2-7 プログラミング演習

月　　日

※動画説明欄から提供プログラムをダウンロード！　→

Lesson のゴール

① スレッドを用いた並列処理のプログラムを２種類の書き方で書けるようになる

② 排他制御と同期制御を行うスレッドのプログラムを書けるようになる

［キーワード］

スレッド、Thread クラス、Runnable インタフェース、
排他制御、同期制御、スタック

▶ 問題

1

文字列を表示するスレッド用クラス PrintGreet1 と、３つのスレッドを生成する実行用クラス Greeting1 を作ります。PrintGreet1 クラスがもつメンバ変数とメソッドは以下の表のとおりです。

メンバ変数	
private String msg	表示する文字列
メソッド	
public PrintGreet1(String msg)	変数 msg に引数を設定する
public void run()	変数 msg を１秒おきに３回表示する

３つのスレッドを使って、「おはよう」「おやすみ」「ありがとう」を３回ずつ表示します。PrintGreet1.java と Greeting1.java にコードを追加して、プログラムを完成させてください。なおこの問題では、Thread クラスを継承して並列処理を実現するものとします。

PrintGreet1.java

```
1  public class PrintGreet1 /* ここを埋める */{
2    private String msg;
3    public PrintGreet1(String msg){
4      this.msg = msg;
5    }
6    public void run(){
7      for(int i = 0; i < 3; i++){
8        System.out.println(msg);
9        try {
10         Thread.sleep(1000);
11       } catch(InterruptedException e) {
12         e.printStackTrace();
13       }
14     }
15   }
16 }
```

Greeting1.java

```
1  public class Greeting1{
2    public static void main(String[] args){
3      PrintGreet1 pg1 = /* ここを埋める */
4      PrintGreet1 pg2 = /* ここを埋める */
5      PrintGreet1 pg3 = /* ここを埋める */
6      /* ここを埋める */
7      /* ここを埋める */
8      /* ここを埋める */
9    }
10 }
```

実行例

```
>java Greeting1
おはよう
おやすみ
ありがとう
おはよう
おやすみ
ありがとう
おはよう
おやすみ
ありがとう
```

2

Runnable インタフェースを実装する方法で問題 1 と同じ動作をするプログラムを作ります。PrintGreet1.java と Greeting1.java をコピーして PrintGreet2.java と Greeting2.java を作成し、クラス名を PrintGreet2 と Greeting2 に変更してください。

実行例

```
>java Greeting2
おはよう
おやすみ
ありがとう
おはよう
おやすみ
ありがとう
おはよう
おやすみ
ありがとう
```

3

複数の文字列を**スタック**（下※参照）形式で保管できるオブジェクトに対して、２つのスレッドも用いて文字列を出し入れするプログラムがあります。

※**スタック**：後に格納したものを先に取り出すデータ構造のこと。スタックにデータを格納することをプッシュ（push）、スタックからデータを取り出すことをポップ（pop）という

プログラムを構成するクラスは、以下の表のとおりです。

MsgBox クラス	３つの文字列を保管するクラス スタックに文字列を格納する pushMsg(String)メソッドとスタックから文字列を取り出す popMsg()メソッドをもつ
PushMsg クラス	MsgBox オブジェクトに５つの文字列を格納するスレッドクラス
PopMsg クラス	MsgBox オブジェクトから５つの文字列を取り出すスレッドクラス
MsgBoxSample クラス	実行用クラス MsgBox オブジェクトを生成して、PushMsg クラスのスレッドと PopMsg クラスのスレッドを開始する

現状のプログラムを実行すると、MsgBox オブジェクトのメンバ変数 msgBox において配列の範囲外にアクセスしてしまう例外 ArrayIndexOutOfBoundsException が発生することがあります。プログラムが正しく動作するように MsgBox.java を修正してください。なお、PushMsg.java、PopMsg.java、MsgBoxSample.java は提供ファイルをそのまま使うものとします。

MsgBox.java

```
1  public class MsgBox{
2    private int index = 0;
3    private String[] msgBox = new String[3];
4    public void pushMsg(String msg){
5      msgBox[index] = msg;
6      System.out.println(msg + "を格納");
7      index++;
8    }
9    public String popMsg(){
10     index--;
11     return msgBox[index];
12   }
13 }
```

PushMsg.java

```
1  public class PushMsg extends Thread{
2    private MsgBox msgBox;
3    String[] msg = {"おはよう", "おやすみ", "ただいま", "おかえり", "ありがとう"};
4
5    public PushMsg(MsgBox msgBox){
6      this.msgBox = msgBox;
7    }
8    public void run(){
9      for(int i = 0; i < 5; i++) {
10       msgBox.pushMsg(msg[i]);
11       try {
12         Thread.sleep((int)(Math.random() * 1000));
13       } catch(InterruptedException e) {
14         e.printStackTrace();
15       }
16     }
17   }
18 }
```

PopMsg.java

```
1  public class PopMsg extends Thread{
2    private MsgBox msgBox;
3
4    public PopMsg(MsgBox msgBox){
5      this.msgBox = msgBox;
6    }
7    public void run(){
8      for(int i = 0; i < 5; i++) {
9        String msg = msgBox.popMsg();
10       System.out.println(msg + "を取り出し");
11       try {
12         Thread.sleep((int)(Math.random() * 1000));
13       } catch(InterruptedException e) {
14         e.printStackTrace();
```

```
15          }
16        }
17      }
18  }
```

MsgBoxSample.java

```
1   public class MsgBoxSample{
2     public static void main(String[] args){
3       MsgBox msgBox = new MsgBox();
4
5       PushMsg push = new PushMsg(msgBox);
6       push.start();
7
8       PopMsg pop = new PopMsg(msgBox);
9       pop.start();
10    }
11  }
```

実行例（プログラム修正前）

```
>java MsgBoxSample
Exception in thread "Thread-1" java.lang.ArrayIndexOutOfBoundsException:
Index -1 out of bounds for length 3
        at MsgBox.popMsg(MsgBox.java:11)
        at PopMsg.run(PopMsg.java:9)
おはようを格納
おやすみを格納
ただいまを格納
おかえりを格納
Exception in thread "Thread-0" java.lang.ArrayIndexOutOfBoundsException:
Index 3 out of bounds for length 3
        at MsgBox.pushMsg(MsgBox.java:5)
        at PushMsg.run(PushMsg.java:10)
```

実行例（プログラム修正後）

```
>java MsgBoxSample
おはようを格納
おはようを取り出し
おやすみを格納
ただいまを格納
ただいまを取り出し
おやすみを取り出し
おかえりを格納
おかえりを取り出し
ありがとうを格納
ありがとうを取り出し
```