https://jagstalk.page.link/examples

```kotlin
fun main() = runBlocking {
    val orders = listOf(...)
    val ordersChannel = produce(CoroutineName("cashier")) {
        orders.forEach { send(it) }
    }
    val espressoMachine = EspressoMachine(this)
    val time = measureTimeMillis {
        coroutineScope {
            launch(CoroutineName("barista-1")) {
                processOrders(ordersChannel, espressoMachine)
            }
            launch(CoroutineName("barista-2")) {
                processOrders(ordersChannel, espressoMachine)
            }
        }
    }
    log("time: $time ms")
    espressoMachine.destroy()
}
```

https://jagstalk.page.link/examples

```
fun main() = runBlocking {
    val orders = listOf(...)
    val ordersChannel = produce(CoroutineName("cashier")) {
        orders.forEach { send(it) }
    }
    val espressoMachine = EspressoMachine(this)
    val time = measureTimeMillis {
        coroutineScope {
            launch(CoroutineName("barista-1")) {
                processOrders(ordersChannel, espressoMachine)
            }
            launch(CoroutineName("barista-2")) {
                processOrders(ordersChannel, espressoMachine)
            }
        }
    }
    log("time: $time ms")
    espressoMachine.destroy()
}
```

https://jagstalk.page.link/examples

```
fun main() = runBlocking {
    val orders = listOf(...)
    val ordersFlow: Flow<Menu.Cappuccino> = orders.asFlow()

    val espressoMachine = EspressoMachine(this)
    val time = measureTimeMillis {
        coroutineScope {
            launch(CoroutineName("barista-1")) {
                processOrders(ordersChannel, espressoMachine)
            }
            launch(CoroutineName("barista-2")) {
                processOrders(ordersChannel, espressoMachine)
            }
        }
    }
    log("time: $time ms")
    espressoMachine.destroy()
}
```