synchronized II

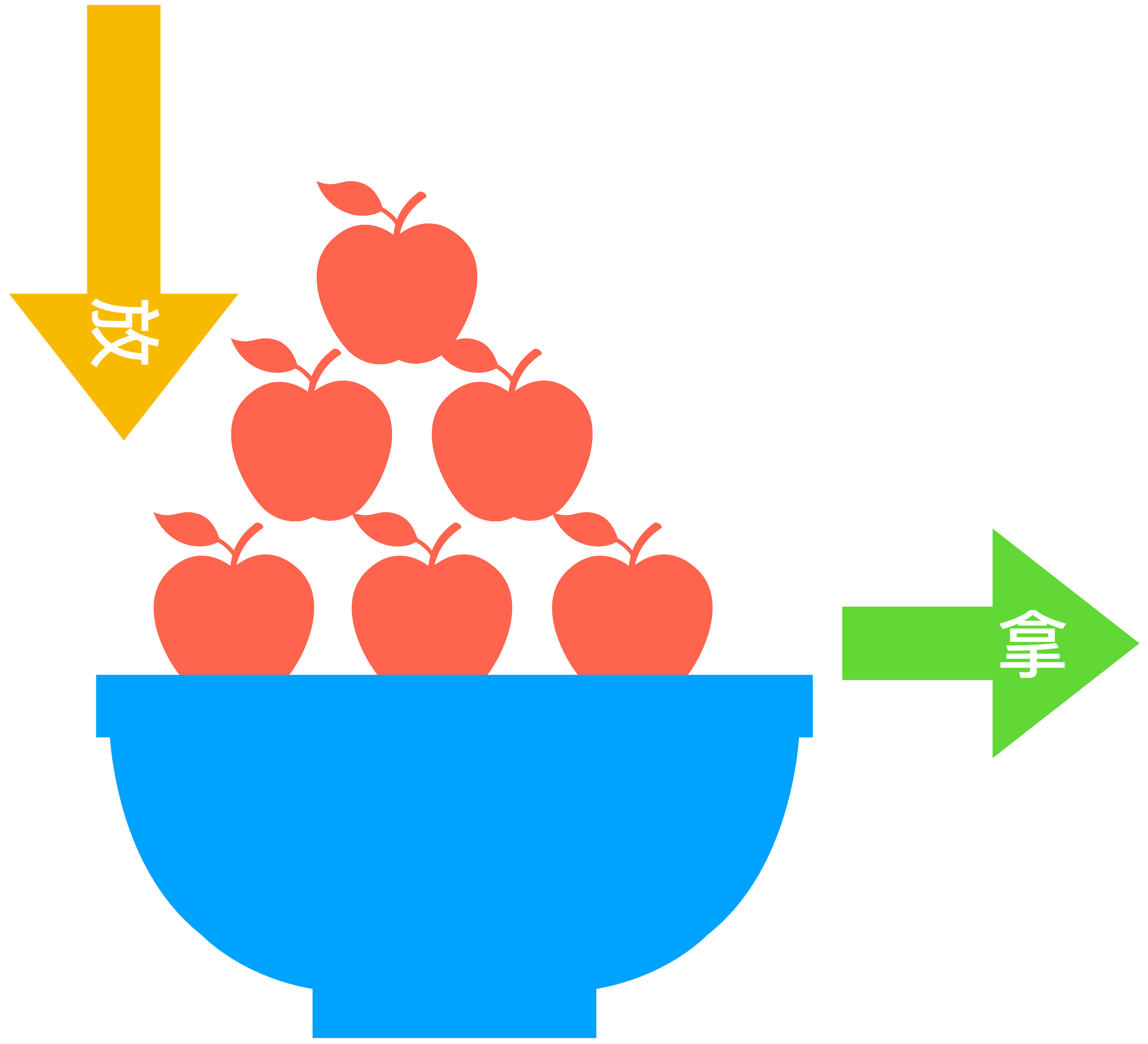
监视器 Monitor Object

monitor object 监视器

```java
private Object obj = new Object();

public void doA() throws InterruptedException{
    synchronized (obj) {
      // 代码写这
    }
}
```

monitor object 监视器



线程同步

- wait()   wait(long ms)   wait(long ms, int ns)
- notify
- notifyAll

java.lang.IllegalMonitorStateException

```java
private Object obj = new Object();

public void doA() throws InterruptedException{

    obj.wait(); ✗

}

public void doB(){

    obj.notify(); ✗

}
```

```java
private Object obj = new Object();

public void doA() throws InterruptedException{
    synchronized(obj){
        // do something
        obj.wait();
        // do something
    }
}
```

获得锁

暂时释放锁
待其他线程notify / notifyAll后，并且重新获得到锁后继续

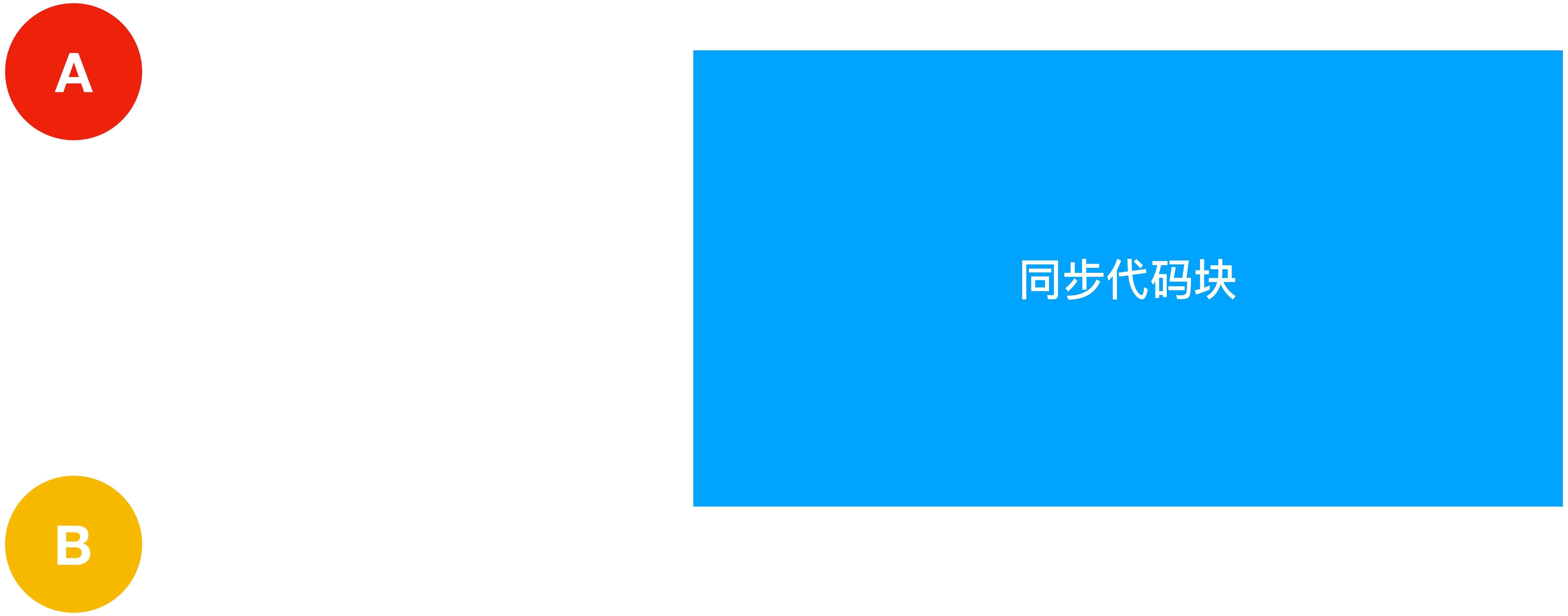
释放锁

```java
public void doB(){
    synchronized(obj){
        // do something
        obj.notify();
        // do something
    }
}
```

获得锁

notify一个wait此监视器(obj)的线程

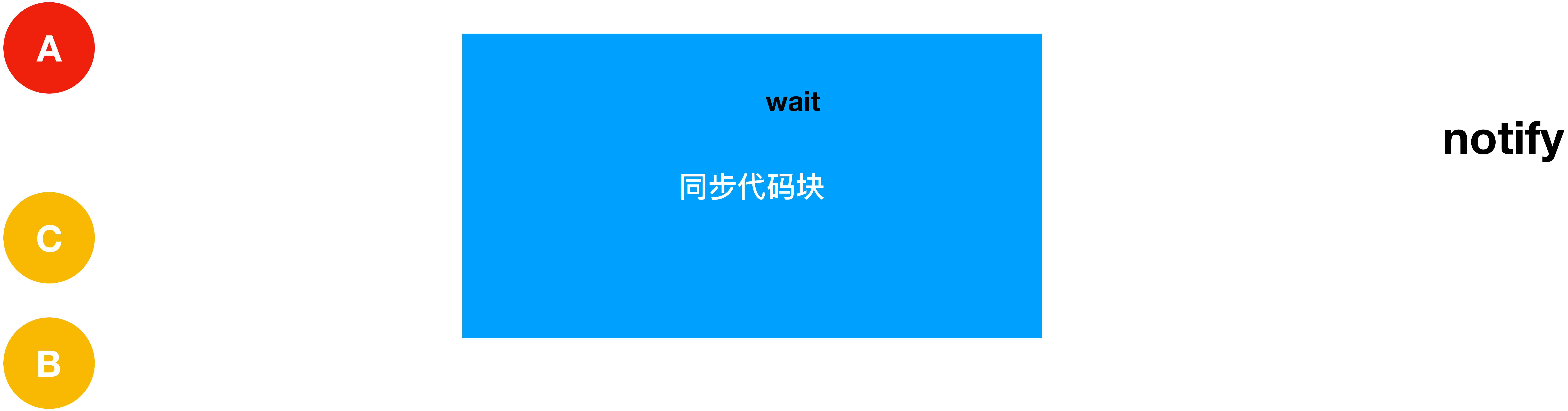
释放锁



A

同步代码块

B



A

C

B

wait

同步代码块

notify

```java
public class MonitorObject {

    private Object obj = new Object();

    public void doA() {
        synchronized(obj){
            System.out.print("1");
            obj.wait();
            System.out.print("2");
        }
    }

    public void doB() throws InterruptedException{
        synchronized(obj){
            System.out.print("3");
            obj.notify();
            Thread.sleep(500);
            System.out.print("4");
        }
    }
```

```java
    public static void main(String[] argvs)
                                throws Exception{
        MonitorObject mo = new MonitorObject();
        Thread t1 = new Thread(new Runnable() {
            @Override
            public void run() {
                mo.doA();
            }
        }, "T1");

        Thread t2 = new Thread(new Runnable() {
            @Override
            public void run() {
                try {
                    mo.doB();
                }catch(InterruptedException ie){
                    throw new RuntimeException(ie);
                }
           }
        }, "T2");

        t1.start();
        Thread.sleep(100);
        t2.start();
    }
}
```

运行结果：1342