```
speaker {
    name = "Sergey Opivalov"
    now = "Build systems"
    before = "Android"
    talks = arrayOf("Mobius Autumn 2022", "Mobius Spring 2023")
    telegram = "sergey_opivalov"
    linkedIn = “sergey-opivalov”
}
```

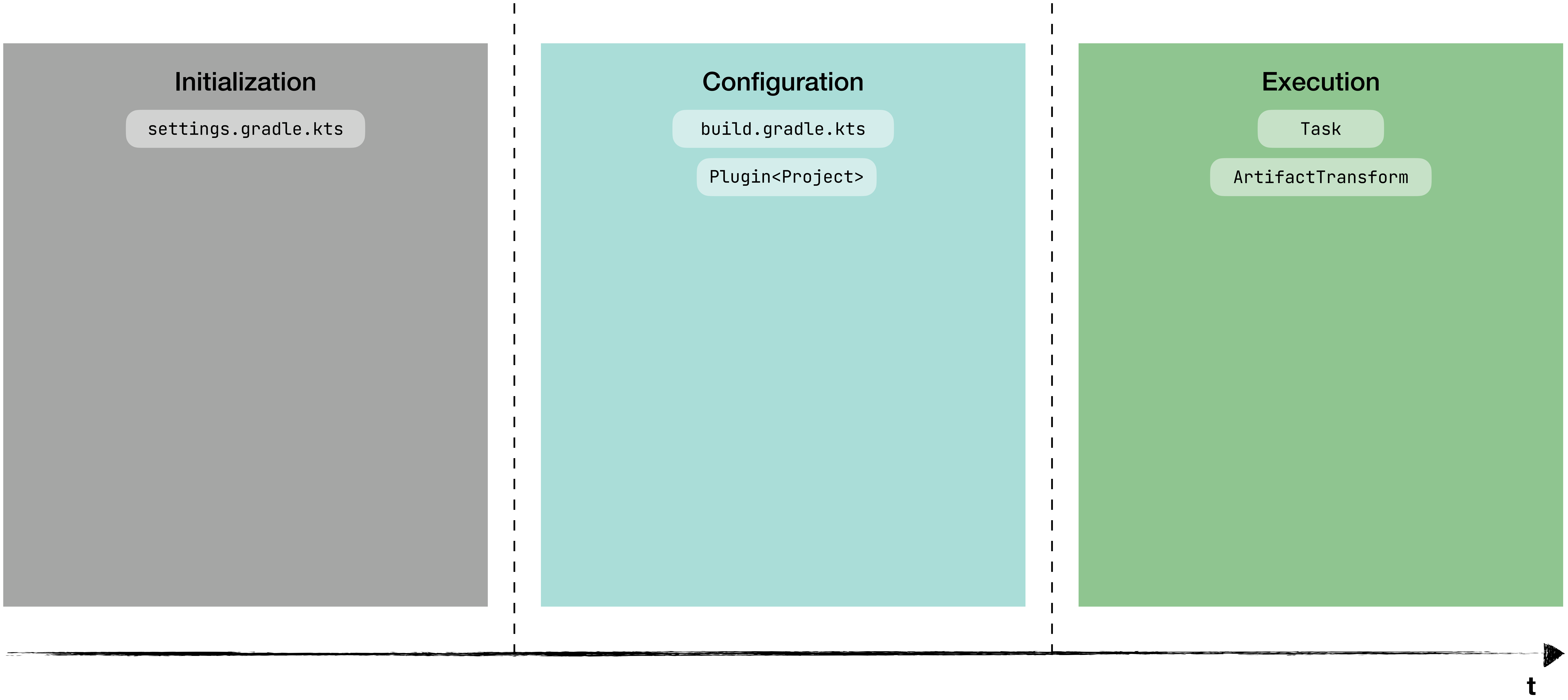
Инкрементальная компиляция Java/Kotlin в Gradle

Управление сложностью состояния

Почему вам стоит включить Gradle Configuration Cache

💎💎💎

Жизненный цикл сборки



Initialization

`settings.gradle.kts`

Configuration

`build.gradle.kts`

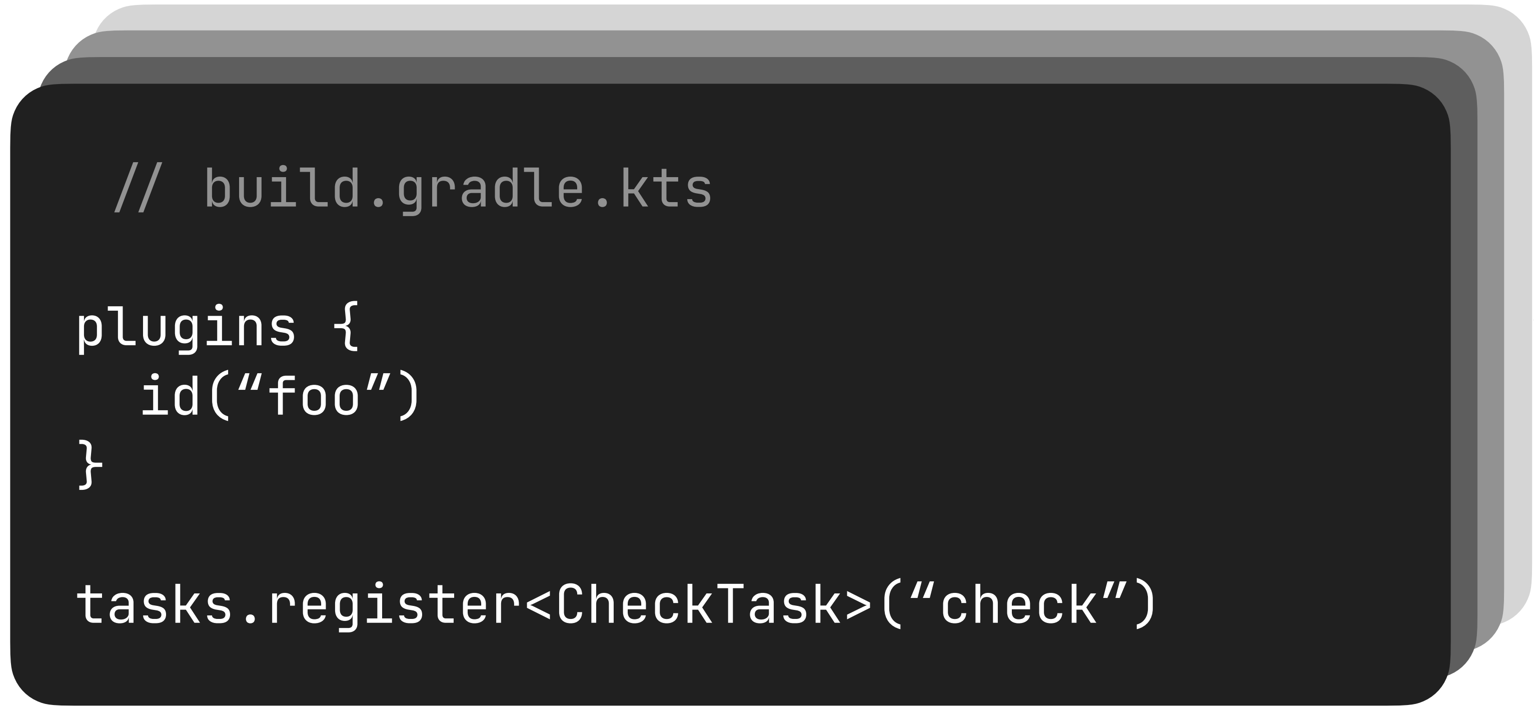
`Plugin<Project>`

Execution

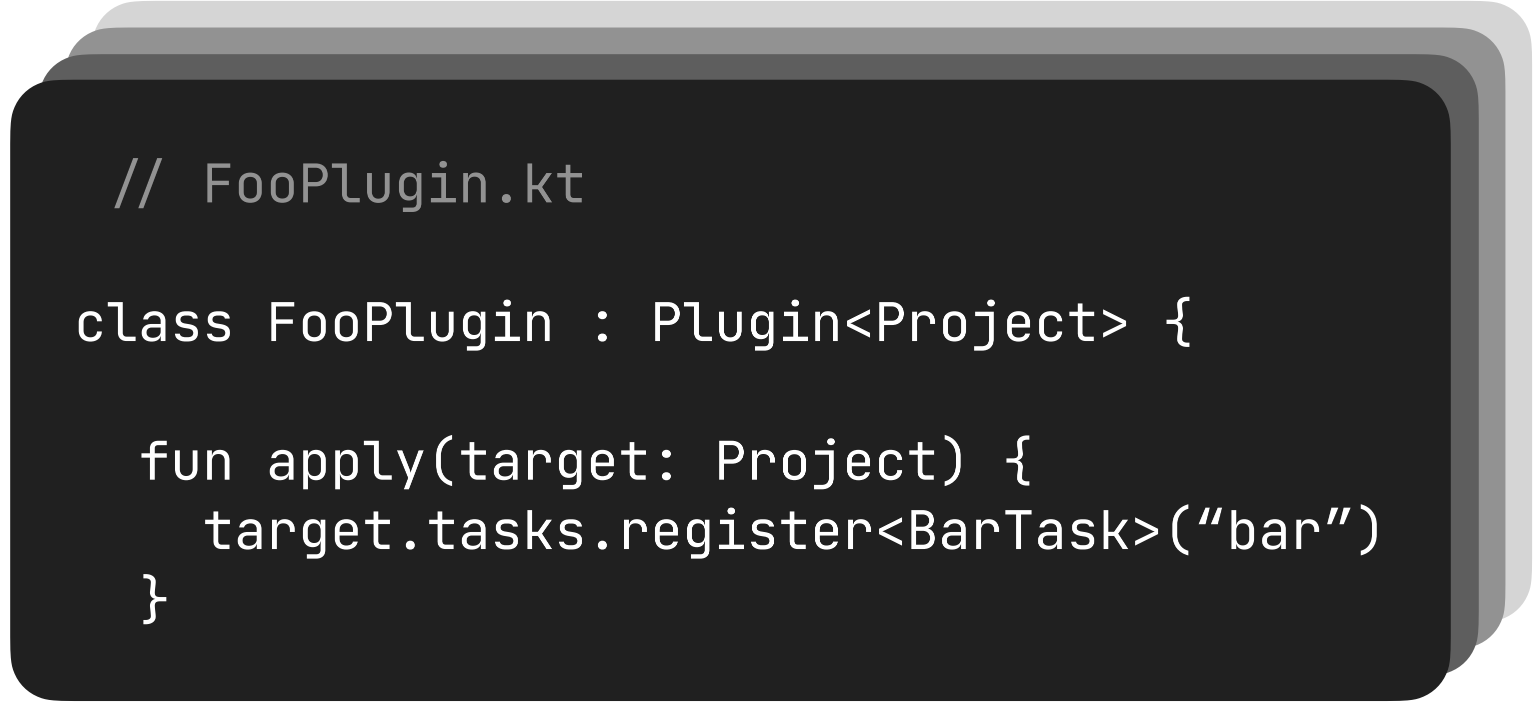
`Task`

`ArtifactTransform`

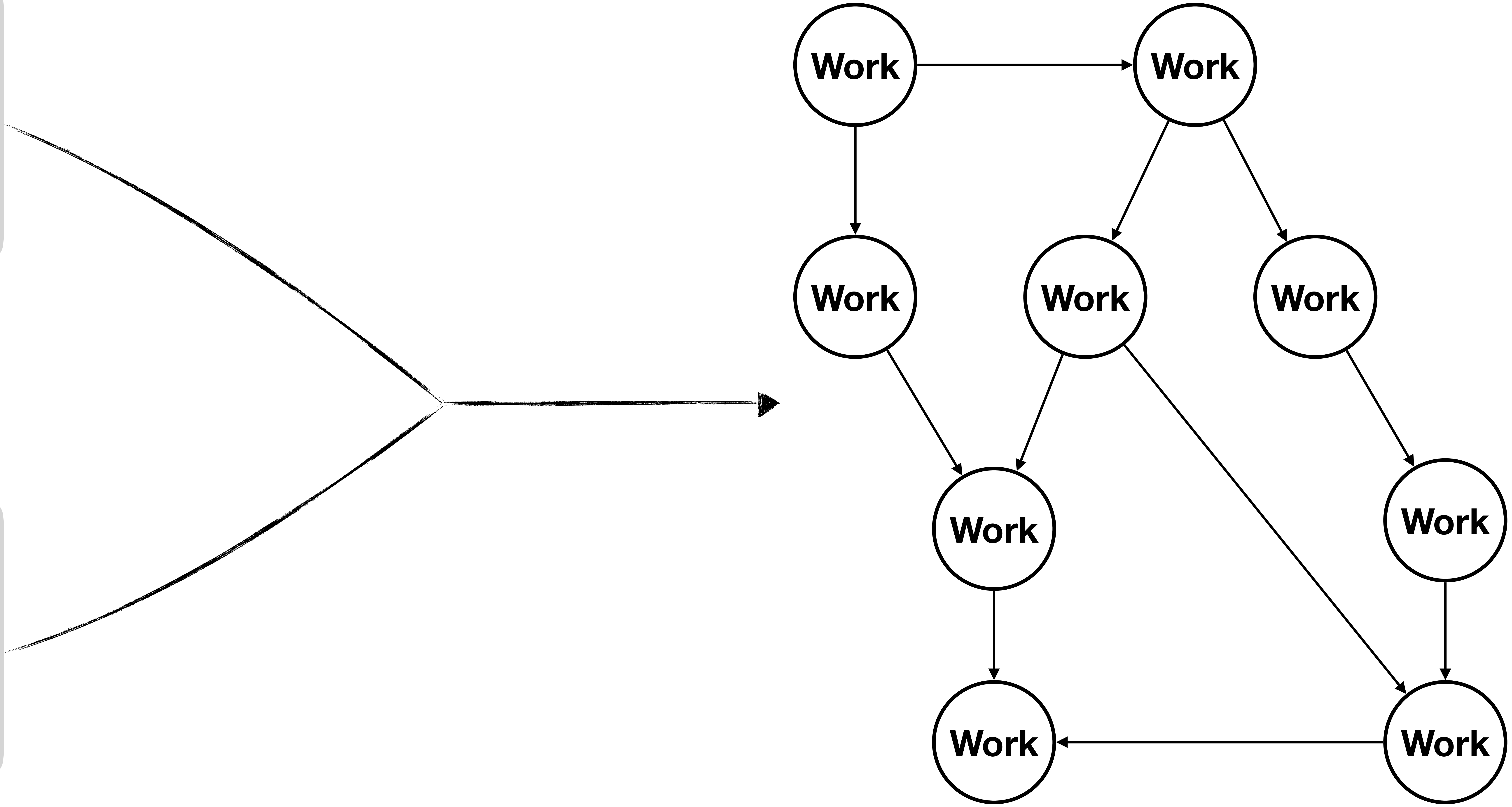
Что такое конфигурация?



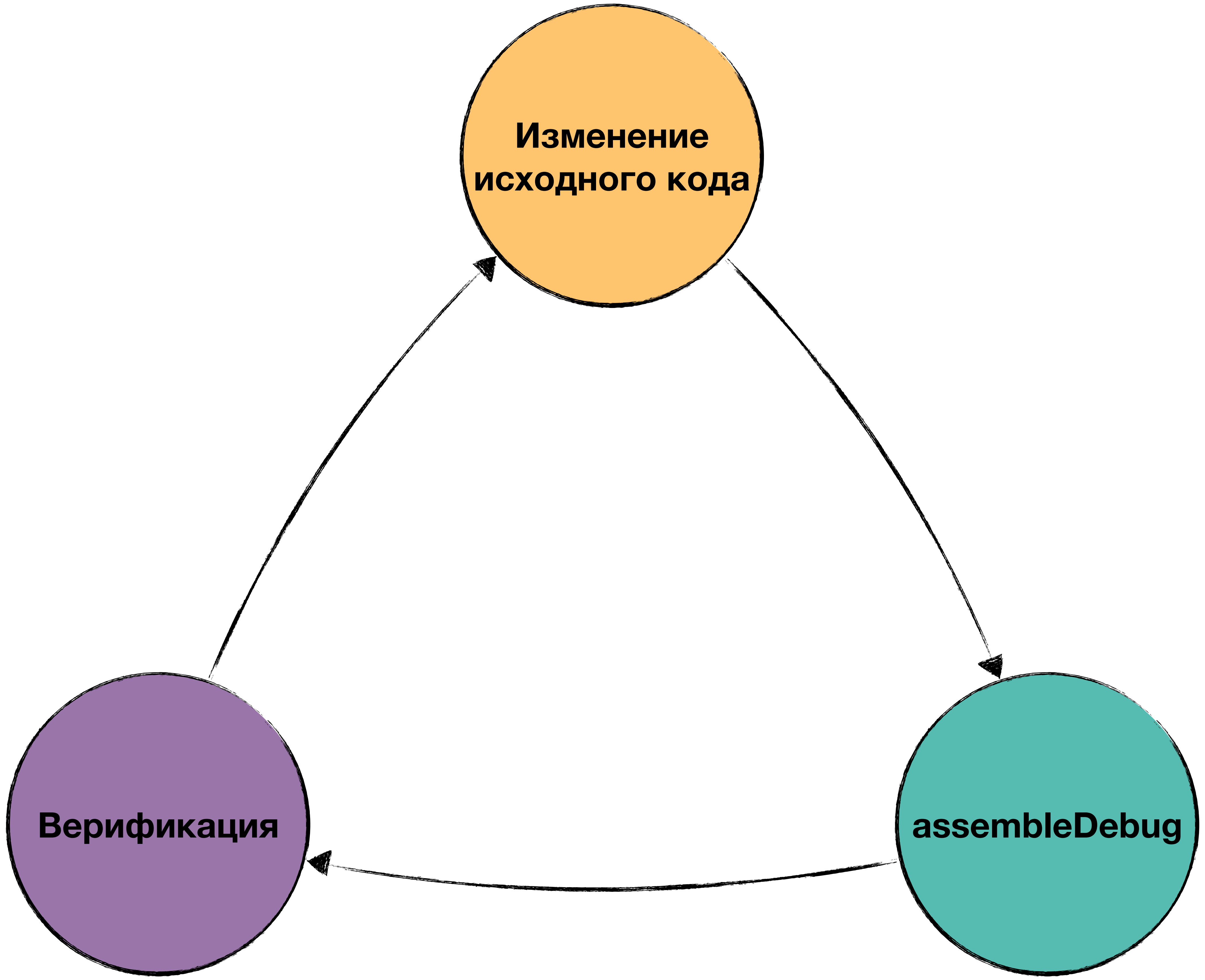
```
 // build.gradle.kts

plugins {
  id(“foo”)
}

tasks.register<CheckTask>(“check”)
```

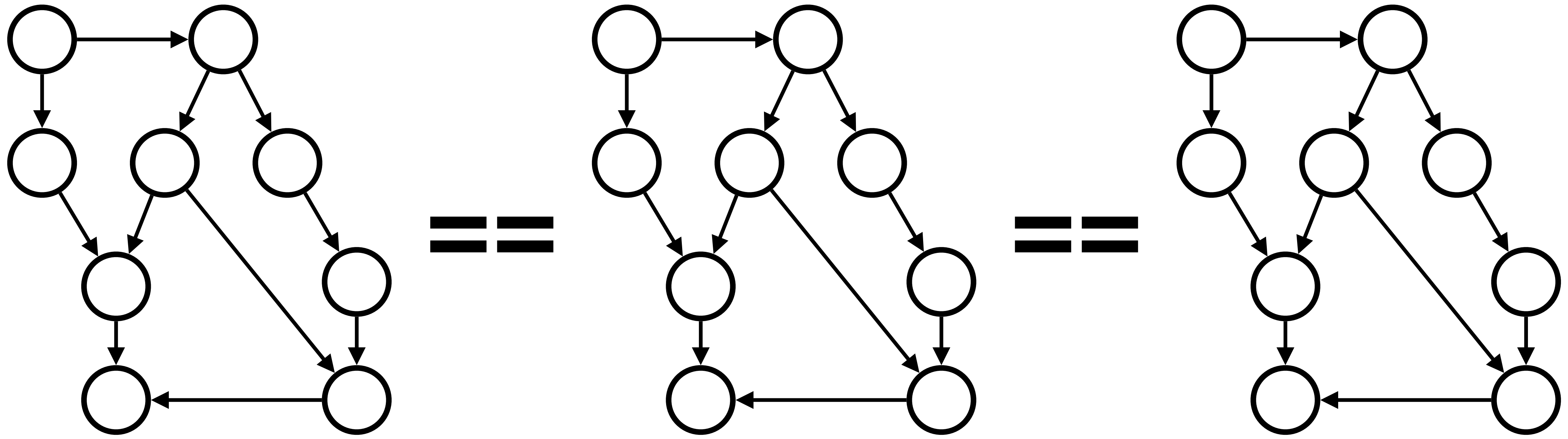


```
 // FooPlugin.kt

class FooPlugin : Plugin<Project> {

  fun apply(target: Project) {
    target.tasks.register<BarTask>(“bar”)
  }
```

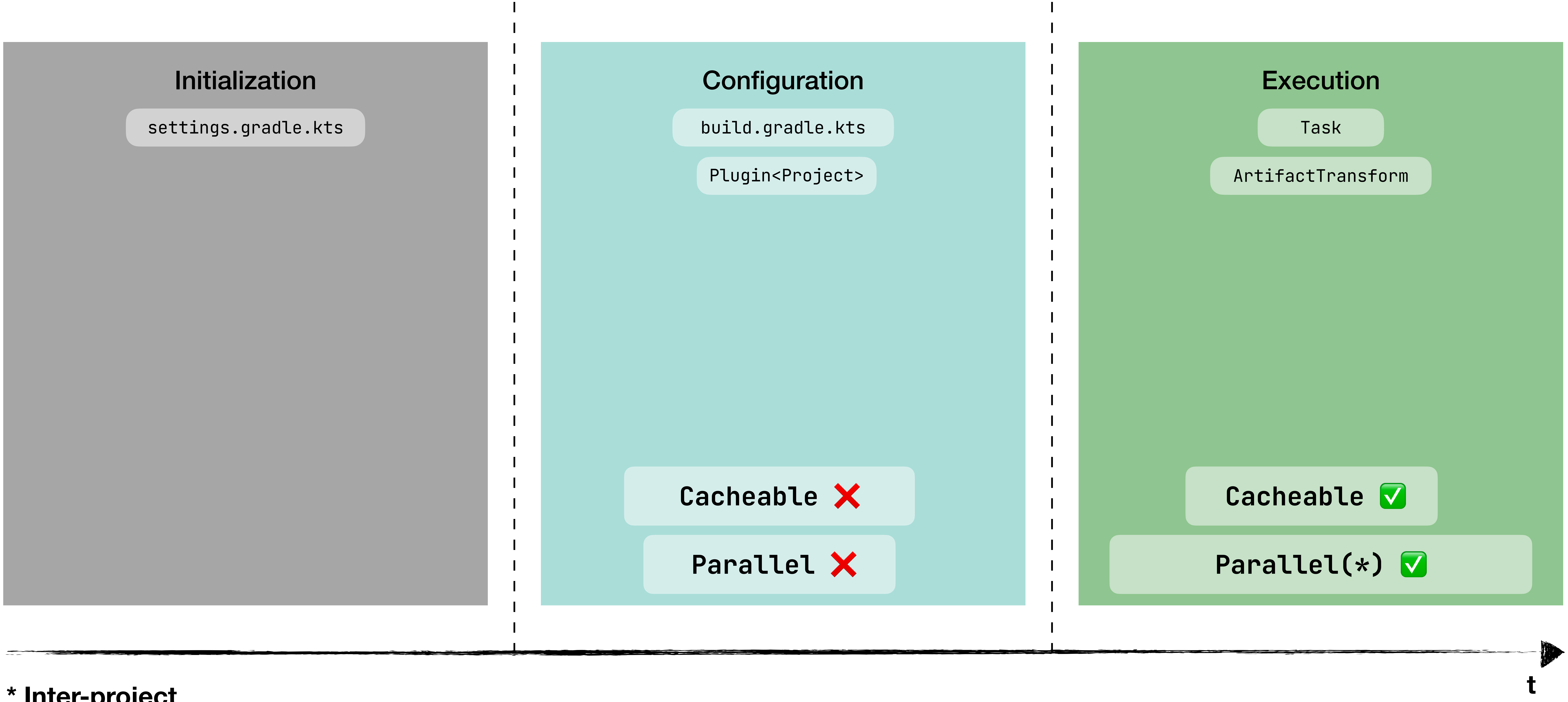


Почему Configuration Cache?





Почему Configuration Cache?



Initialization

`settings.gradle.kts`

Configuration

`build.gradle.kts`

`Plugin<Project>`

`Cacheable` ❌

`Parallel` ❌

Execution

`Task`

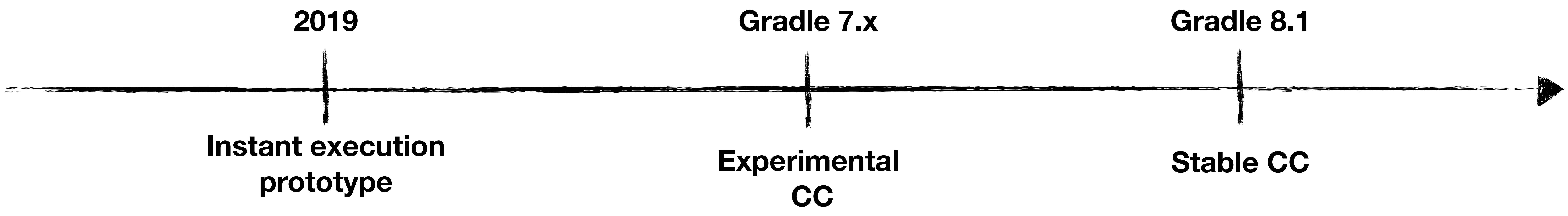
`ArtifactTransform`

`Cacheable` ✅

`Parallel(*)` ✅

* Inter-project

Эволюция CC



2019

Instant execution prototype

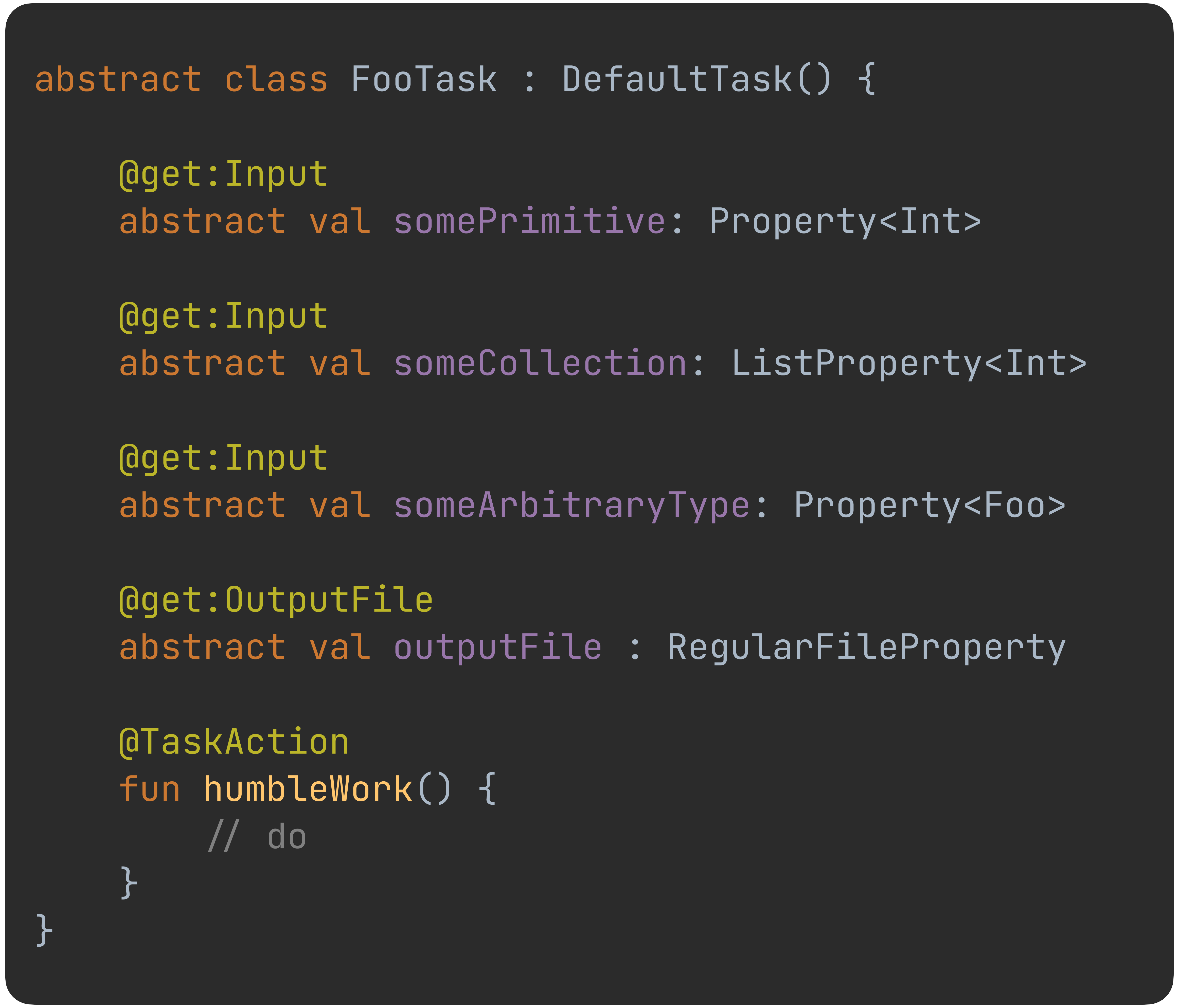
Gradle 7.x

Experimental CC

Gradle 8.1

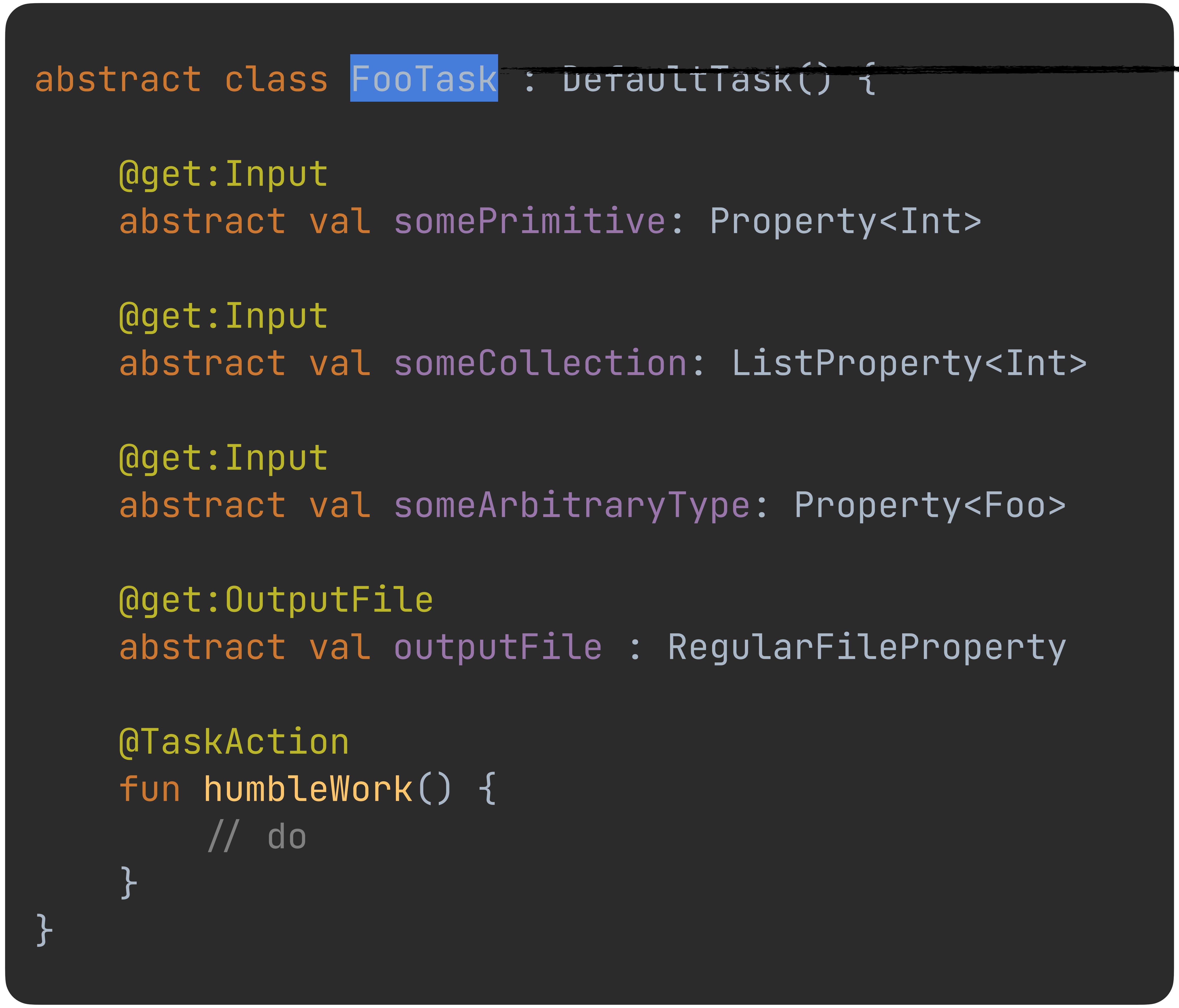
Stable CC

Модель Inputs/Outputs

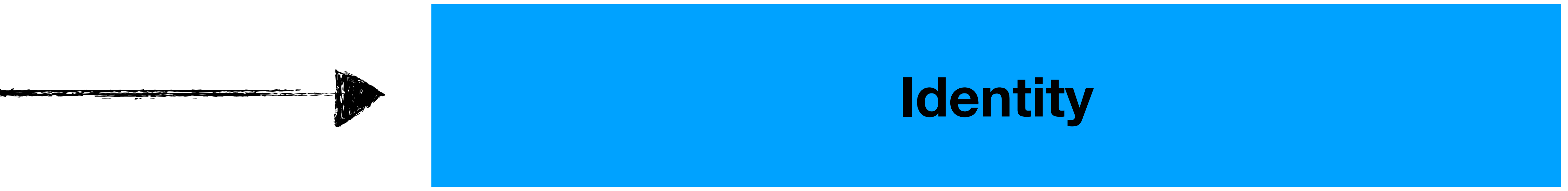


```
abstract class FooTask : DefaultTask() {

    @get:Input
    abstract val somePrimitive: Property<Int>

    @get:Input
    abstract val someCollection: ListProperty<Int>

    @get:Input
    abstract val someArbitraryType: Property<Foo>

    @get:OutputFile
    abstract val outputFile : RegularFileProperty

    @TaskAction
    fun humbleWork() {
        // do
    }
}
```

Модель Inputs/Outputs

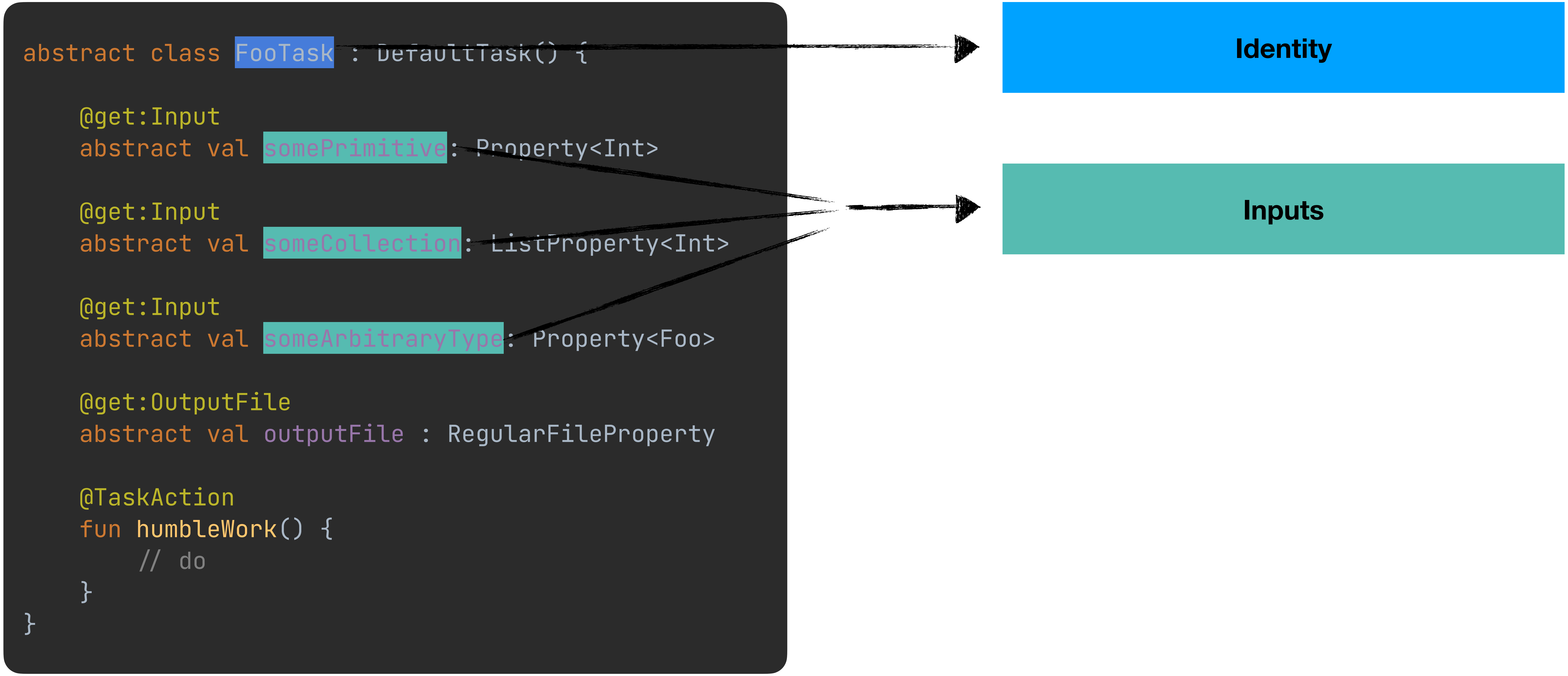


```
abstract class FooTask : DefaultTask() {

    @get:Input
    abstract val somePrimitive: Property<Int>

    @get:Input
    abstract val someCollection: ListProperty<Int>

    @get:Input
    abstract val someArbitraryType: Property<Foo>

    @get:OutputFile
    abstract val outputFile : RegularFileProperty

    @TaskAction
    fun humbleWork() {
        // do
    }
}
```



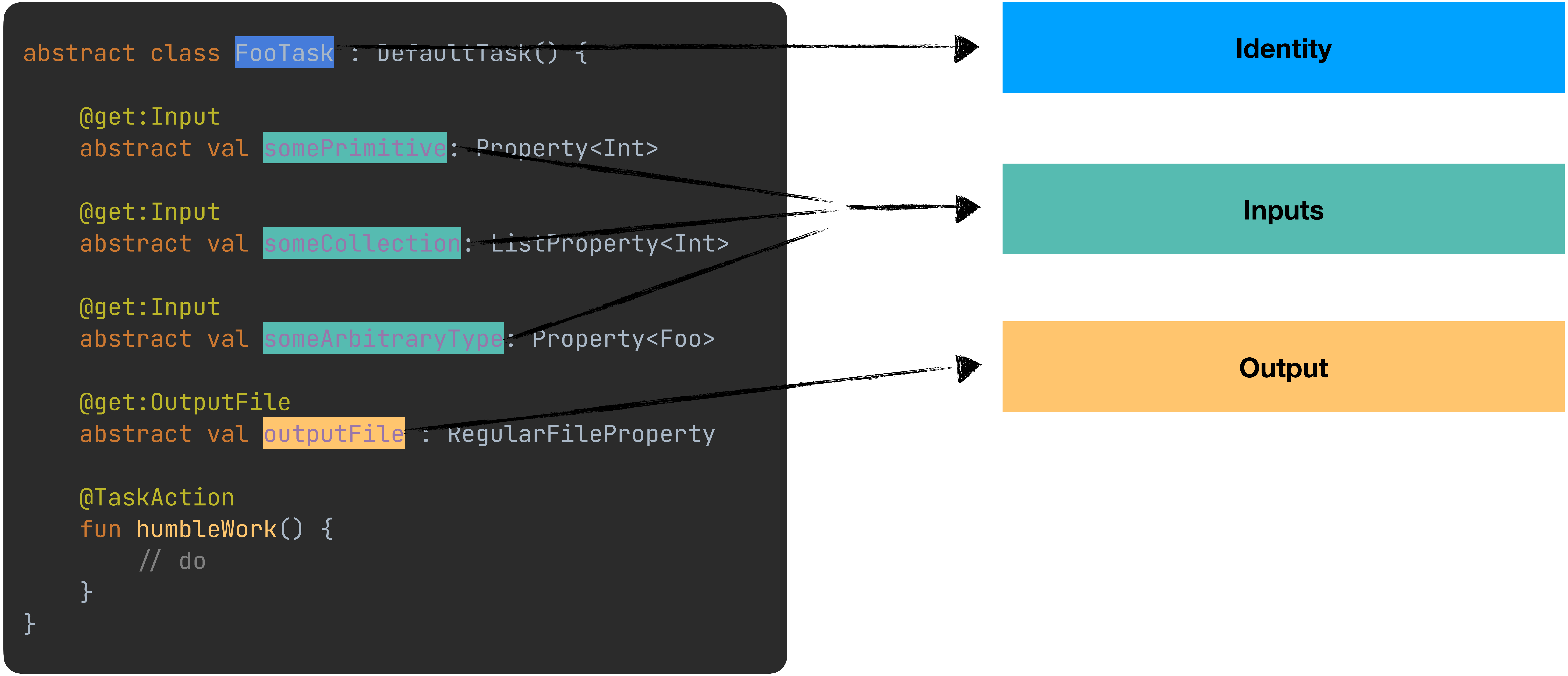
Identity

Модель Inputs/Outputs



```
abstract class FooTask : DefaultTask() {

    @get:Input
    abstract val somePrimitive: Property<Int>

    @get:Input
    abstract val someCollection: ListProperty<Int>

    @get:Input
    abstract val someArbitraryType: Property<Foo>

    @get:OutputFile
    abstract val outputFile : RegularFileProperty

    @TaskAction
    fun humbleWork() {
        // do
    }
}
```

Модель Inputs/Outputs



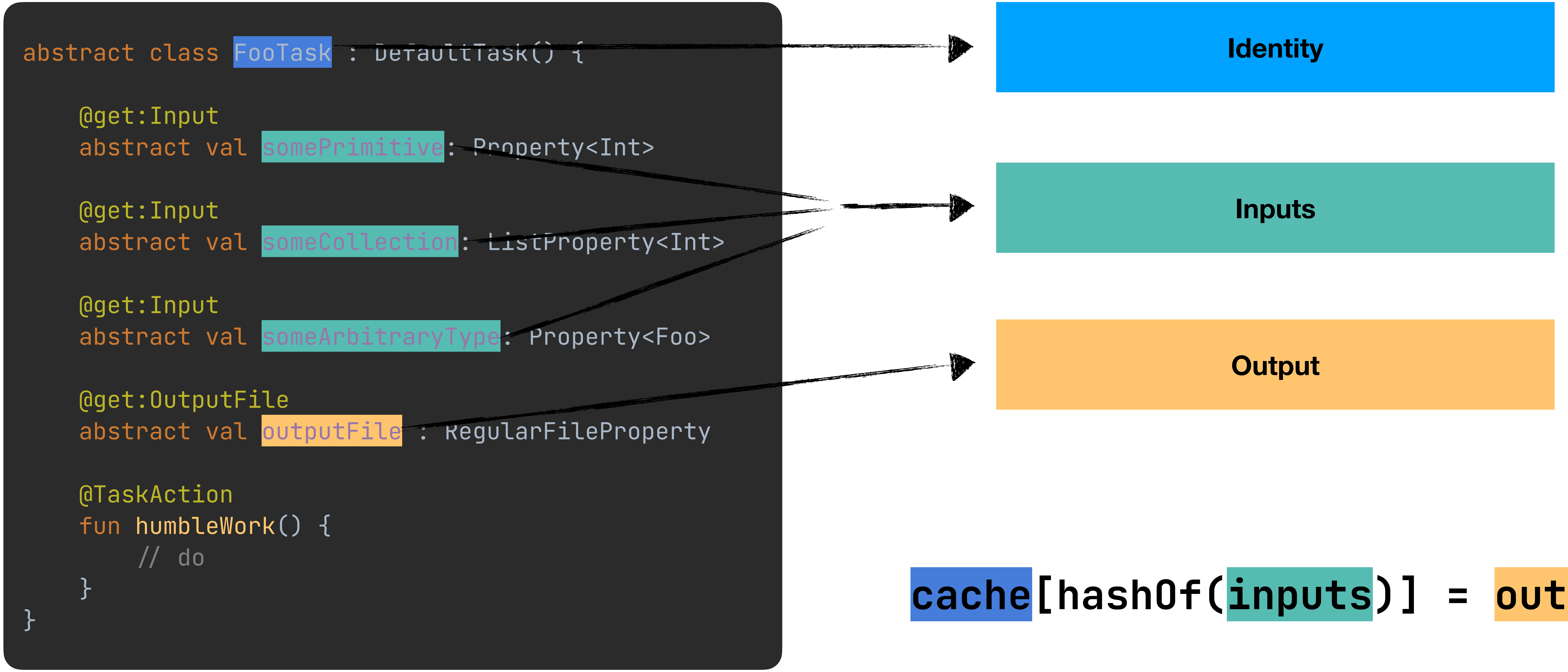
```kotlin
abstract class FooTask : DefaultTask() {

    @get:Input
    abstract val somePrimitive: Property<Int>

    @get:Input
    abstract val someCollection: ListProperty<Int>

    @get:Input
    abstract val someArbitraryType: Property<Foo>

    @get:OutputFile
    abstract val outputFile : RegularFileProperty

    @TaskAction
    fun humbleWork() {
        // do
    }
}
```

Identity

Inputs

Output

Модель Inputs/Outputs



```
abstract class FooTask : DefaultTask() {

    @get:Input
    abstract val somePrimitive: Property<Int>

    @get:Input
    abstract val someCollection: ListProperty<Int>

    @get:Input
    abstract val someArbitraryType: Property<Foo>

    @get:OutputFile
    abstract val outputFile : RegularFileProperty

    @TaskAction
    fun humbleWork() {
        // do
    }
}
```

Identity

Inputs

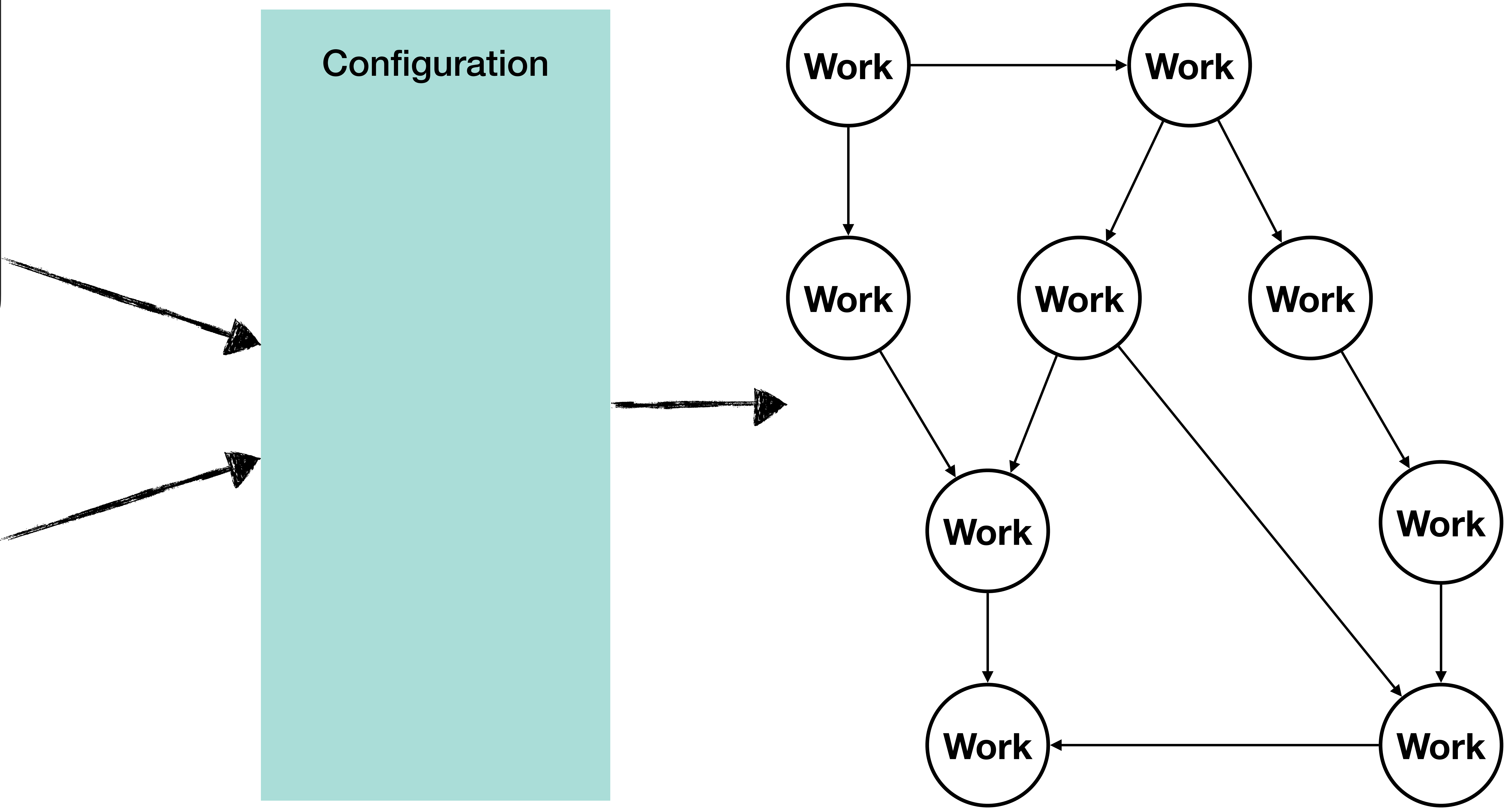
Output

```
cache[hashOf(inputs)] = output
```

Inputs/Outputs фазы конфигурации

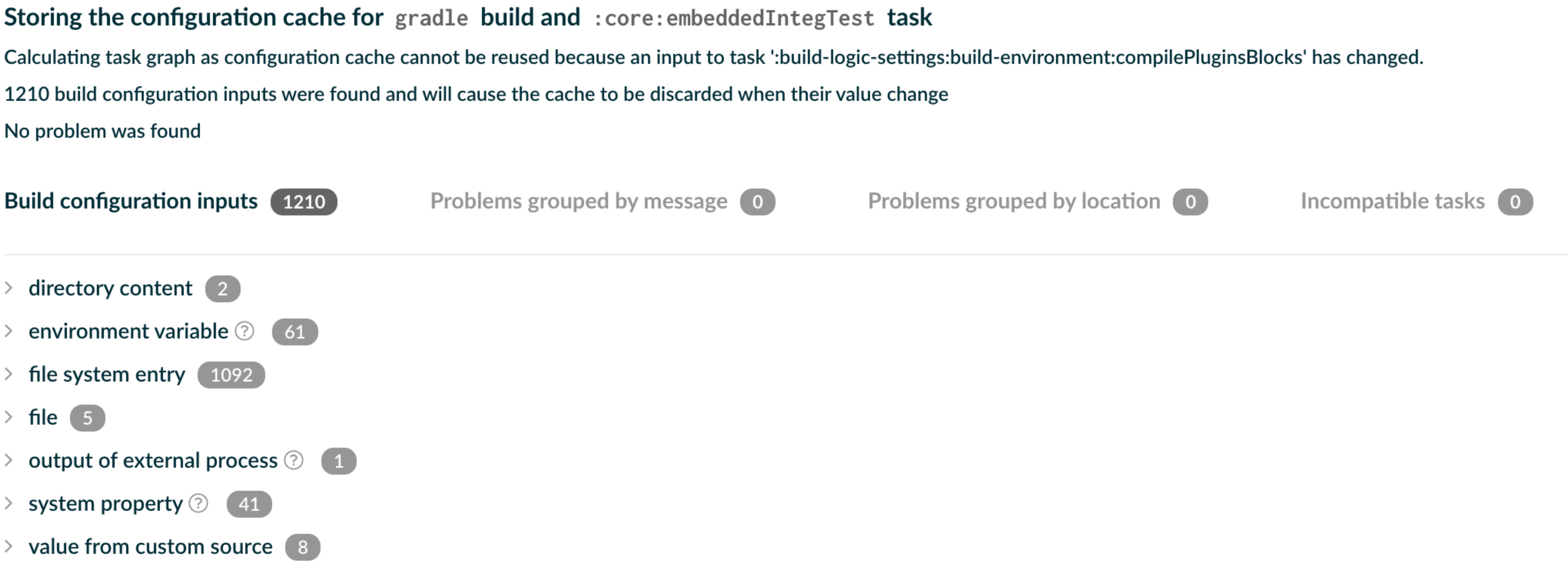
```
 // Static inputs

./gradlew —version
./gradlew foo —offline
./gradlew foo —include-build included
./gradlew foo -x bar
./gradlew bar
…

ConfigurationCacheKey.kt
```

```
 // Discoverable inputs

BuildLogic#System.getProperty(…)
BuildLogic#System.getEnv(…)
BuildLogic#file(…)
JVM
Dynamic deps versions
Build scripts
…

ConfigurationCacheFingerprint.kt
```



CC отчет

**<PROJECT_ROOT_DIR>/build/reports/configuration-cache/
5h2evejkvnvovzfhl85ajf25n/configuration-cache-report.html**



Storing the configuration cache for `gradle` build and `:core:embeddedIntegTest` task

Calculating task graph as configuration cache cannot be reused because an input to task ':build-logic-settings:build-environment:compilePluginsBlocks' has changed.

1210 build configuration inputs were found and will cause the cache to be discarded when their value change

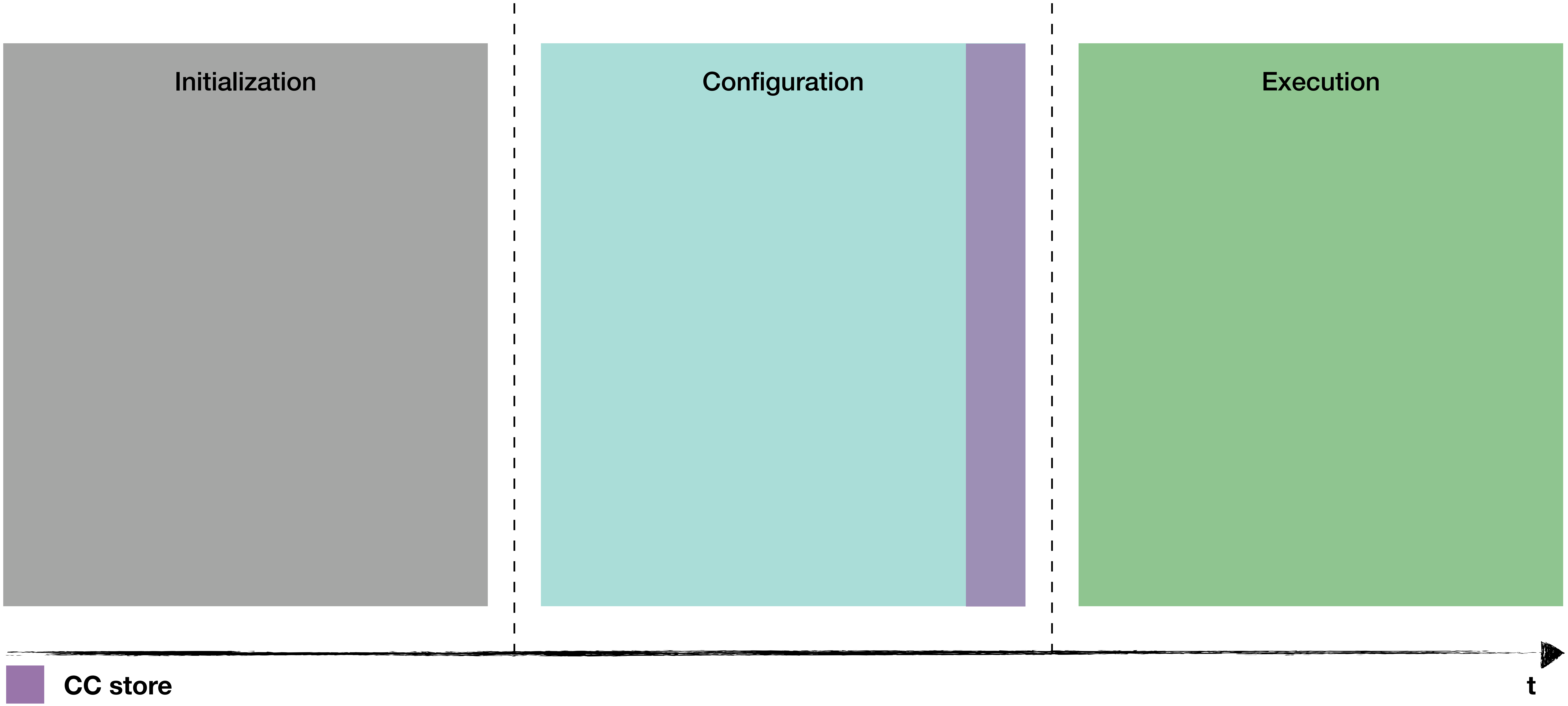
No problem was found

Build configuration inputs 1210 | Problems grouped by message 0 | Problems grouped by location 0 | Incompatible tasks 0

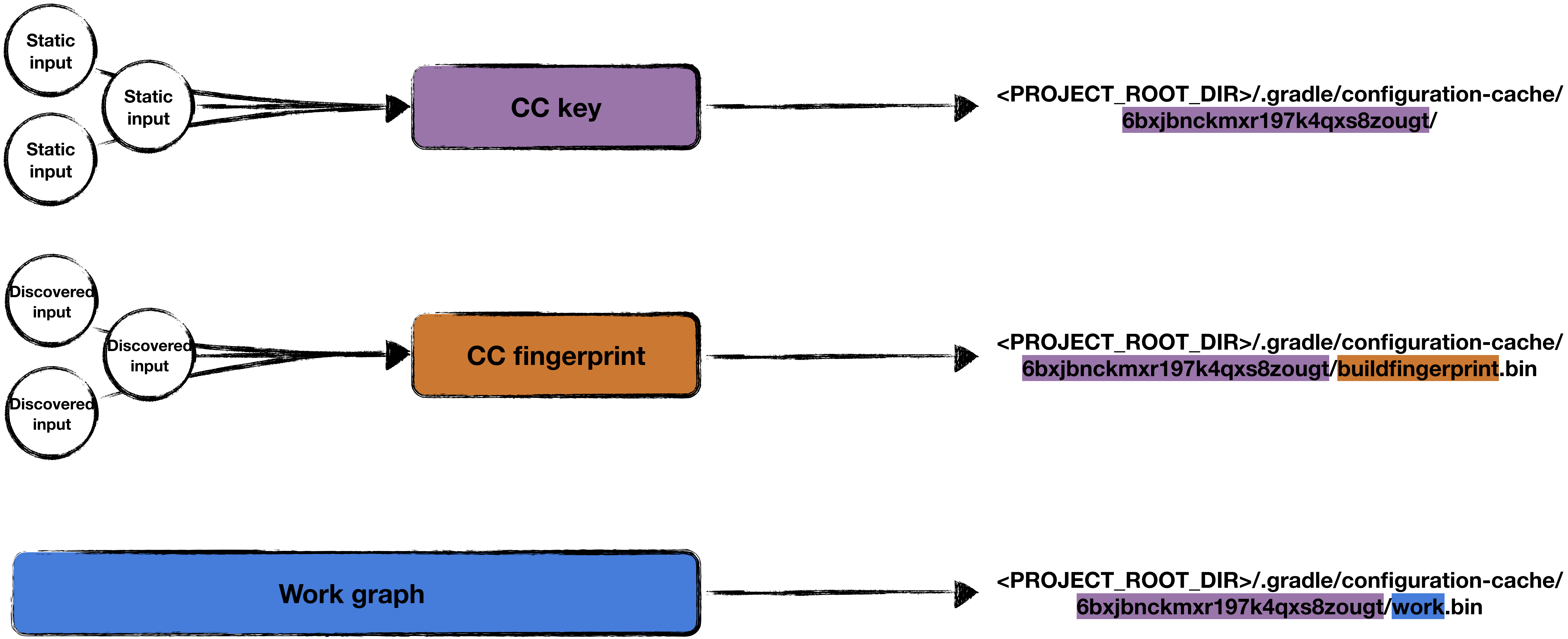
- directory content 2
- environment variable ? 61
- file system entry 1092
- file 5
- output of external process ? 1
- system property ? 41
- value from custom source 8

Improvements in the Build Configuration Input Tracking

CC miss



CC store

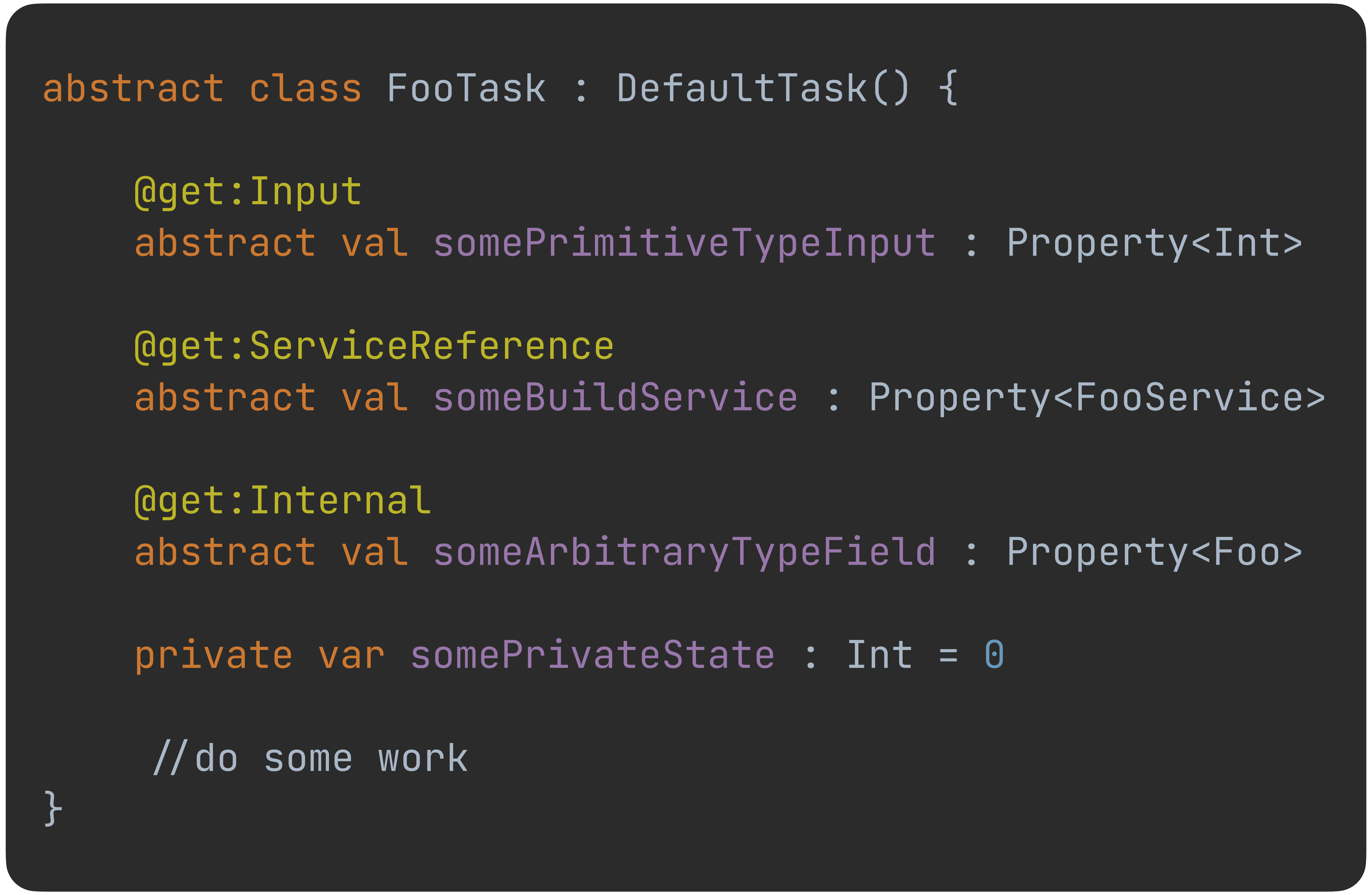


<PROJECT_ROOT_DIR>/.gradle/configuration-cache/
6bxjbnckmxr197k4qxs8zougt/

<PROJECT_ROOT_DIR>/.gradle/configuration-cache/
6bxjbnckmxr197k4qxs8zougt/buildfingerprint.bin

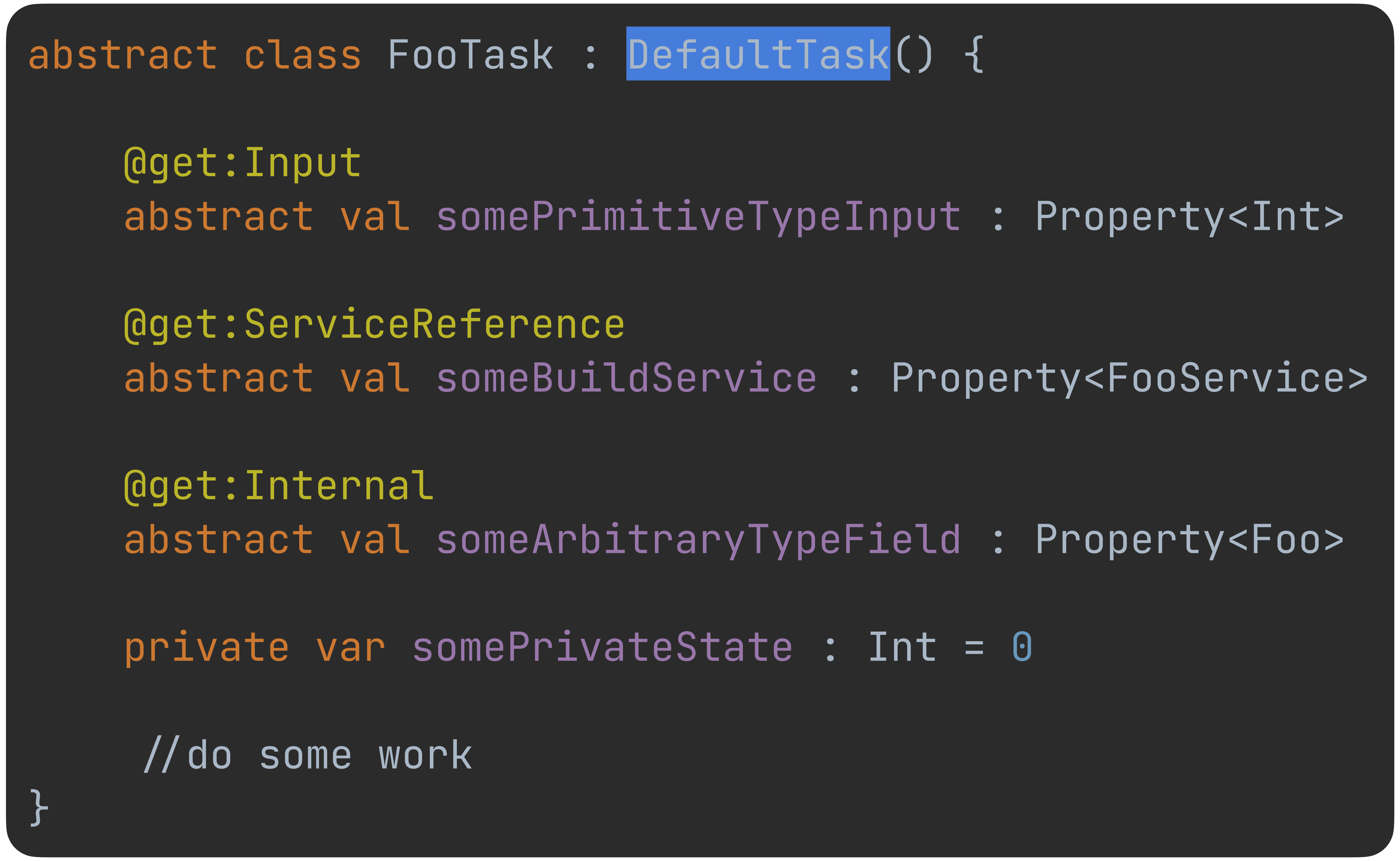
<PROJECT_ROOT_DIR>/.gradle/configuration-cache/
6bxjbnckmxr197k4qxs8zougt/work.bin

CC сериализация



```kotlin
abstract class FooTask : DefaultTask() {

    @get:Input
    abstract val somePrimitiveTypeInput : Property<Int>

    @get:ServiceReference
    abstract val someBuildService : Property<FooService>

    @get:Internal
    abstract val someArbitraryTypeField : Property<Foo>

    private var somePrivateState : Int = 0

     //do some work
}
```

Сериализация Task

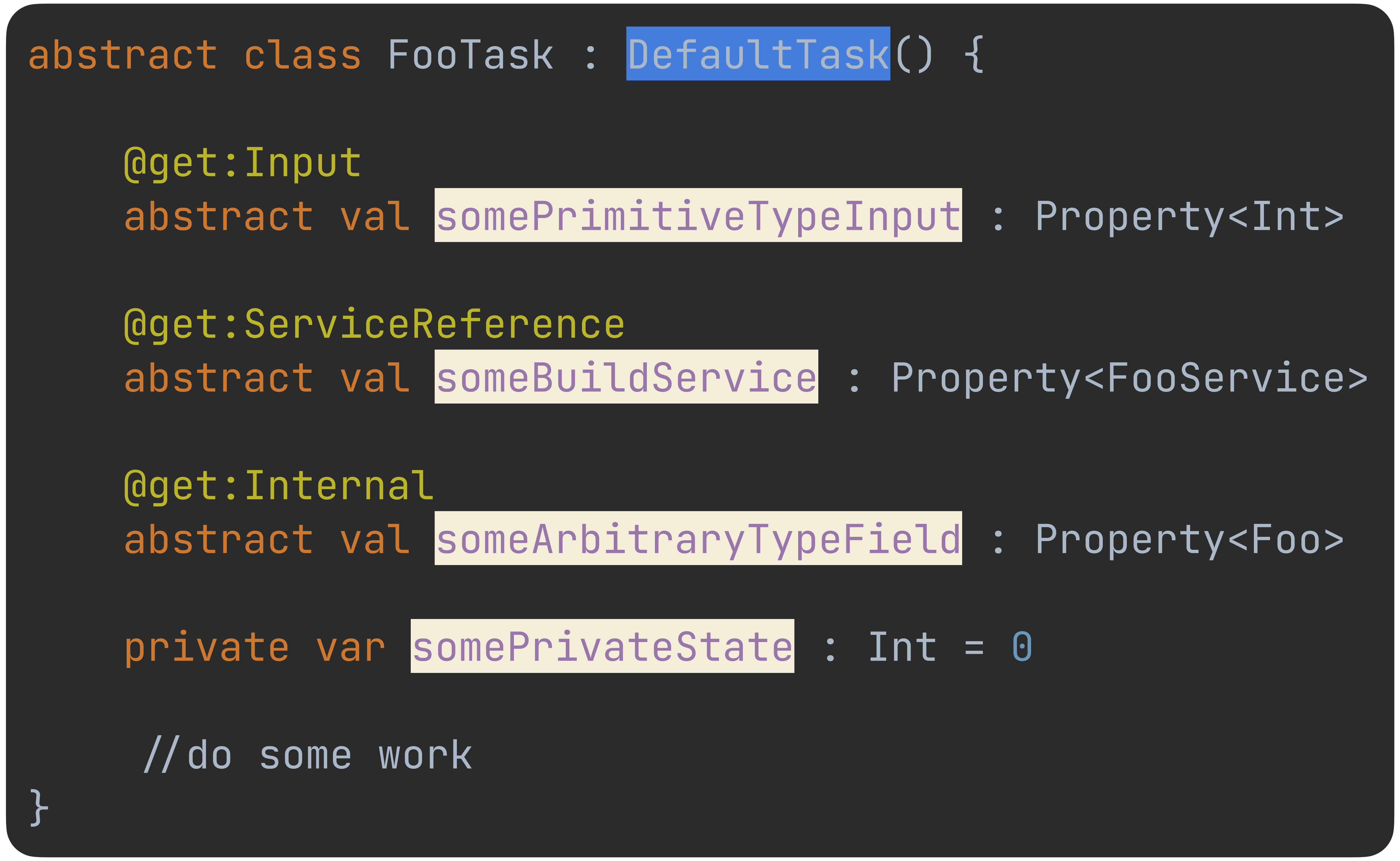


```kotlin
abstract class FooTask : DefaultTask() {

    @get:Input
    abstract val somePrimitiveTypeInput : Property<Int>

    @get:ServiceReference
    abstract val someBuildService : Property<FooService>

    @get:Internal
    abstract val someArbitraryTypeField : Property<Foo>

    private var somePrivateState : Int = 0

     //do some work
}
```

TaskNodeCodec

Relevant fields walking

Сериализация Task

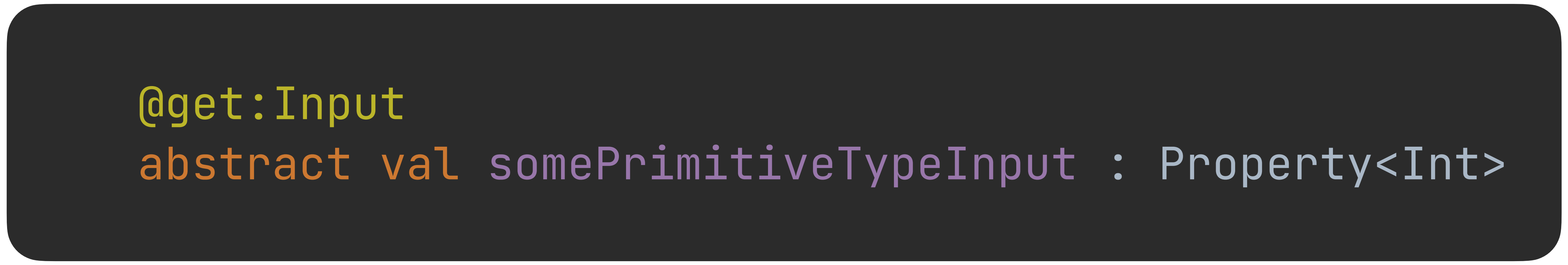


```
abstract class FooTask : DefaultTask() {

    @get:Input
    abstract val somePrimitiveTypeInput : Property<Int>

    @get:ServiceReference
    abstract val someBuildService : Property<FooService>

    @get:Internal
    abstract val someArbitraryTypeField : Property<Foo>

    private var somePrivateState : Int = 0

     //do some work
}
```

TaskNodeCodec

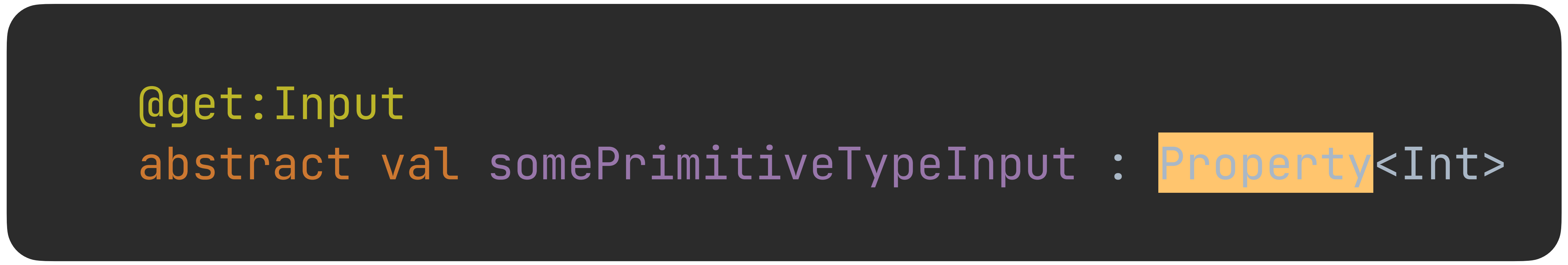
Relevant fields walking

Сериализация Provider<Int>

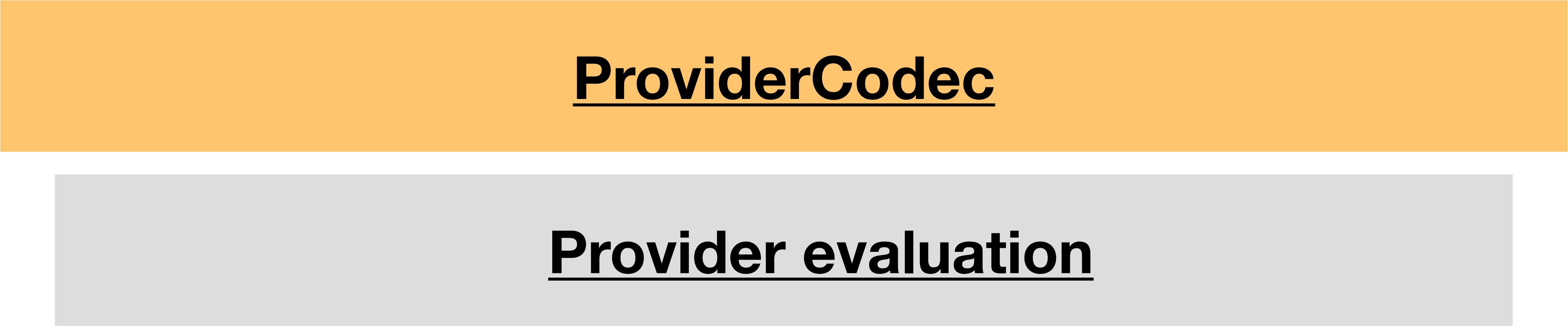


```
@get:Input
abstract val somePrimitiveTypeInput : Property<Int>
```

Сериализация Provider<Int>



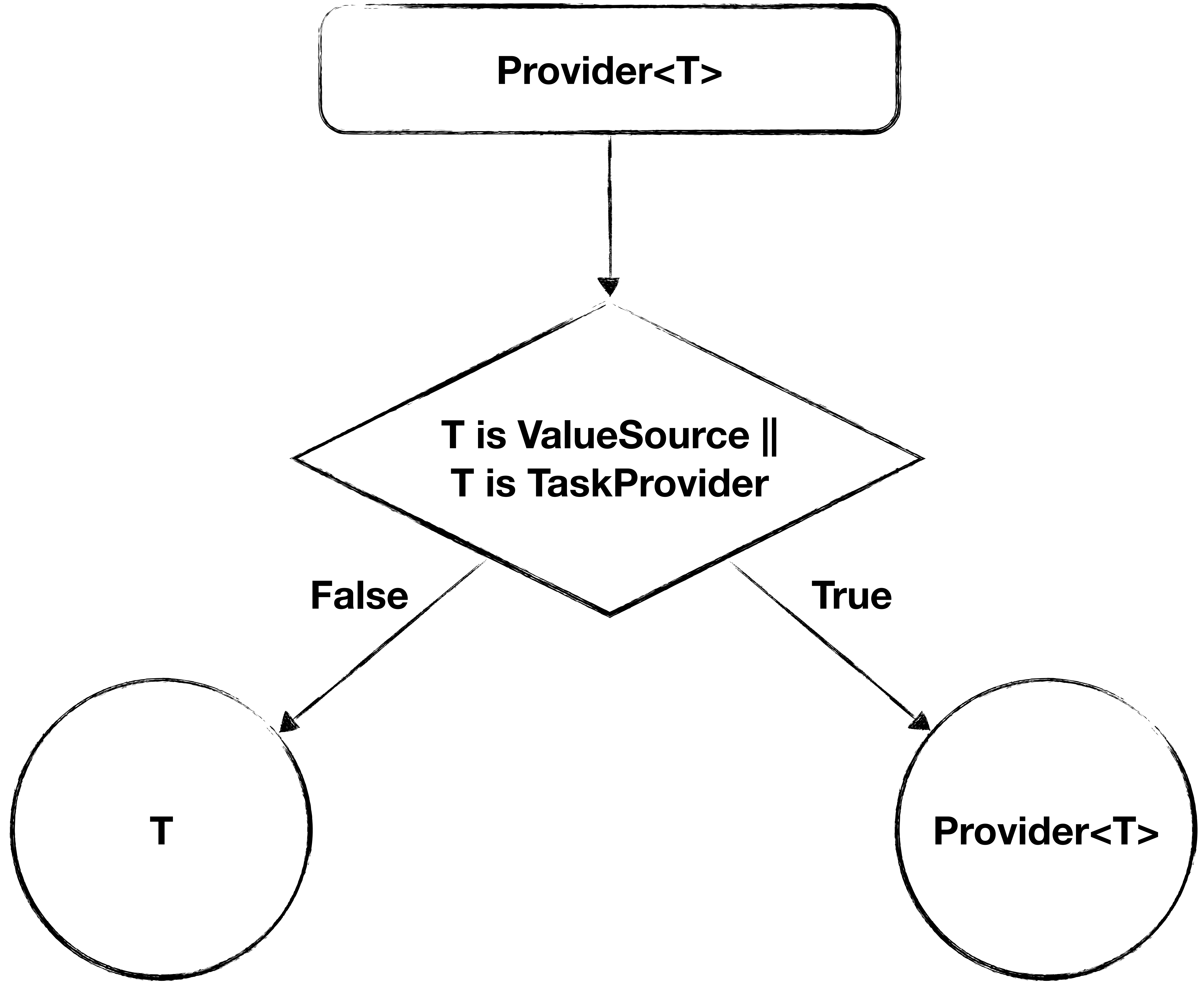
```
@get:Input
abstract val somePrimitiveTypeInput : Property<Int>
```



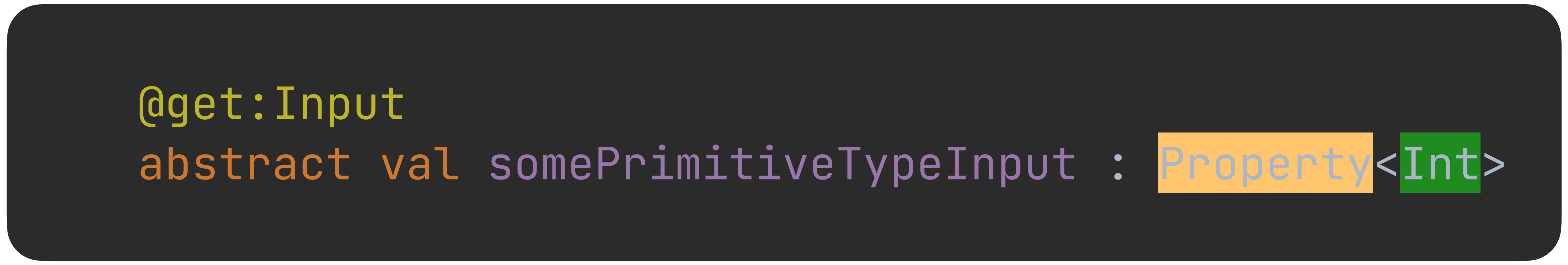
ProviderCodec

Provider evaluation

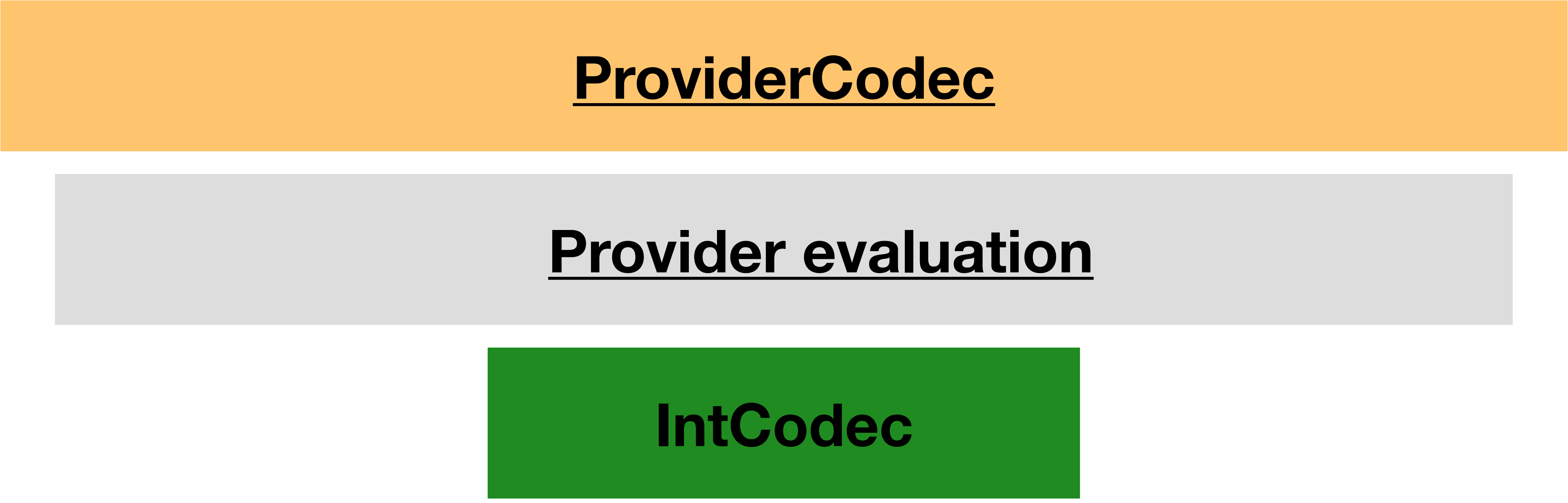
Эвалюация Provider



Сериализация Provider<Int>



```
@get:Input
abstract val somePrimitiveTypeInput : Property<Int>
```

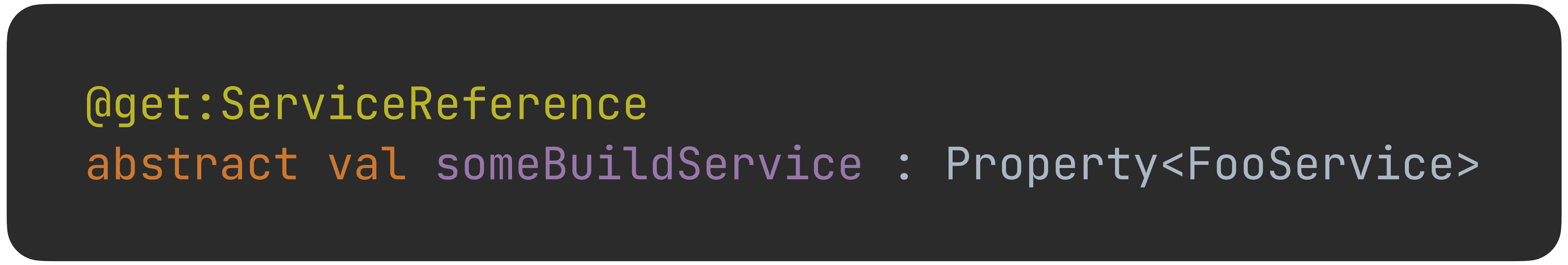


ProviderCodec

Provider evaluation

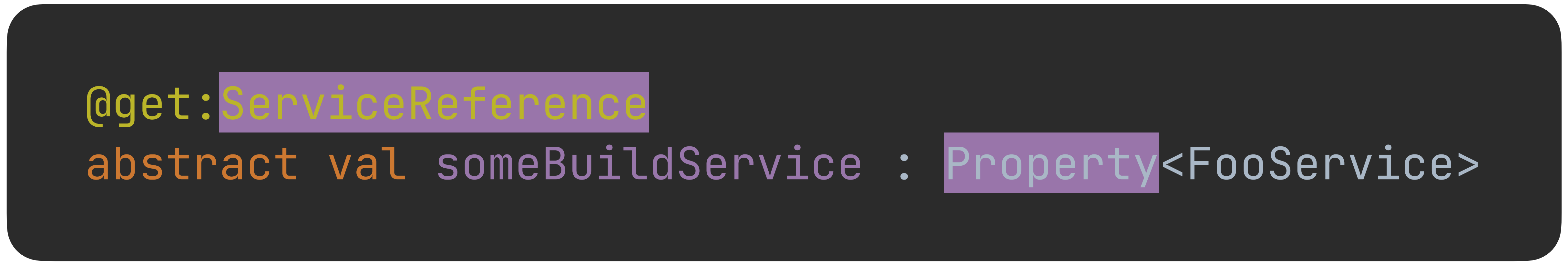
IntCodec

Сериализация Provider<BuildService>



```
@get:ServiceReference
abstract val someBuildService : Property<FooService>
```

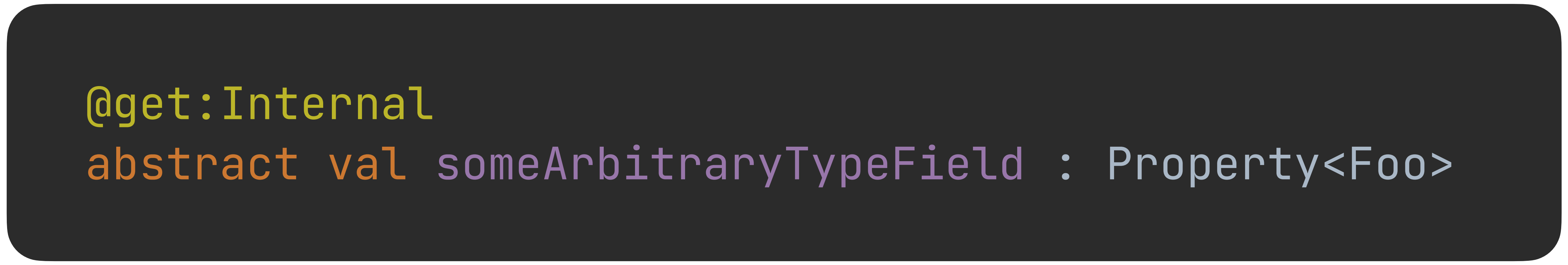
Сериализация Provider<BuildService>



```kotlin
@get:ServiceReference
abstract val someBuildService : Property<FooService>
```

BuildServiceProviderCodec

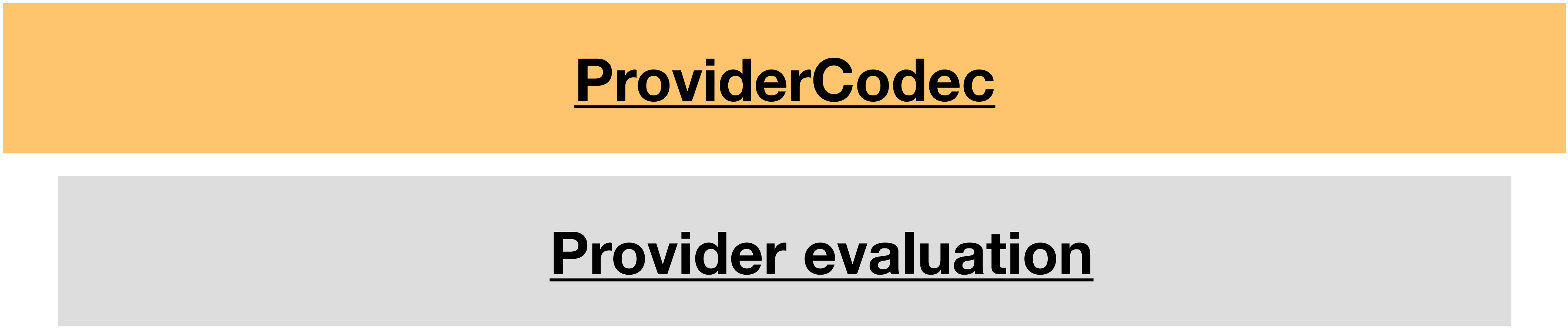
Сериализация Provider<Bean>



```
@get:Internal
abstract val someArbitraryTypeField : Property<Foo>
```

Сериализация Provider<Bean>

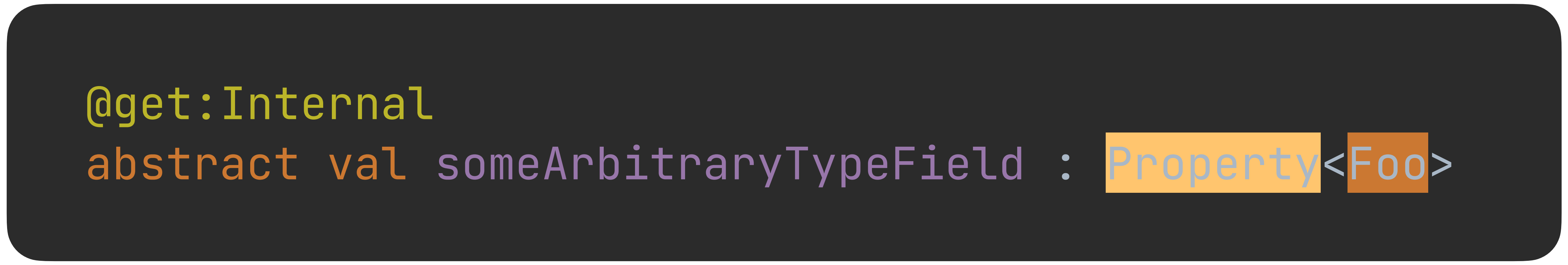
```
@get:Internal
abstract val someArbitraryTypeField : Property<Foo>
```



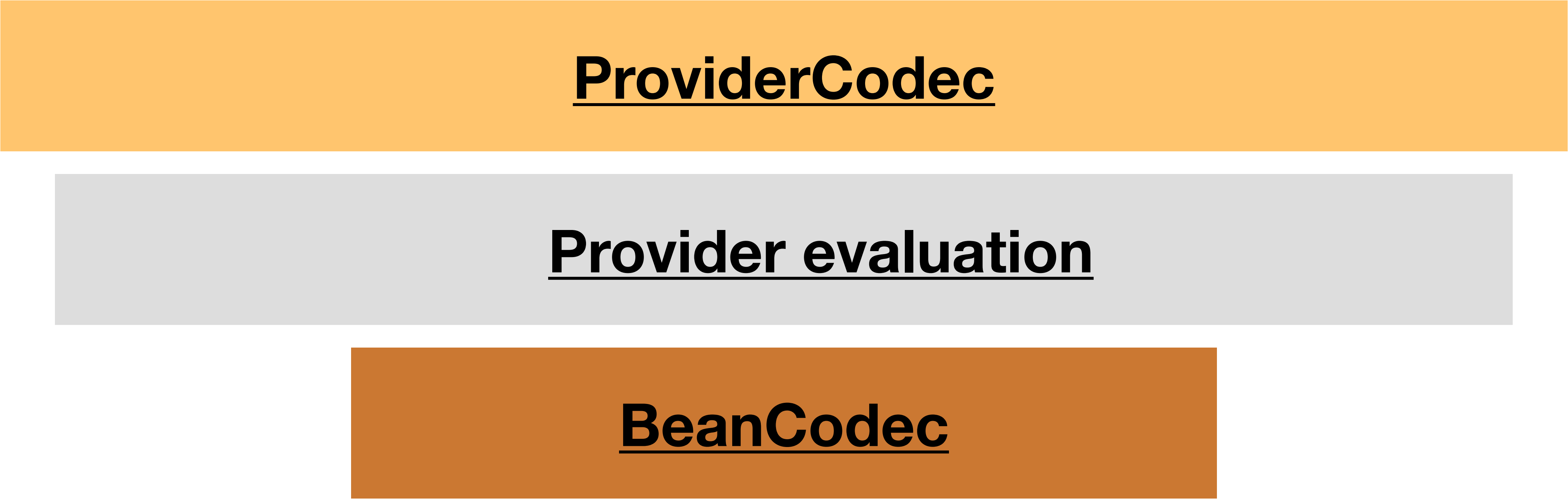
ProviderCodec

Provider evaluation

Сериализация Provider<Bean>



```kotlin
@get:Internal
abstract val someArbitraryTypeField : Property<Foo>
```



ProviderCodec

Provider evaluation

BeanCodec

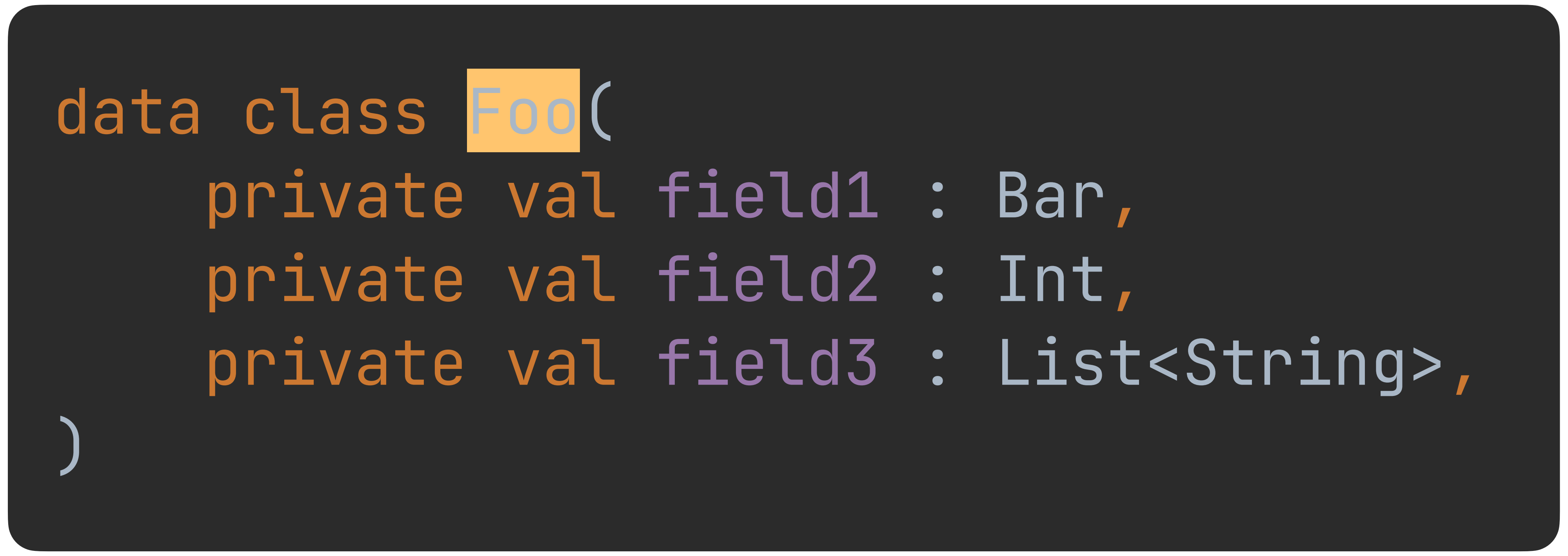
Сериализация произвольных типов(aka Bean)

```kotlin
data class Foo(
    private val field1 : Bar,
    private val field2 : Int,
    private val field3 : List<String>,
)
```

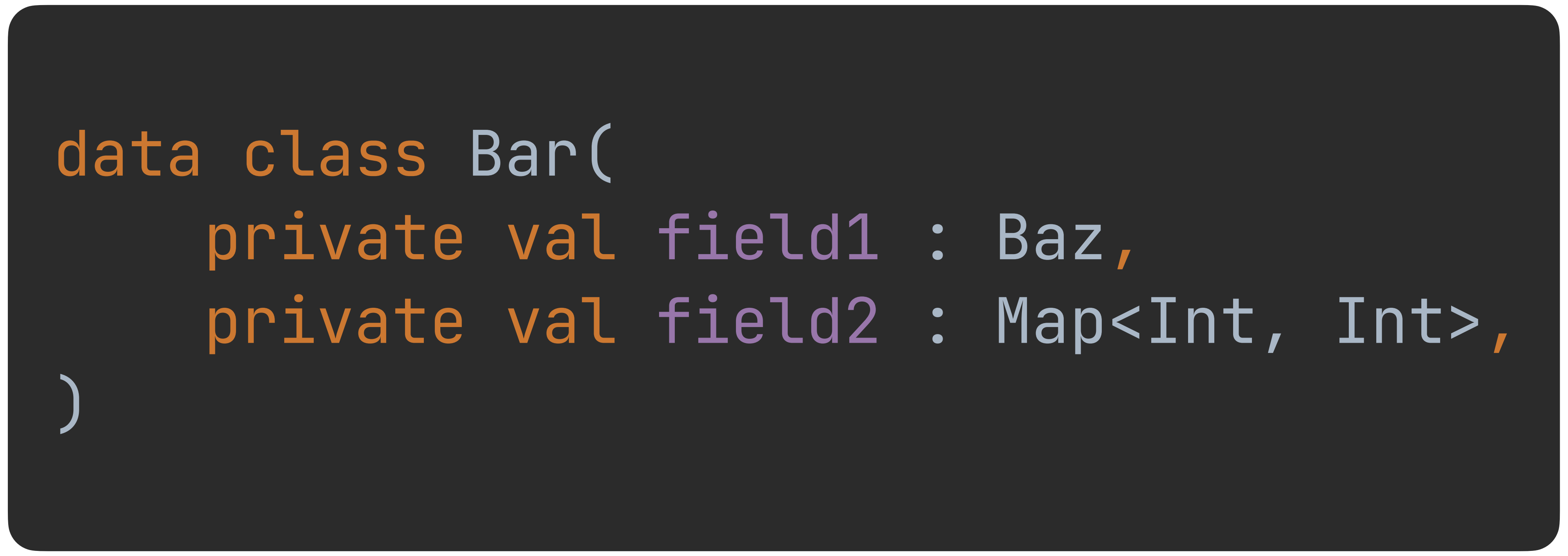
```kotlin
data class Bar(
    private val field1 : Baz,
    private val field2 : Map<Int, Int>,
)
```

```kotlin
data class Baz(
    .
    .
    .
)
```

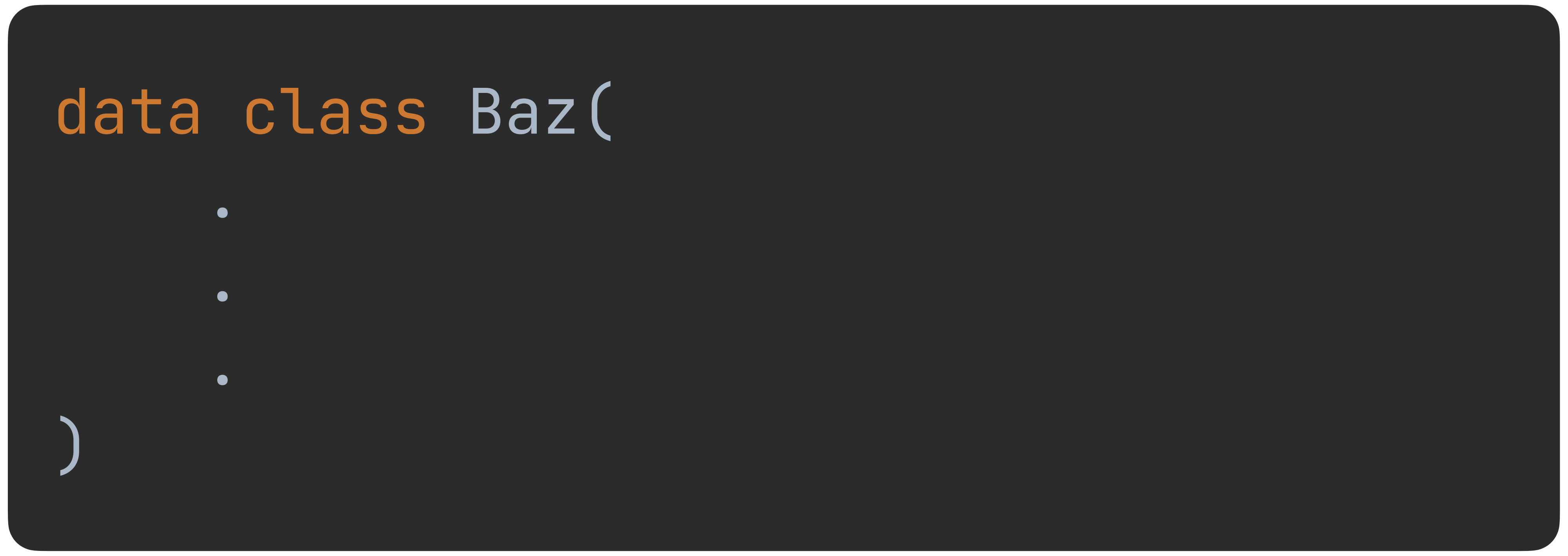
Сериализация произвольных типов(aka Bean)



```
data class Foo(
    private val field1 : Bar,
    private val field2 : Int,
    private val field3 : List<String>,
)
```



```
data class Bar(
    private val field1 : Baz,
    private val field2 : Map<Int, Int>,
)
```



```
data class Baz(
    .
    .
    .
)
```

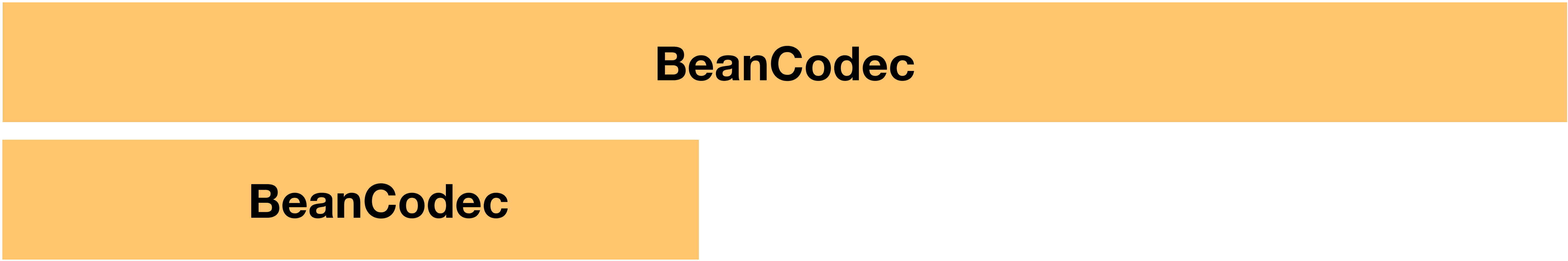
BeanCodec

Сериализация произвольных типов(aka Bean)

```
data class Foo(
    private val field1 : Bar,
    private val field2 : Int,
    private val field3 : List<String>,
)
```

```
data class Bar(
    private val field1 : Baz,
    private val field2 : Map<Int, Int>,
)
```

```
data class Baz(
    .
    .
    .
)
```

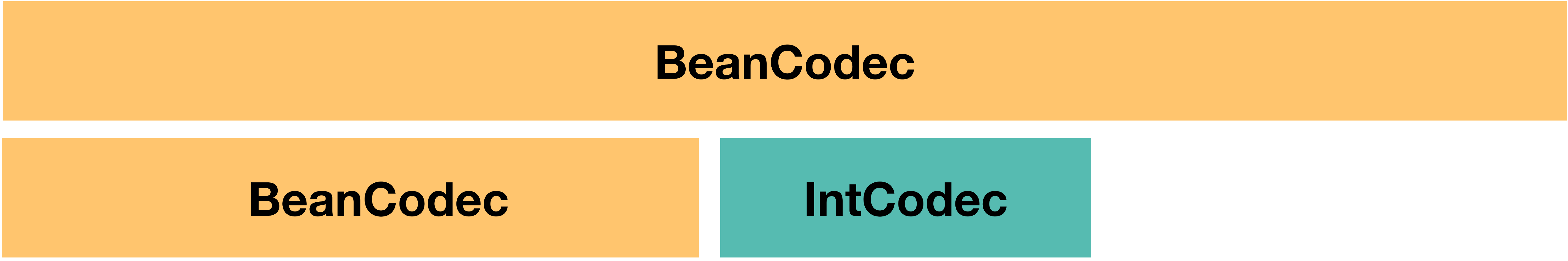


Сериализация произвольных типов(aka Bean)

```
data class Foo(
    private val field1 : Bar,
    private val field2 : Int,
    private val field3 : List<String>,
)
```

```
data class Bar(
    private val field1 : Baz,
    private val field2 : Map<Int, Int>,
)
```

```
data class Baz(
    .
    .
    .
)
```

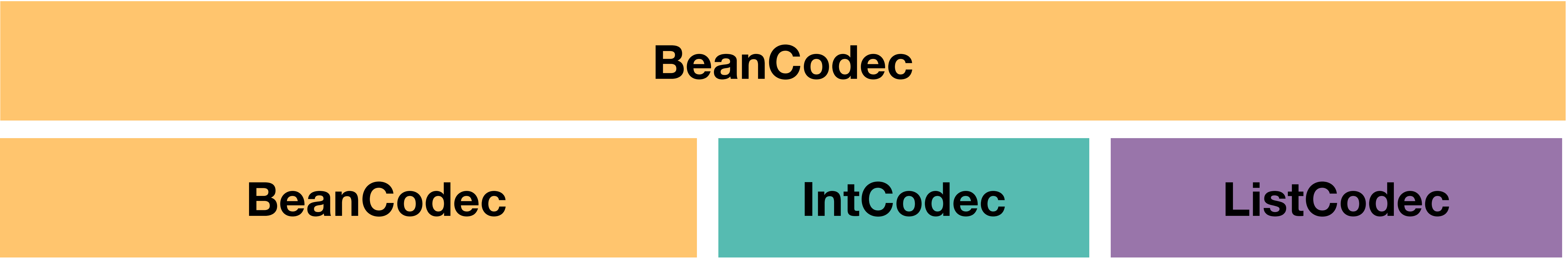


Сериализация произвольных типов(aka Bean)

```
data class Foo(
    private val field1 : Bar,
    private val field2 : Int,
    private val field3 : List<String>,
)
```

```
data class Bar(
    private val field1 : Baz,
    private val field2 : Map<Int, Int>,
)
```

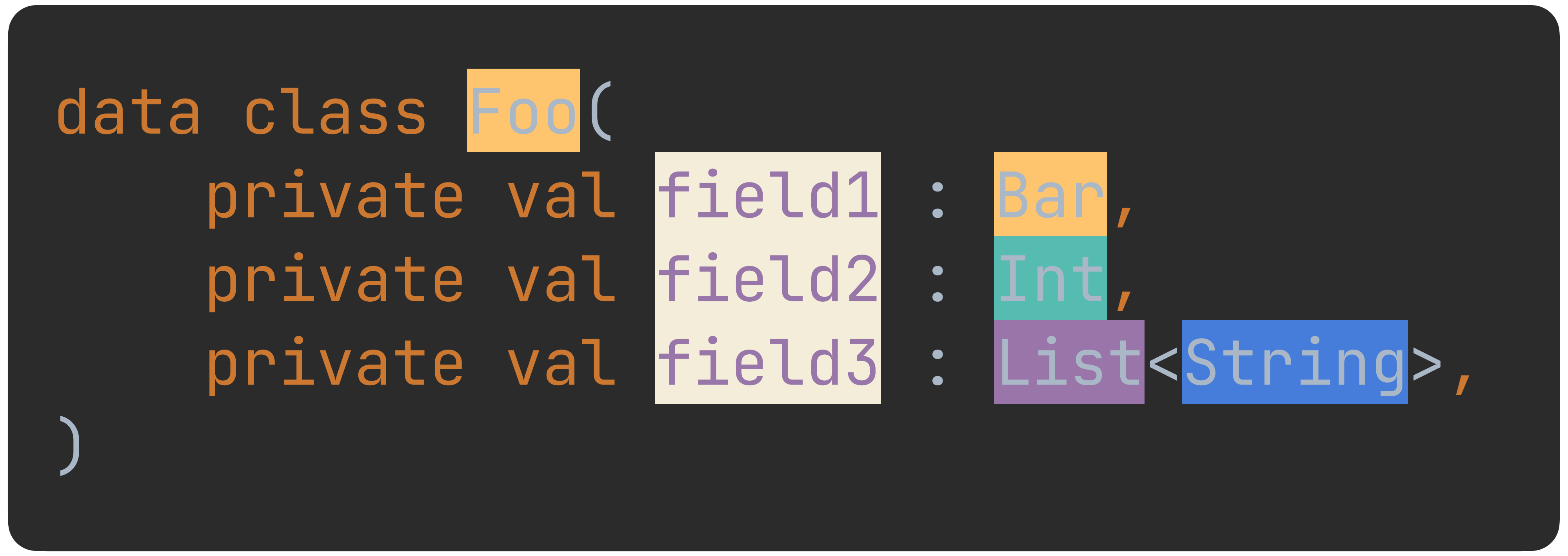
```
data class Baz(
    .
    .
    .
)
```



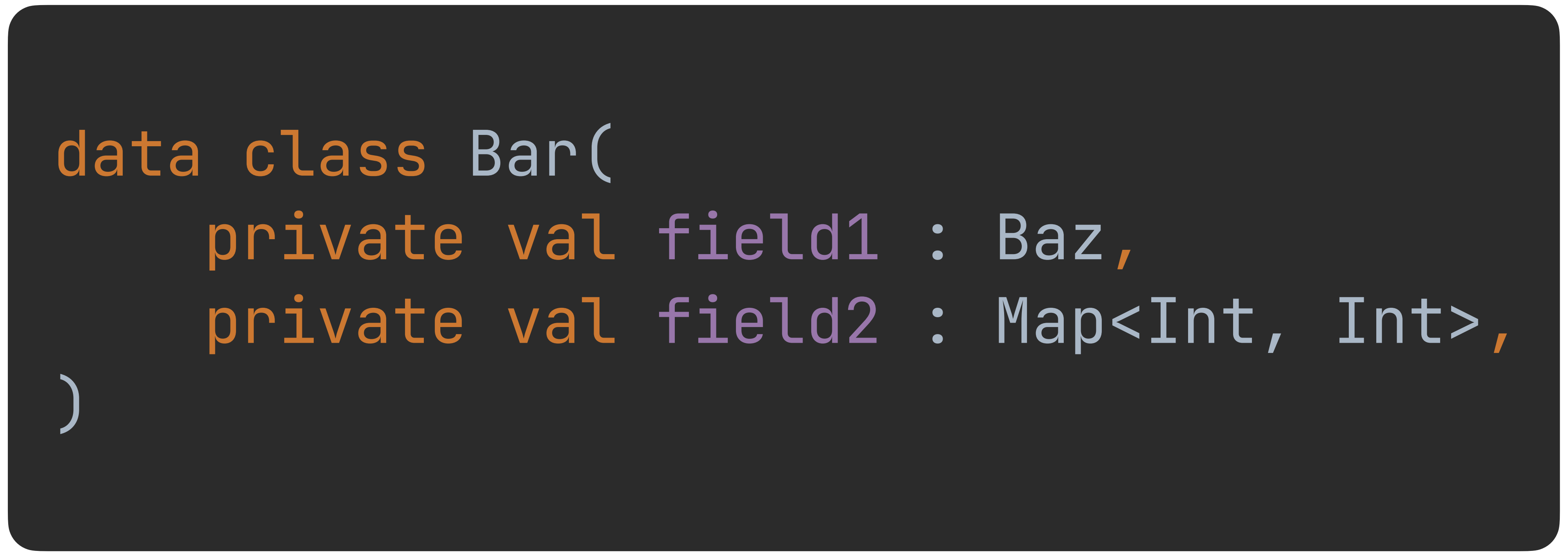
BeanCodec

BeanCodec IntCodec ListCodec

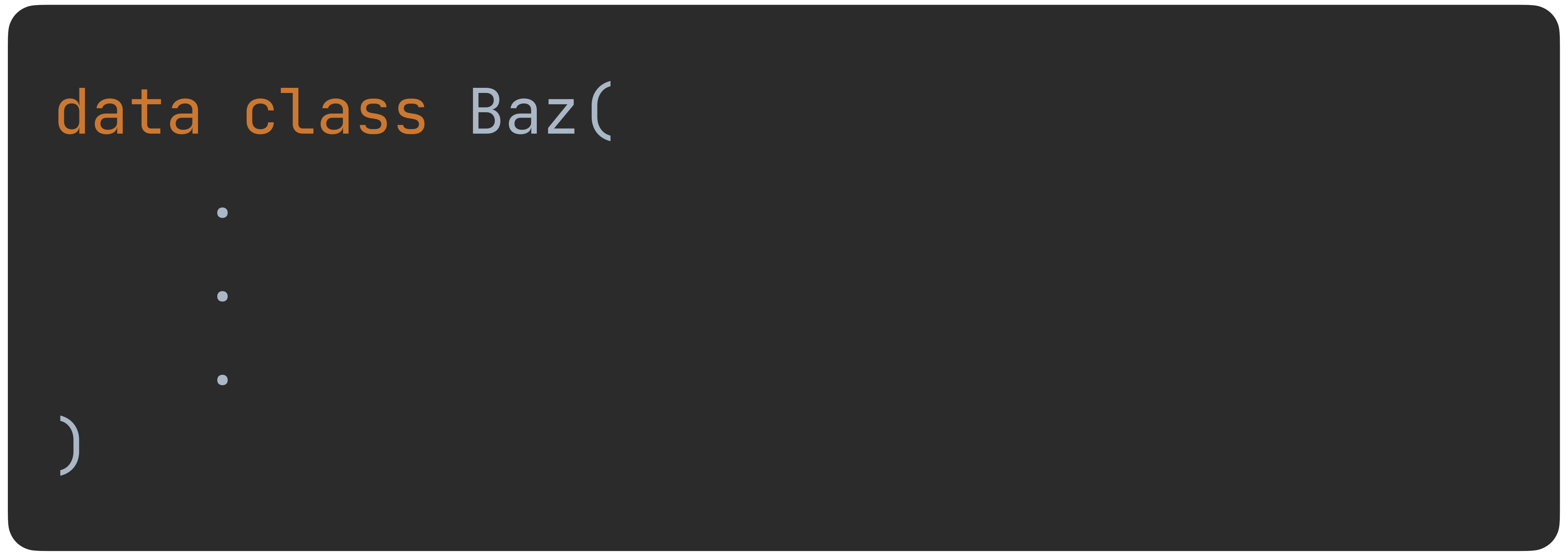
Сериализация произвольных типов(aka Bean)



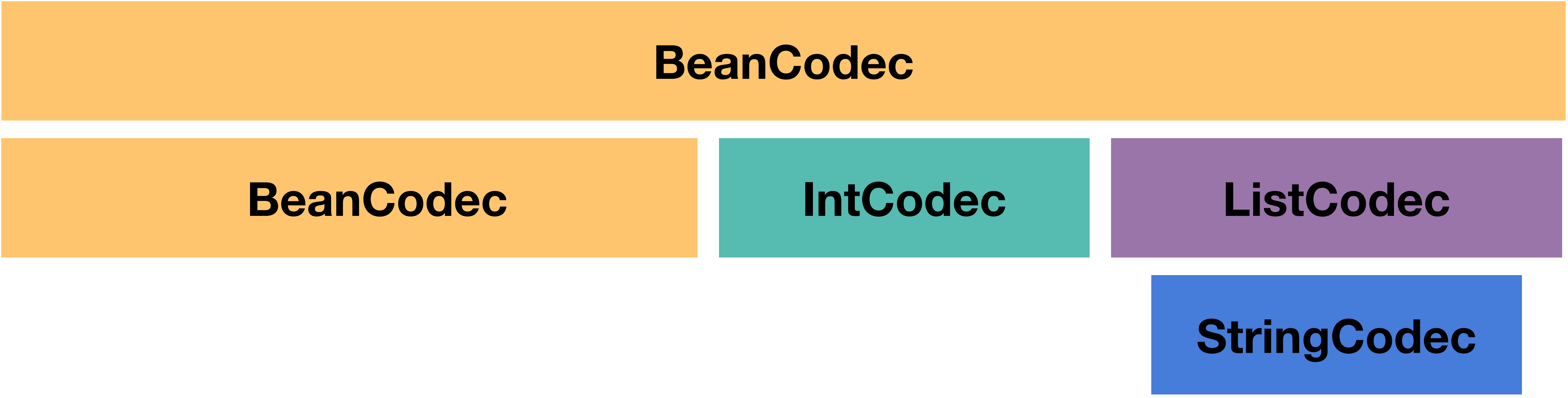
```
data class Foo(
    private val field1 : Bar,
    private val field2 : Int,
    private val field3 : List<String>,
)
```



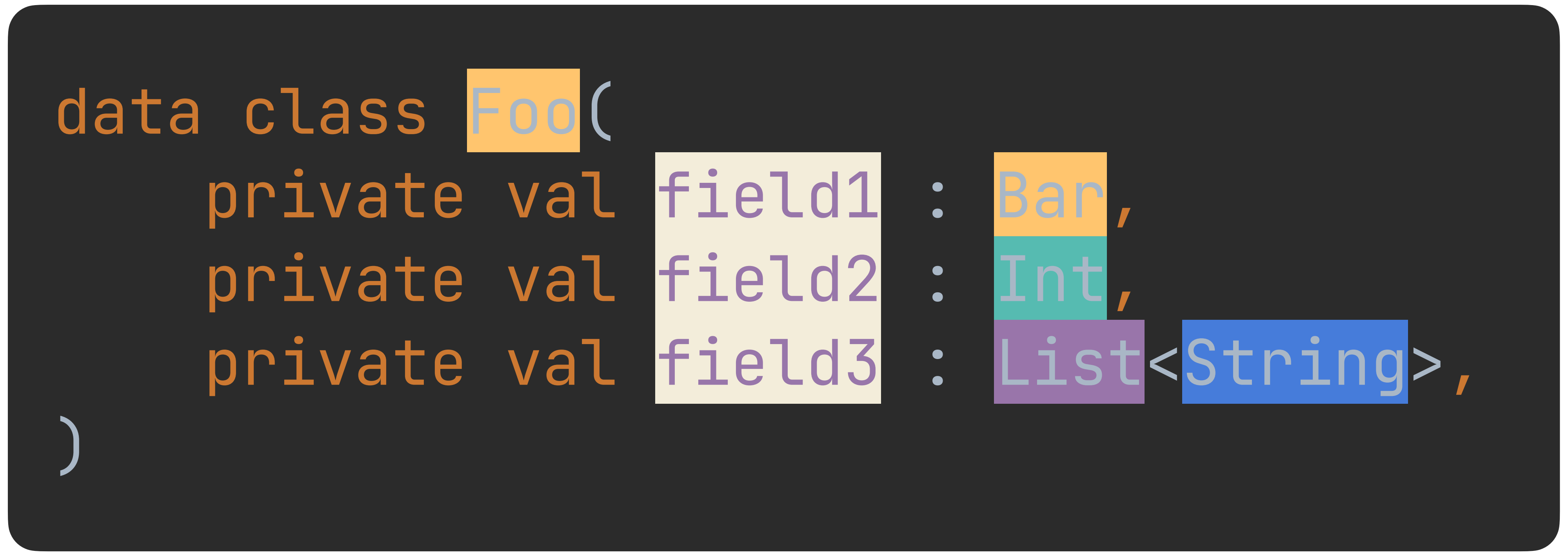
```
data class Bar(
    private val field1 : Baz,
    private val field2 : Map<Int, Int>,
)
```



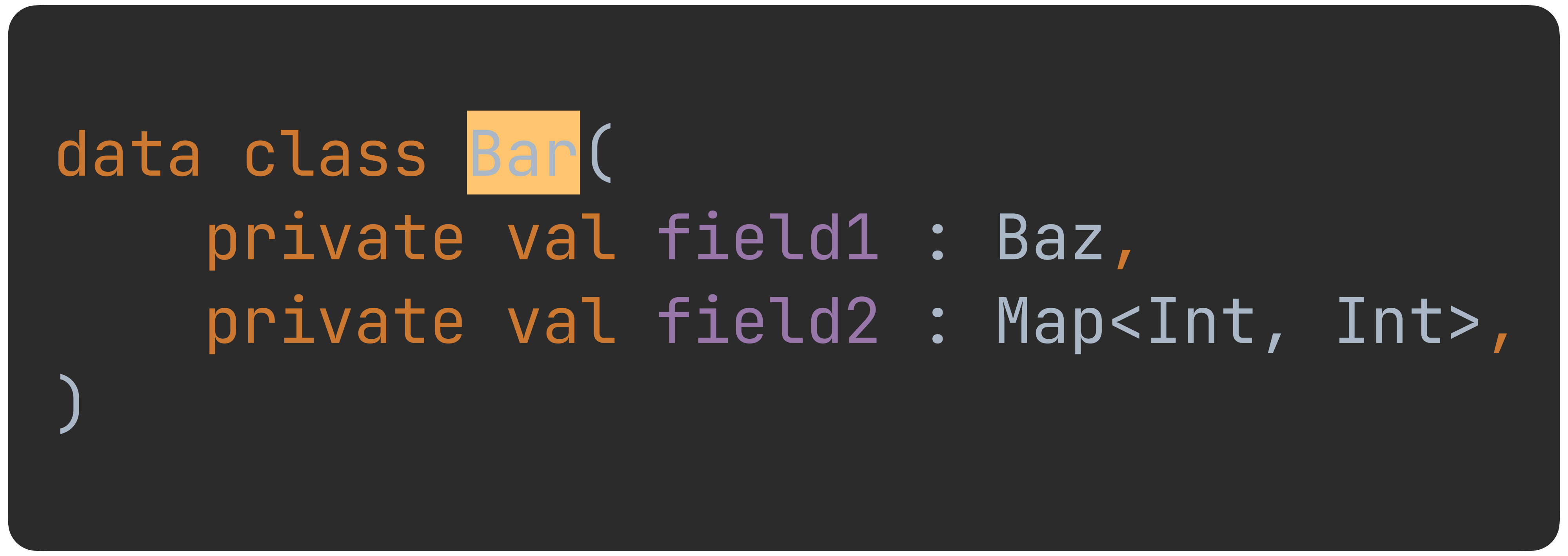
```
data class Baz(
    .
    .
    .
)
```



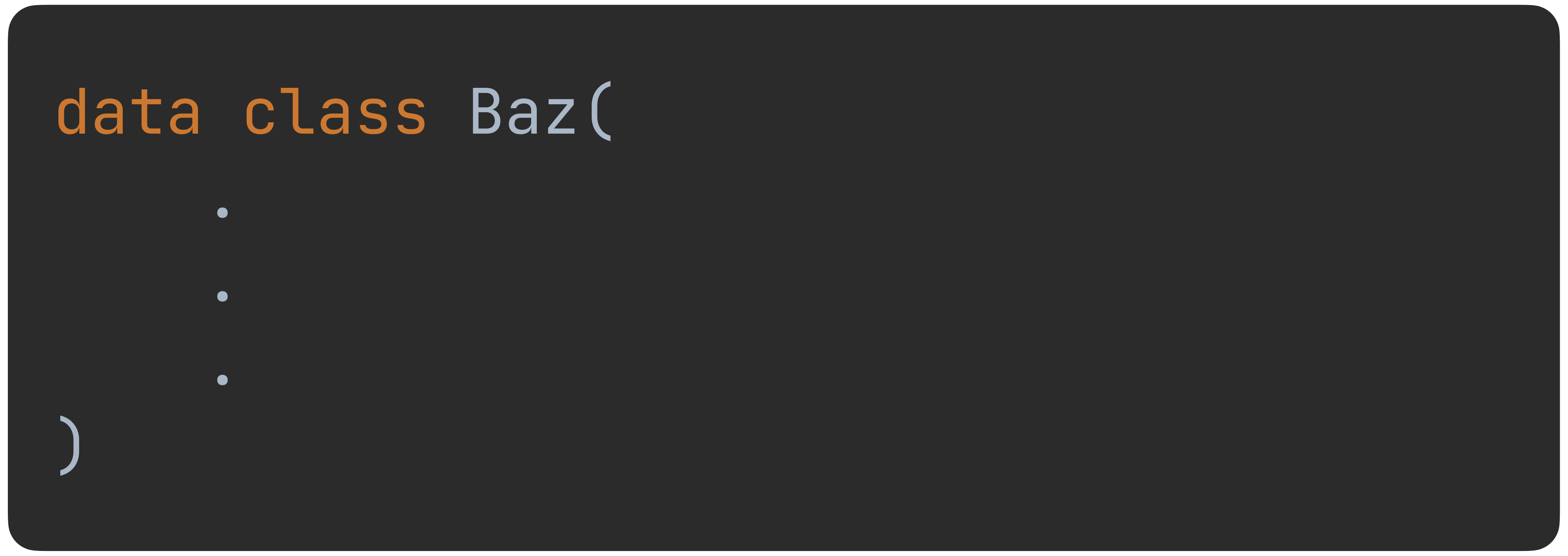
Сериализация произвольных типов(aka Bean)



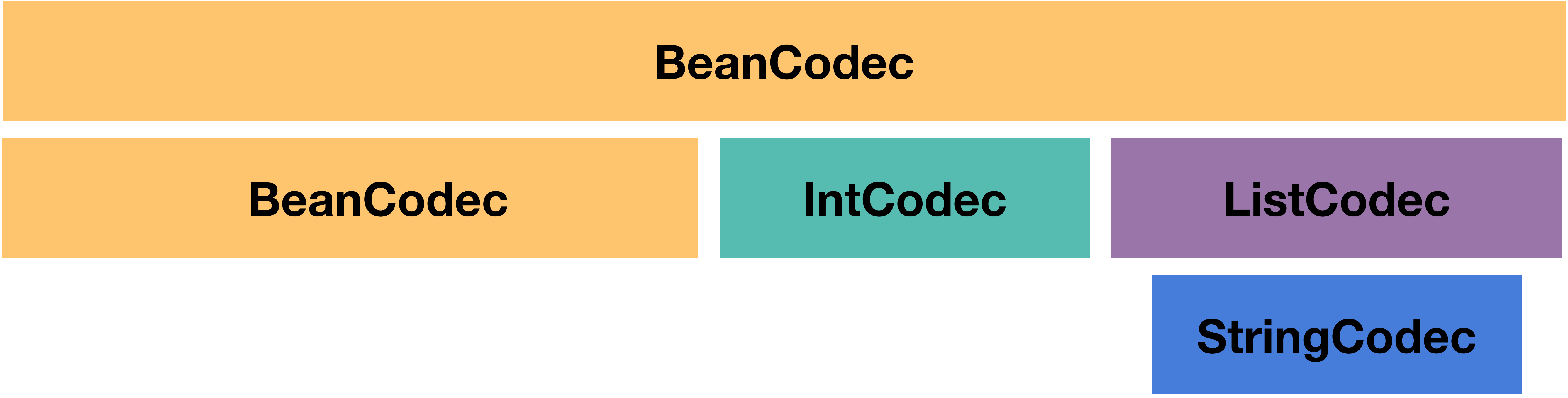
```
data class Foo(
    private val field1 : Bar,
    private val field2 : Int,
    private val field3 : List<String>,
)
```



```
data class Bar(
    private val field1 : Baz,
    private val field2 : Map<Int, Int>,
)
```



```
data class Baz(
    .
    .
    .
)
```



BeanCodec

BeanCodec

IntCodec

ListCodec

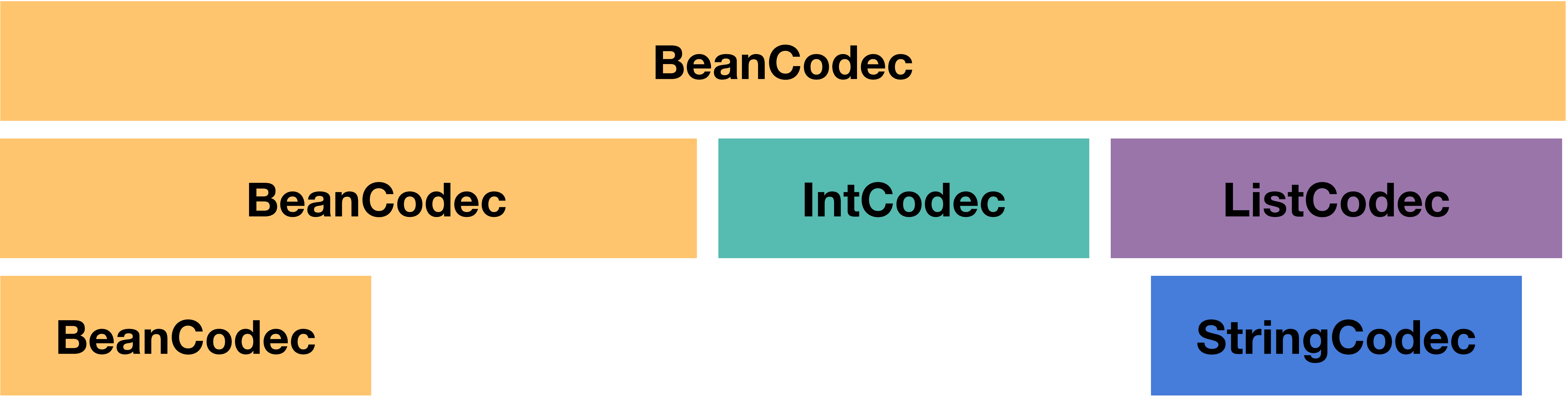
StringCodec

Сериализация произвольных типов(aka Bean)

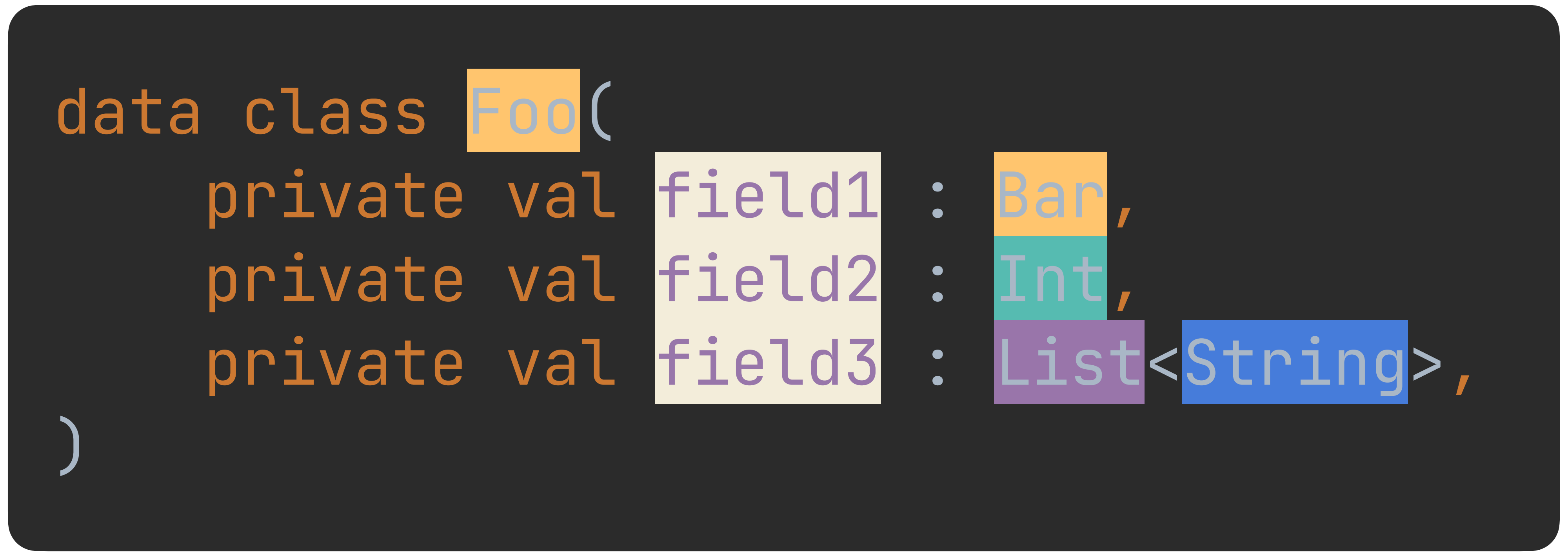
```
data class Foo(
    private val field1 : Bar,
    private val field2 : Int,
    private val field3 : List<String>,
)
```

```
data class Bar(
    private val field1 : Baz,
    private val field2 : Map<Int, Int>,
)
```

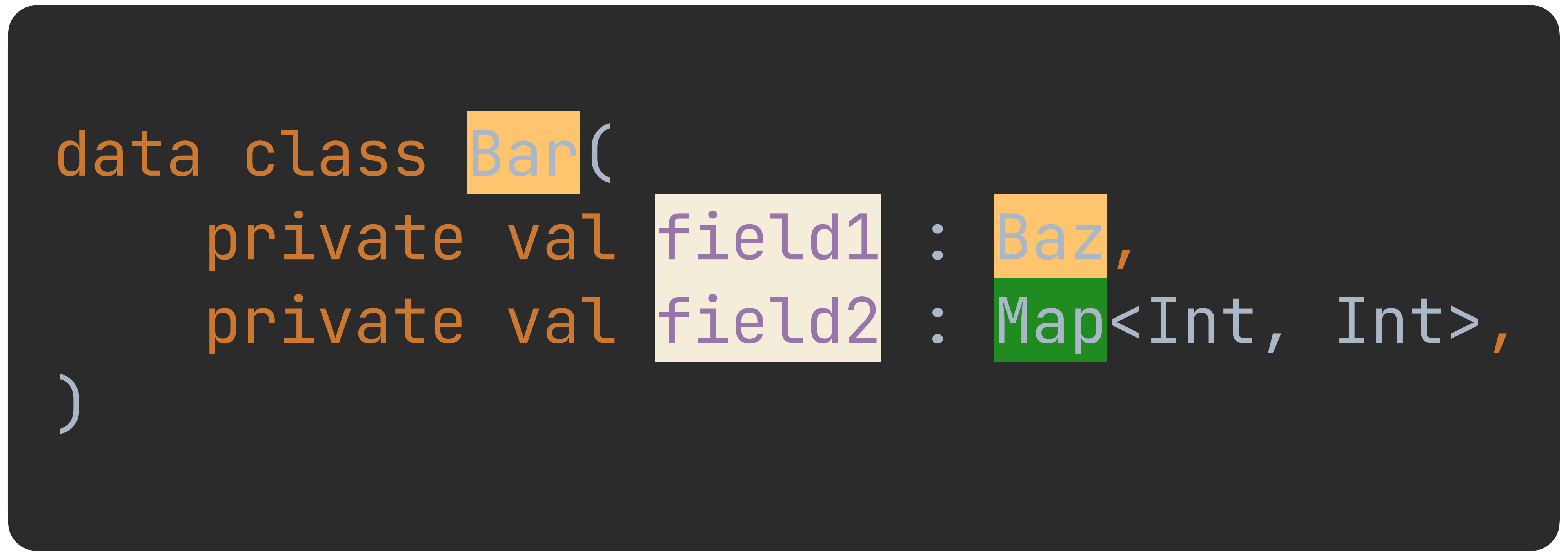
```
data class Baz(
    .
    .
    .
)
```



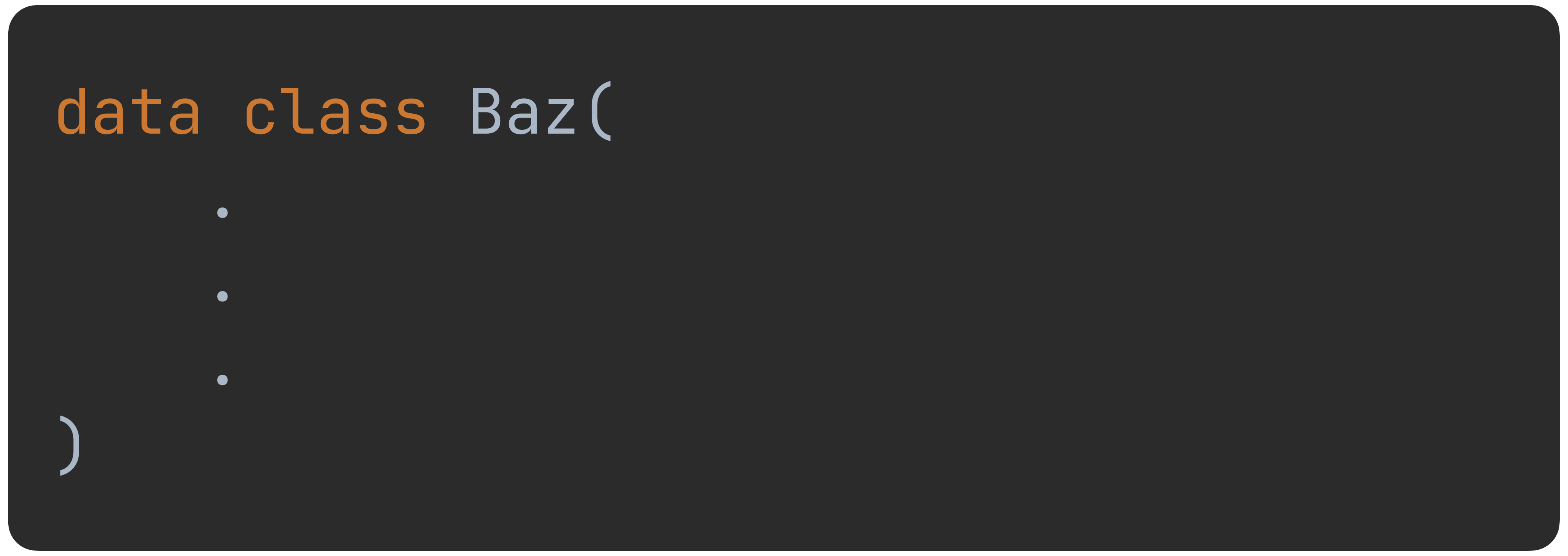
Сериализация произвольных типов(aka Bean)



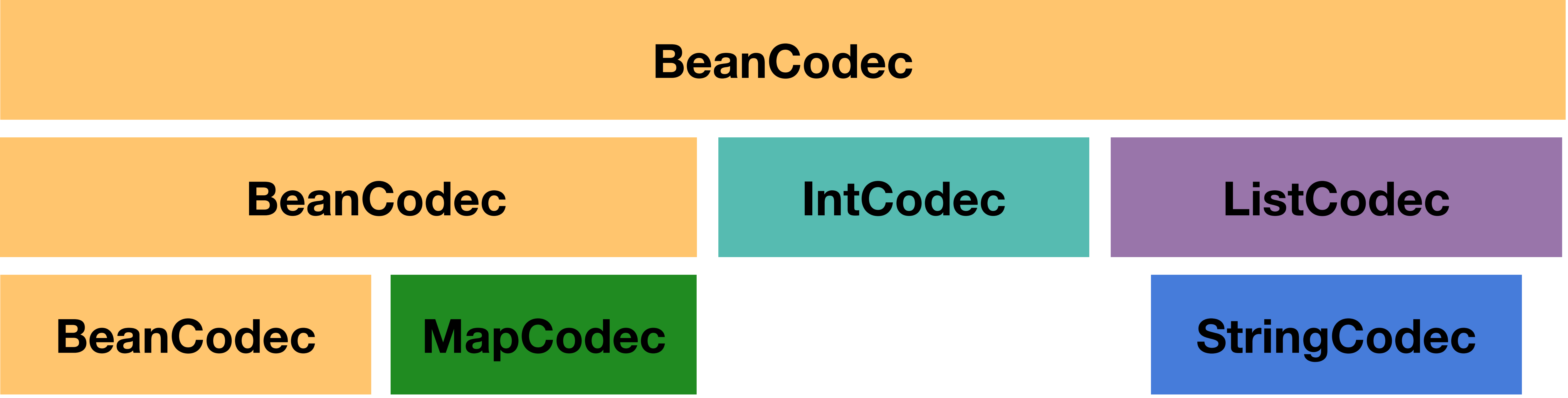
```
data class Foo(
    private val field1 : Bar,
    private val field2 : Int,
    private val field3 : List<String>,
)
```



```
data class Bar(
    private val field1 : Baz,
    private val field2 : Map<Int, Int>,
)
```



```
data class Baz(
    .
    .
    .
)
```

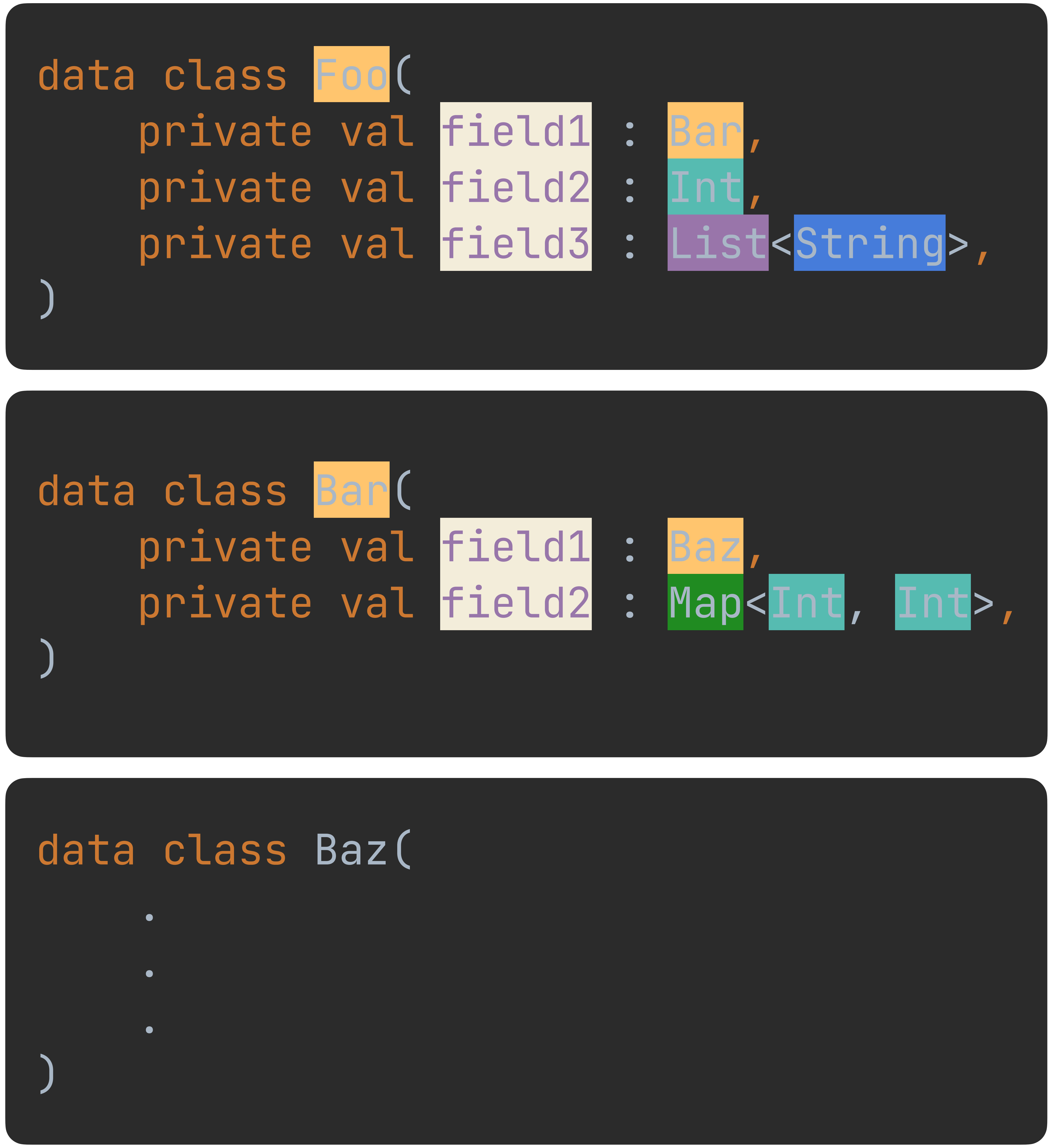


BeanCodec

BeanCodec IntCodec ListCodec

BeanCodec MapCodec StringCodec

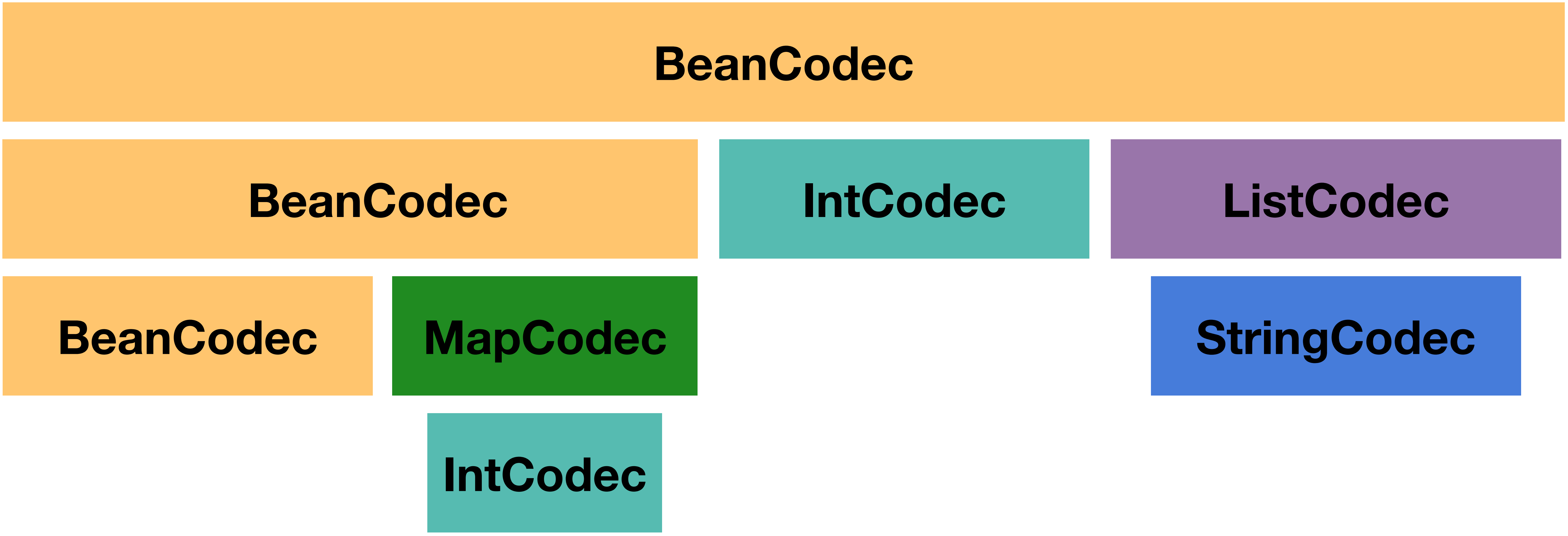
Сериализация произвольных типов(aka Bean)



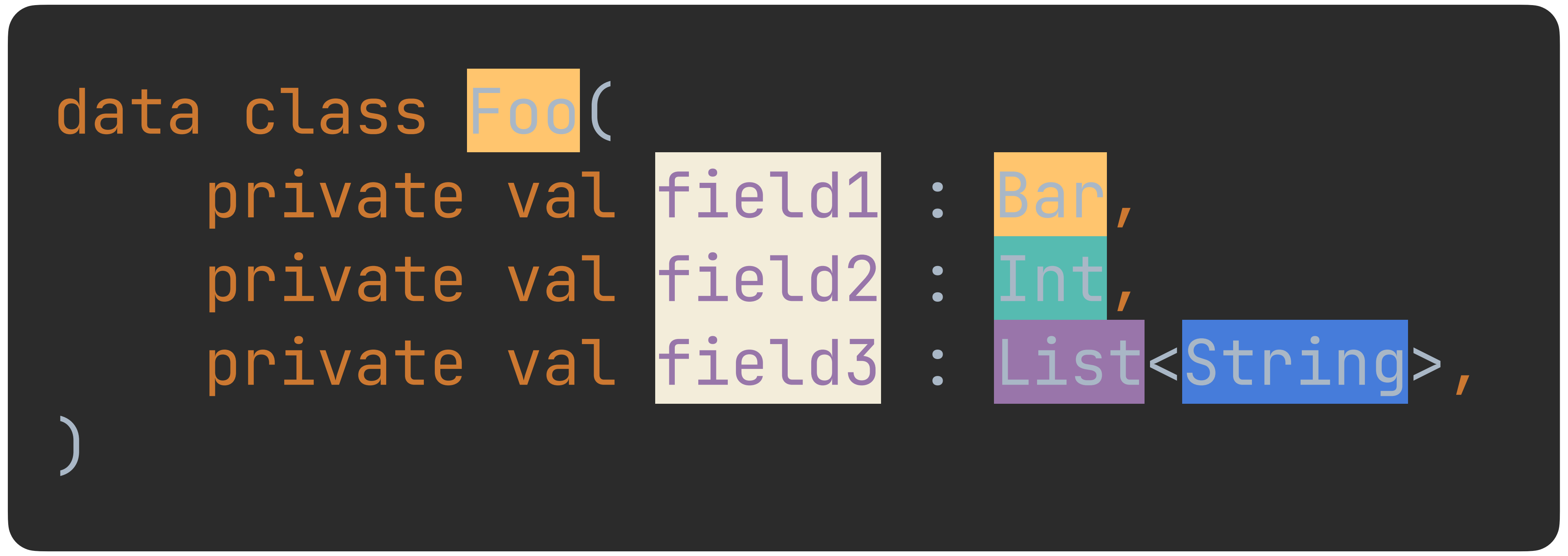
```
data class Foo(
    private val field1 : Bar,
    private val field2 : Int,
    private val field3 : List<String>,
)
```

```
data class Bar(
    private val field1 : Baz,
    private val field2 : Map<Int, Int>,
)
```

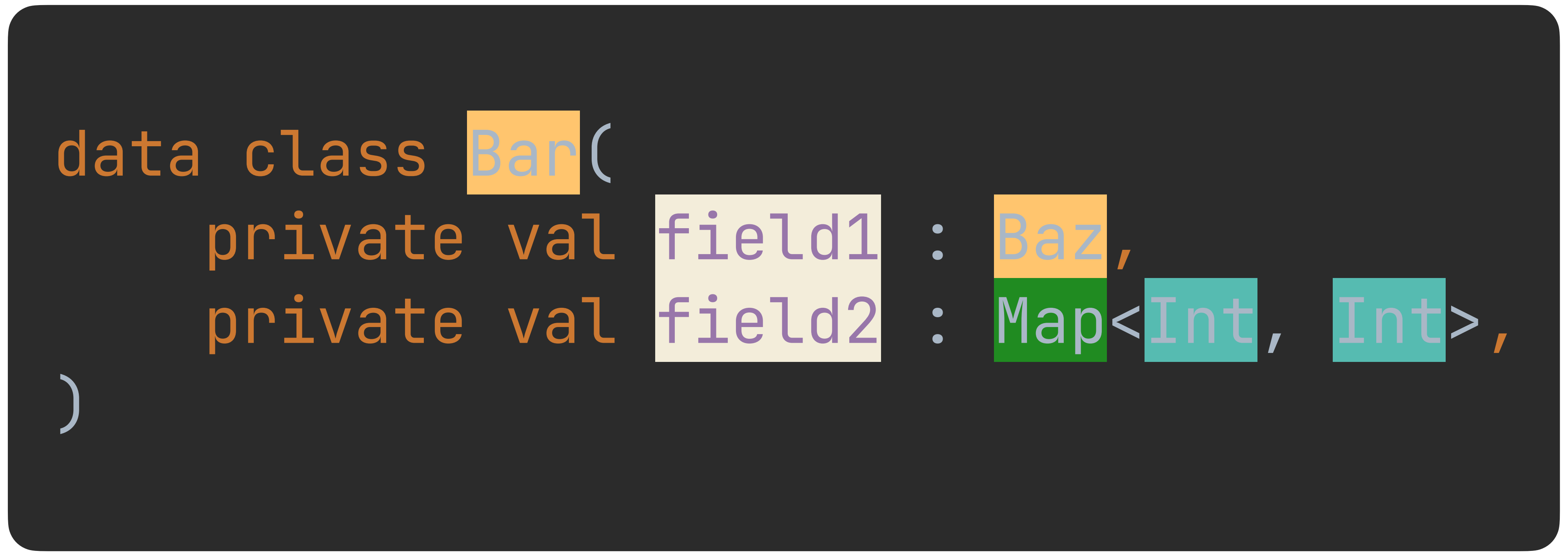
```
data class Baz(
    .
    .
    .
)
```



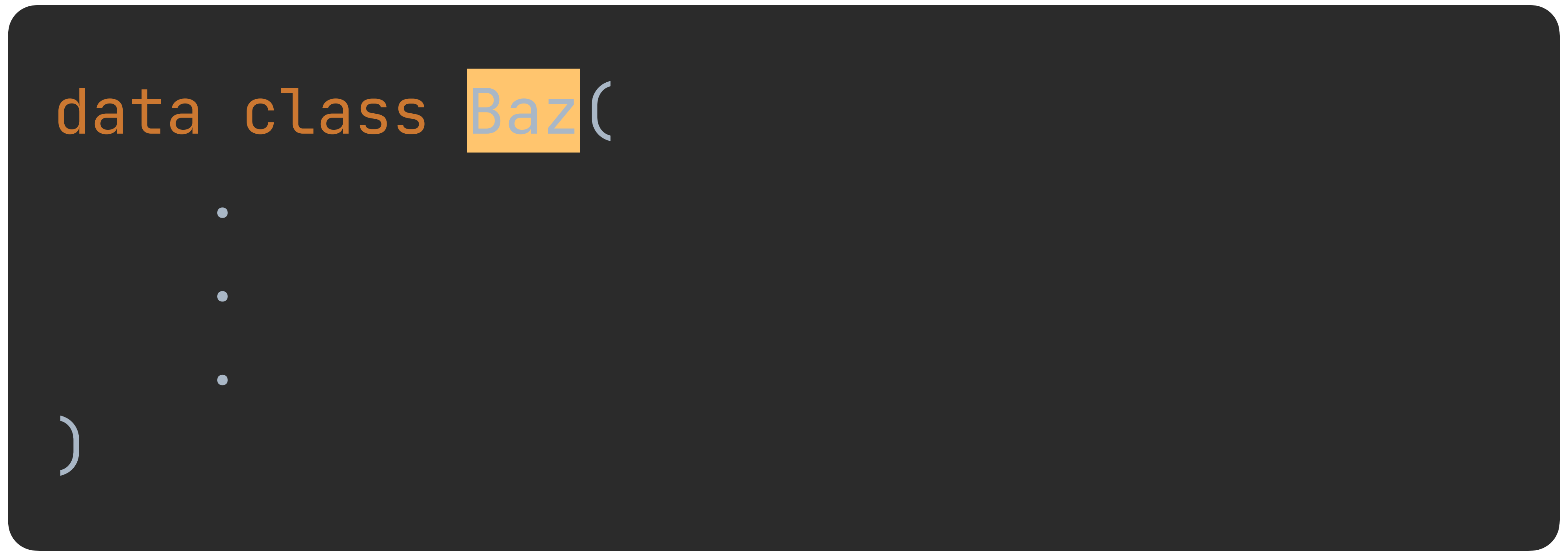
Сериализация произвольных типов(aka Bean)



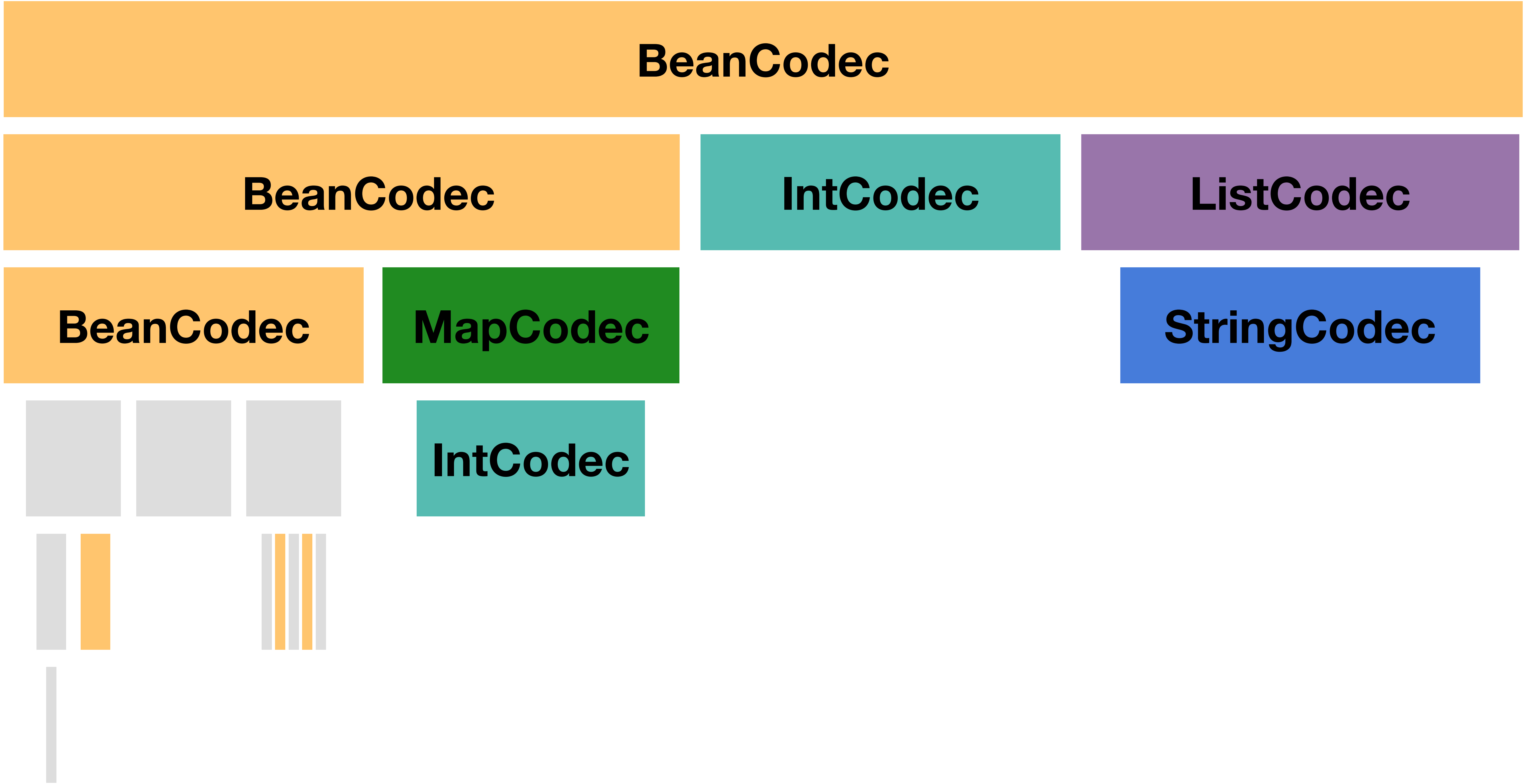
```
data class Foo(
    private val field1 : Bar,
    private val field2 : Int,
    private val field3 : List<String>,
)
```



```
data class Bar(
    private val field1 : Baz,
    private val field2 : Map<Int, Int>,
)
```



```
data class Baz(
    .
    .
    .
)
```

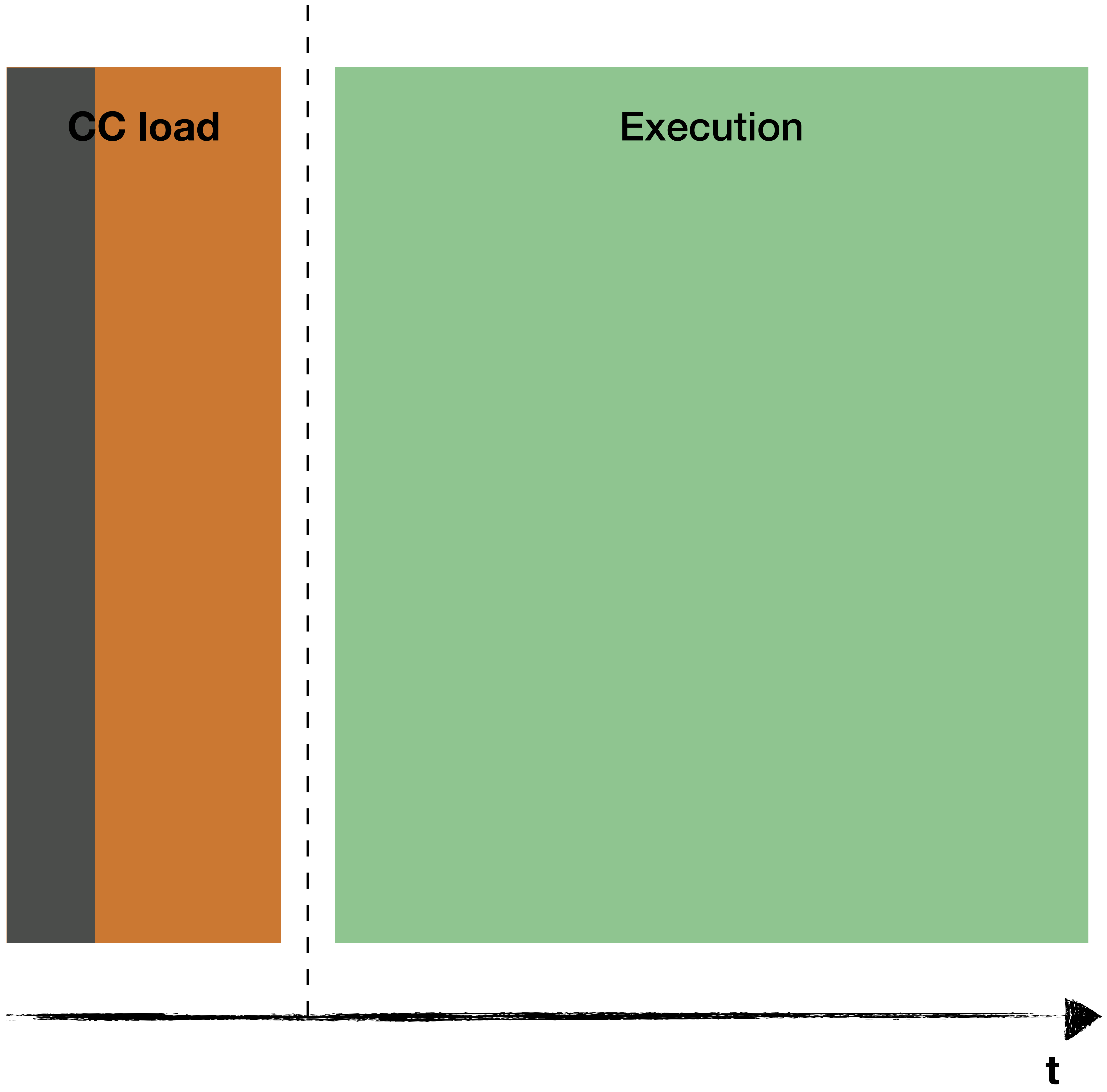


StackOverflowError ❌

BeanCodec reentrancy ✅

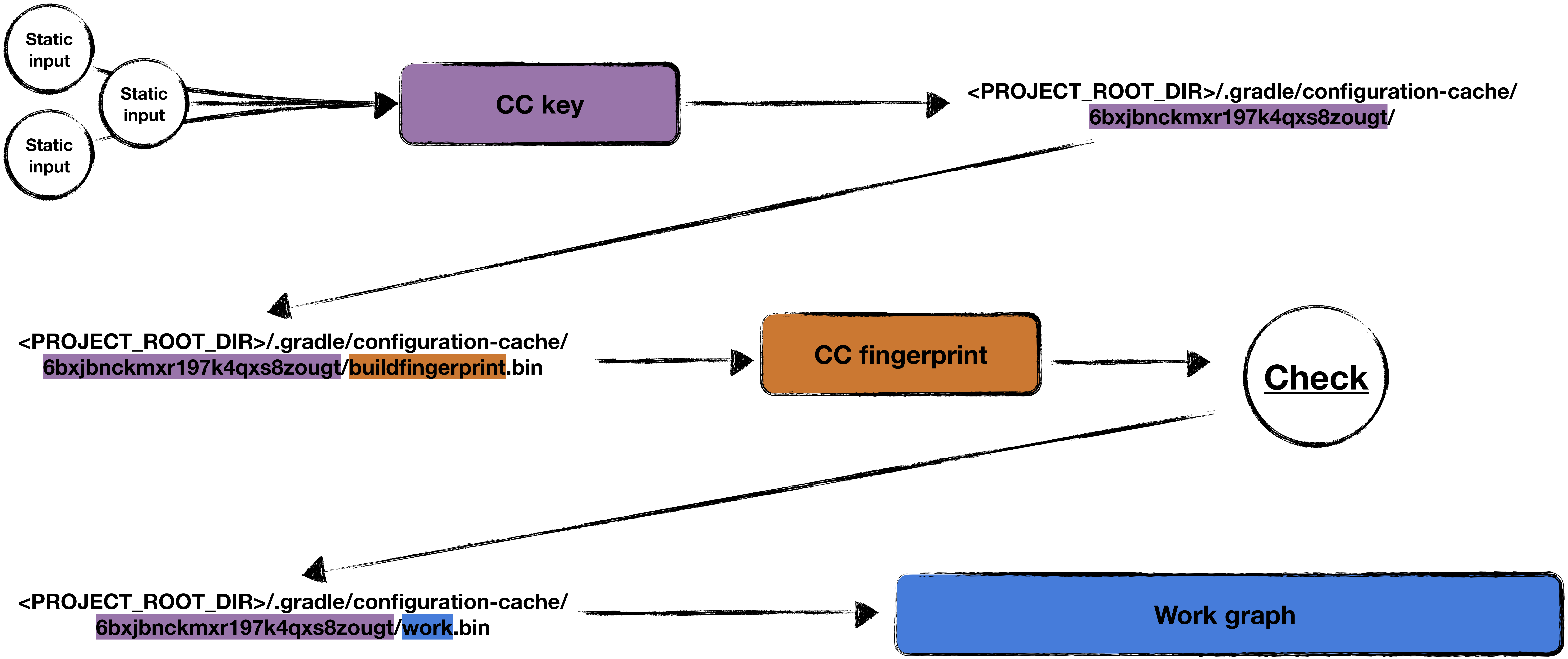
Coroutines beyond concurrency

CC hit



⬛ Fingerprint check

CC load

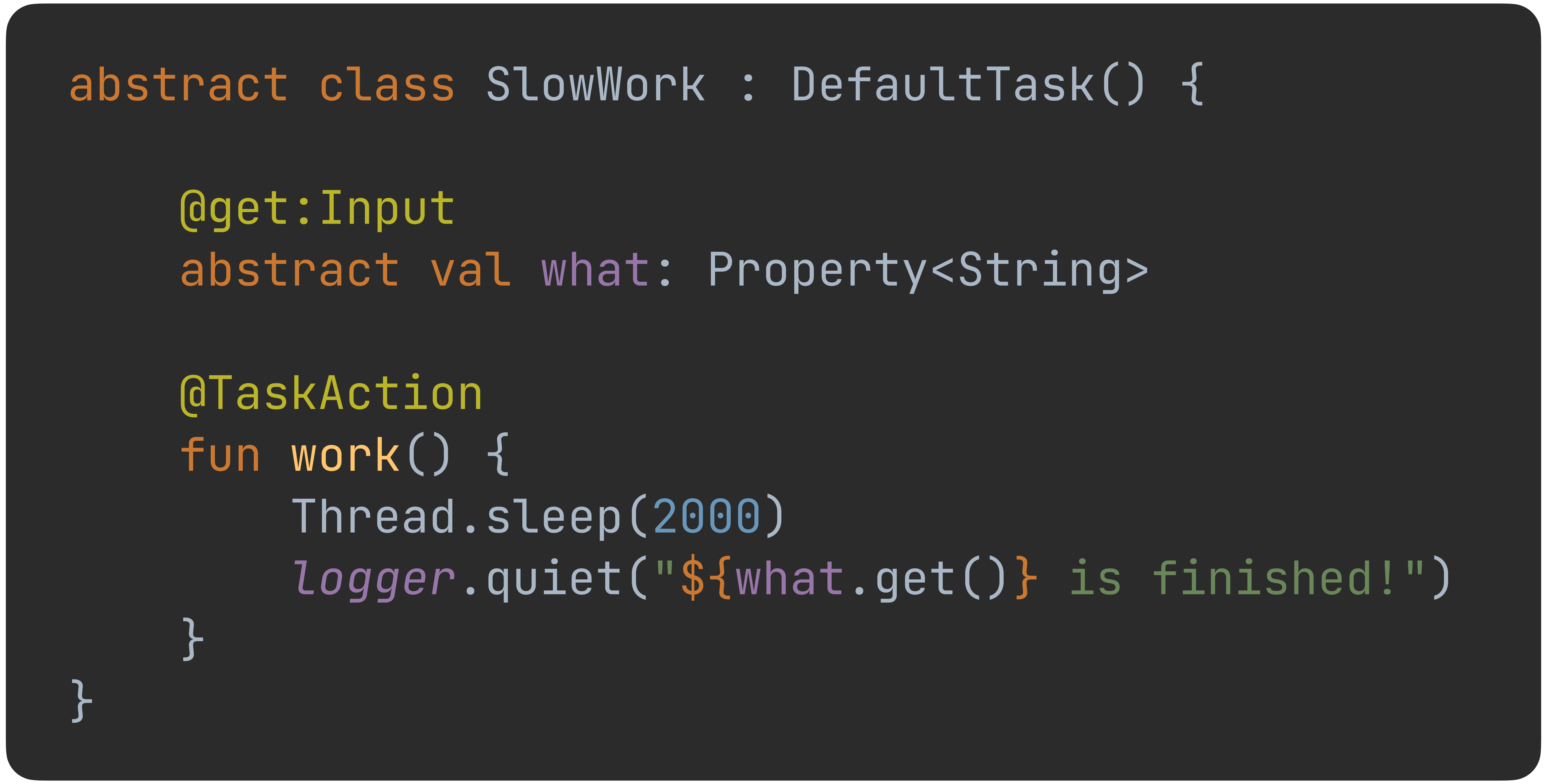


<PROJECT_ROOT_DIR>/.gradle/configuration-cache/
6bxjbnckmxr197k4qxs8zougt/

<PROJECT_ROOT_DIR>/.gradle/configuration-cache/
6bxjbnckmxr197k4qxs8zougt/buildfingerprint.bin

<PROJECT_ROOT_DIR>/.gradle/configuration-cache/
6bxjbnckmxr197k4qxs8zougt/work.bin

Параллелизм внутри проектов



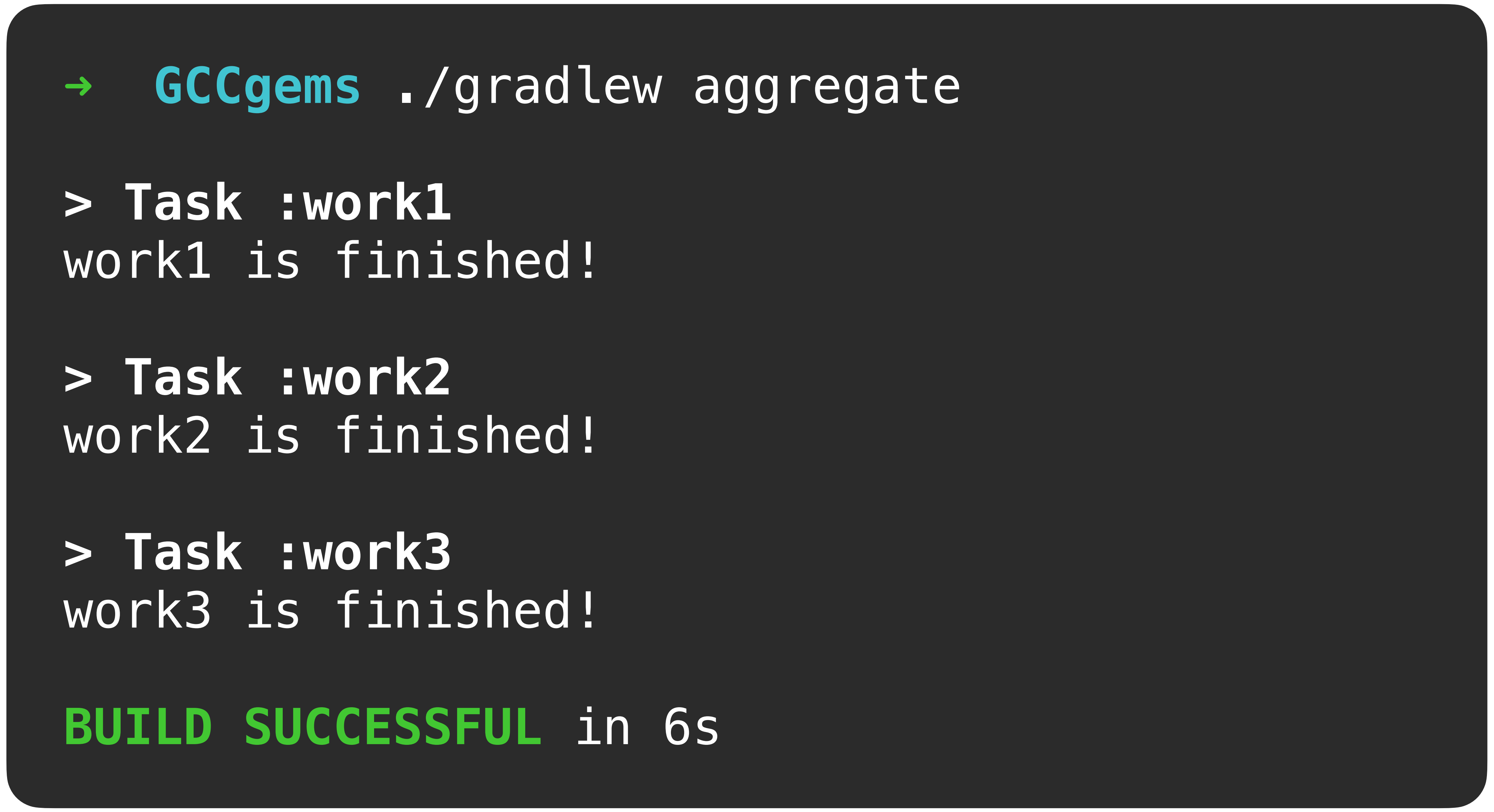
```kotlin
abstract class SlowWork : DefaultTask() {

    @get:Input
    abstract val what: Property<String>

    @TaskAction
    fun work() {
        Thread.sleep(2000)
        logger.quiet("${what.get()} is finished!")
    }
}
```

Параллелизм внутри проектов

```kotlin
// build.gradle.kts
val work1 = tasks.register("work1", SlowWork::class) {
    what = "work1"
}
val work2 = tasks.register("work2", SlowWork::class) {
    what = "work2"
}
val work3 = tasks.register("work3", SlowWork::class) {
    what = "work3"
}
tasks.register("aggregate", DefaultTask::class) {
    dependsOn(work1, work2, work3)
}
```

💎 Параллелизм внутри проектов

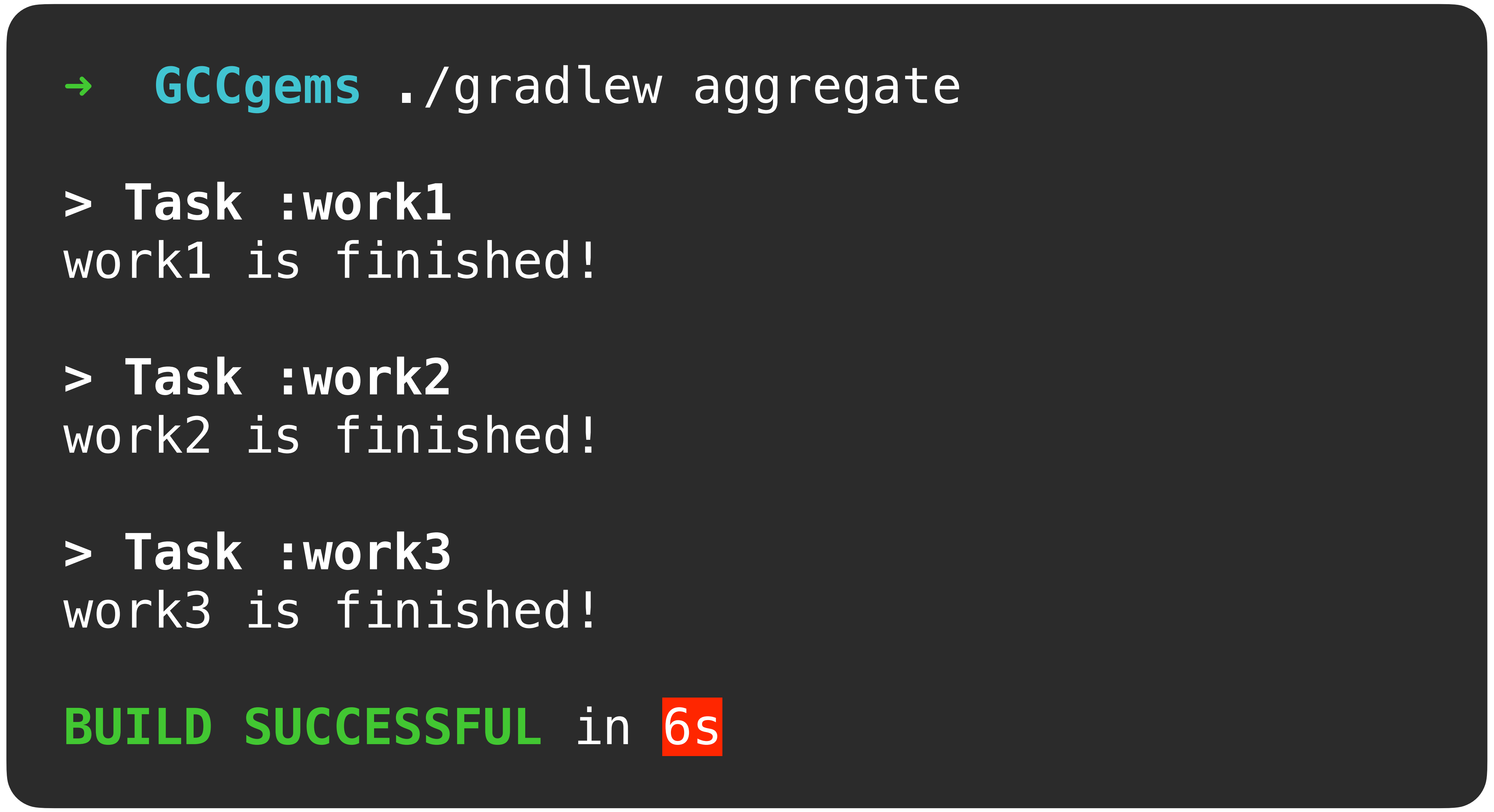
Vintage



```
➜  GCCgems ./gradlew aggregate

> Task :work1
work1 is finished!

> Task :work2
work2 is finished!

> Task :work3
work3 is finished!

BUILD SUCCESSFUL in 6s
```

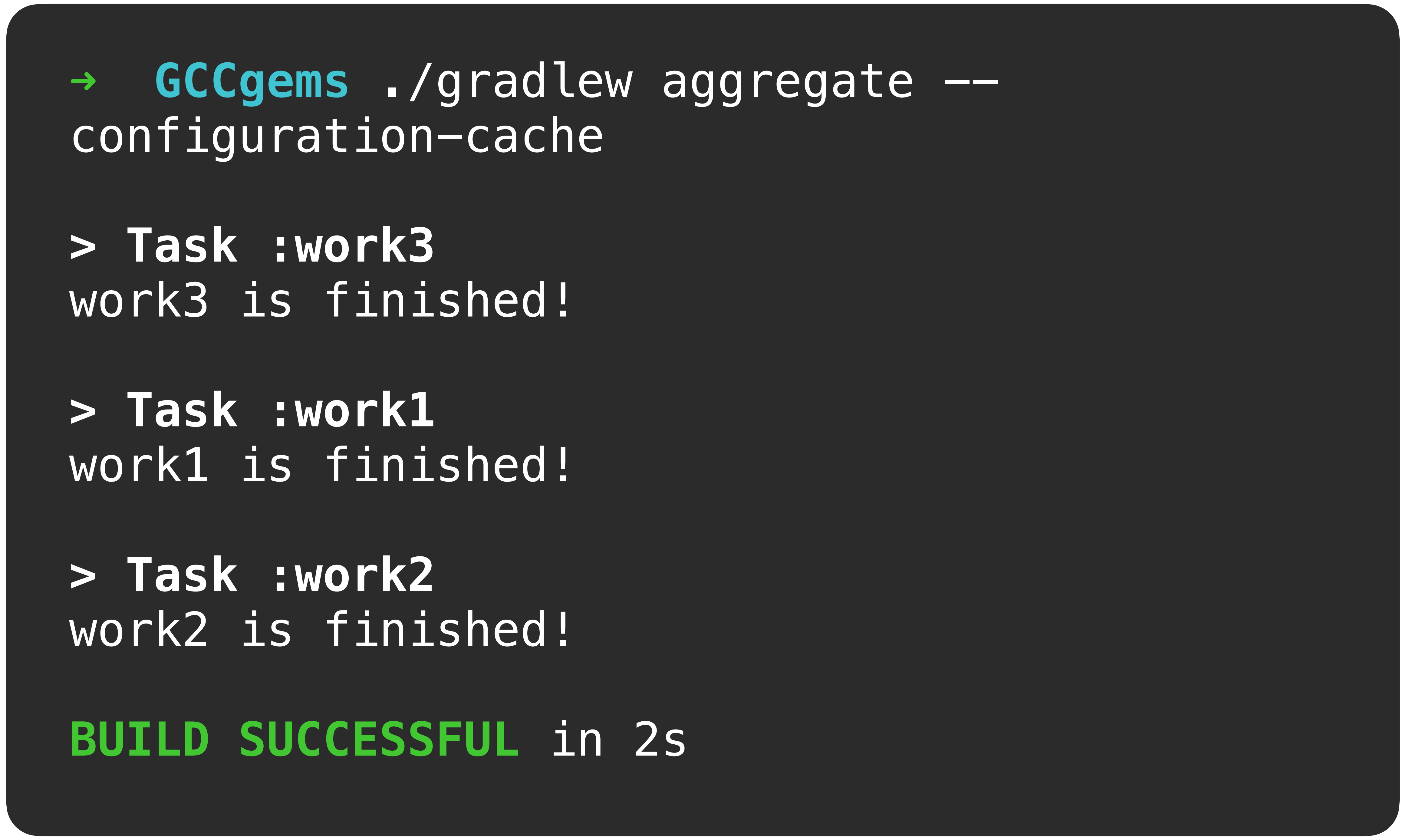
💎 Параллелизм внутри проектов

Vintage



```
➜  GCCgems ./gradlew aggregate

> Task :work1
work1 is finished!

> Task :work2
work2 is finished!

> Task :work3
work3 is finished!

BUILD SUCCESSFUL in 6s
```

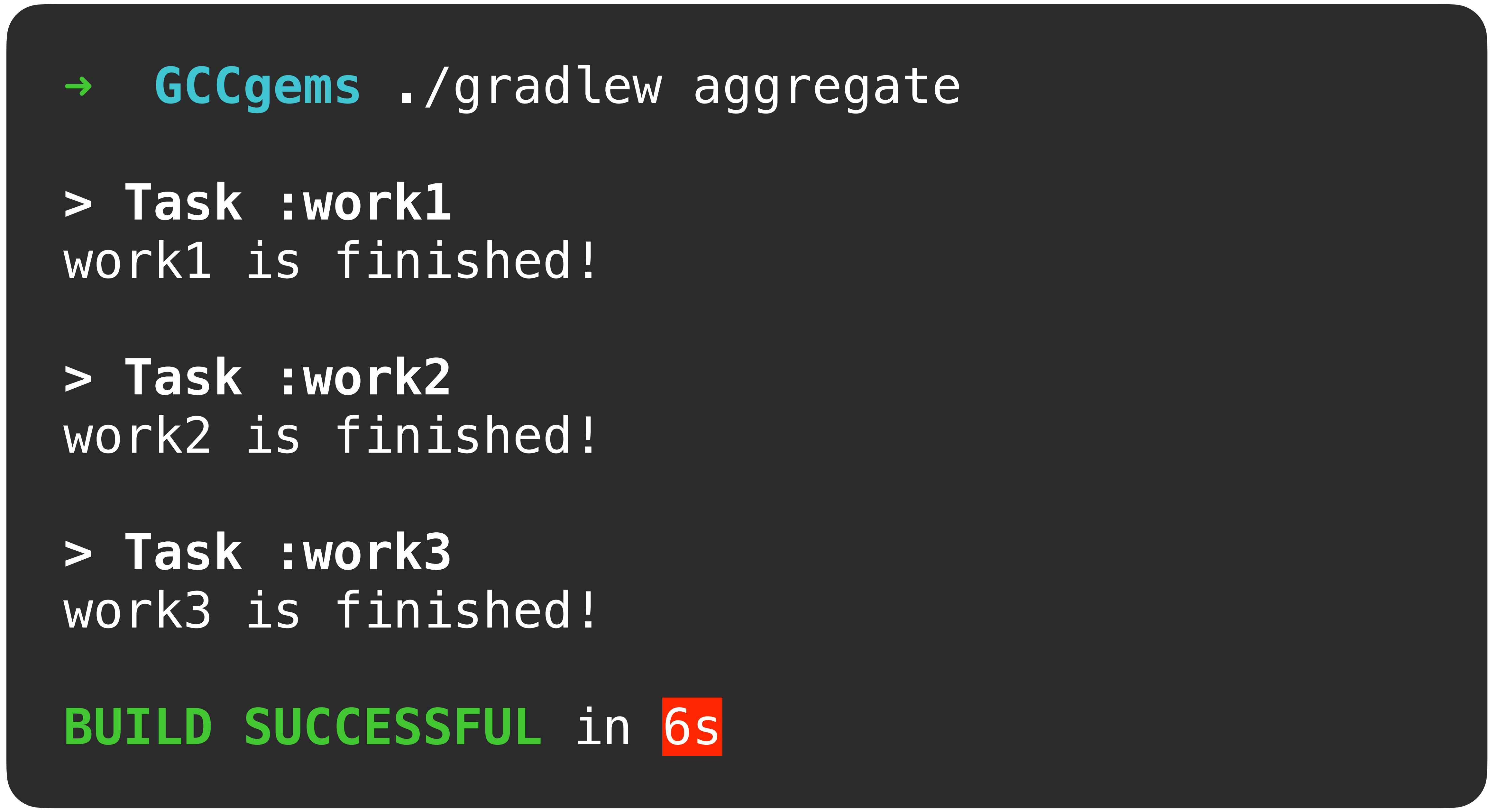
CC



```
➜  GCCgems ./gradlew aggregate --
configuration-cache

> Task :work3
work3 is finished!

> Task :work1
work1 is finished!

> Task :work2
work2 is finished!

BUILD SUCCESSFUL in 2s
```

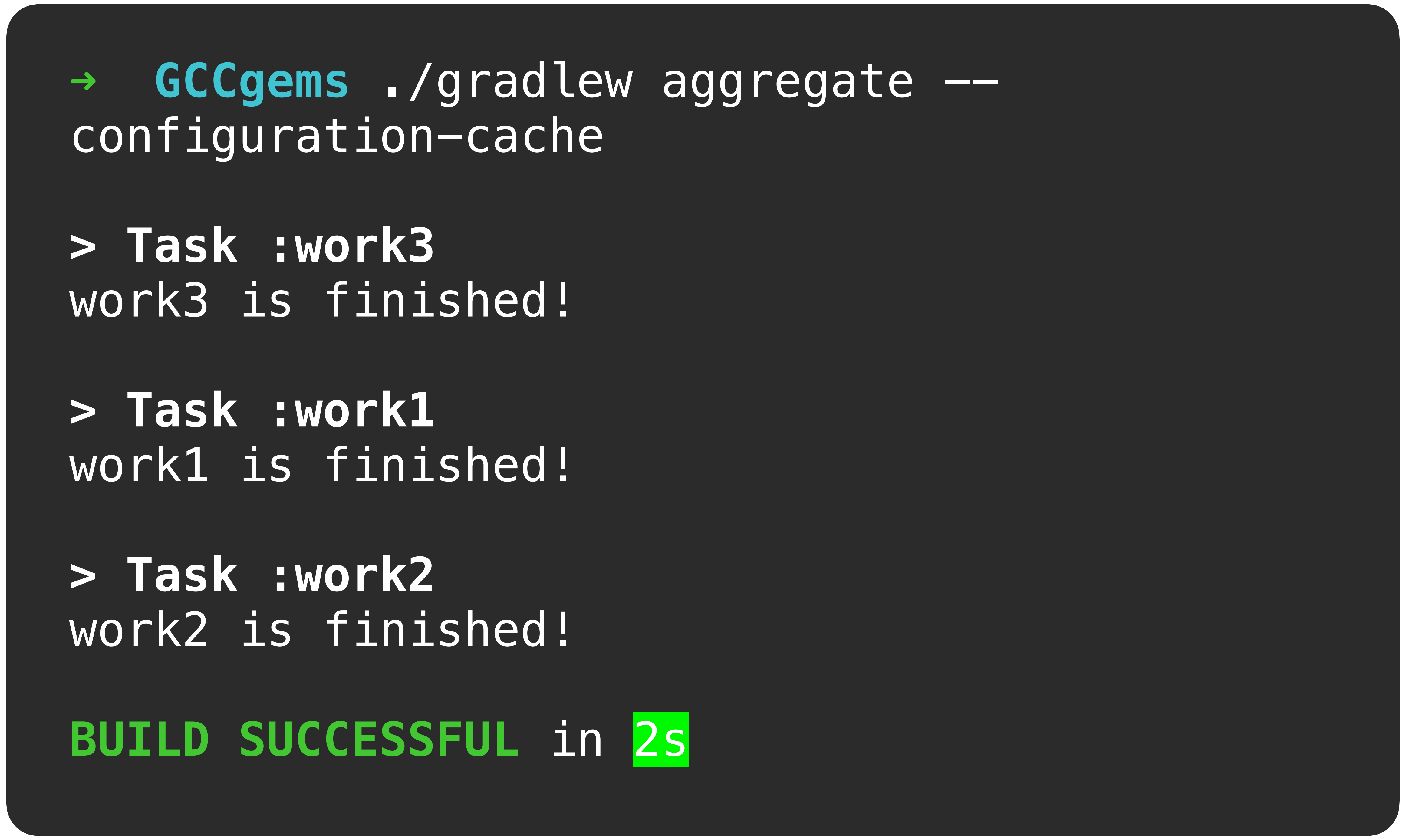
Параллелизм внутри проектов

Vintage



```
➜  GCCgems ./gradlew aggregate

> Task :work1
work1 is finished!

> Task :work2
work2 is finished!

> Task :work3
work3 is finished!

BUILD SUCCESSFUL in 6s
```

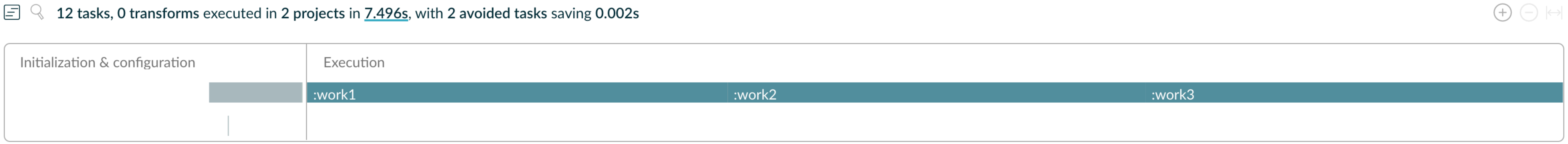
CC



```
➜  GCCgems ./gradlew aggregate --
configuration-cache

> Task :work3
work3 is finished!

> Task :work1
work1 is finished!

> Task :work2
work2 is finished!

BUILD SUCCESSFUL in 2s
```

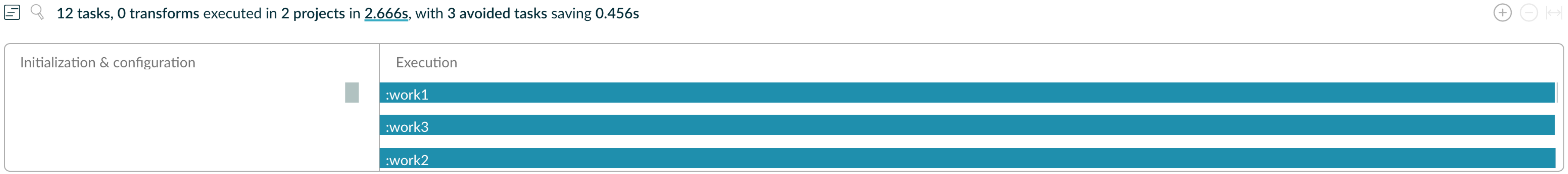
💎 Параллелизм внутри проектов

Vintage



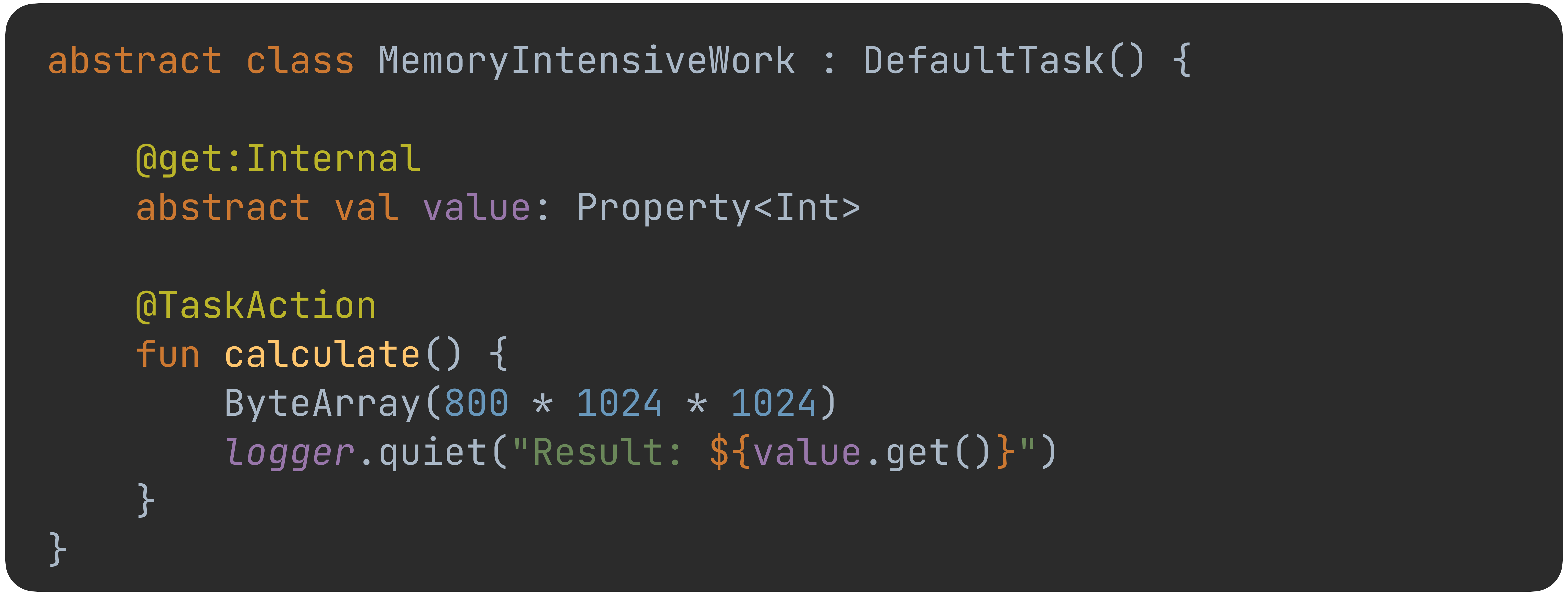
12 tasks, 0 transforms executed in 2 projects in 7.496s, with 2 avoided tasks saving 0.002s

CC



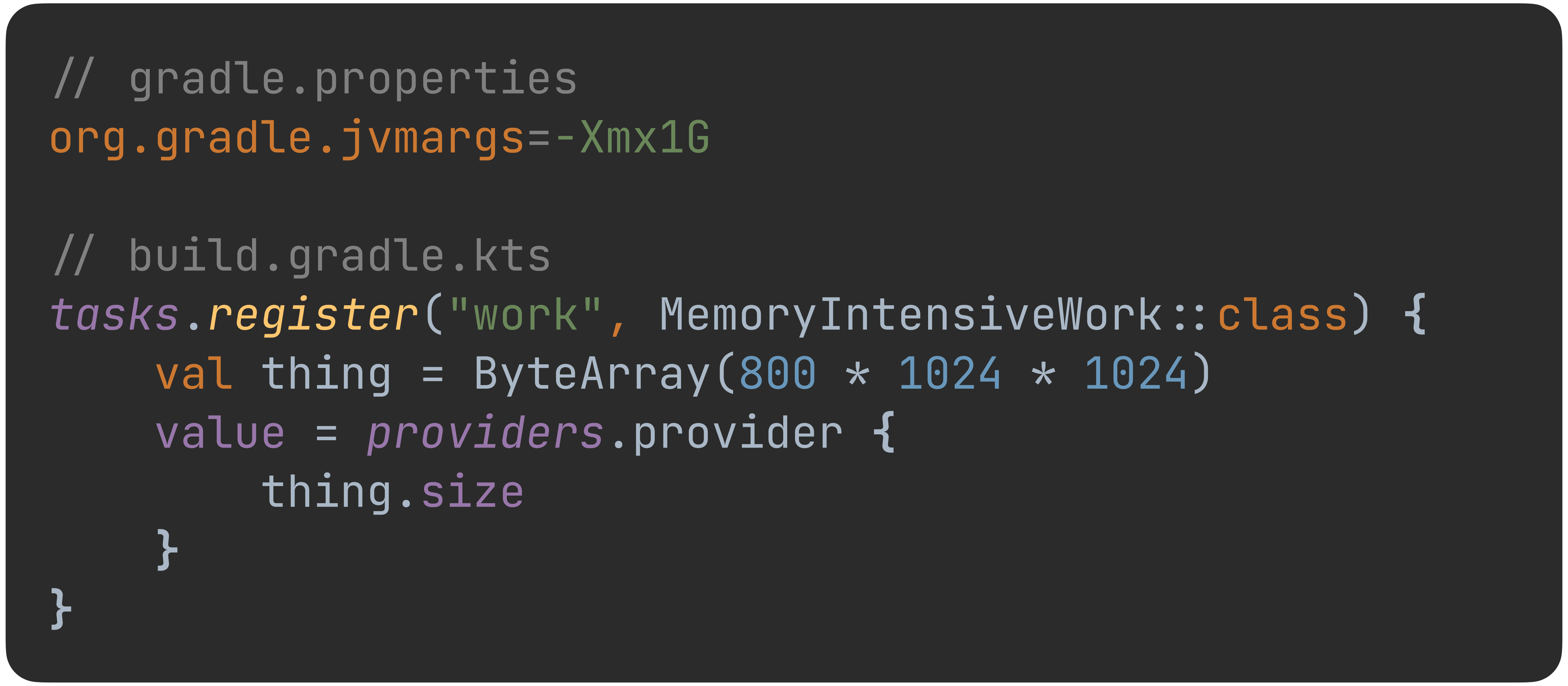
12 tasks, 0 transforms executed in 2 projects in 2.666s, with 3 avoided tasks saving 0.456s

💎 Ранее освобождение памяти



```
abstract class MemoryIntensiveWork : DefaultTask() {

    @get:Internal
    abstract val value: Property<Int>

    @TaskAction
    fun calculate() {
        ByteArray(800 * 1024 * 1024)
        logger.quiet("Result: ${value.get()}")
    }
}
```

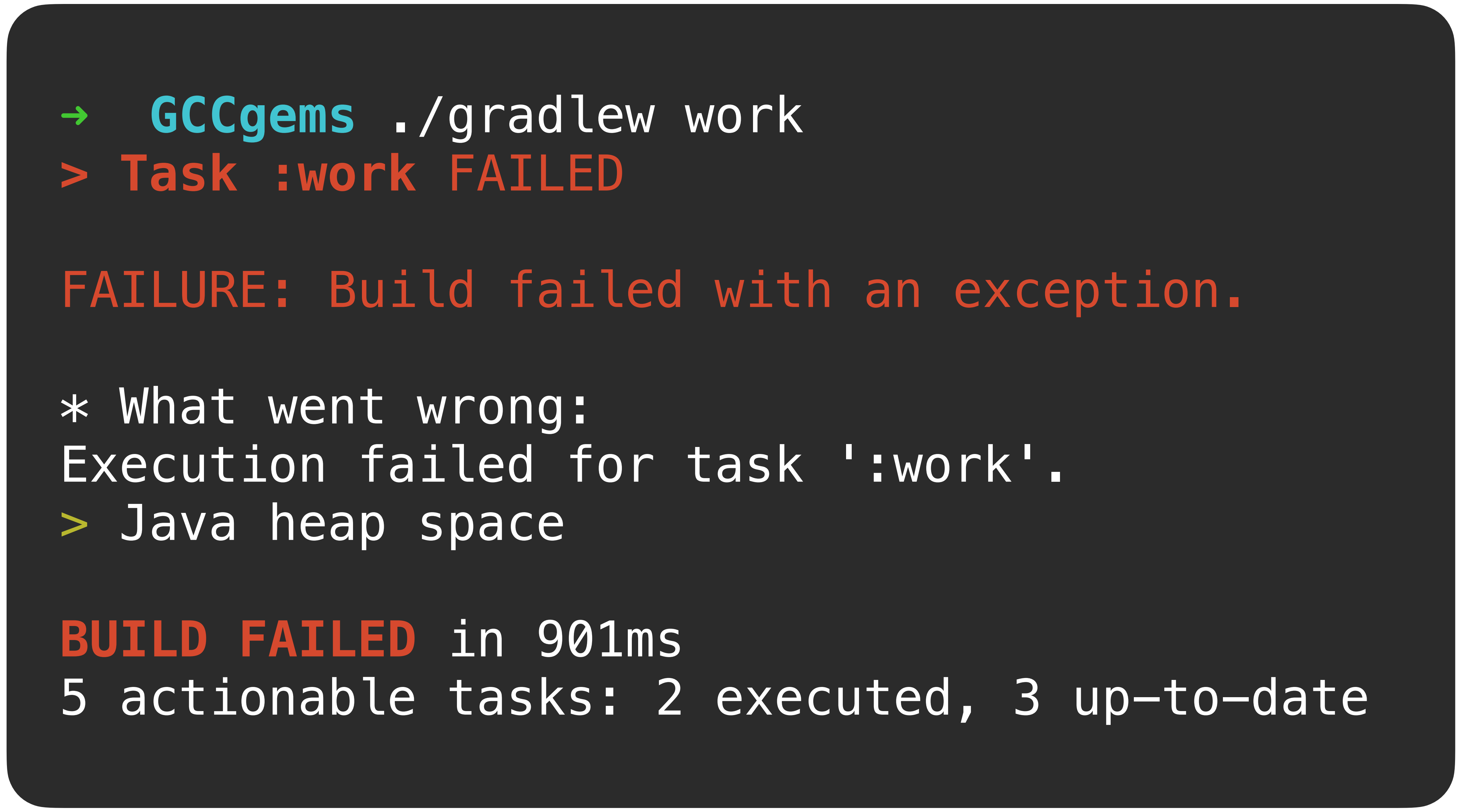
💎 Ранее освобождение памяти



```
// gradle.properties
org.gradle.jvmargs=-Xmx1G

// build.gradle.kts
tasks.register("work", MemoryIntensiveWork::class) {
    val thing = ByteArray(800 * 1024 * 1024)
    value = providers.provider {
        thing.size
    }
}
```

💎 Ранее освобождение памяти

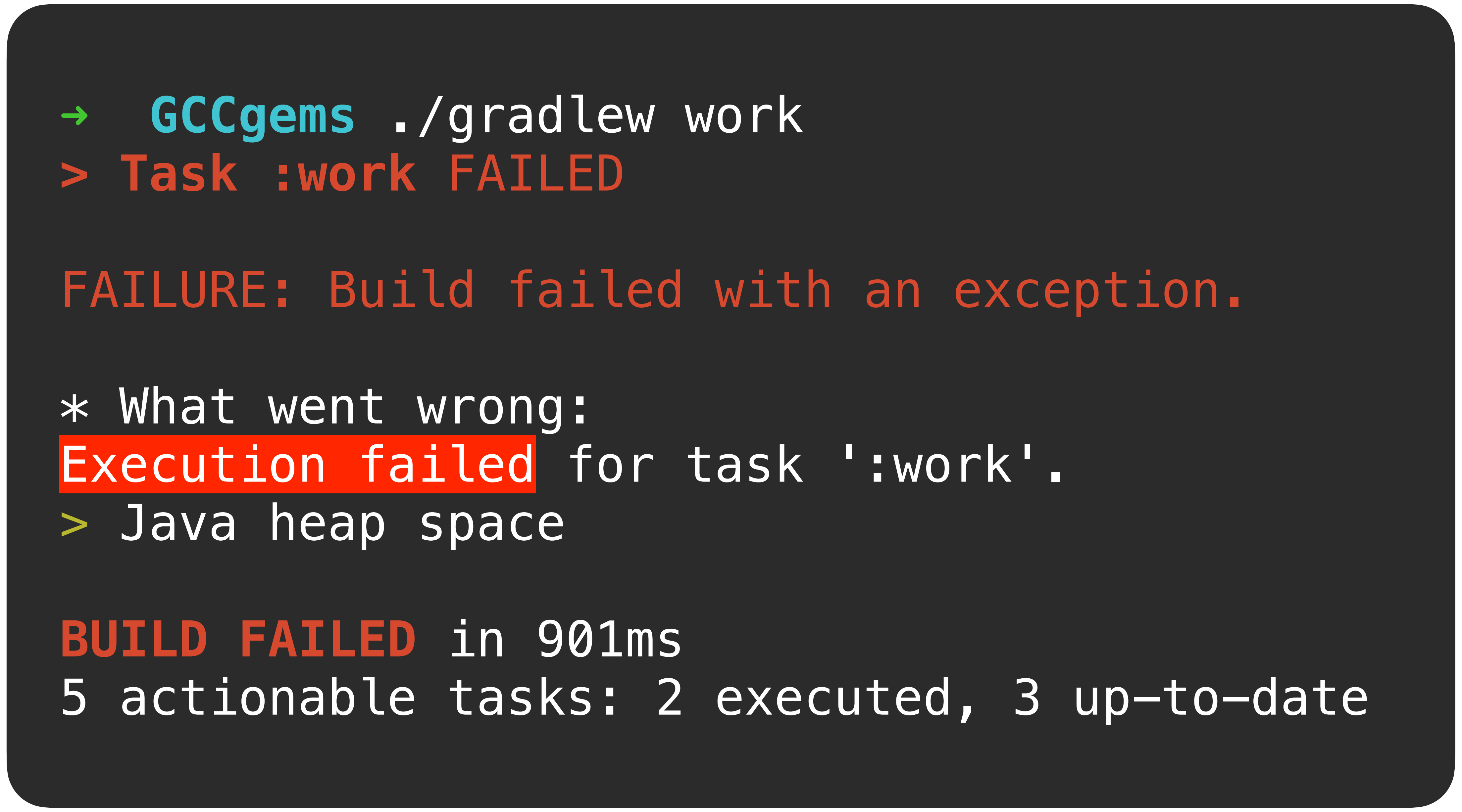
Vintage



```
➜  GCCgems ./gradlew work
> Task :work FAILED

FAILURE: Build failed with an exception.

* What went wrong:
Execution failed for task ':work'.
> Java heap space

BUILD FAILED in 901ms
5 actionable tasks: 2 executed, 3 up-to-date
```

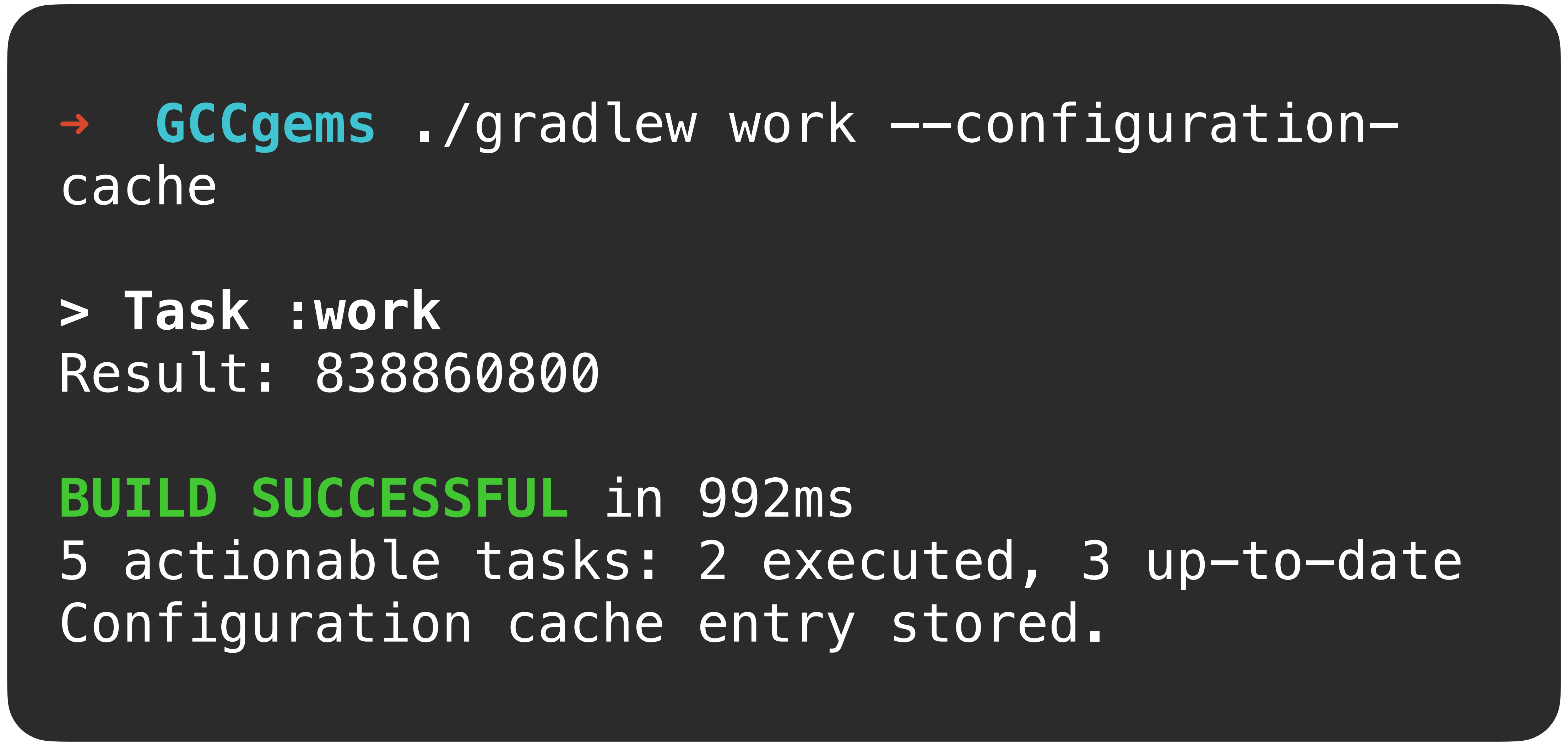
Ранее освобождение памяти

Vintage



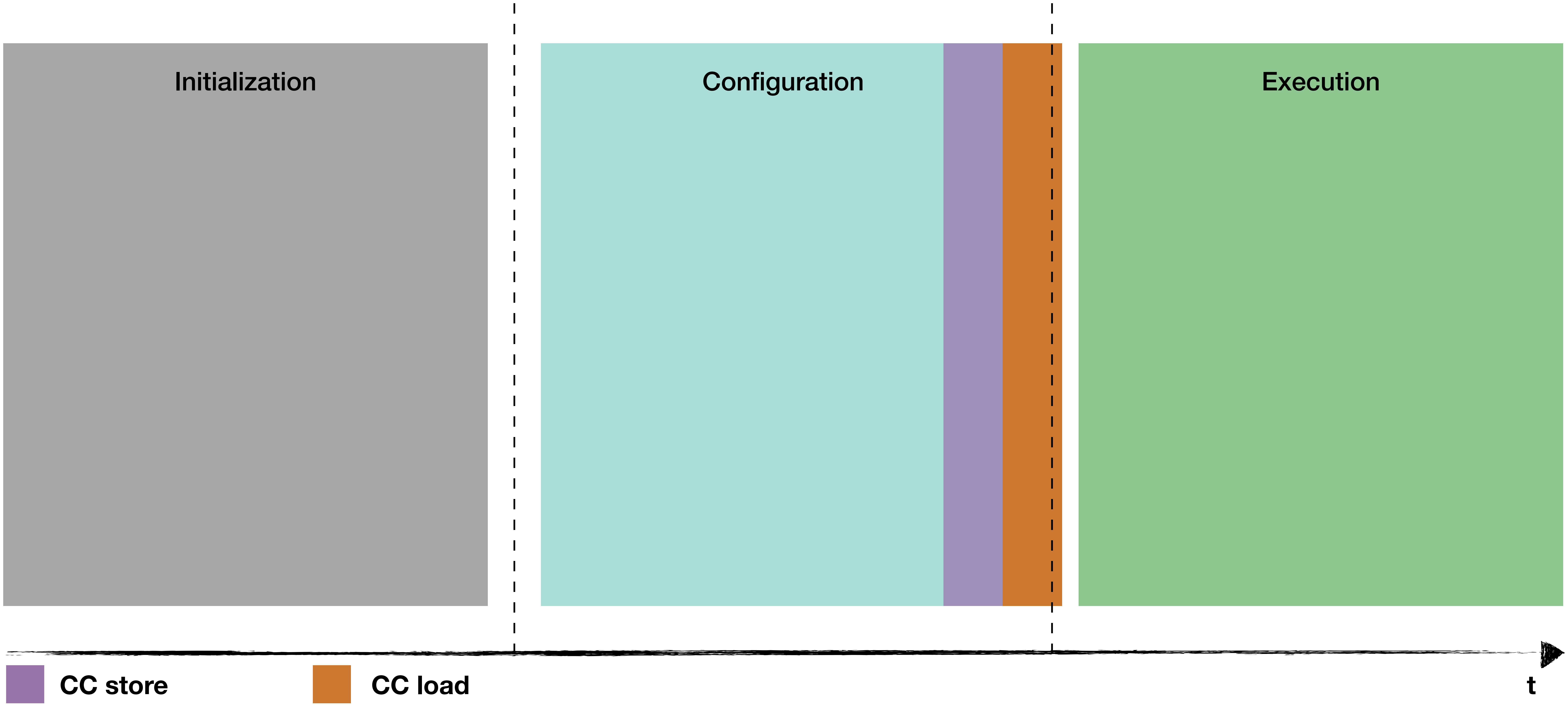
```
➜  GCCgems ./gradlew work
> Task :work FAILED

FAILURE: Build failed with an exception.

* What went wrong:
Execution failed for task ':work'.
> Java heap space

BUILD FAILED in 901ms
5 actionable tasks: 2 executed, 3 up-to-date
```

CC

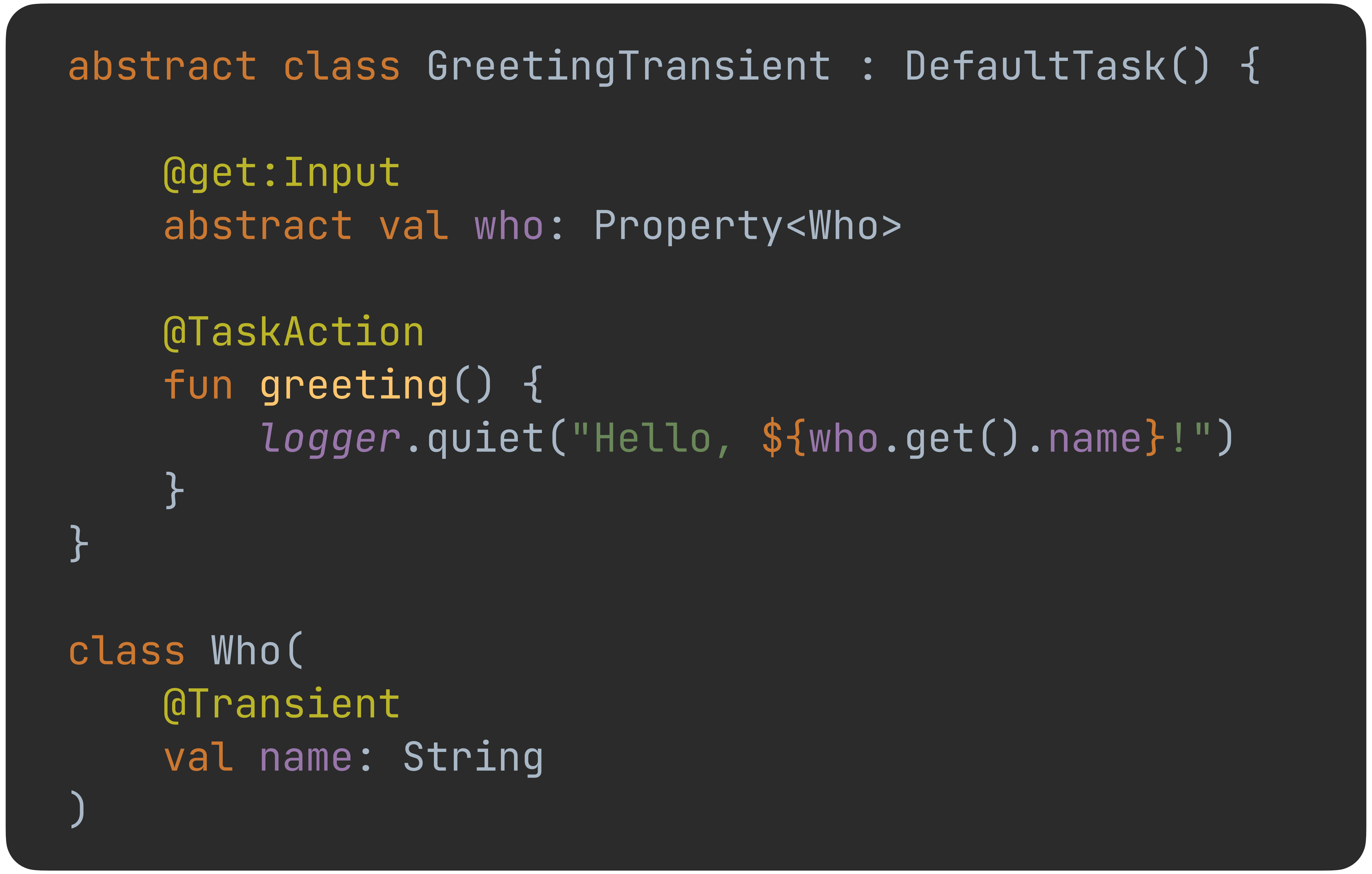


```
➜  GCCgems ./gradlew work --configuration-
cache

> Task :work
Result: 838860800

BUILD SUCCESSFUL in 992ms
5 actionable tasks: 2 executed, 3 up-to-date
Configuration cache entry stored.
```

“Load after store”



💎 Ранее обнаружение ошибок десеарилизации



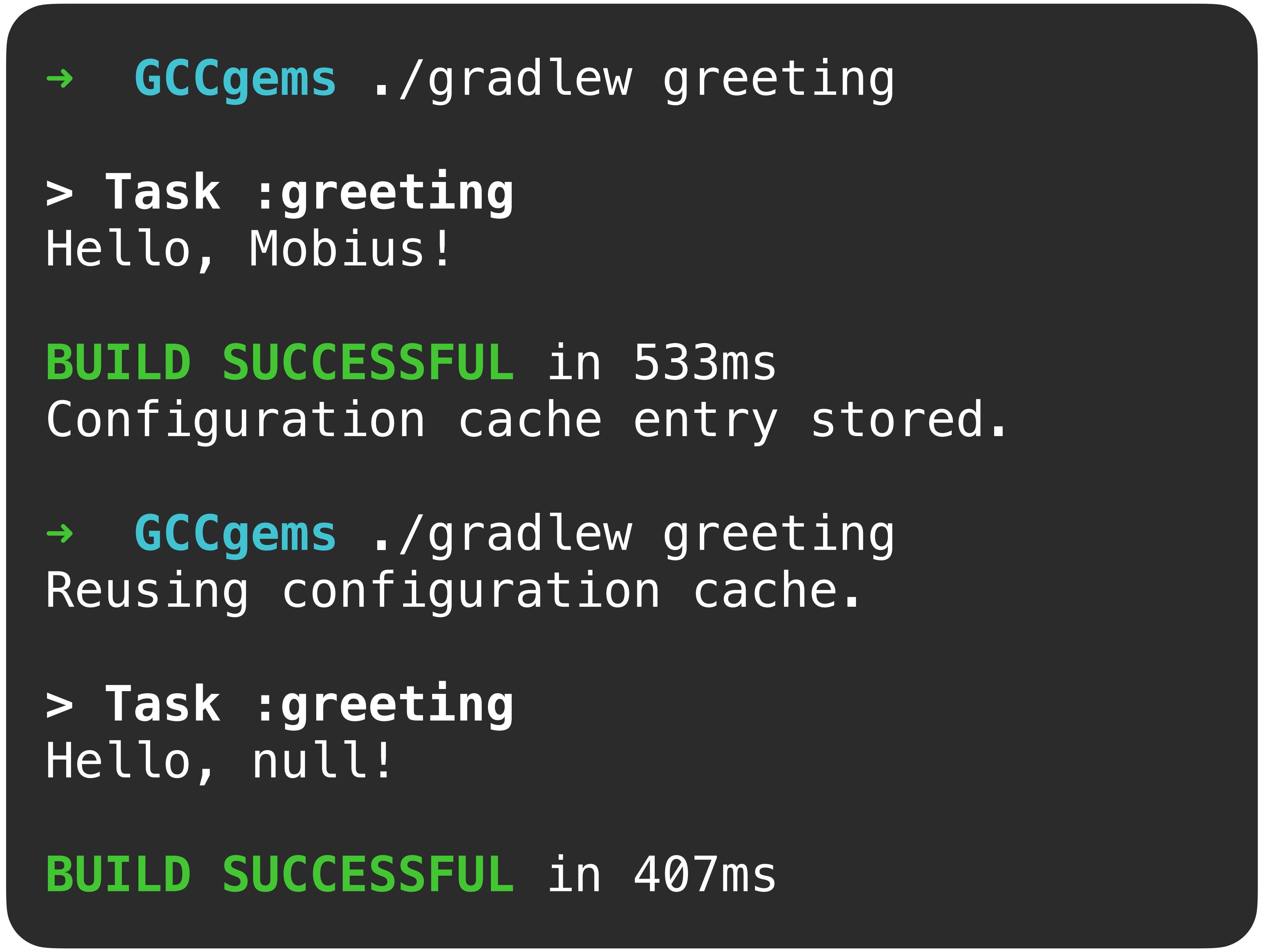
```kotlin
abstract class GreetingTransient : DefaultTask() {

    @get:Input
    abstract val who: Property<Who>

    @TaskAction
    fun greeting() {
        logger.quiet("Hello, ${who.get().name}!")
    }
}

class Who(
    @Transient
    val name: String
)
```

💎 Ранее обнаружение ошибок десеарилизации

```kotlin
// build.gradle.kts
tasks.register("greeting", GreetingTransient::class) {
    who = Who("Mobius")
}
```

💎 Ранее обнаружение ошибок десеарилизации

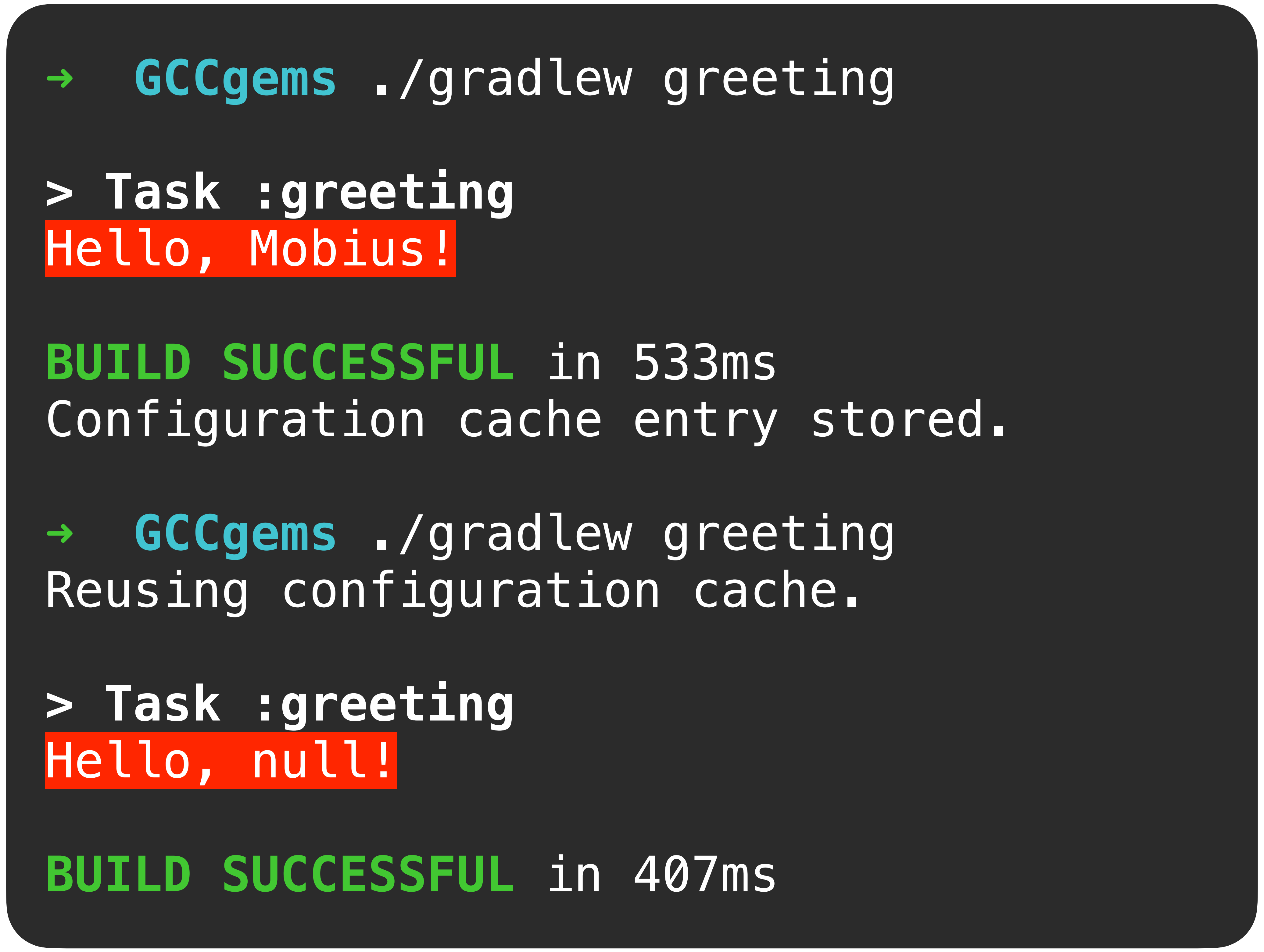
No LAS



```
➜  GCCgems ./gradlew greeting

> Task :greeting
Hello, Mobius!

BUILD SUCCESSFUL in 533ms
Configuration cache entry stored.

➜  GCCgems ./gradlew greeting
Reusing configuration cache.

> Task :greeting
Hello, null!

BUILD SUCCESSFUL in 407ms
```

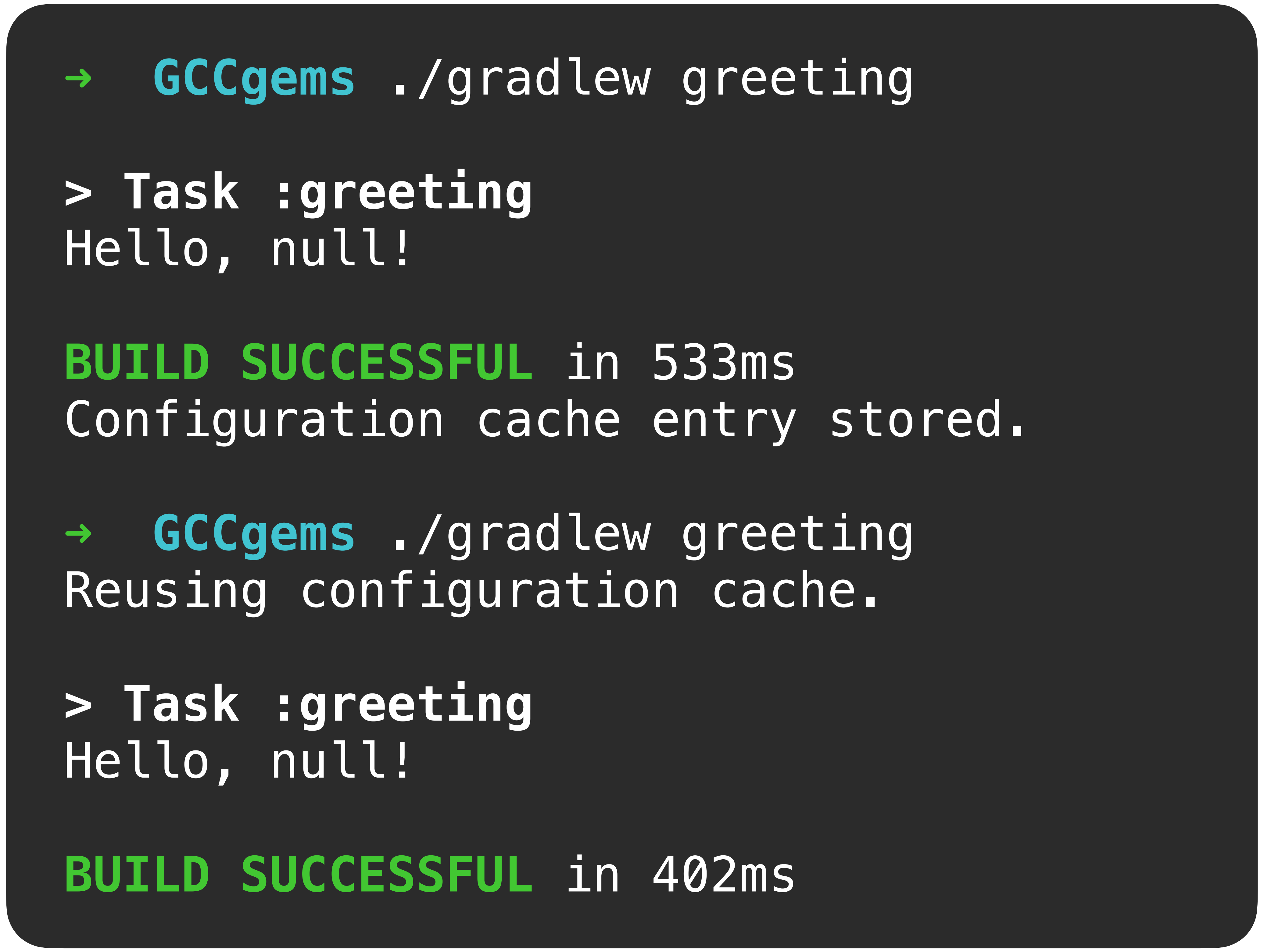
💎 Ранее обнаружение ошибок десеарилизации

No LAS



```
➜  GCCgems ./gradlew greeting

> Task :greeting
Hello, Mobius!

BUILD SUCCESSFUL in 533ms
Configuration cache entry stored.

➜  GCCgems ./gradlew greeting
Reusing configuration cache.

> Task :greeting
Hello, null!

BUILD SUCCESSFUL in 407ms
```

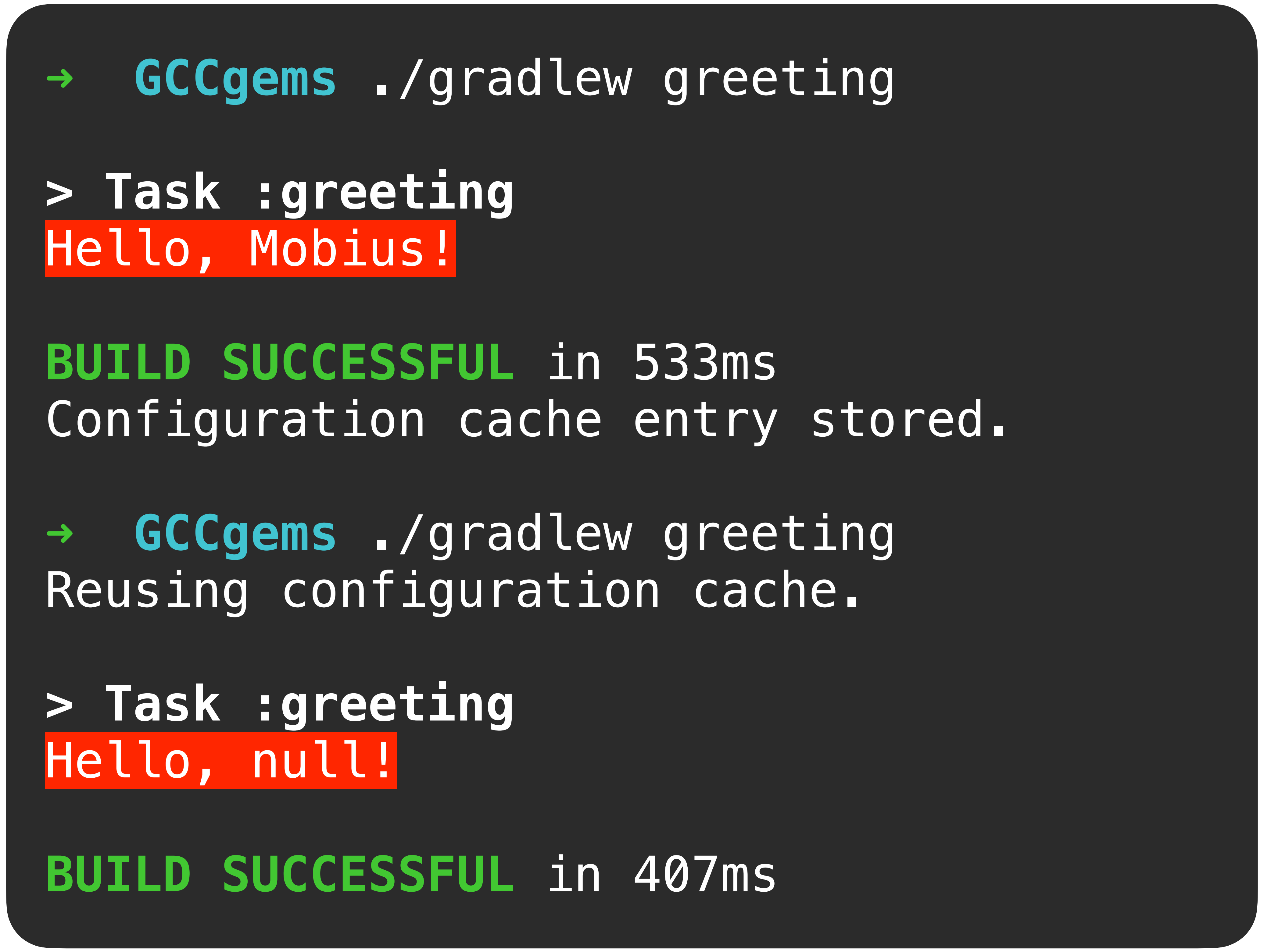
LAS



```
➜  GCCgems ./gradlew greeting

> Task :greeting
Hello, null!

BUILD SUCCESSFUL in 533ms
Configuration cache entry stored.

➜  GCCgems ./gradlew greeting
Reusing configuration cache.

> Task :greeting
Hello, null!

BUILD SUCCESSFUL in 402ms
```

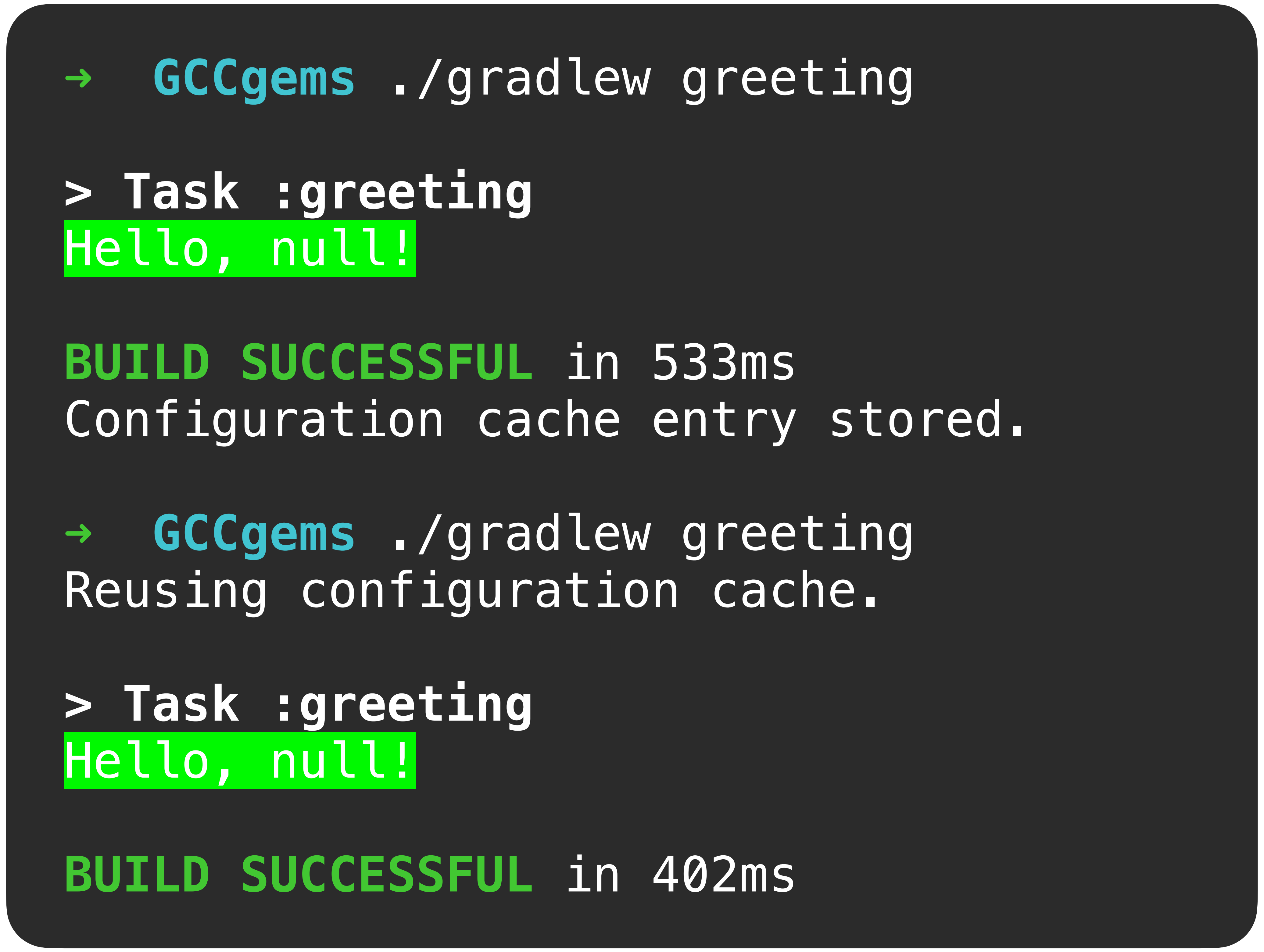
Ранее обнаружение ошибок десеарилизации

No LAS



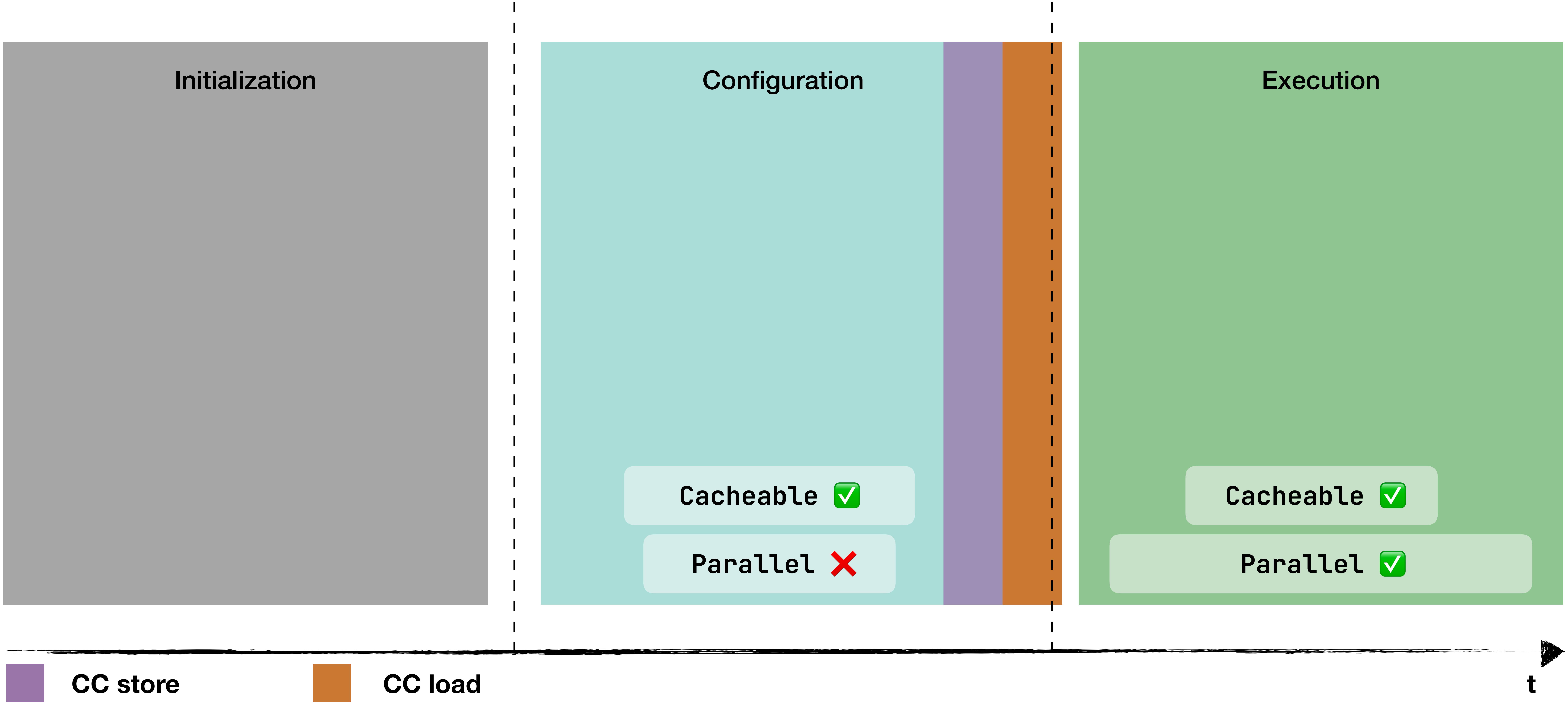
```
➜  GCCgems ./gradlew greeting

> Task :greeting
Hello, Mobius!

BUILD SUCCESSFUL in 533ms
Configuration cache entry stored.

➜  GCCgems ./gradlew greeting
Reusing configuration cache.

> Task :greeting
Hello, null!

BUILD SUCCESSFUL in 407ms
```

LAS



```
➜  GCCgems ./gradlew greeting

> Task :greeting
Hello, null!

BUILD SUCCESSFUL in 533ms
Configuration cache entry stored.

➜  GCCgems ./gradlew greeting
Reusing configuration cache.

> Task :greeting
Hello, null!

BUILD SUCCESSFUL in 402ms
```

Жизненный цикл сборки с CC



Initialization

Configuration

Cacheable ✅

Parallel ❌

Execution

Cacheable ✅

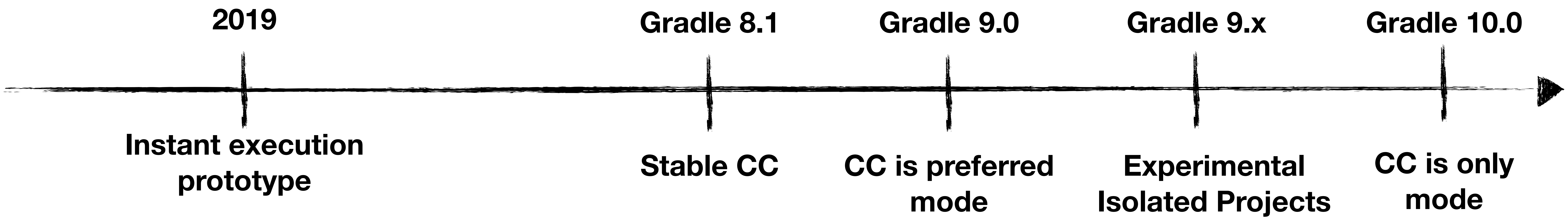
Parallel ✅

🟪 CC store

🟧 CC load

t

Будущее CC



2019

Instant execution prototype

Gradle 8.1

Stable CC

Gradle 9.0

CC is preferred mode

Gradle 9.x

Experimental Isolated Projects

Gradle 10.0

CC is only mode

Параллельная конфигурация ✅

Isolated Projects loading…