ました:1
ケーキセット1個出ました
ケーキ出ました:4
ケーキセット0個出ました
コーヒー出ました:1

```

```

つづき
ケーキセット1個出ました
コーヒー出ました:1
ケーキセット1個出ました
コーヒー出ました:1
ケーキセット1個出ました
...

ケーキ出ました:2
ケーキセット2個出ました
コーヒー出ました:3
ケーキセット0個出ました
コーヒー出ました:4
ケーキセット0個出ました
コーヒー出ました:5
ケーキセット0個出ました
Thread-0 これ以上コーヒーをおけません
ケーキ出ました:2
ケーキセット2個出ました
コーヒー出ました:4
ケーキセット0個出ました
コーヒー出ました:5
ケーキセット0個出ました
Thread-0 これ以上コーヒーをおけません
ケーキ出ました:2
ケーキセット2個出ました
コーヒー出ました:4
...

```

【提出方法】

作成したソースコードをポータルにアップロードすること。

提出期限：CoursePower に記載