Aprenda Programação Funcional

Aprenda Programação Funcional

Como Pensar Funcionalmente para Trabalhar com Códigos Complexos

Jack Widman, Ph.D.

ALTA BOOKS
GRUPO EDITORIAL
Rio de Janeiro, 2024

Aprenda Programação Funcional

ISBN: 978-85-508-1962-4

Impresso no Brasil — 1ª Edição, 2024 — Edição revisada conforme o Acordo Ortográfico da Língua Portuguesa de 2009.

Dados Internacionais de Catalogação na Publicação (CIP)
(BENITEZ Catalogação Ass. Editorial, MS, Brasil)

W633a 1.ed. Widman, Jack
Aprenda programação funcional : como pensar funcionalmente para trabalhar com códigos complexos / Jack Widman ; tradução Renan Amorim. – 1.ed. – Rio de Janeiro : Alta Books, 2024.
160 p.; il.; 15,7 x 23 cm.

Título original: Learning functional programming.
ISBN 978-85-508-1962-4

1. Algoritmos. 2. Ciência da computação. 3. Java (Linguagem de programação para computadores). 4. Programação (Computadores). I. Amorim, Renan. II. Título.

10-2024/112 CDD 004

Índice para catálogo sistemático:
1. Linguagem de programação : Ciência da computação 004

Aline Graziele Benitez – Bibliotecária – CRB-1/3129

Produção Editorial: Grupo Editorial Alta Books
Diretor Editorial: Anderson Vieira
Vendas Governamentais: Cristiane Mutüs
Gerência Comercial: Claudio Lima

Produtora Editorial: Isabella Gibara
Tradução: Renan Amorim
Copidesque: Aline Amaral
Revisão: André Cavanha; Alessandro Thomé
Diagramação: Joyce Matos
Revisão Técnica: Mario Adaniya
(Professor de Ciência da Computação no Centro Universitário Filadélfia - UniFil)

ALTA BOOKS
GRUPO EDITORIAL

Rua Viúva Cláudio, 291 — Bairro Industrial do Jacaré
CEP: 20.970-031 — Rio de Janeiro (RJ)
Tels.: (21) 3278-8069 / 3278-8419
www.altabooks.com.br — altabooks@altabooks.com.br
Ouvidoria: ouvidoria@altabooks.com.br

Editora **afiliada à:**
ASSOCIADO

Para as minhas três filhas incríveis: Katherine, Annie e Victoria.
E para Andrea, cujo apoio e amor me incentivaram de tantas formas.

Sumário

Prefácio

No decorrer dos últimos anos, a programação funcional (PF) passou por um período de renascença. Várias empresas estão procurando por programadores que tenham experiência em programação funcional. Muitas linguagens que não foram projetadas originalmente para serem funcionais vêm evoluindo com o passar do tempo para incluírem capacidades funcionais. Linguagens como Java, JavaScript e Python, para citar apenas algumas, vêm se tornando cada vez mais funcionais. O incentivo para os programadores adquirirem experiência funcional parece se dever, em parte, à aparente melhoria no processo de desenvolvimento, incluindo uma sensação de que, quando adotamos o método funcional, temos menos bugs e produzimos códigos mais expansivos e robustos. Se isso é verdade ou não e se a proporção da produção de códigos funcionais aumentará, só o tempo dirá. Por enquanto, consideraremos a programação funcional como um de vários paradigmas, cada um deles com suas próprias vantagens e desvantagens.

Quem Deveria Usar Este Livro?

Basicamente, todos os programadores. Se não tem experiência em PF, mas já ouvir falar nela e ficou curioso, ou até se comprou esta publicação sem a menor noção do que seja a PF, este livro será útil para você. Programadores de PF experientes também tirarão proveito dele. Este livro analisa as raízes da Teoria das Categorias da PF de uma forma jamais vista em outros livros sobre PF. Por fim, os programadores que têm certa experiência no uso da PF, mas que desejam ir além e se aprofundar nos conceitos e teorias que a compõem, encontrarão muitas coisas que poderão lhes ser úteis.

Como Este Livro Está Organizado

Queremos demonstrar, por meio de várias linguagens de programação, como os construtos funcionais podem melhorar nossos códigos. Mas, à medida que o livro avança, a linguagem de programação Scala é usada com mais frequência nos exemplos. O motivo disso é que o autor acredita que, devido à facilidade com que ideias funcionais conseguem ser expressas em Scala, o leitor terá muito mais facilidade de entender tais ideias se estas forem transmitidas de um modo mais natural, o que é possível fazer com a Scala. Para uma breve introdução à Scala, veja o apêndice.

Convenções Utilizadas Neste Livro

As seguintes convenções tipográficas são usadas neste livro:

Itálico

Indica novos termos, URLs, endereços de e-mail, nomes de arquivos e extensões de arquivos.

`Fonte monoespaçada`

Usada para a listagem de programas e para fazer referência a elementos dos programas em parágrafos, como nomes de variáveis ou funções, bancos de dados, tipos de dados, variáveis de ambiente, declarações e palavras-chave.

Este elemento representa uma dica ou sugestão.

Este elemento representa uma observação geral.

Este elemento representa um alerta ou atenção.

Temos um site para este livro, onde listamos erratas, exemplos e outras informações adicionais. Você pode acessar essa página utilizando o seguinte endereço: *https://oreil.ly/learning-fp (conteúdo em inglês)*.

CAPÍTULO 1

O que É Programação Funcional?

Programação funcional? Functores? Monoides, monads? "Eu não sou nenhum matemático!", você talvez diga. "Como vou aprender esses conceitos esotéricos? E por que deveria querer fazer isso?" Essas preocupações são totalmente compreensíveis. Mas a verdade é que você não precisa ser um matemático para ser um programador funcional.

Os conceitos fundamentais da programação funcional são fáceis de entender quando apresentados de forma clara e direta. E é esse o objetivo deste livro: tornar a programação funcional compreensível e prática. Mais especificamente, vou ensiná-lo a *pensar* como um programador funcional. Mas por que você desejaria aprender programação funcional?

Imagine o seguinte: são 22h e você está totalmente travado, tentando resolver um bug de um programa que precisa entregar na manhã seguinte. O problema parece girar em torno de uma variável chamada *ratio*. O problema é que, dependendo das condições do sistema que está modelando, essa variável continua mudando. Sua frustração aumenta. Ou você tem um prazo no trabalho e seu microsserviço tem um bug elusivo que está lutando para encontrar. O problema parece estar em dois loops for aninhados nos quais as variáveis são modificadas de uma forma bastante intrincada. A lógica é complexa, e você não consegue encontrar a solução. Quem dera houvesse uma maneira de escrever programas de uma forma em que os valores das variáveis não mudassem! É aí que entra a programação funcional.

As variáveis cujos valores mudam com frequência são uma considerável fonte de bugs em programas. Pode ser difícil acompanhar os valores delas porque elas podem mudar a qualquer momento.