실력 상승! 연습 문제

Q1 다음과 같이 Thread 클래스를 상속한 클래스 MyThread가 있다. 이때 새로운 쓰레드 객체를 생성한 후 쓰레드를 실행하는 코드를 작성하시오(참조 변수명은 자유롭게 작성 가능).

```
class MyThread extends Thread {
    @Override
    public void run() {
        for(int i = 1; i <= 5; i++) {
            try {Thread.sleep(1000);} catch (InterruptedException e) {}
            System.out.println(i + "초");
        }
    }
}
```

```
public static void main(String[] args) {

    // MyThread 객체를 이용한 쓰레드의 실행

    MyThread mt = new MyThread();
    mt.start();

}
```

실행 결과

```
1초
2초
3초
4초
5초
```

Q2 다음과 같이 Runnable 인터페이스를 구현한 클래스 MyRunnable이 있다. 이때 새로운 쓰레드 객체를 생성한 후 쓰레드를 실행하는 코드를 작성하시오(참조 변수명은 자유롭게 작성 가능).

```
class MyRunnable implements Runnable {
    @Override
    public void run() {
        for(int i = 1; i <= 5; i++) {
            try {Thread.sleep(1000);} catch (InterruptedException e) {}
            System.out.println(i + "초");
        }
    }
}
```

```
public static void main(String[] args) {
    // MyRunnable 객체를 이용한 쓰레드의 실행

    MyRunnable mr = new MyRunnable();
    Thread t = new Thread(mr);
    t.start();

}
```

실행 결과

```
1초
2초
3초
4초
5초
```

Q3 다음은 현재 쓰레드의 객체 참조 방법과 실행 쓰레드의 개수 그리고 쓰레드의 이름을 지정하거나 가져오는 기능을 포함한 코드다. 실행 결과를 쓰시오.

```
public static void main(String[] args) {

    // 현재 쓰레드 객체 가져오기 + 쓰레드 이름 가져오기 + 쓰레드의 개수
    Thread curThread = Thread.currentThread();
    System.out.println(curThread.getName());
    System.out.println(curThread.activeCount());

    // 새로운 쓰레드 생성 + 쓰레드 이름 가져오기
    Thread t1 = new Thread();
    System.out.println(t1.getName());

    // 두 번째 쓰레드 생성 + 쓰레드 이름 설정 + 쓰레드 이름 가져오기
    Thread t2 = new Thread();
    t2.setName("두 번째 쓰레드");
    System.out.println(t2.getName());

}
```

실행 결과

```
main
1
Thread-0
두번째 쓰레드
```

Q4 다음은 main 쓰레드 내에서 익명 이너 클래스 방법으로 쓰레드 객체를 생성해 쓰레드를 실행하는 코드다. 쓰레드는 데몬 쓰레드로 설정했다. 이때 실행 결과를 쓰시오.

```
public class EX04 {
    public static void main(String[] args) {
        Thread t = new Thread() {
            public void run() {
                for(int i = 1; i <= 5; i++) {
                    try {Thread.sleep(1000);} catch (InterruptedException e) {}
                    System.out.println(i);
                }
            }
        };
        t.setDaemon(true);
        t.start();

        try {Thread.sleep(3500);} catch (InterruptedException e) {}
    }
}
```

실행 결과

```
1
2
3
```

Q5 클래스 MyData 내의 modifyData() 메서드는 동기화돼 있다. 다음 코드의 실행 결과를 작성하시오.

```
class MyData {
    int data;
    synchronized void modifyData() {
        data++;
    }
}
class MyThread extends Thread {
    MyData myData;
    public MyThread(MyData myData) {
        this.myData = myData;
    }
    @Override
    public void run() {
        for(int i = 0; i < 10000; i++) {
            myData.modifyData();
        }
        System.out.println(myData.data);
    }
}
```

```
public static void main(String[] args) {
    MyData md = new MyData();
    MyThread mt1 = new MyThread(md);
    mt1.start();
    // 쓰레드 준비 과정 + 동기화 준비 과정을 위한 짧은 시간
    try {Thread.sleep(100);} catch (InterruptedException e) {}
    MyThread mt2 = new MyThread(md);
    mt2.start();
}
```

실행 결과

```
10000
20000
```

Q6 쓰레드에서 공유 객체로 사용할 MyData 클래스의 내부에는 메서드 동기화와 블록 동기화가 섞여 있다. 이때 2개의 쓰레드가 공유 객체를 사용할 때 동시 사용 가능 여부를 표기하시오.

```
class MyData {
    String name = "홍길동";
    synchronized void abc() {
    }
    void bcd() {
        synchronized(this) {
        }
    }
    void cde() {
        synchronized (name) {
        }
    }
}
```

Thread1	Thread2	동시 사용(가능/불가능)
abc()	bcd()	불가능
bcd()	cde()	가능
cde()	abc()	가능

Q7 다음 Thread 클래스는 내부에 Thread.sleep() 메서드를 호출하고 있다. 이때 다음과 같이 외부에서 interrupt() 메서드를 호출할 때 실행 결과를 쓰시오.

```
class MyThread extends Thread {
    @Override
    public void run() {
        try {
            System.out.println("첫 번째 출력");
            Thread.sleep(1000);
            System.out.println("두 번째 출력");
            Thread.sleep(1000);
            System.out.println("세 번째 출력");
            Thread.sleep(1000);
        } catch (InterruptedException e) {
            System.out.println("쓰레드 종료");
        }
    }
}
```

```
public static void main(String[] args) {
    MyThread mt = new MyThread();
    mt.start();
    try {Thread.sleep(1500);} catch (InterruptedException e) {}
    mt.interrupt();
}
```

실행 결과

```
첫 번째 출력
두 번째 출력
쓰레드 종료
```

Q8 클래스 MyData 내부에 "A"와 "B"를 출력하는 메서드를 1개씩 포함하고 있다. ThreadA와 ThreadB는 각각 MyData 내의 메서드를 0.5초 간격으로 5번 호출하는 클래스다. 이때 다음 코드의 실행 결과가 A → B → A → B와 같이 순서대로 나오도록 코드를 완성하시오.

```
class MyData {
    boolean flag = false;
    synchronized void printA(){

        if(flag)
        try { wait(); } catch (InterruptedException e) {}

        System.out.println("A");

        flag=true;
        notify();

    }
    synchronized void printB() {

        if(!flag)
        try { wait(); } catch (InterruptedException e) {}

        System.out.println("B");

        flag=false;
        notify();

    }
}

class ThreadA extends Thread {
    MyData myData;
    public ThreadA (MyData myData) {
        this.myData = myData;
    }
    @Override
    public void run() {
        for(int i = 0; i < 5; i++) {
            myData.printA();
```

```
            try {Thread.sleep(500);} catch (InterruptedException e) {}
        }
    }
}

class ThreadB extends Thread {
    MyData myData;
    public ThreadB (MyData myData) {
        this.myData = myData;
    }
    @Override
    public void run() {
        for(int i = 0; i < 5; i++) {
            myData.printB();
            try {Thread.sleep(500);} catch (InterruptedException e) {}
        }
    }
}
```

```
public static void main(String[] args) {
    MyData myData = new MyData();
    Thread t1 = new ThreadA(myData);
    Thread t2 = new ThreadB(myData);

    t1.start();
    t2.start();
}
```

실행 결과

```
A
B
A
B
A
B
A
B
A
B
```