



Introduction 01

문제 11-1의 해답

- ... (1) Java 프로그램에서는 적어도 한 개의 쓰레드가 동작하고 있다.
- ×... (2) Thread 클래스의 run 메소드를 호출하면 새로운 쓰레드가 기동한다.
 - 새로운 쓰레드를 기동하는 것은 run 메소드가 아니라 start 메소드입니다.
- ×... (3) Runnable 인터페이스에는 start 메소드와 run 메소드가 선언되어 있다.
 - Runnable 인터페이스에 선언되어 있는 것은 run 메소드뿐입니다.
- ... (4) 동일한 인스턴스의 메소드가 복수의 쓰레드로부터 호출되는 경우도 있다.
 - 따라서 쓰레드의 배타제어가 필요한 것입니다.
- ... (5) Thread 클래스의 한 개 인스턴스 메소드가 여러 쓰레드로부터 호출되는 경우도 있다.
 - Thread 클래스의 인스턴스 메소드라고 해서 다른 클래스의 인스턴스 메소드와 다를 게 없습니다. 여러 쓰레드로부터 호출되는 경우도 있을 수 있습니다.
- ×... (6) sleep 메소드를 실행하면 지정한 시간만큼 모든 쓰레드가 일시 정지한다.
 - 일시 정지하는 것은 sleep 메소드를 호출한 쓰레드(현재의 쓰레드)뿐입니다.
- ×... (7) 어떠한 쓰레드가 synchronized 메소드를 실행하고 있는 동안에는 다른 모든 쓰레드가 실행을 정지한다.
 - 실행을 정지하는 것은 같은 인스턴스의 락을 취하려고 했던 쓰레드 뿐입니다.
- ×... (8) sleep 메소드를 실행하면 그 쓰레드는 지정한 시간만큼 wait 셋에 들어간다.
 - sleep 메소드로는 쓰레드가 wait 셋에 들어가지 않습니다. wait 셋에 들어가는 것은 wait 메소드를 실행했을 때입니다.

✕... (9) wait 메소드를 호출하는 문은 synchronized 메소드 안에 적어야 한다.

→ wait 메소드를 호출하는 문은 synchronized 메소드 안에 적으나, synchronized 블록 안에 적으나, 또한 거기에서 호출하는 별도의 메소드 안에 적으나 아무 상관이 없습니다. 요컨대 wait 메소드를 실행하는 스레드가 실행할 때 대상이 되는 인스턴스의 락을 취하고 있으면 되는 것입니다.

○... (10) notifyAll 메소드는 java.lang.Object 클래스의 인스턴스 메소드이다.

문제 11-2의 해답

클래스 라이브러리 안에서 스레드의 배타제어가 적절하게 이뤄지고 있기 때문입니다. 즉, 한 개의 스레드가 Good!이라고 하는 문자열을 표시하고 있는 중간에 Nice!를 표시하는 별도의 스레드는 자신이 표시할 수 있기를 기다립니다.

문제 11-3의 해답

메인 스레드가 호출하고 있는 것이 start 메소드가 아니라 run 메소드이기 때문입니다.

`new PrintThread("*").run();`라는 문에서는 `PrintThread` 클래스의 인스턴스를 생성하고 그 인스턴스의 `run` 메소드를 실행하고 있습니다. 그러나 `run` 메소드는 새로운 스레드가 아닌 메인 스레드가 실행하고 있습니다.

1000개의 *가 표시된 다음에서야 다음 문이 실행되는 것입니다.

```
new PrintThread("+").run();
```

결국, 모든 표시는 메인 스레드라는 한 개의 스레드가 실행하게 되며, 이 프로그램은 싱글 스레드 프로그램이 됩니다.



문제1-4의 해답

확인용 프로그램은 다음과 같습니다. 단, 타이밍에 따라서는 체크까지 오랜 시간이 걸릴 수 있습니다.

리스트 11-1 예금잔고가 마이너스가 될 가능성이 있음을 확인하는 Main 클래스 (Main.java)

```
public class Main {  
    public static void main(String[] args) {  
        Bank bank = new Bank("A Bad Bank", 1000); // 1000원의 은행계좌를 만든다  
        new ClientThread(bank).start();  
        new ClientThread(bank).start();  
    }  
}
```

⇒ 예제파일 경로 : 부록CD/src/Introduction1/A11-4

리스트 11-2 1000원 인출과 예금을 반복하는 ClientThread 클래스 (ClientThread.java)

```
public class ClientThread extends Thread {  
    private Bank bank;  
    public ClientThread(Bank bank) {  
        this.bank = bank;  
    }  
    public void run() {  
        while (true) {  
            boolean ok = bank.withdraw(1000);  
            if (ok) {  
                bank.deposit(1000);  
            }  
        }  
    }  
}
```

⇒ 예제파일 경로 : 부록CD/Introduction1/A11-4

그림 11-1 실행 예

예금잔고가 마이너스입니다! money = - 1000
(CTRL+C로 종료)

문제1-5의 해답

`t.sleep(1000);`을 실행했을 때 일시 정지하는 것이 `t`와 관련된 쓰레드가 아니라 이 문을 실행한 쓰레드입니다.

```
t.sleep(1000);
```

이 호출하는 것은 `t`의 인스턴스 메소드가 아닌 `Thread`의 클래스 메소드입니다. 즉, 다음 문을 실행하는 것과 똑같아집니다.

```
Thread.sleep(1000);
```

새로 기동한 쓰레드를 일시 정지하고 싶다면 `MyThread` 클래스의 `run` 메소드 안에서 `sleep` 메소드를 호출합니다.

잠깐! 한 마디 : 인스턴스를 경유하여 클래스 메소드를 호출하는 것은 오해를 초래한다

이 문제처럼 인스턴스를 매개하여 클래스 메소드를 호출하는 방식은 문법상으로 틀린 곳이 없지만 프로그램을 읽는 사람에게 오해를 불러일으킬 수 있습니다. 그러므로 실제 프로그램에서 이렇게 적는 건 삼가하기 바랍니다.

```
t.sleep(1000);    X 오해를 불러일으키는 방식
```

```
Thread.sleep(1000); O 올바른 방식
```

문제 11-6의 해답

○... (1) `x.iA()`;

와

`x.iA()`;

`synchronized`가 아닌 메소드는 복수의 쓰레드에서 언제든지 실행할 수 있습니다.

○... (2) `x.iA()`;

와

`x.iB()`;

`synchronized`가 아닌 메소드는 복수의 쓰레드에서 언제든지 실행할 수 있습니다.



○... (3) x.iA();

와

x.iSyncA();

synchronized가 아닌 메소드는 복수의 쓰레드에서 언제든지 실행할 수 있습니다. 가령 다른 synchronized 메소드를 실행하고 있는 쓰레드가 있더라도 실행 가능합니다.

✕... (4) x.iSyncA();

와

x.iSyncA();

같은 인스턴스의 synchronized한 인스턴스 메소드는 동시에 한 개의 쓰레드에서만 실행됩니다.

✕... (5) x.iSyncA();

와

x.iSyncB();

동일한 인스턴스의 synchronized한 인스턴스 메소드는 동시에 한 개의 쓰레드에서만 실행됩니다.

○... (6) x.iSyncA();

와

y.iSyncA();

인스턴스가 다르면 락이 다르기 때문에 synchronized한 인스턴스 메소드라도 여러 쓰레드에서 동시에 실행할 수 있습니다.

○... (7) x.iSyncA();

와

y.iSyncB();

인스턴스가 다르면 락이 다르기 때문에 synchronized한 인스턴스 메소드라도 여러 쓰레드에서 동시에 실행할 수 있습니다.

○... (8) x.iSyncA();

와

Something.cA();

원래부터 클래스 메소드가 synchronized 메소드가 아니므로 실행 가능합니다.

○... (9) x.iSyncA();

와

Something.cSyncA();

synchronized 한 인스턴스 메소드와 synchronized한 클래스 메소드는 락이 다르기 때문에 복수 쓰레드에서 동시에 실행할 수 있습니다.

×... (10) Something.cSyncA();

와

Something.cSyncA();

synchronized한 클래스 메소드끼리는 복수 쓰레드에서 동시에 실행할 수 없습니다.

×... (11) Something.cSyncA();

와

Something.cSyncB();

synchronized한 클래스 메소드끼리는 복수 쓰레드에서 동시에 실행할 수 없습니다.

×... (12) x.cSyncA();

와

y.cSyncB();

x.cSyncA()가 호출하는 메소드는 Something.cSyncA();이고, y.cSyncB();가 호출하는 메소드는 Something.cSyncB();이므로 양측 모두 synchronized한 클래스가 됩니다. 따라서 복수 쓰레드에서 동시에 실행할 수 없습니다.

주의 ... 문제 (12)에서는 해설을 위해 「클래스 메소드를 인스턴스를 경유하여 호출한다」고 하는 문을 적었습니다. 그러나 읽는 사람의 오해를 불러일으킬 우려가 있으므로 실제 프로그램에서는 사용하지 않는 것이 좋습니다.