

실습문제 - Thread 문제

* 다음 괄호안에 알맞은 답을 적으세요.

1. 자바 스레드는 Thread 클래스를 상속받는 방법과 () 인터페이스를 구현 하는 방법으로 만들 수 있다.
2. Thread 클래스를 상속받아 스레드로 실행할 코드는 () 메소드에 작성한다. 스레드 객체 생성 후 () 메소드를 호출해야 스레드로 작동한다.
- 3.()를 상속받는 모든 클래스는 wait(), notify() 메소드를 가지고 있으며, 한 스레드가 wait()를 호출하면 다른 스레드가 notify()를 호출할 때까지 대기한다. 이런 방식으로 두 스레드가 서로 대화하면서 공유 데이터에 대한 정상적인 접근을 유지한다.
4. 자바 스레드를 만들기 위해서 Thread를 상속받아 오버라이딩해야 하는 메소드는?
5. 자바 스레드가 sleep()하는 동안 발생할 수 있는 예외에 대비하여 try~catch 블록을 만들어야 하는데 이 예외는 무엇인가?
6. 한번 종료한 스레드의 Thread 객체를 이용하여 스레드를 다시 생성할 수 있는가?
7. Thread 클래스의 run() 메소드에 작성된 return문은 어떤 역할을 하는가?
8. 자바에서 2개 이상의 스레드가 동시에 실행할 가능성이 있는 코드 블록에, 스레드가 하나씩 순차적으로 실행되도록 지시하는 키워드는 무엇인가?
9. 스레드 동기화 어떤 경우에 필요한가?
 - ① 멀티스레드가 공유 데이터에 접근할 때
 - ② 스레드의 우선순위를 높일 때
 - ③ 스레드가 종료하는 시간을 맞출 때
 - ④ 스레드를 타이머처럼 작동 시킬 때
10. 어떤 객체 a의 wait() 메소드를 호출하여 대기 상태인 스레드는 어떤 경우에 깨어나서 실행을 계속할 수 있는가? 1
 - ① 다른 스레드가 객체 a의 notify()를 호출할 때
 - ② 객체 a가 소멸될 때
 - ③ 일정한 시간이 지나면
 - ④ 다른 스레드가 객체 a의 wakeup()을 호출할 때

11. 다음은 Runnable 인터페이스를 이용하여 스레드를 작성한 코드이다. 스레드는 작동 10초 후에 종료한다. 빈칸을 채워라

```
public class MyThread implements Runnable{
    public void run() {
        try {
            _____ // 1초동안 잠잔다.
        } catch (InterruptedException e) {
            return ;
        }
    }
}

public static void main(String[] args){
    Thread th=new Thread(_____);
    _____ // 스레드를 실행한다.
}
```

12. 스레드에 대한 설명 중 틀린 것은 무엇입니까?

- ① 자바 애플리케이션은 메인(main) 스레드가 main() 메소드를 실행시킨다.
- ② 작업 스레드 클래스는 Thread 클래스를 상속해서 만들 수 있다.
- ③ Runnable 객체는 스레드가 실행해야 할 코드를 가지고 있는 객체라고 볼 수 있다.
- ④ 스레드 실행을 시작하려면 run() 메소드를 호출해야 한다.

13. 스레드의 우선순위에 대한 설명 중 틀린 것은 무엇입니까?

- ① 우선순위가 높은 스레드가 실행 기회를 더 많이 가질 수 있다.
- ② 우선순위는 1부터 10까지 줄 수 있는데, 디폴트는 5이다.
- ③ Thread 클래스는 NORM_PRIORITY, MIN_PRIORITY, MAX_PRIORITY 상수를 제공한다.
- ④ 1은 가장 높은 우선순위이기 때문에 다른 스레드보다 실행 기회를 더 많이 갖는다.

14. 동기화 메소드와 동기화 블록에 대한 설명 중 틀린 것은 무엇입니까? 2

- ① 동기화 메소드와 동기화 블록은 싱글(단일) 스레드 환경에서는 필요 없다.
- ② 스레드가 동기화 메소드를 실행할 때 다른 스레드는 일반 메소드를 호출할 수 없다.
- ③ 스레드가 동기화 메소드를 실행할 때 다른 스레드는 다른 동기화 메소드를 호출할 수 없다.
- ④ 스레드가 동기화 블록을 실행할 때 다른 스레드는 다른 동기화 메소드를 호출할 수 없다.

15. 스레드 일시 정지 상태에 대한 설명 중 틀린 것은 무엇입니까?

- ① 일시 정지 상태는 BLOCKED, WAITING, TIMED_WAITING이 있다.
- ② 스레드가 동기화 메소드를 실행할 때 다른 스레드가 동기화 메소드를 호출하게 되면 BLOCKED 일시 상태가 된다.

- ③ 동기화 메소드는 내에서 wait()를 호출하면 WAITING 일시 정지 상태가 된다.
- ④ yield() 메소드를 호출하면 TIMED_WAITING 일시 정지 상태가 된다.

16. 스레드 상태 제어를 하는 메소드에 대한 설명 중 틀린 것은 무엇입니까?

- ① yield() 메소드를 호출한 스레드는 동일한 우선순위나 높은 우선순위의 스레드에게 실행 기회를 양보하고 자신은 실행 대기 상태가 된다.
- ② sleep() 메소드를 호출한 스레드는 주어진 시간 동안 일시 정지 상태가 된다.
- ③ stop() 메소드는 스레드를 즉시 종료시키기 때문에 스레드 안정성에 좋지 못하다.
- ④ join() 메소드를 호출한 스레드가 종료할 때까지 join() 메소드를 멤버로 가지는 스레드는 일시 정지 상태가 된다. // 내용 뒤바뀜

17. interrupt() 메소드를 호출한 효과에 대한 설명 중 틀린 것은 무엇입니까?

- ① 일시 정지 상태에서 InterruptedException을 발생 시킨다.
- ② 스레드는 즉시 종료한다.
- ③ 실행 대기 상태에서 호출되면 일시 정지 상태가 될 때까지 InterruptedException이 발생하지 않는다.
- ④ 아직 InterruptedException이 발생하지 않았다면 interrupted(), isInterrupted() 메소드는 true를 리턴한다.

18. wait()와 notify() 메소드에 대한 설명 중 틀린 것은 무엇입니까?

- ① 스레드가 wait()를 호출하면 일시 정지 상태가 된다.
- ② 스레드가 notify()를 호출하면 wait()로 일시 정지 상태에 있던 다른 스레드가 실행 대기 상태가 된다.
- ③ wait()와 notify()는 동기화 메소드 또는 블록에서 호출할 필요가 없다.
- ④ 스레드가 wait(long maillis)를 호출하면 notify()가 호출되지 않아도 주어진 시간이 지나면 자동으로 실행 대기 상태가 된다.

19. while문으로 반복적인 작업을 하는 스레드를 종료시키는 방법에 대한 설명 중 최선의 방법이 아닌 것은?

- ① stop() 메소드를 호출해서 즉시 종료시킨다.
- ② 조건식에 boolean 타입의 stop 플래그를 이용해서 while문을 빠져 나가게 한다.
- ③ 스레드가 반복적으로 일시 정지 상태가 된다면 InterruptedException을 발생시켜 예외 처리 코드에서 break문으로 while문을 빠져나가게 한다.
- ④ 스레드가 일시 정지 상태로 가지 않는다면 isInterrupted()나 interrupted() 메소드의 리턴값을 조사해서 true일 경우 break문으로 while문을 빠져나가게 한다.

20. 메인 스레드에서 1초 후 MovieThread의 interrupt() 메소드를 호출해서 MovieThread를 안전하게 종료하고 싶습니다. 비어 있는 부분에 적당한 코드를 작성하세요.

```
class MovieThread extends Thread{
    @Override
    public void run() {
        while(true){
            System.out.println("동영상을 재생합니다.");
            _____
        }
    }
}

public class Quiz20{
    public static void main(String[] args) {
        MovieThread movieThread =new MovieThread ();
        movieThread.start();
        try { Thread.sleep(1000); } catch (InterruptedException e) { }

        movieThread.interrupt
    }
}
```

21. 메인 스레드가 종료하면 MovieThread도 같이 종료되게 만들고 싶습니다. 비어 있는 부분에 적당한 코드를 넣어보세요.

```
class MovieThread extends Thread{
    @Override
    public void run() {
        while(true){
            System.out.println("동영상을 재생합니다.");
            try { Thread.sleep(1000); } catch (InterruptedException e) { }
        }
    }
}

public class Quiz21{
    public static void main(String[] args) {
        Thread thread=new MovieThread();
        _____
        thread.start();

        try { Thread.sleep(3000); } catch (InterruptedException e) { }
    }
}
```