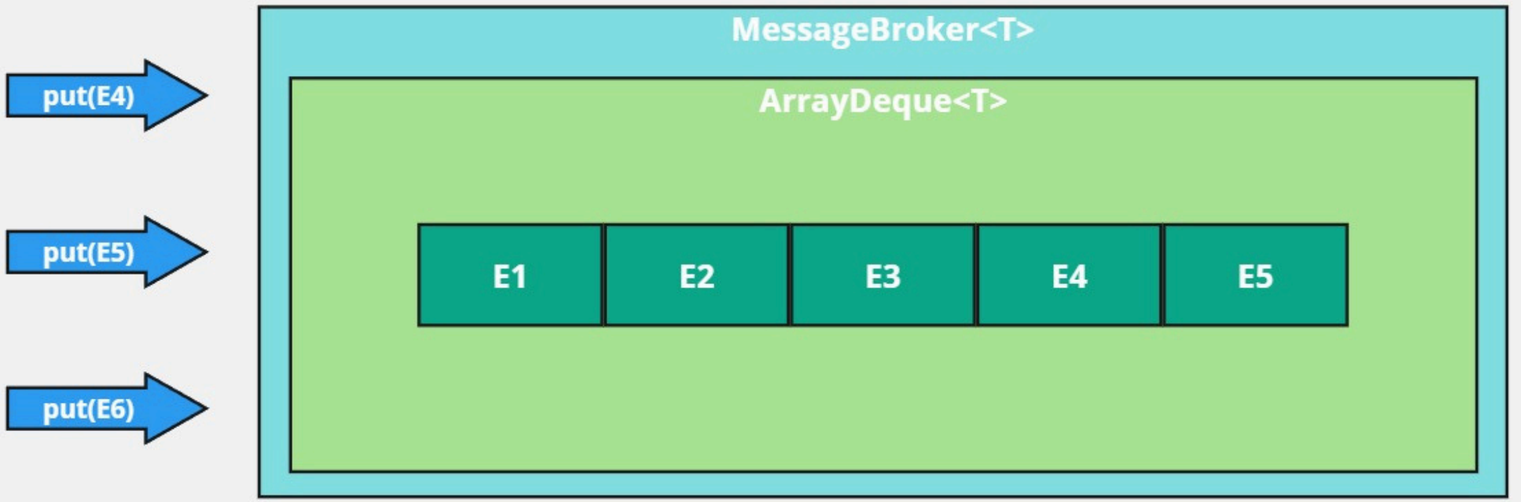
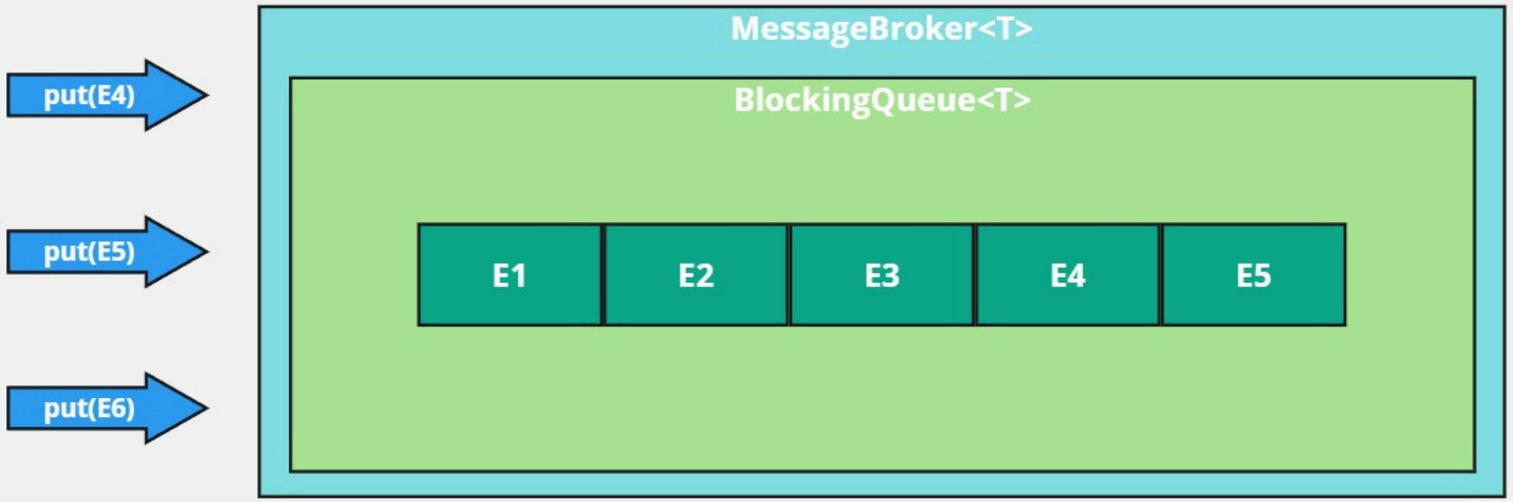
JAVA

МНОГОПОТОЧНОСТЬ.

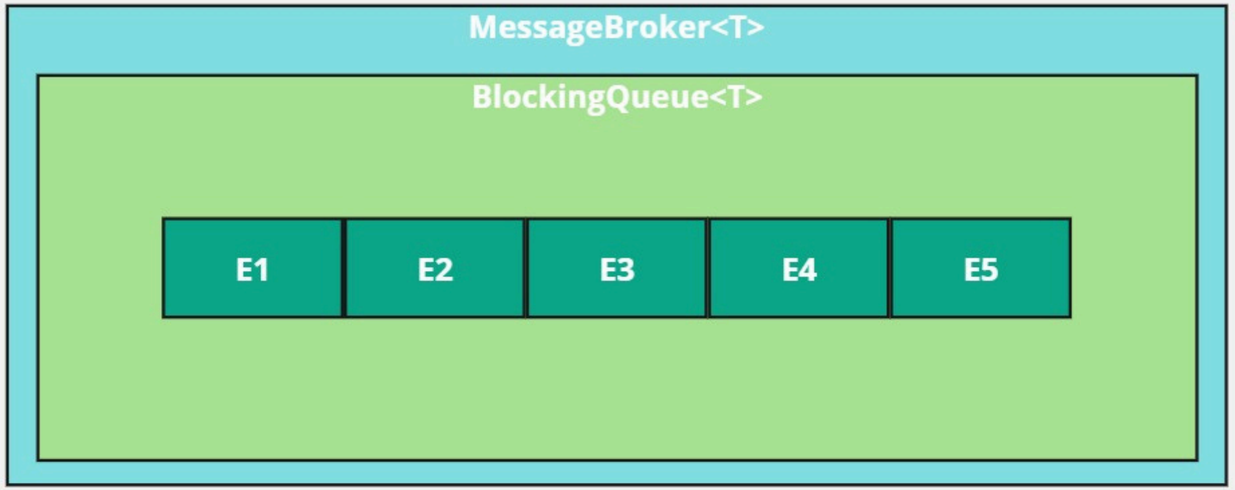
BlockingQueue. *Пример*

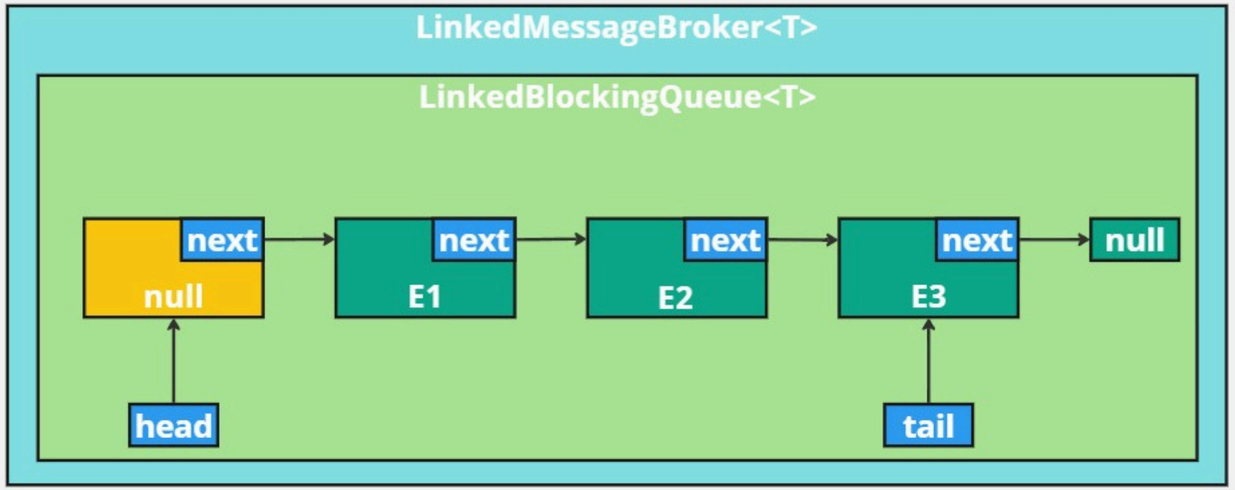


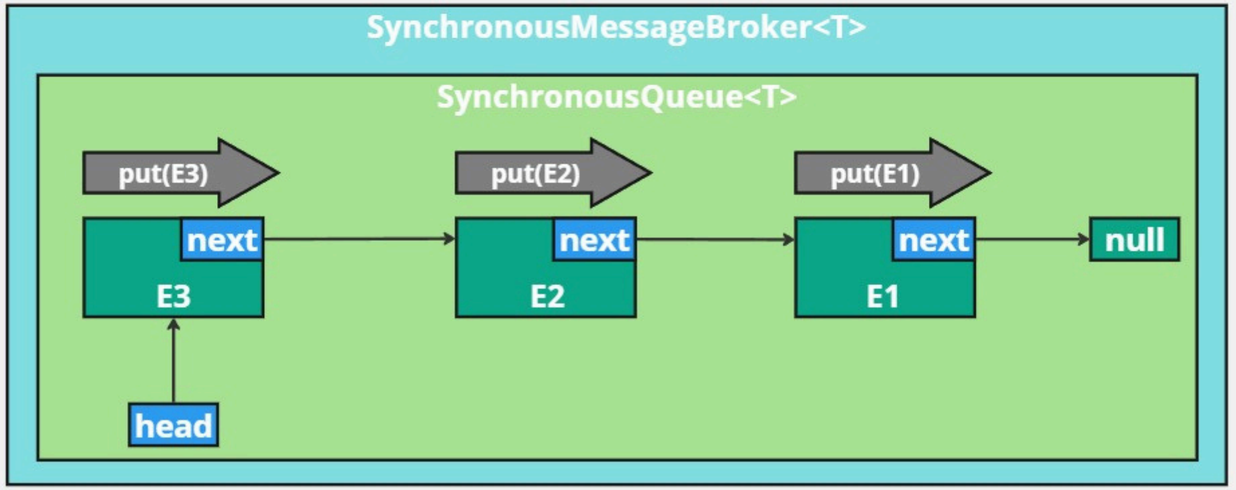

```java
public final class MessageBroker<T> {
    private final Queue<T> messages;
    private final int capacity;

    public MessageBroker(final int capacity) {
        this.messages = new ArrayDeque<>(capacity);
        this.capacity = capacity;
    }

    public synchronized void put(final T message)
            throws InterruptedException {
        while (messages.size() >= capacity) {
            wait();
        }
        messages.add(message);
        notifyAll();
    }

    public synchronized T take()
            throws InterruptedException {
        while (messages.isEmpty()) {
            wait();
        }
        final T message = messages.poll();
        notifyAll();
        return message;
    }
}
```

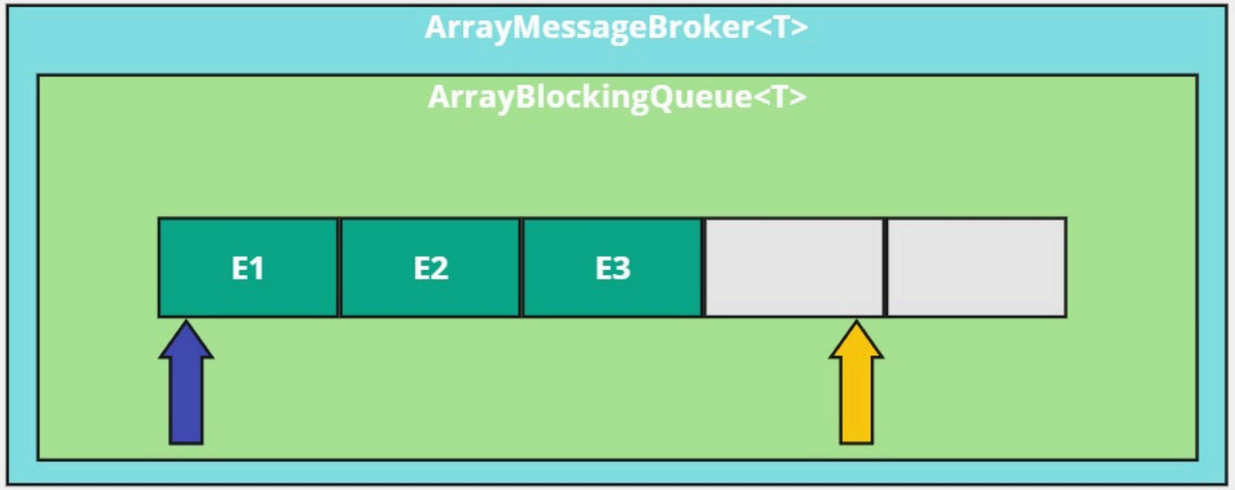



```java
public abstract class MessageBroker<T> {
    private final BlockingQueue<T> messages;

    public MessageBroker(final BlockingQueue<T> messages) {
        this.messages = messages;
    }

    public final void put(final T message)
            throws InterruptedException {
        messages.put(message);
    }

    public final T take()
            throws InterruptedException {
        return messages.take();
    }
}
```


MessageBroker<T>
BlockingQueue<T>
E1
E2
E3
E4
E5


LinkedMessageBroker<T>
LinkedBlockingQueue<T>
next
null
next
E1
next
E2
next
E3
null
head
tail


SynchronousMessageBroker<T>
SynchronousQueue<T>
put(E3)
put(E2)
put(E1)
next
E3
next
E2
next
E1
null
head


ArrayMessageBroker<T>
ArrayBlockingQueue<T>
E1
E2
E3