

Programação Funcional — BCC222

Aula 0

Porque aprender Haskell?

Lucília Camarão de Figueiredo

Departamento de Ciência da Computação

Universidade Federal de Ouro Preto

Porque aprender Haskell?

- ▶ É importante aprender diversas linguagens ao longo da carreira
- ▶ Aprender a pensar de maneira diferente sobre programas
 - ▶ Programação imperativa é baseada na noção de estado

```
total = 0;  
for (i = 1; i = 10; ++i)  
total = total + i;
```

- ▶ Programação Functional é baseada na noção de valor

```
sum [1..10]
```

- ▶ Aprender a operar sobre estruturas de dados de maneira geral

Programação Funcional no mundo real

- ▶ Google MapReduce, Sawzall
- ▶ Ericsson AXE phone switch
- ▶ Jane Street Capital
- ▶ Credit Suisse
- ▶ Morgan Stanley
- ▶ Standard Chartered
- ▶ Twitter
- ▶ Facebook
- ▶ Yahoo
- ▶ Erlang
- ▶ Scala
- ▶ Microsoft F#

Programação Funcional é o futuro

Características de linguagens funcionais têm sido incorporadas em outras linguagens

- ▶ **Garbage collection** (Java, C#, Python, Perl, Ruby, Javascript)
- ▶ **Higher-order functions** (Java, C#, Python, Perl, Ruby, Javascript)
- ▶ **Generics** (Java, C#)
- ▶ **List comprehensions** (C#, Python, Perl 6, Javascript)
- ▶ **Type classes** (C++ “concepts”)

Bibliografia

Livro Texto:

Haskell: The Craft of Functional Programming, Second Edition,
Simon Thompson, Addison-Wesley, 1999.

Thompson, Capítulos 1,2,3 (pp. 1–52): Aulas 1 e 2.

Thompson, Capítulos 4,5,7 (pp. 53–95, 115–134): Aula 3 e 4.

Thompson, Capítulos 6,8 (pp. 96–114, 135–151): Aulas 5 e 6.

Thompson, Capítulos 9,10,11 (pp. 152–209): Aulas 7 e 8

Prova 1

Thompson, Capítulos 12,13,14 (pp. 210–279): Aulas 11,12,13

Thompson, Capítulos 15,16 (pp. 280–336): Aulas 14

Thompson, Capítulos 17,18,19 (pp. 337–435): Aulas 15

Prova 2

Outras referências bibliográficas: no site da disciplina

<http://www.dcc.ufmg.br/~lucilia/cursos/func>

Mais informação sobre Haskell: Haskell Wikibook

Haskell - Wikibooks, collection of open-content textbooks - Windows Internet Explorer

http://en.wikibooks.org/wiki/Haskell

nr shevit

http://tweb/lt... #1494 (ponic... American Expr... Port 25 : Rand... http://research... Haskell - W... x

Рыковод

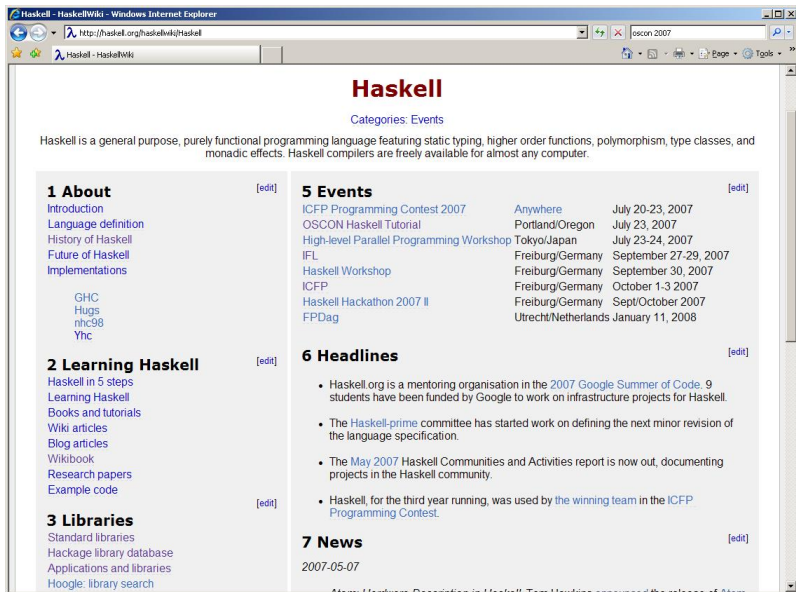
Haskell Basics [edit] - Getting set up - Variables and functions - Lists and tuples - Next steps - Type basics - Simple input and output - Type declarations edit this chapter	Elementary Haskell [edit] - Recursion - Pattern matching - More about lists - Control structures - List processing - More on functions - Higher order functions edit this chapter	Intermediate Haskell [edit] - Modules - Indentation - More on datatypes - Class declarations - Classes and types - Keeping track of State edit this chapter	Monads [edit] - Understanding monads - Advanced monads - Additive monads (MonadPlus) - Monad transformers - Practical monads edit this chapter
---	--	--	---

Advanced Track [edit]

This section will introduce wider functional programming concepts such as different data structures and type theory. It will also cover more practical topics like concurrency.

Advanced Haskell [edit] - Arrows - Understanding arrows - Continuation passing style (CPS) - Mutable objects - Zipers - Applicative Functors - Concurrency edit this chapter	Fun with Types [edit] - Existentially quantified types - Polymorphism - Advanced type classes - Phantom types - Generalised algebraic data-types (GADT) - Datatype algebra edit this chapter	Wider Theory [edit] - Denotational semantics - Equational reasoning - Program derivation - Category theory - The Curry-Howard isomorphism edit this chapter	Haskell Performance [edit] - Graph reduction - Laziness - Strictness - Algorithm complexity - Parallelism - Choosing data structures edit this chapter
---	---	--	---

Mais informação sobre Haskell: haskell.org



The screenshot shows a web browser window titled "Haskell - HaskellWiki - Windows Internet Explorer". The address bar shows "http://haskell.org/haskellwiki/Haskell". The page content is as follows:

Haskell

Categories: [Events](#)

Haskell is a general purpose, purely functional programming language featuring static typing, higher order functions, polymorphism, type classes, and monadic effects. Haskell compilers are freely available for almost any computer.

1 About

[Introduction](#)
[Language definition](#)
[History of Haskell](#)
[Future of Haskell](#)
[Implementations](#)

[GHC](#)
[Hugs](#)
[nhc98](#)
[Yhc](#)

2 Learning Haskell

[Haskell in 5 steps](#)
[Learning Haskell](#)
[Books and tutorials](#)
[Wiki articles](#)
[Blog articles](#)
[Wikibook](#)
[Research papers](#)
[Example code](#)

3 Libraries

[Standard libraries](#)
[Hackage library database](#)
[Applications and libraries](#)
[Hoogle: library search](#)

5 Events

ICFP Programming Contest 2007	Anywhere	July 20-23, 2007
OSCON Haskell Tutorial	Portland/Oregon	July 23, 2007
High-level Parallel Programming Workshop	Tokyo/Japan	July 23-24, 2007
IFL	Freiburg/Germany	September 27-29, 2007
Haskell Workshop	Freiburg/Germany	September 30, 2007
ICFP	Freiburg/Germany	October 1-3 2007
Haskell Hackathon 2007 II	Freiburg/Germany	Sept/October 2007
FPDag	Utrecht/Netherlands	January 11, 2008

6 Headlines

- Haskell.org is a mentoring organisation in the [2007 Google Summer of Code](#). 9 students have been funded by Google to work on infrastructure projects for Haskell.
- The [Haskell-prime](#) committee has started work on defining the next minor revision of the language specification.
- The [May 2007 Haskell Communities and Activities](#) report is now out, documenting projects in the Haskell community.
- Haskell, for the third year running, was used by [the winning team](#) in the [ICFP Programming Contest](#).

7 News

2007-05-07

More Haskell: Specifying in Haskell Tom Hickey announced the release of *Haskell*

Avaliação

Aulas Práticas: 20%

Prova 1: 30%

Prova 2: 50%