

import java.util.concurrent.Semaphore;

笔记本: 哲学家进餐

创建时间: 2022/6/26 16:41

更新时间: 2022/6/26 16:42

作者: zt277vw1

URL: about:blank

```
import java.util.concurrent.Semaphore;

/**
 * @author Lenovo
 */
public class PhilosopherMeal {
    //五支筷子，定义信号量
    static final Semaphore[] mutex = {new Semaphore(1),
    new Semaphore(1),
    new Semaphore(1),
    new Semaphore(1),
    new Semaphore(1)};

    static class Philosopher extends Thread {
        String[] name={"阿基米德","亚里士多德","柏拉图","孔子","芝诺"};
        Philosopher(String number) {
            super.setName(number);
        }

        @Override
        public void run() {
            do {
                try {
                    Integer i = Integer.parseInt(super.getName());
                    //偶数号先拿左边的筷子，奇数号的先拿右边的筷子，防止死锁
                    if(i%2==0) {
                        //偶数号哲学家
                        //先拿左手边的筷子
                        mutex[i].acquire();
                        //再拿拿右边的筷子
                        mutex[(i + 1) % 5].acquire();
                        //谁先抢到谁第一个开吃
                        System.out.println("哲学家 " + name[i] + " 吃饭!");
                        //吃完放下左手的筷子
                        mutex[i].release();
                        //再放下右手的筷子，开始思考
                        mutex[(i + 1) % 5].release();
                        //模拟延迟
                        Thread.sleep(2000);
                    } else {
                        //奇数号哲学家
                        //先拿右手边的筷子
                        mutex[(i + 1) % 5].acquire();
                        //再拿左边的筷子
                        mutex[i].acquire();
                        //谁先抢到谁第一个开吃
                        System.out.println("哲学家 " + name[i] + " 吃饭!");
                        //吃完放下右手的筷子
```

```

mutex[(i + 1) % 5].release();
//再放下左手的筷子, 开始思考
mutex[i].release();
//模拟延迟
Thread.sleep(2000);
}

} catch (InterruptedException e) {
System.out.println("异常");

while (true);

}

public static void main(String[] args) {
//五个哲学家
Philosopher p0 = new Philosopher("0");
Philosopher p1 = new Philosopher("1");
Philosopher p2 = new Philosopher("2");
Philosopher p3 = new Philosopher("3");
Philosopher p4 = new Philosopher("4");

p0.start();
p1.start();
p2.start();
p3.start();
p4.start();
}
}

```

```

哲学家 阿基米德 吃饭!
哲学家 柏拉图 吃饭!
哲学家 亚里士多德 吃饭!
哲学家 孔子 吃饭!
哲学家 芝诺 吃饭!
哲学家 亚里士多德 吃饭!
哲学家 孔子 吃饭!
哲学家 阿基米德 吃饭!
哲学家 柏拉图 吃饭!
哲学家 芝诺 吃饭!

```