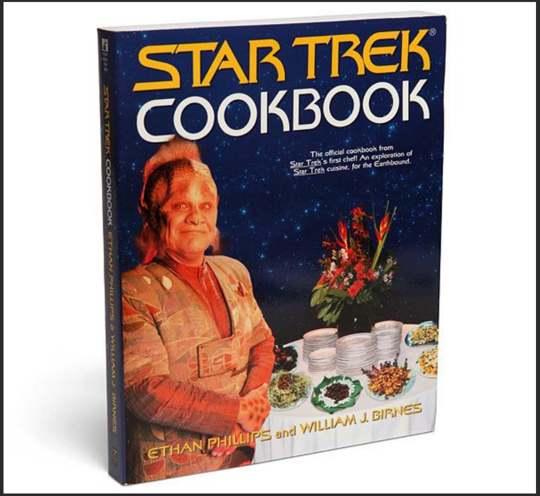
Factory Pattern & Template Method

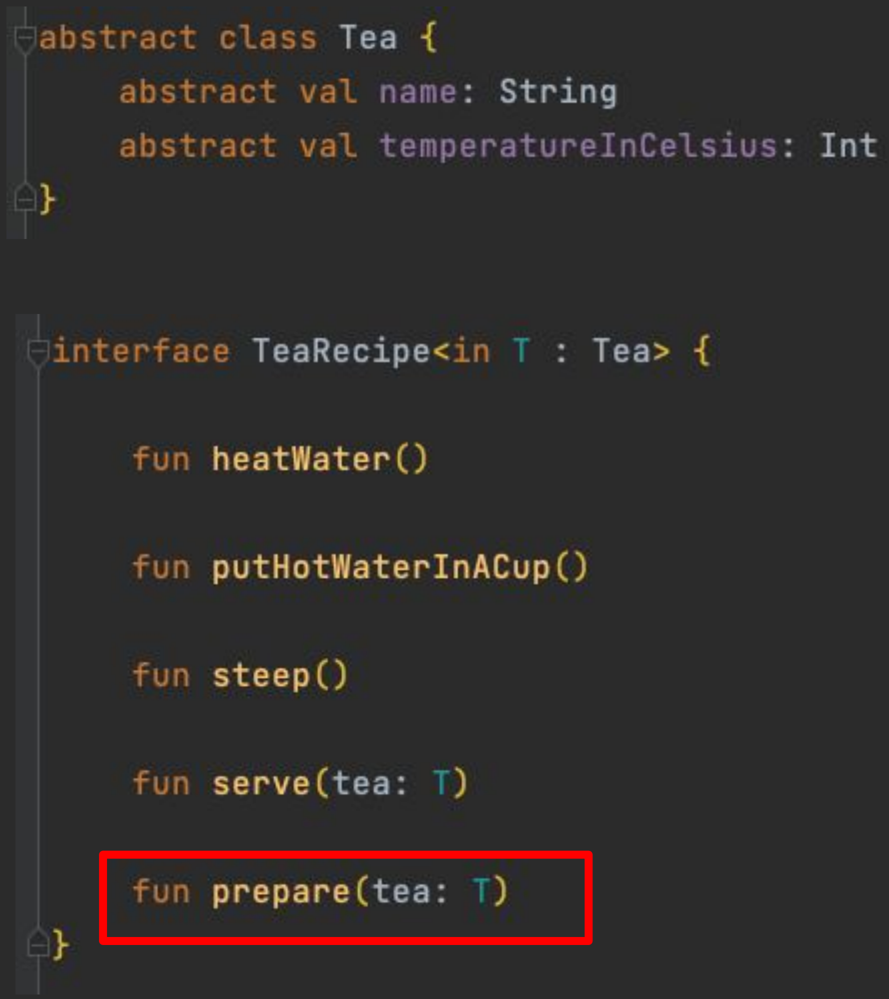
By Moro

10 Find out the basic steps to make tea


STAR TREK COOKBOOK
The official cookbook from Star Trek's first chef! An exploration of Star Trek cuisine, for the Earthbound.
ETHAN PHILLIPS and WILLIAM J. BIRNES

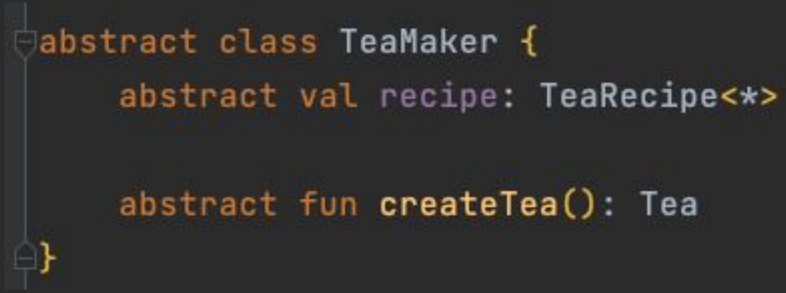
```kotlin
abstract class Tea {
    abstract val name: String
    abstract val temperatureInCelsius: Int
}
```

```kotlin
interface TeaRecipe<in T : Tea> {

    fun heatWater()

    fun putHotWaterInACup()

    fun steep()

    fun serve(tea: T)

    fun prepare(tea: T)
}
```



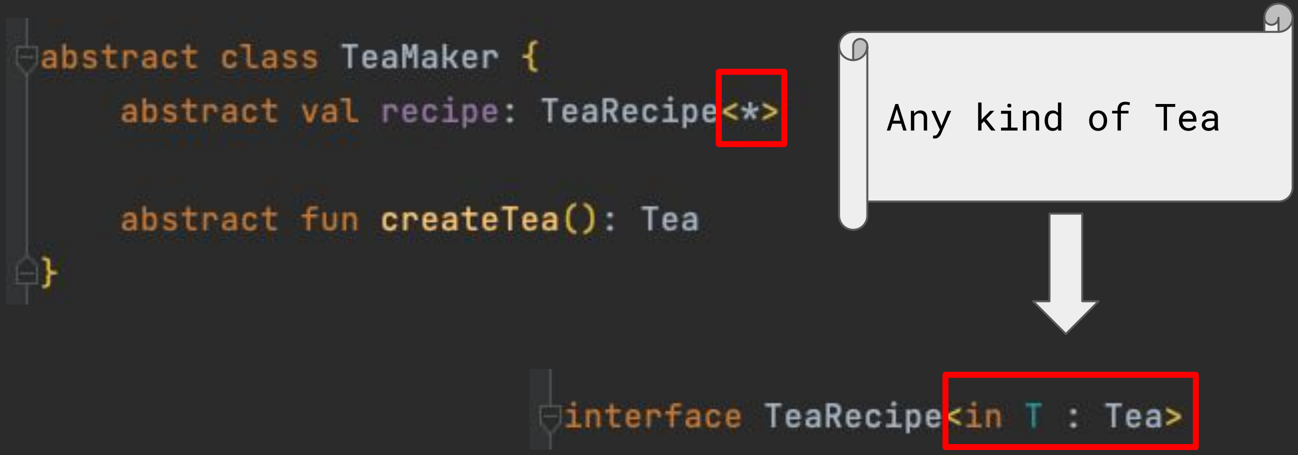
```kotlin
abstract class Tea {
    abstract val name: String
    abstract val temperatureInCelsius: Int
}


interface TeaRecipe<in T : Tea> {

    fun heatWater()

    fun putHotWaterInACup()

    fun steep()

    fun serve(tea: T)

    fun prepare(tea: T)
}
```

This method will be our ***TemplateMethod***



```
abstract class TeaMaker {
    abstract val recipe: TeaRecipe<*>

    abstract fun createTea(): Tea
}
```

2º Create our Factory



```kotlin
abstract class TeaMaker {
    abstract val recipe: TeaRecipe<*>

    abstract fun createTea(): Tea
}
```

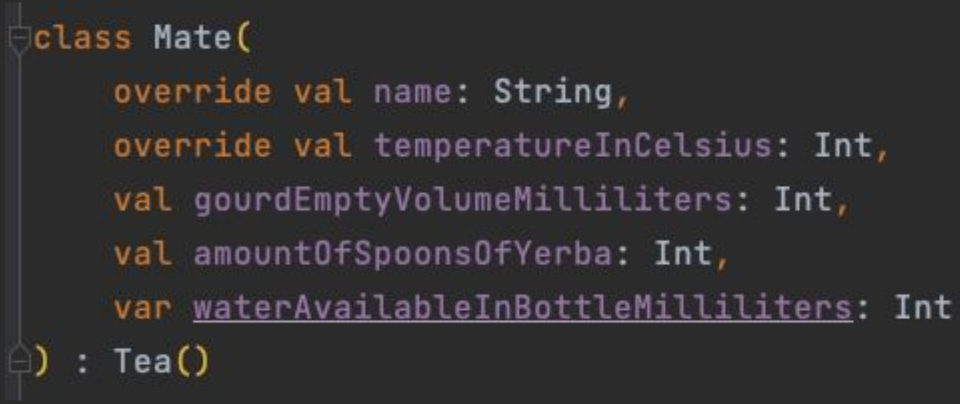
```kotlin
interface TeaRecipe<in T : Tea>
```

2º Create our Factory

What is your favorite Tea?

A specific kind of Mate, called Chimarrão

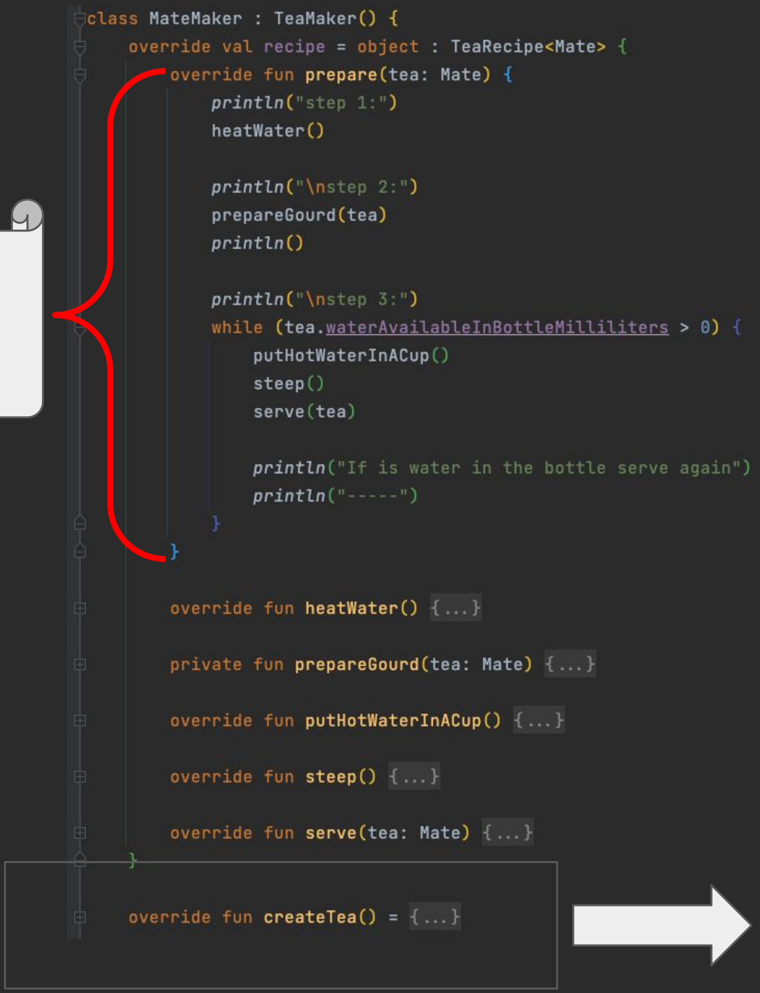
3º Create the new kind of tea



```kotlin
class Mate(
    override val name: String,
    override val temperatureInCelsius: Int,
    val gourdEmptyVolumeMilliliters: Int,
    val amountOfSpoonsOfYerba: Int,
    var waterAvailableInBottleMilliliters: Int
) : Tea()
```

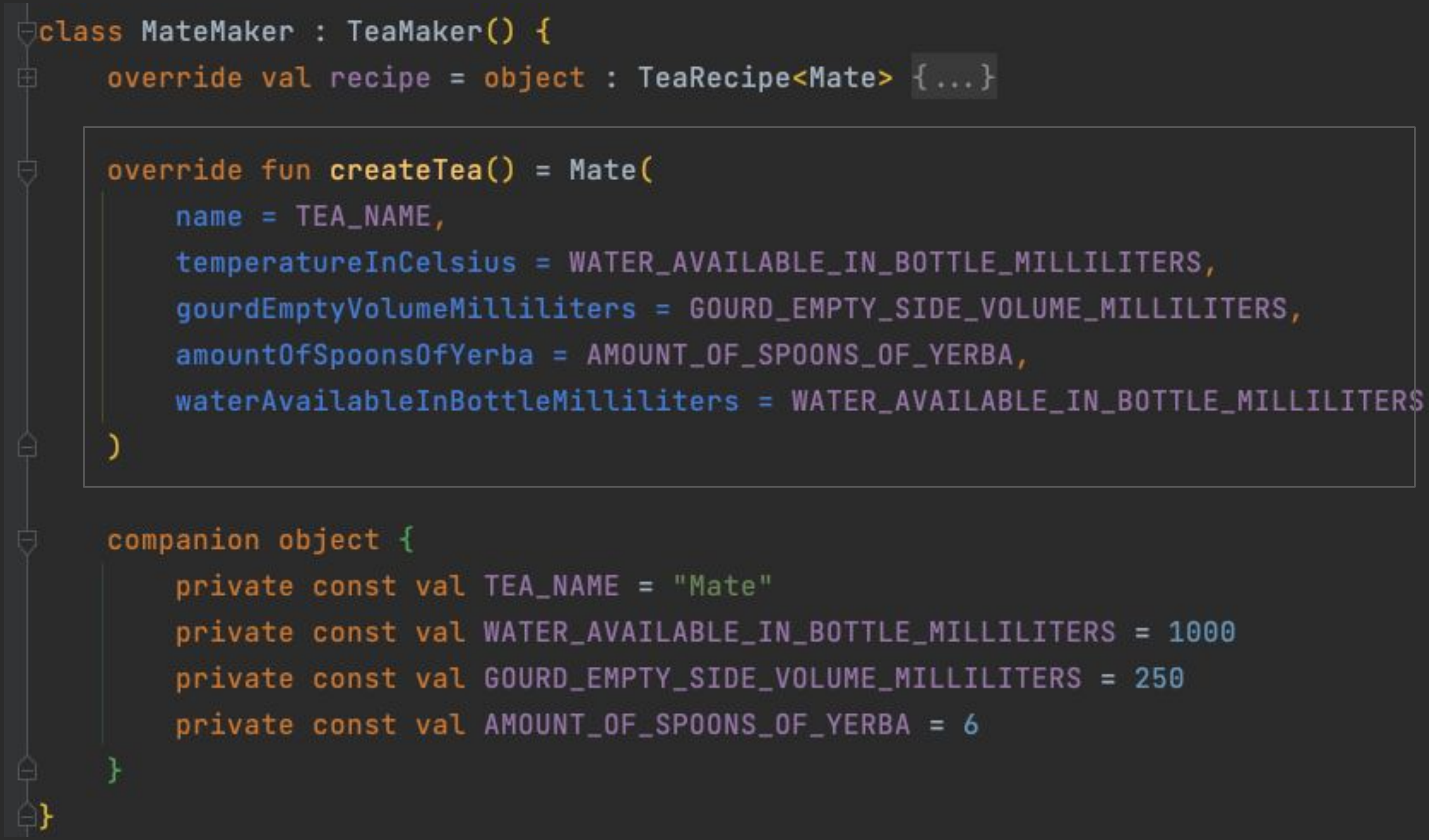
4º Create a Factory of Mates

This method keeps the steps sorted



```kotlin
class MateMaker : TeaMaker() {
    override val recipe = object : TeaRecipe<Mate> {
        override fun prepare(tea: Mate) {
            println("step 1:")
            heatWater()

            println("\nstep 2:")
            prepareGourd(tea)
            println()

            println("\nstep 3:")
            while (tea.waterAvailableInBottleMilliliters > 0) {
                putHotWaterInACup()
                steep()
                serve(tea)

                println("If is water in the bottle serve again")
                println("-----")
            }
        }

        override fun heatWater() {...}

        private fun prepareGourd(tea: Mate) {...}

        override fun putHotWaterInACup() {...}

        override fun steep() {...}

        override fun serve(tea: Mate) {...}
    }

    override fun createTea() = {...}
```

4º Create a Factory of Mates



```kotlin
class MateMaker : TeaMaker() {
    override val recipe = object : TeaRecipe<Mate> {...}

    override fun createTea() = Mate(
        name = TEA_NAME,
        temperatureInCelsius = WATER_AVAILABLE_IN_BOTTLE_MILLILITERS,
        gourdEmptyVolumeMilliliters = GOURD_EMPTY_SIDE_VOLUME_MILLILITERS,
        amountOfSpoonsOfYerba = AMOUNT_OF_SPOONS_OF_YERBA,
        waterAvailableInBottleMilliliters = WATER_AVAILABLE_IN_BOTTLE_MILLILITERS
    )

    companion object {
        private const val TEA_NAME = "Mate"
        private const val WATER_AVAILABLE_IN_BOTTLE_MILLILITERS = 1000
        private const val GOURD_EMPTY_SIDE_VOLUME_MILLILITERS = 250
        private const val AMOUNT_OF_SPOONS_OF_YERBA = 6
    }
}
```

```kotlin
fun main() {
    val teaFactory = MateMaker()
    val tea = teaFactory.createTea() // instance of Mate
    val teaRecipe = teaFactory.recipe
    teaRecipe.prepare(tea)
}
```

```kotlin
fun main() {
    val teaFactory = GreenTeaMaker()
    val tea = teaFactory.createTea() // instance of GreenTea
    val teaRecipe = teaFactory.recipe
    teaRecipe.prepare(tea)
}
```

By Moro