Vitor Amadeu Souza

Introdução ao

Kotlin

Parte IV

© 2025 by Cerne Tecnologia e Treinamento Ltda.

© 2025 by Vitor Amadeu Souza

Nenhuma parte desta publicação poderá ser reproduzida sem autorização prévia e escrita de **Cerne Tecnologia e Treinamento Ltda.** Este livro publica nomes comerciais e marcas registradas de produtos pertencentes a diversas companhias. O editor utiliza as marcas somente para fins editoriais e em benefício dos proprietários das marcas, sem nenhuma intenção de atingir seus direitos.

Fevereiro de 2025

Direitos reservados por:
Cerne Tecnologia e Treinamento Ltda
Produção: Cerne Tecnologia e Treinamento
E-mail da Empresa: cerne@cerne-tec.com.br
Home Page: www.cerne-tec.com.br.com.br
Atendimento ao Consumidor: sac@cerne-tec.com.br
Contato com o Autor: vitor@cerne-tec.com.br

FEITO NO BRASIL

“Se quer ir rápido, vá sozinho. Se quer ir longe, vá acompanhado. ”

Provérbio africano

Cerne Tecnologia

A Cerne Tecnologia tem uma equipe preparada para desenvolvimento de projetos eletrônicos em diversas áreas: Médica, Entretenimento, Industrial, Robótica, Científica, Automobilística, Aeronáutica, etc. Trabalhamos com tecnologia microcontrolada usando o PIC, ARM, AVR, 8051, dsPIC, PIC24, PIC32 além do Arduino, Raspberry, Beaglebone etc. Desenvolvemos o projeto desde sua concepção até a entrega do produto final, passando pelas etapas de esquema elétrico, protótipo e desenvolvimento de circuito impresso.

Desenvolvemos aplicativos para smartphones/tablets Android, iOS, Blackberry, Windows Phone e no desenvolvimento de softwares a nível PC para plataforma Windows, usando ferramentas como o Visual Basic, C# e C++.

Atuamos na parte de montagem de placas, onde podemos fornecer ambos os serviços de desenvolvimento de projetos e produção ou apenas um destes.

Desenvolvemos esquemas elétricos e layout de PCI, tanto em tecnologia convencional como SMD.

Temos a flexibilidade de customizar um de nossos produtos, de modo a atender a uma necessidade específica do cliente, tornando o custo de desenvolvimento menor se comparado a construção de um projeto desde a sua fase inicial.

Desenvolvemos e fornecemos kits didáticos para diversos microcontroladores além de apostilas, livros e e-books.

Na hora de desenvolver um projeto ou equipar seu laboratório não hesite em nos contatar. Entre em contato conosco através do endereço cerne-tec.com.br para obter mais informações.

Cerne
Conhecimento para o Desenvolvimento

cerne-tec.com.br

Sumário

Capítulo I – Metodologia de desenvolvimento **6**

1. Introdução 6

Capítulo II – Programação em Kotlin **7**

1. none 7
2. all 8
3. any 9
4. map 10
5. filterpp 11
6. run 12
7. also 13
8. let 14
9. when 16
10. if 17
11. try-catch-finally 18
12. Ranges com Passo 19

Capítulo I

Metodologia de desenvolvimento

1. Introdução

Este livro tem como objetivo ensinar a programação utilizando Kotlin, explorando seus recursos de forma prática e aplicada. Ao longo desta série, diversas aplicações serão desenvolvidas, permitindo uma compreensão aprofundada da linguagem e de suas principais funcionalidades. Cada volume abordará temas específicos, conforme detalhado no sumário, proporcionando um aprendizado progressivo e estruturado.

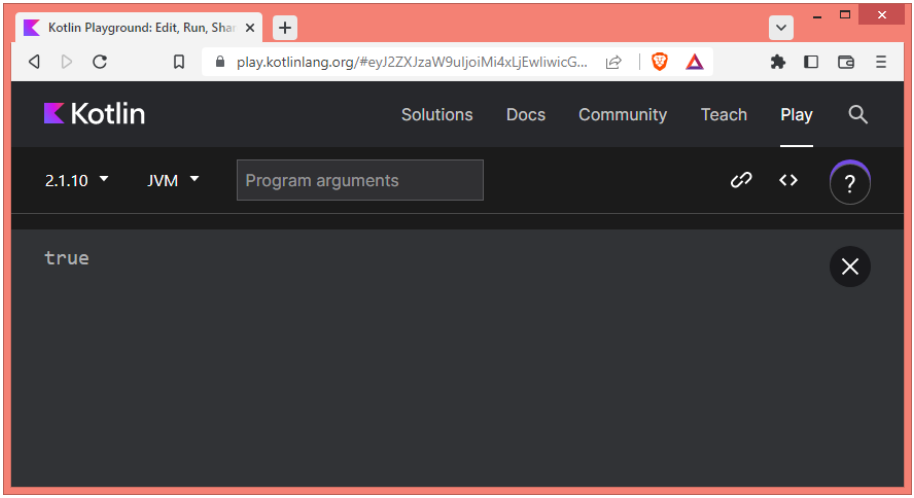
Capítulo II

Programação em Kotlin

1. none

O código cria uma lista chamada numbers com os valores 1, 3, 5 e 7. Em seguida, a função none é usada para verificar se nenhum dos elementos da lista é par. O predicado it % 2 == 0 dentro da função none verifica se um número é divisível por 2 (ou seja, se é par). Como todos os números da lista são ímpares, a função none retorna true.

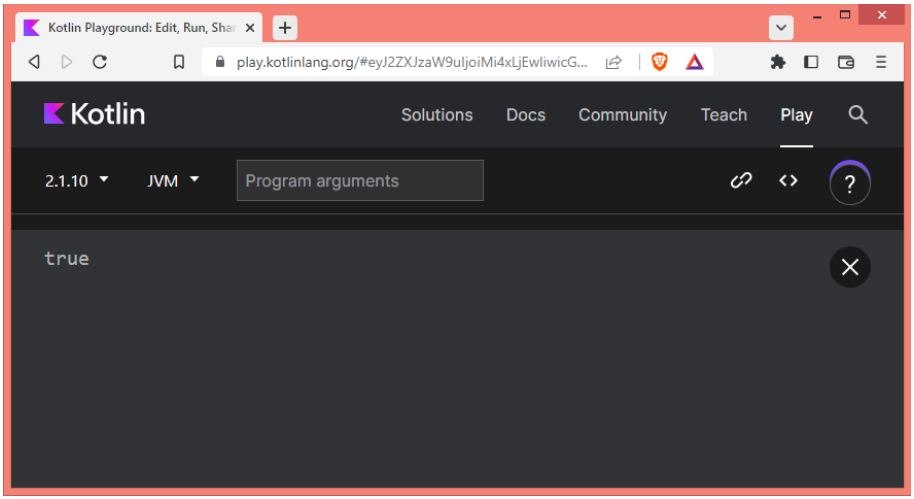
```
fun main() {
    val numbers = listOf(1, 3, 5, 7)
    val noEven = numbers.none { it % 2 == 0 }
    println(noEven)
}
```



2. all

O código cria uma lista chamada numbers, contendo os valores 2, 4, 6 e 8. A função all é usada para verificar se todos os elementos da lista atendem ao predicado fornecido. No caso, o predicado é it % 2 == 0, que verifica se o número é par. Como todos os números da lista são pares, a função all retorna true.

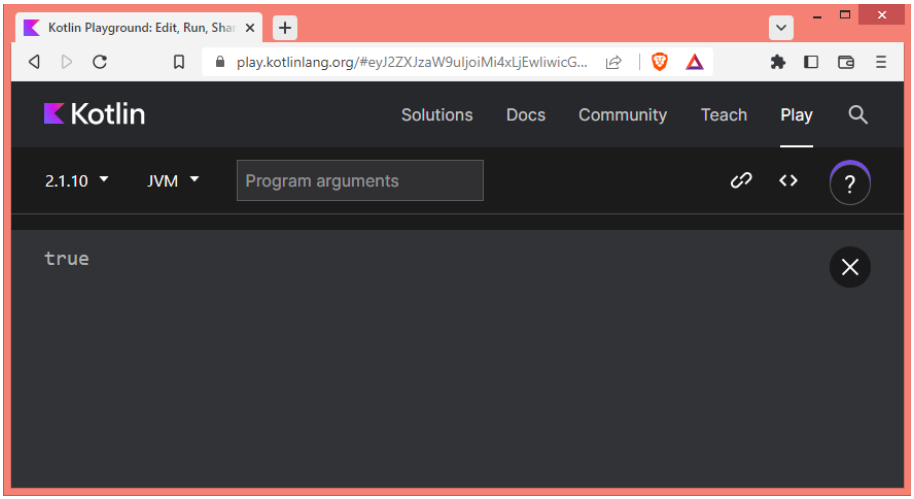
```
fun main() {
    val numbers = listOf(2, 4, 6, 8)
    val allEven = numbers.all { it % 2 == 0 }
    println(allEven)
}
```



3. any

O código cria uma lista chamada numbers com os valores 1, 2, 3, 4 e 5. A função any é utilizada para verificar se pelo menos um elemento da lista atende ao predicado fornecido. O predicado it % 2 == 0 verifica se o número é par. Como existem os números 2 e 4 na lista (que são pares), a função any retorna true.

```
fun main() {
    val numbers = listOf(1, 2, 3, 4, 5)
    val hasEven = numbers.any { it % 2 == 0 }
    println(hasEven)
}
```



4. map

O código cria uma lista chamada numbers com os valores 1, 2, 3, 4 e 5. A função map é utilizada para transformar cada elemento da lista, aplicando a operação it * it, que eleva cada número ao quadrado. O resultado é uma nova lista com os números ao quadrado.

```
fun main() {
    val numbers = listOf(1, 2, 3, 4, 5)
    val squaredNumbers = numbers.map { it * it }
    println(squaredNumbers)
}
```