```
fun main() {
    val orders = listOf(...)
    val time = measureTimeMillis {
        launch { processOrders(orders) }
        launch { processOrders(orders) }
    }
    log("time: $time ms”)
}

private fun processOrders(orders: List<Menu.Cappuccino>) {
    orders.forEach {
        val groundBeans = grindCoffeeBeans(it.beans())
        val espresso = pullEspressoShot(groundBeans)
        val steamedMilk = steamMilk(it.milk())
        val cappuccino = makeCappuccino(it, espresso, steamedMilk)
        log("serve: $cappuccino")
    }
}
```

https://jagstalk.page.link/examples

https://jagstalk.page.link/examples

```
fun main() {
    val orders = listOf(...)
    val time = measureTimeMillis {
        launch { processOrders(orders) }
        launch { processOrders(orders) }
    }
    log("time: $time ms”)
}

private fun processOrders(orders: List<Menu.Cappuccino>) {
    orders.forEach {
        val groundBeans = grindCoffeeBeans(it.beans())
        val espresso = pullEspressoShot(groundBeans)
        val steamedMilk = steamMilk(it.milk())
        val cappuccino = makeCappuccino(it, espresso, steamedMilk)
        log("serve: $cappuccino")
    }
}
```

https://jagstalk.page.link/examples

```
fun main() = runBlocking {
    val orders = listOf(...)
    val time = measureTimeMillis {
        launch { processOrders(orders) }
        launch { processOrders(orders) }
    }
    log("time: $time ms”)
}

private fun processOrders(orders: List<Menu.Cappuccino>) {
    orders.forEach {
        val groundBeans = grindCoffeeBeans(it.beans())
        val espresso = pullEspressoShot(groundBeans)
        val steamedMilk = steamMilk(it.milk())
        val cappuccino = makeCappuccino(it, espresso, steamedMilk)
        log("serve: $cappuccino")
    }
}
```