

Chapter 7

マルチスレッドプログラミング(1)

以下の指示に従って、クラス、オブジェクトを作成・実行せよ。なお、文中では個別に指示しないが、変更を行ったら適宜コンパイルすること。

7.1 マルチスレッドプログラミング入門

7.1.1 最初のマルチスレッドプログラミング

- まず、次のクラスを作成せよ。

```
class ThreadMain {  
    public static void main(String[] args) {  
        ThreadSmpl shoriA = new ThreadSmpl("A");  
        ThreadSmpl shoriB = new ThreadSmpl("B");  
        shoriA.start();  
        shoriB.start();  
        System.out.println("Done.");  
    }  
}
```

- このクラスではThreadSmplクラスのオブジェクトが生成されているが、このクラスは次の仕様で作成すること。
 - Threadクラスを継承せよ。

- Stringクラスのフィールドnameを作成する。また、コンストラクタの引数の文字列をnameフィールドに代入せよ。
- ThreadMainクラスのmainメソッドでstartメソッドが実行される際には、「nameフィールドの値+“:”+繰り返し回数+“回目”」という文字列を画面に繰り返し表示するよう処理を実装せよ。ただし、繰り返しは10回行うものとする。¹
- ThreadMainクラスを実行するとどのような結果が得られるか。このプログラムがシングルスレッドプログラミングであれば“Done.”は必ず最後に表示されるはずだが、これらがどの段階で表示されているかを確認せよ。（処理系等により多少結果が代わることがある）

7.1.2 別の方法を用いた実現方法

- ThreadSmplクラスをコピーし、ThreadSmpl2という名前にせよ。（コンストラクタの定義についてもあわせて変更せよ）
- ThreadSmpl2クラスはThreadクラスを継承せず、Runnableインターフェイスを実装することとする。そのようにクラスを変更せよ。
- 7.1.1のThreadMainクラスをコピーしてThreadMain2クラスとせよ。ThreadMain2クラスのmainメソッドではThreadSmplオブジェクトの代わりにThreadSmpl2オブジェクトを生成し、ThreadMainクラスのmainメソッドを実行したときに同じ結果が得られるよう、その内容を変更せよ。

7.1.3 スレッドの休止・待機

- ThreadSmplクラスをコピーし、ThreadSmpl3という名前にせよ。（コンストラクタの定義についてもあわせて変更せよ）
- コンストラクタの引数を二つに変更せよ。ただし、第一引数はStringクラスのオブジェクト、第二引数はint型の値とする。また、これに伴い、フィールドにint型変数*i*を追加し、コンストラクタでは第二引数の値をフィールド*i*に代入することとする。

¹この処理は何という名前のメソッドに実装する必要があるだろうか。startメソッドではないことに注意。

- 「nameフィールドの値+“:”+繰り返し回数+“回目”」を繰り返し表示する際に、一度表示するたびに*i* ミリ秒休止するよう処理を追加せよ。
- 7.1.1のThreadMainクラスをコピーしてThreadMain3クラスとせよ。また、ThreadMain3クラスのmainメソッドではThreadSmplオブジェクトの代わりにThreadSmpl3オブジェクトを生成し、shoriAオブジェクトの休止時間を1000ミリ秒、shoriBオブジェクトの休止時間を1500ミリ秒となるよう変更を行え。また、ThreadMain3クラスを実行し、その様子を確認せよ。
- ThreadMain3では、shoriAおよびshoriBのそれぞれのクラスで定義されたスレッドが終了したのちに“Done.”が表示されるよう処理を追加せよ。