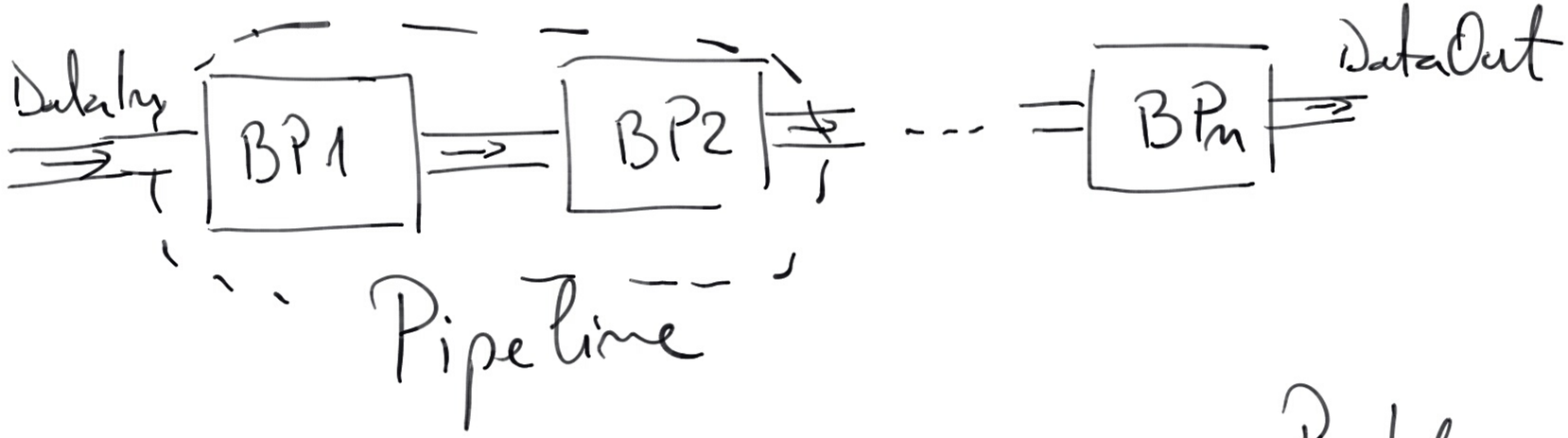
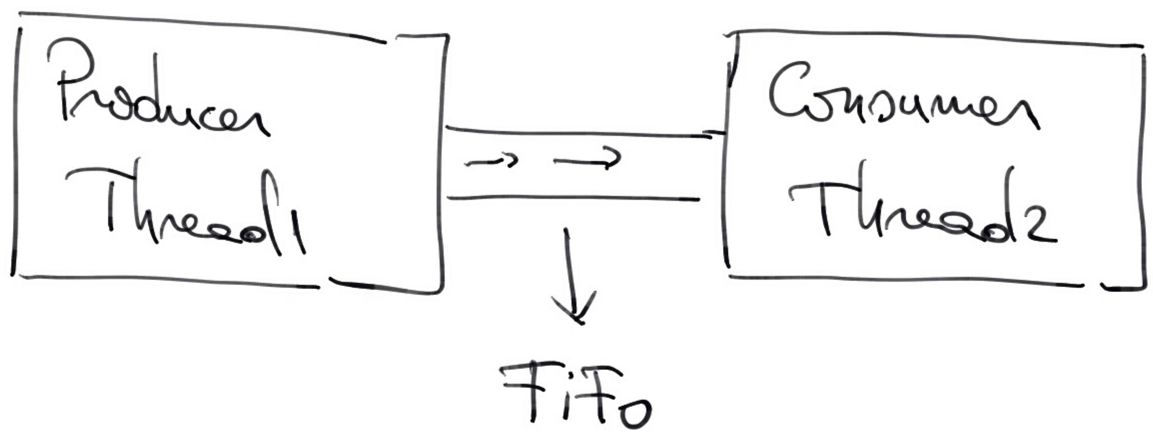
(8) Pentru a realiza un thread, var. 2:
Crearea unei clase care să implementeze interfața
java.lang.Runnable

```
public class MyRunnable implements Runnable {
    public void run() {
        //...
    }
}
```

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



Problema se rezolvă cu un sistem de bariere.



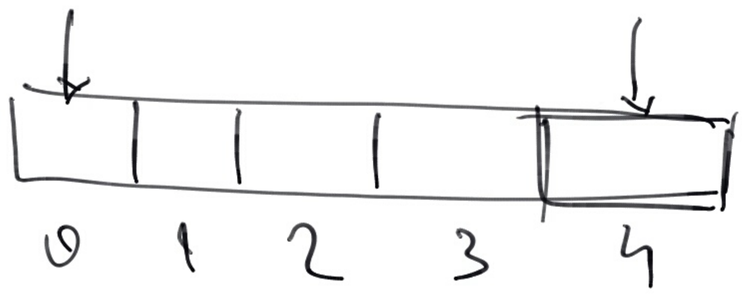
```
public class Fifo {
    private Object[] mQueue;
    private int mReadPointer;
    private int mWritePointer;
    private int mSize;

public Fifo(int capacity) {
        mQueue = new Object[capacity];
}

public synchronized void push(Object obj) throws InterruptedException {
    while (mSize == mQueue.length) {
        this.wait(); // eliberează monitorul (this)
    }
    mQueue[mWritePointer++] = obj;
    mWritePointer = mWritePointer % mQueue.length;
    mSize++;
                                        this.notifyAll();
                                    }
```

```
public synchronized Object pop() throws InterruptedException {
        while (mSize == 0) {
            this.wait();
        }
        Object toReturn = mQueue[mReadPointer++];
        mReadPointer = mReadPointer % mQueue.length;
        mSize--;
        this.notifyAll();
        return toReturn;
    }
}
```

Check out java.concurrent



Object o = ...;

if (o instanceof Message)

↳ verifică dacă obiectul referit de o este de tip Message

String s = "aaa";

if (s instanceof Message)

↳ eroare de compilare