



Bacharelado em Sistemas de Informação

Paradigmas de Programação

Prof. Eduardo Heredia



Sobre a disciplina

- Os Paradigmas de Programação estabelecem a visão que o programador possui sobre a estruturação e a execução de um programa.
- Exemplos de Paradigmas
 - Programação Imperativa
 - Procedural
 - Orientada a Objetos
 - Programação Declarativa
 - Lógico
 - Funcional
- Neste curso abordaremos a programação declarativa, com ênfase na programação funcional.

Metodologia

- Aulas teóricas expositivas
- Exploração prática dos conceitos
- ADOs complementando a teoria
- EPs para exercício da programação

Critérios para Aprovação na Disciplina

- Frequência mínima de 75% nas aulas da disciplina
- Média $\geq 6,0$, conquistada nas seguintes atividades:
 - 2 provas individuais, com cola oficial
 - P_1 em 02/10/2022 (previsão)
 - P_2 em 27/11/2022 (previsão)
 - n ADOs (atividade discente orientada)
 - Aplicadas ao longo do semestre
 - Não serão aceitas ADOs fora do prazo
 - m EPs (exercício de programação)
 - Aplicados ao longo do semestre
 - Não serão aceitas EPs fora do prazo

Cálculo da Média Final


$$\text{Média Final} = \frac{P_1 + P_2}{2} \cdot 5 + \frac{\sum_1^n \text{ADO}_n}{n} \cdot 2 + \frac{\sum_1^m \text{EP}_m}{m} \cdot 3$$

Ética Acadêmica

- Não plagiem
- Não deixem de creditar autoria quando apropriado
- Não enviem qualquer trabalho que não seja de própria autoria
- Não enviem trabalhos que sejam baseados em trabalhos de outros colegas
 - A troca de ideias é salutar, mas usar o texto/código alheio como base de sua própria produção é considerado plágio
- Em caso de plágio
 - A nota de todos os alunos envolvidos será zerada!
 - Não será possível realizar uma atividade substitutiva
 - Não há limite de data para detecção do plágio

Referências

<https://www.swi-prolog.org>



Robust, mature, free. **Prolog for the real world.**


SWI Prolog

HOME	DOWNLOAD	DOCUMENTATION	TUTORIALS	COMMUNITY	USERS	WIKI
------	----------	---------------	-----------	-----------	-------	------

SWI-Prolog offers a comprehensive free Prolog environment. Since its start in 1987, SWI-Prolog development has been driven by the needs of real world applications. SWI-Prolog is widely used in research and education as well as commercial applications. Join over a million users who have downloaded SWI-Prolog. [more ...](#)

Referências

<https://wiki.haskell.org>



Navigation

[Haskell](#)
[Wiki community](#)
[Recent changes](#)
[Random page](#)

Tools

[What links here](#)
[Related changes](#)
[Special pages](#)
[Printable version](#)
[Permanent link](#)
[Page information](#)
[Cite this page](#)

Page

Discussion

Read

View source

View history

Introduction

Haskell is a computer programming language. In particular, it is a *polymorphically statically typed, lazy, purely functional* language, quite different from most other programming languages. The language is named for **Haskell Brooks Curry**, whose work in mathematical logic serves as a foundation for functional languages. Haskell is based on the *lambda calculus*, hence the lambda we use as a logo.

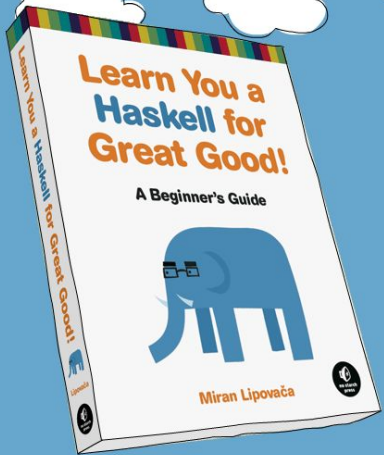
Contents

hide

- 1 Why use Haskell?
- 2 What is functional programming?
- 3 What's good about functional programming?
 - 3.1 Quicksort in Haskell
 - 3.2 Quicksort in C
 - 3.3 Ease of understanding
 - 3.4 Brevity

Referências

<http://haskell.tailorfontela.com.br/chapters>



Hey yo! This is **Learn You a Haskell**, the funkiest way to learn Haskell, which is the best functional programming language around. You may have heard of it. This guide is meant for people who have programmed already, but have yet to try functional programming.

The whole thing is completely free to read online, but it's also available in print and I encourage you to buy as many copies as you can afford!

To contact me, shoot me an email to: `bonus at learnyouahaskell dot com`! You can also find me idling on `#haskell` where I go by the name **BONUS**.

Got questions? READ THE FAQ

Buy it!
You know you want to!

Read it online!
For free!

inglês
Google Translate
SVTIV.

Referências

<http://book.realworldhaskell.org/>

Real World Haskell

by Bryan O'Sullivan, Don Stewart, and John Goerzen

[Why functional programming? Why Haskell?](#)



1. [Getting started](#)



2. [Types and functions](#)



3. [Defining types, streamlining functions](#)



4. [Functional programming](#)



5. [Writing a library: working with JSON data](#)



6. [Using typeclasses](#)



7. [Input and output](#)



8. [Efficient file processing, regular expressions, and file name matching](#)



9. [I/O case study: a library for searching the filesystem](#)



10. [Code case study: parsing a binary data format](#)



11. [Testing and quality assurance](#)



12. [Barcode recognition](#)



Conteúdo Programático (resumo)

- Computabilidade
 - Máquina de Turing
 - Cálculo Lambda
- Linguagens de Programação
 - Semântica
- Paradigmas de Programação
 - Paradigma Lógico
 - Paradigma Funcional