

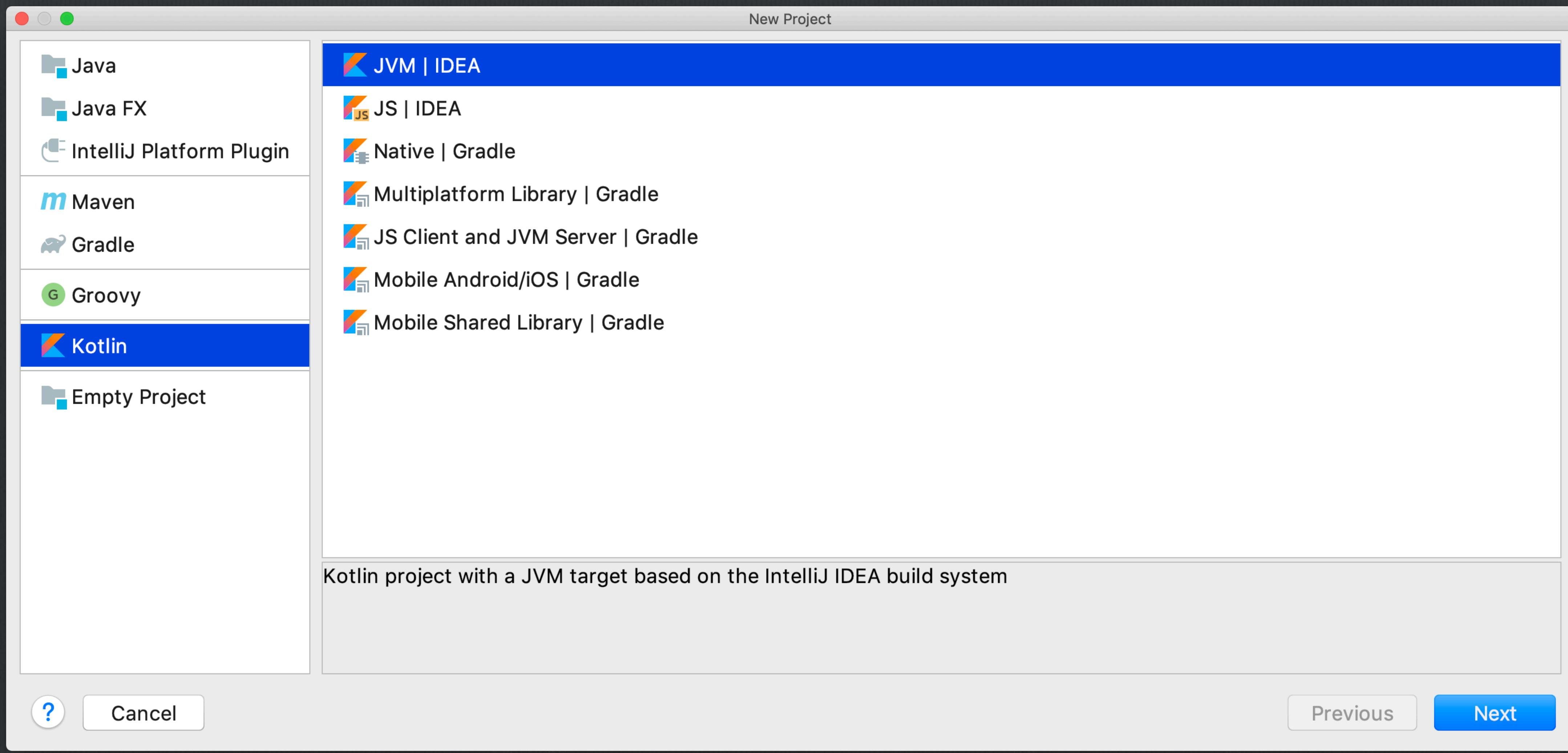


# Kotlin

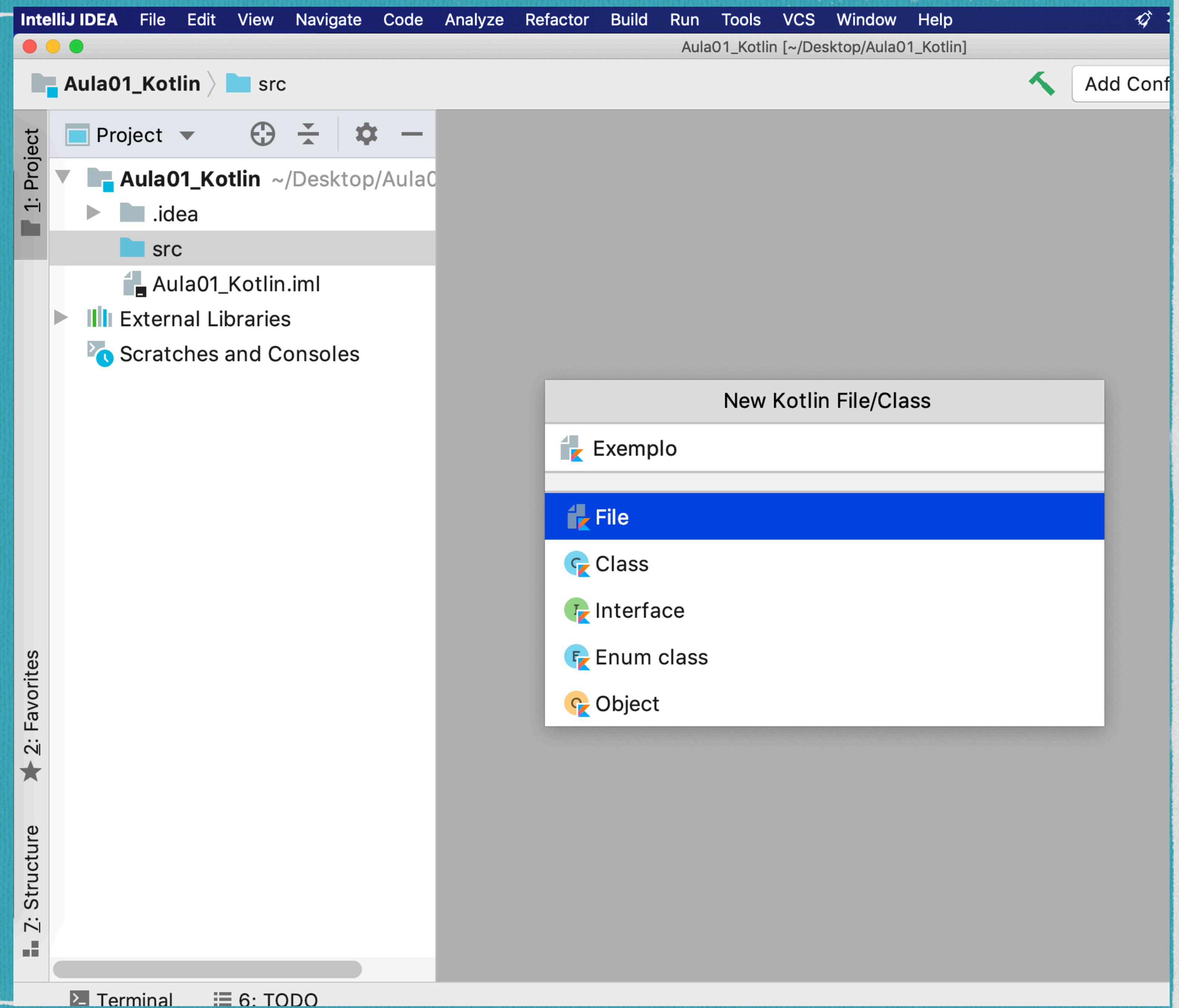
Rafael Vieira Coelho  
[rafaelvc2@gmail.com](mailto:rafaelvc2@gmail.com)

<https://kotlinlang.org>

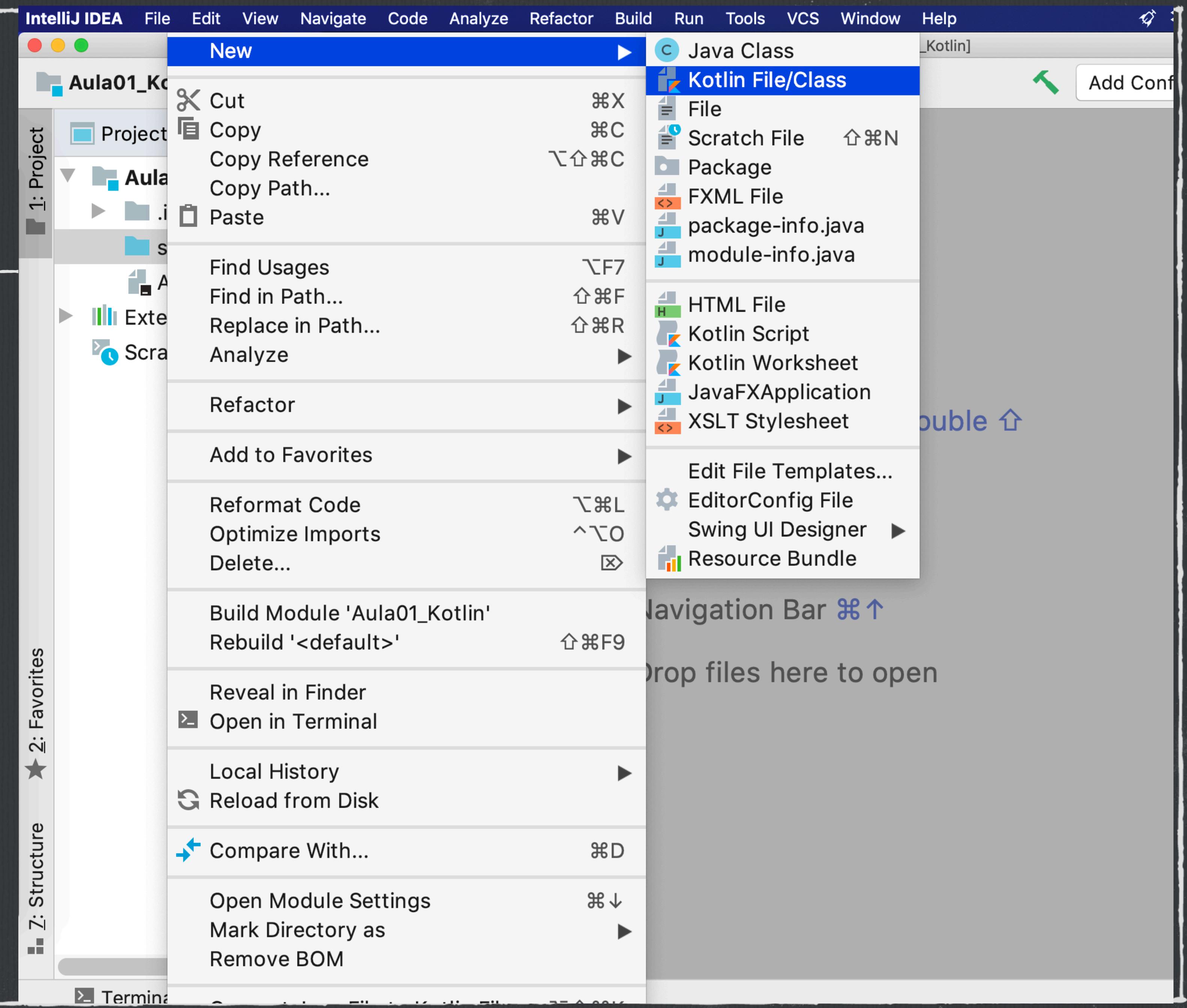
# 1) Criando um Projeto Kotlin no IntelliJ



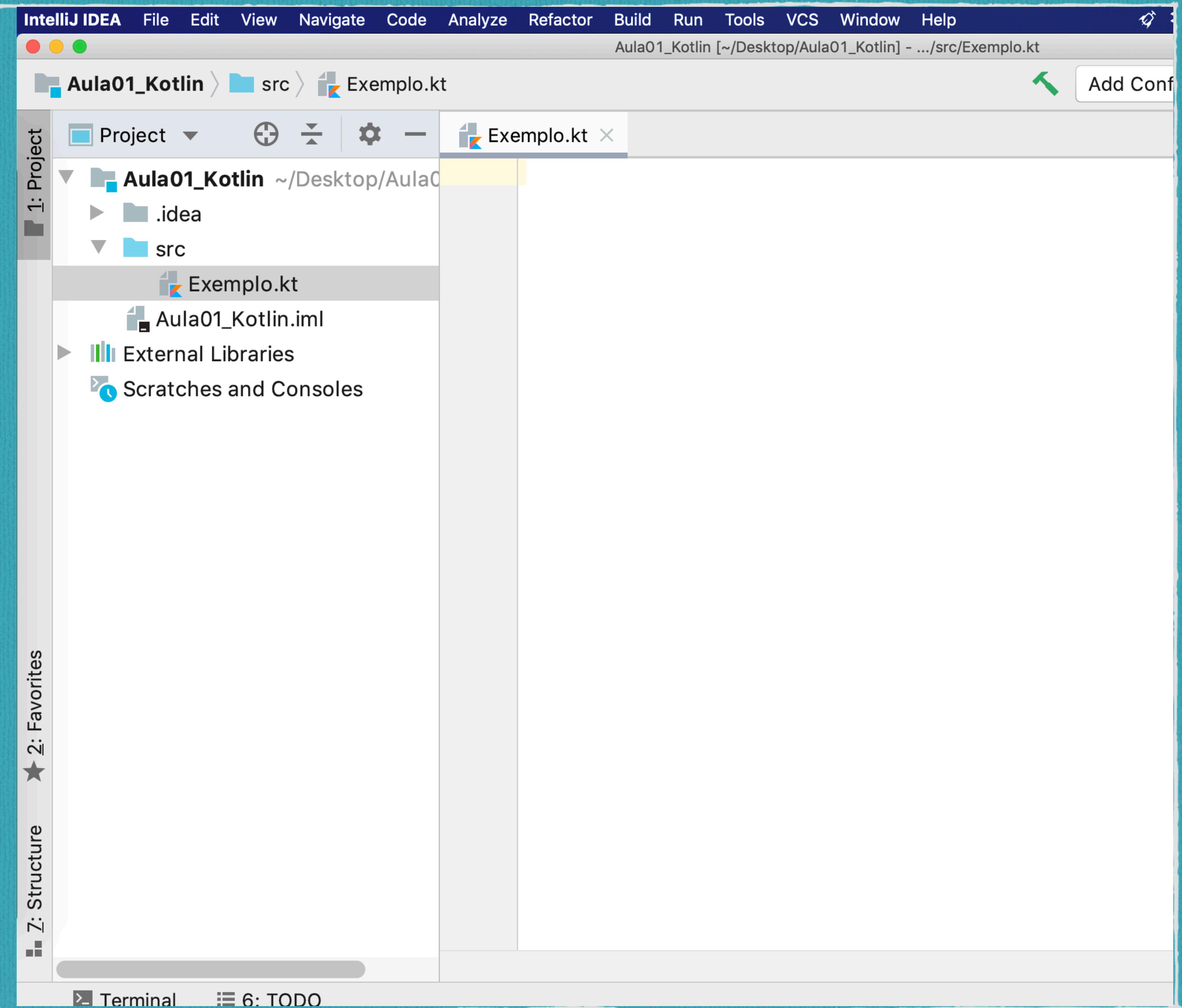
## 2) Criando um novo arquivo .kt



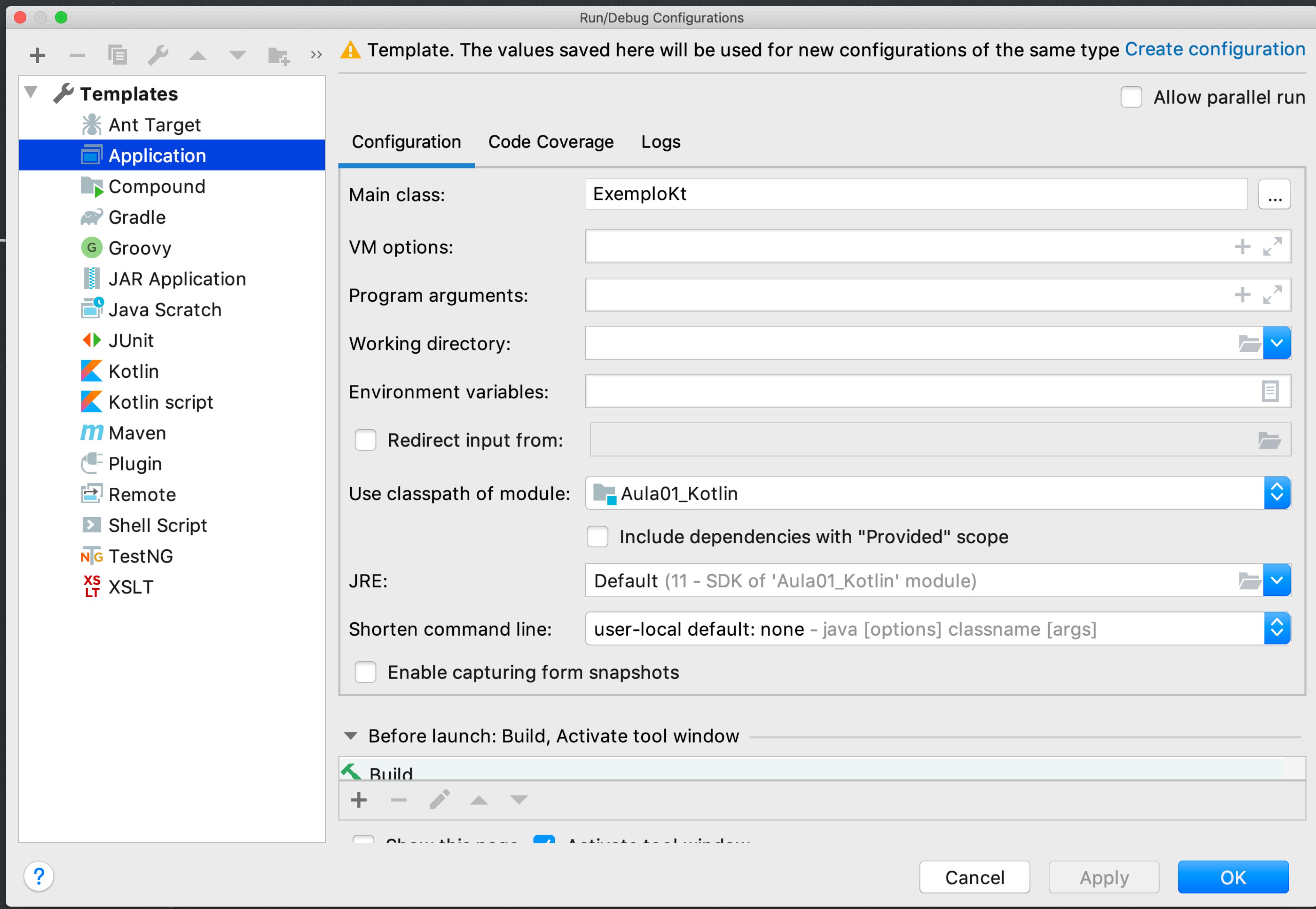
## 2) Criando um novo arquivo .kt



## 2) Criando um novo arquivo .kt



# 3) Configurando para Executar



The screenshot shows the IntelliJ IDEA interface with the following details:

- Project Bar:** Aula01\_Kotlin > src > Exemplo.kt
- Project View (1: Project):** Shows the project structure with Aula01\_Kotlin containing .idea, out, and src folders. src contains Exemplo.kt and Aula01\_Kotlin.iml. External Libraries and Scratches and Consoles are also listed.
- Code Editor:** The file Exemplo.kt contains the following code:

```
1 fun main() {
2     println("Hello World")
3 }
```
- Run Tab (2: Run):** The run configuration is set to ExemploKt. The output window shows:

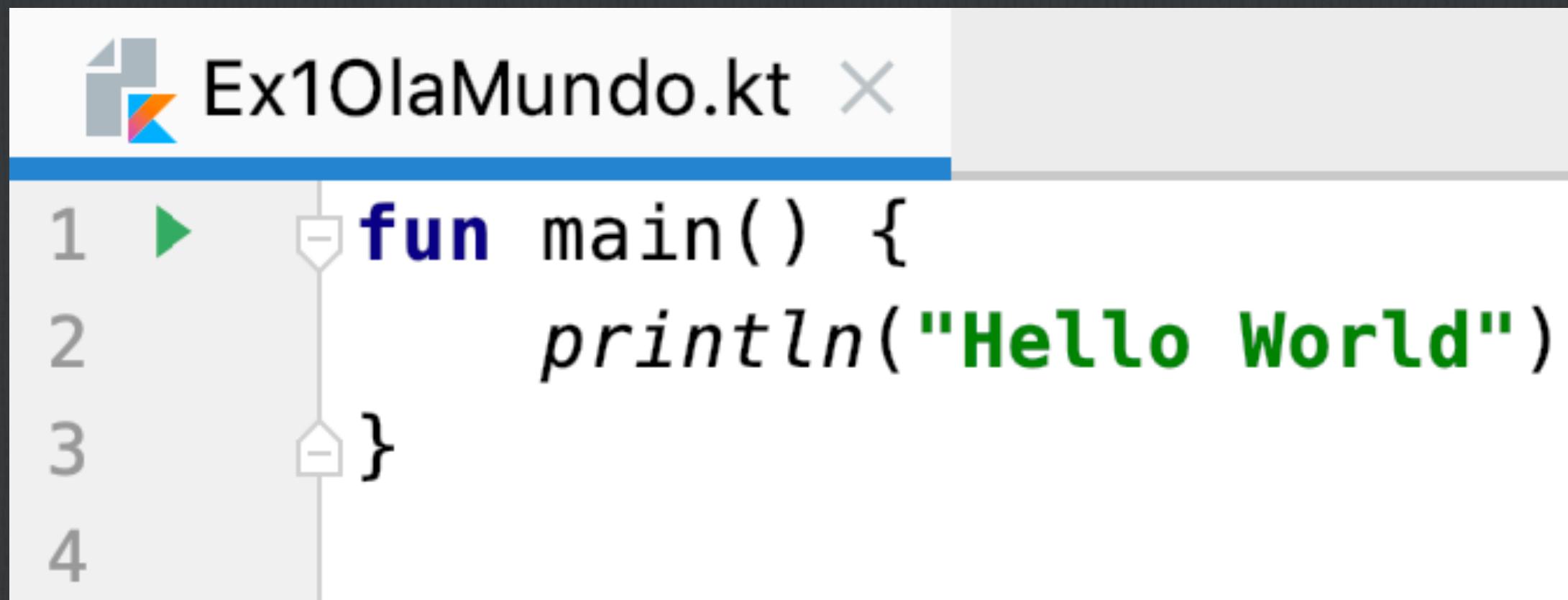
```
"/Applications/IntelliJ IDEA CE.app/Contents/jbr/Contents/Home/bin/java" "-javaagent:/Applications/IntelliJ IDEA CE.app/...
Hello World

Process finished with exit code 0
```
- Bottom Navigation:** 0: Messages, 4: Run, 6: TODO, Terminal, Event Log.
- Bottom Status:** Build completed successfully in 8 s 587 ms (moments ago), 5:1 LF, UTF-8, 4 spaces, lock icon.

**Section Title:** 4) Programa Olá Mundo

## 4) Programa Olá Mundo

---

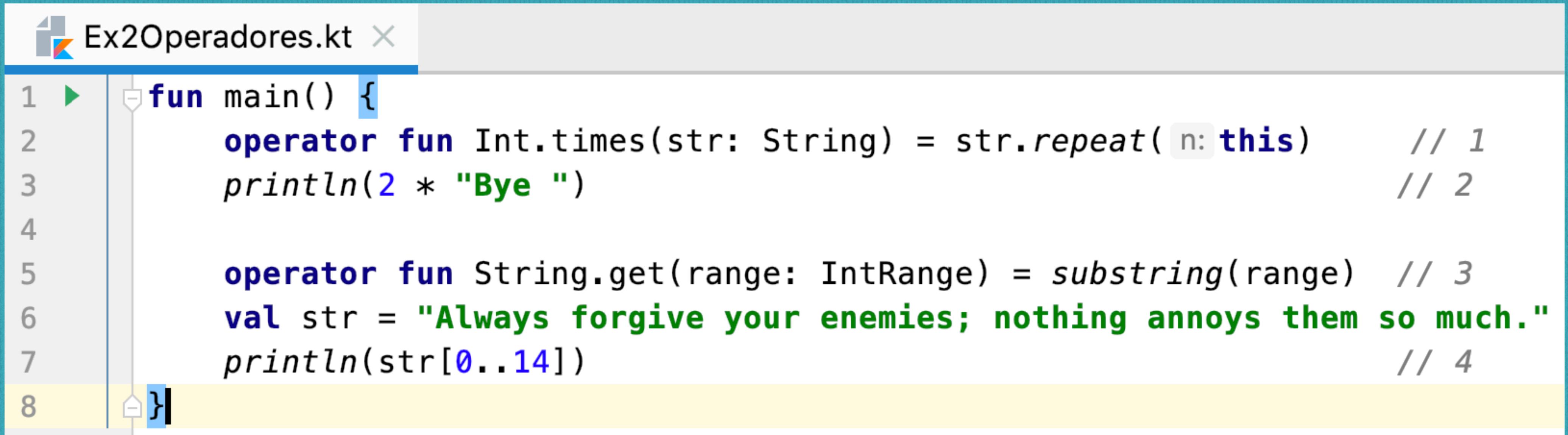


The image shows a screenshot of a code editor window titled "Ex1OláMundo.kt". The code editor has a dark theme. The file contains the following code:

```
1 >  fun main() {  
2       println("Hello World")  
3   }  
4
```

The first line starts with a green arrow icon followed by the word "fun". The second line contains the method name "main" and the third line contains the "println" function with the string "Hello World". The fourth line is an empty line.

# 5) Operadores

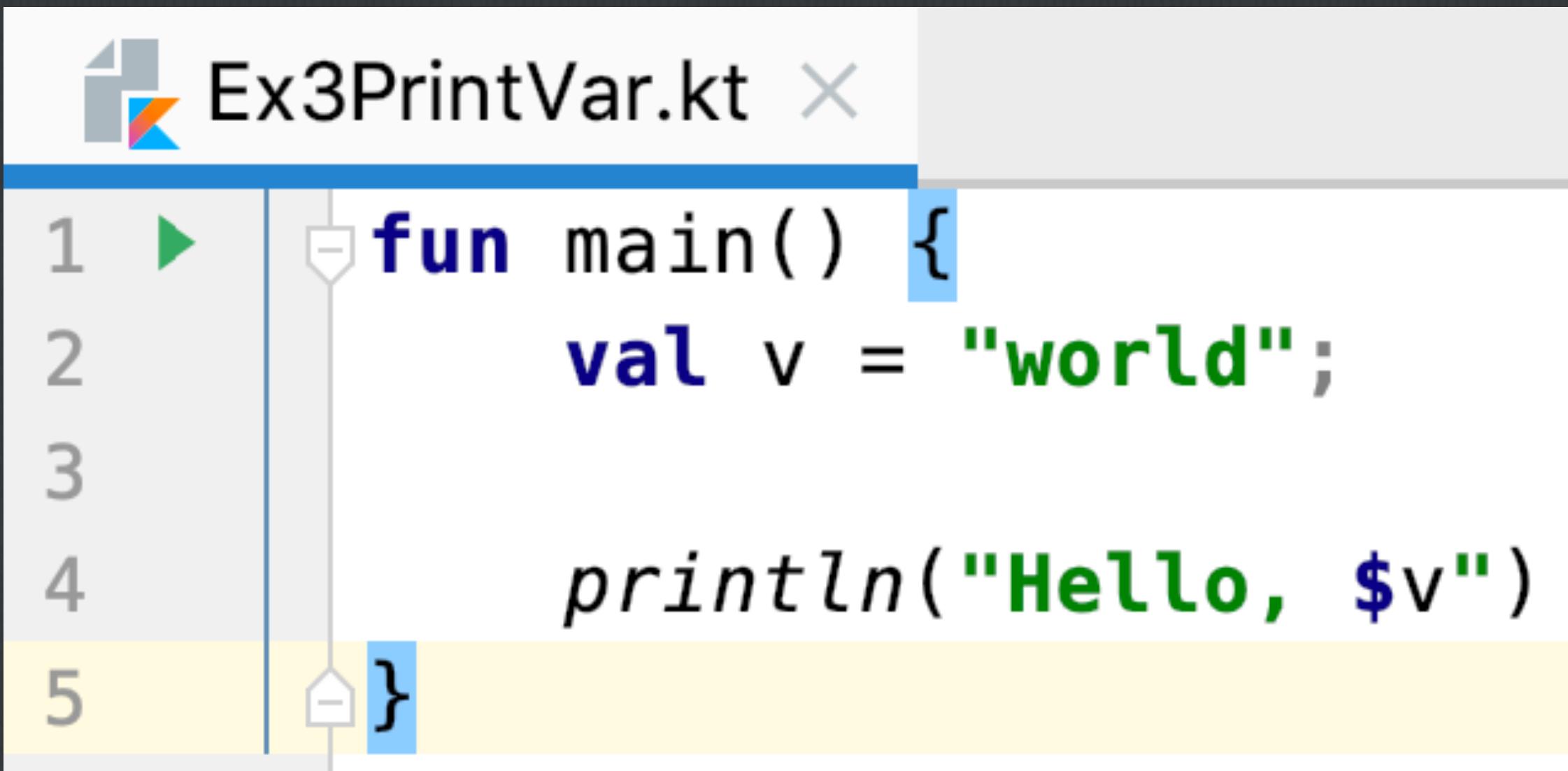


The screenshot shows a code editor window with the file name "Ex2Operadores.kt" at the top. The code defines a main function that contains two operator functions: one for Int and one for String. The Int operator function "times" repeats a string based on the value of the Int. The String operator function "get" returns a substring based on the specified range. A val declaration creates a string "str" with a quote from Sun Tzu, and a println statement prints a portion of it.

```
1 > | fun main() {  
2 |     operator fun Int.times(str: String) = str.repeat( n: this)      // 1  
3 |     println(2 * "Bye ")  
4 |  
5 |     operator fun String.get(range: IntRange) = substring(range) // 3  
6 |     val str = "Always forgive your enemies; nothing annoys them so much."  
7 |     println(str[0..14])                                         // 4  
8 }|
```

## 6) Imprimindo Conteúdo de Variável

---



```
Ex3PrintVar.kt ×
1 ► fun main() {
2     val v = "world";
3
4     println("Hello, $v")
5 }
```

Aula02\_Kotlin > src > Ex3PrintVar.kt

Project

1: Project

Aula02\_Kotlin ~/Dropbox/Livros/Ling  
|.idea  
| out  
| src  
| Ex1OlaMundo.kt  
| Ex2Operadores.kt  
| Ex3PrintVar.kt  
Aula02\_Kotlin.iml  
External Libraries  
Scratches and Consoles

Ex1OlaMundo.kt Ex2Operadores.kt Ex3PrintVar.kt

Ant

```
1 fun main() {  
2     val v = "world";  
3     println("Hello, $v")  
4 }  
5
```

main()

Run: Ex3PrintVarKt

Z: Structure

"/Applications/IntelliJ IDEA CE.app/Contents/jbr/Contents/Home/bin/java" "-javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea\_rt.jar=53148:/Applications/IntelliJ IDEA CE.app/Contents/bin  
Hello, world  
Process finished with exit code 0

2: Favorites

4: Run 6: TODO Terminal 0: Messages Event Log

Build completed successfully in 9 s 290 ms (moments ago) 4:25 LF UTF-8 4 spaces

## 6) Imprimindo Conteúdo de Variável

## 7) Programa Principal (main)

---

- Podemos declarar a função main com ou sem argumentos:

```
fun main() { ... } // starting from Kotlin 1.3  
  
fun main(args: Array<String>) { ... }
```

# Main com Argumentos

```
fun main(args: Array<String>) {  
    val name = if (args.size > 0) args[0] else "Kotlin"  
    println("Hello, $name!")  
}
```

# Arrays

---

```
fun main(args: Array<String>) {  
    val name = if (args.size > 0) args[0] else "Kotlin"  
    println("Hello, $name!")  
}
```

# Expressão if

```
fun main(args: Array<String>) {  
    val name = if (args.size > 0) args[0] else "Kotlin"  
    println("Hello, $name!")  
}
```

# String Templates

---

```
"...$variable..."  
"...${functionCall(...)}..."
```

```
fun main(args: Array<String>) {  
    println("Hello, ${args.getOrNull(0)}!")  
}
```

# Pergunta: O que é mostrado na tela se não são passados argumentos?

```
1 fun main(args: Array<String>) {  
2     println("Hello, ${args.getOrNull(0)}!")  
3 }
```

- Hello, null!
- Hello, !
- Será lançada a exceção NPE (Null Pointer Exception)

# Pergunta: O que é mostrado na tela se não são passados argumentos?

```
1 fun main(args: Array<String>) {  
2     println("Hello, ${args.getOrNull(0)}!")  
3 }
```

- Hello, null!**
- Hello, !
- Será lançada a exceção NPE (Null Pointer Exception)