



Thread

1초마다 자신의 이름을 출력하는 Thread 를 2개 생성. 단, 이름을 출력할 때 교대로 출력하는 것을 보장해야한다.

- 동기화 처리를 해야하기 때문에 공유객체를 써야한다.
- `wait, notify` 를 써야 한다.

```
package lecture0715;

class toName {
    public synchronized void printName(String toName) {

        try {
            for (int i = 0; i < 10; i++) {
                Thread.sleep(1000);
                System.out.println(Thread.currentThread().getName());
                notify();
                wait();
            }
        } catch (Exception e) {
            e.printStackTrace(); // 에러 메시지의 발생 근원지를 찾아 단계별로 에러 출력
        }
    }
}

class ThreadEx_10 implements Runnable {
    toName tn = new toName();

    @Override
    public void run() {
        tn.printName(Thread.currentThread().getName());
    }
}

public class ThreadExam10 {
    public static void main(String[] args) {
        ThreadEx_10 r = new ThreadEx_10();
        Thread t1 = new Thread(r);
        Thread t2 = new Thread(r);

        t1.setName("첫번째 Thread!!");
        t2.setName("두번째 Thread!!");

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

```
}  
}
```

2. Thread가 세개 이상 있을때 순서대로 찍는 LOGIC을 구현해보자!!

3. Thread 1,2 순서대로 출력하고 1초 쉬고 다시 순서대로 출력하고 1초 쉬는 LOGIC을 구현해보자!!