ＪａｖａプログラミングⅡ

１１回目　スレッド　課題

確認○×問題

次の各文は正しいか誤っているか答えなさい。

⑴　スレッドは、１つの実行箇所をもつ一連の処理の流れである
⑵　マルチスレッドで各スレッドの処理は並行して実行される
⑶　Java はマルチスレッド処理を記述できない
⑷　スレッドを生成する場合、Thread クラスを継承し、かつ Runnable インタフェースを実装する必要がある
⑸　スレッドで実行する処理は Thread クラスまたは Runnable インタフェースの run() メソッドをオーバーライドして記述する
⑹　複数のスレッドは、それを開始した順番に終了する
⑺　同期とは、複数のスレッドの処理を互いに排他的に行うことである
⑻　メソッドの修飾子に synchronized を付加するとそのメソッドの処理は排他的になる

課題１　２のｎ（１～３０）乗を順次求めて一覧表示する処理をスレッドで実行しなさい。

（実行例）

```
>java Assignments11_1
結果をお待ちください。
２の１乗は２です。
２の２乗は４です。
２の３乗は８です。
      :
２の 30 乗は 1073741824 です。
 -- Press any key to exit (Input ″c″ to continue) --
```

（コード例）

```
class TwoToPowerOf extends Thread{

    ここに、２のｎ（１～３０）乗を順次求めて
    一覧表示する処理がスレッドで実行されるように記述する。

}

class Assignments11_1{
    public static void main(String[] args){
        // 別スレッドを実行
        TwoToPowerOf ttp=new TwoToPowerOf();
        ttp.start();
        // メインスレッドはここで終わり
        System.out.println(″結果をお待ちください。″);
    }
}
```

課題２　フィボナッチ数の計算と表示を行うスレッドを宣言しなさい。メインメソッドでキーボードから求めたいフィボナッチ数列の項を入力した後、スレッドを生成しフィボナッチ数を求めなさい。以下に、フィボナッチ数の計算をスレッドで行うクラスの宣言とメインメソッドのコードを示す。

（フィボナッチ数列）
1, 1, 2, 3, 5, 8, ・・・・・

漸化式
$n_1 = 1$、$n_2 = 1$、$n_k = n_{k-1} + n_{k-2}$ （k≧3）

（フィボナッチ数列をスレッドで計算するクラス）

```
class Fibonacci extends Thread
{
    private int v1=1, v2=0, v3;
    private int n; // 求めたい項（１以上）

    // 求めたい項をコンストラクタで設定
    public Fibonacci(int i){
        n=i;
    }

    // 与えられた項のフィボナッチ数を求める
    public int getTerm(){
        for(int i=0;i<n;i++){
            v3=v1+v2;
            v1=v2;
            v2=v3;
        }
        return v3;
    }

    // フィボナッチ数を別スレッドで計算
    public void run(){
        ここにフィボナッチ数を計算し、結果を表示するコードを
        記述してください
        メンバの getTerm() をうまく用いましょう
    }
}
```

（メインメソッド）

```
class Assignment11_2{
    public static void main(String[] args) throws IOException{
        InputStreamReader isr=new InputStreamReader(System.in);
        BufferedReader br=new BufferedReader(isr);

        System.out.println("フィボナッチの求めたい項を入力してください。");
        String str=br.readLine();
        int n=Integer.parseInt(str);

        ・・・・
        ここにフィボナッチを計算するスレッドを作成して
        処理を開始するコードを記述してください

        System.out.println("結果をお待ちください。");
    }
}
```

（実行例）

```
>java Assignment11_2
フィボナッチの求めたい項を入力してください。
30                                          ←（キーボード入力）
結果をお待ちください。                      ←（メインメソッドはここで終了）
フィボナッチ数列 30 項目は 832040 です。    ←（スレッドによる出力）
 -- Press any key to exit (Input "c" to continue) --
```

課題３　銀行口座への操作には、預金や払い戻し、振込みなどがある。一般に、複数の利用者が同一の口座へ並行してアクセスすることは容易に想定される。例えば、ある預金者 A が本人の口座へ預金を行うのと並行して、他の利用者がその預金者 A の口座へ振り込みを行うなど。このように並行して起こる処理はスレッドを用いると比較的容易に記述できる。次の内容をシミュレーションするコードをスレッドを用いて作成しなさい。

シミュレーションの内容：

１）　会社Ａがある銀行に口座 a を持つ。

２）　会社Ａの３人の社員が並行して次のように口座 a へ預金と払戻を行う。

社員１　１００万円預金して７５万円払戻を行う

社員２　２０万円預金して２５万円払戻を行う

社員３　５０万円預金して２０万円払戻を行う

３）　銀行は各社員からの処理を整合性を取りながら実行する(synchronized)。

ヒント）教科書 p. 483 の Sample7「車会社と運転手」の例題を参考にしなさい。
会社クラス　→　会社 A の銀行口座クラス
運転手クラス（スレッド）　→　社員クラス（スレッド）

課題４　次は、キーボードから２つの整数を入力して加算をスレッドで実行するクラスです。

（加算をスレッドで計算するクラス）

```
import java.io.*;
class Addition extends Thread
{
    public void run(){
        int num1=0,num2=0;

        // キーボードの準備
        InputStreamReader isr=new InputStreamReader(System.in);
        BufferedReader br=new BufferedReader(isr);

        // ２つの整数の入力
        System.out.println("【加算】２つの整数を入力してください。");
        try{
            num1=Integer.parseInt(br.readLine());
            num2=Integer.parseInt(br.readLine());
        }catch(IOException e){
        }

        // 結果出力
        System.out.println(num1+"+"+num2+"="+(num1+num2));
    }
}
```

この例を参考にして、

１．キーボードから数値を入力して、なんらかの計算を行い、結果を表示するをスレッドで（run()メソッド内で）実行するクラスを各自宣言しなさい。

２．メインメソッドからこのスレッドを実行しなさい。

例の加算のスレッドを実行するメインメソッドを参考として以下に示します。

（メインメソッド）

```
class Assignment11_4
{
    public static void main(String[] args){
        // 別スレッドを実行
        System.out.println("スレッドを開始します。");
        Addition add=new Addition();
        add.start();

        // メインスレッドはここで終わり
        System.out.println("メインメソッドを終了します。");
    }
}
```