

課題 1 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

$$n_1 = 1, n_2 = 1, n_k = n_{k-1} + n_{k-2} \quad (k \geq 3)$$

(フィボナッチ数列を計算するクラス)

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

    // 求めたい項は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;
    }
}
```

```

(メインメソッド)
import java.io.*;
class Assignment11_1{
    public static void main(String[] args) throws IOException{
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

        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);

        // ここで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。

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

```

(実行例)

```

>java Assignment11_1
フィボナッチの求めたい項を入力してください。
30
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2 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)