

Introdução

Paradigma de Programação Lógico

Marco A L Barbosa



Este trabalho está licenciado com uma Licença Creative Commons - Atribuição-Compartilhual 4.0 Internacional.

<http://github.com/malbarbo/na-func>

Conteúdo

Introdução

Imperativo vs declarativo

Funcional vs lógico

Prolog

Primeiros passos

Introdução

Introdução

No paradigma de **programação declarativo**, as estruturas e os elementos do programa são escritos de maneira a especificar a lógica da computação sem descrever o fluxo de controle.

Consequência da transparência referencial.

Imperativo vs declarativo

Imperativo vs declarativo

- ▶ Imperativo

- ▶ Modelo de computação baseado em sequência passo a passo de comandos
- ▶ Atribuições destrutivas
- ▶ Ordem de execução é crucial, os comandos só podem ser entendidos no contexto das computações anteriores devido aos efeitos colaterais
- ▶ Controle é responsabilidade do programados
- ▶ Exemplos: Java, C, Pascal

Imperativo vs declarativo

- ▶ Declarativo
 - ▶ Modelo de computação baseado em um sistema onde as relações são especificadas diretamente em termos da entrada
 - ▶ Atribuição não destrutiva
 - ▶ A ordem de execução não importa (não tem efeitos colaterais)
 - ▶ O programador não é responsável pelo controle
 - ▶ Exemplos: SQL, Prolog, Haskell
- ▶ Alguns autores consideram “como” (imperativo) vs “o que” (declarativo)

Funcional vs lógico

Funcional vs lógico

- ▶ Os dois principais paradigmas declarativos são o funcional e o lógico
- ▶ Funcional
 - ▶ Baseado em declaração e aplicação de funções (cálculo lambda)
 - ▶ Todos os parâmetro de uma função precisam estar instanciados
 - ▶ Clara distinção entre entrada e saída
- ▶ Lógico
 - ▶ Baseado no cálculo de predicados
 - ▶ Objetos e relações
 - ▶ A computação é feita usando um mecanismo de inferência lógico
 - ▶ A computação pode ser realizada com variáveis não instanciadas

Prolog

Prolog

- ▶ Para estudar o paradigma lógico vamos utilizar a linguagem Prolog
- ▶ Existem muitas implementações
- ▶ Vamos utilizar o SWI-Prolog

Primeiros passos

Instalação e execução

- ▶ Instalação

```
$ apt-get install swi-prolog
```

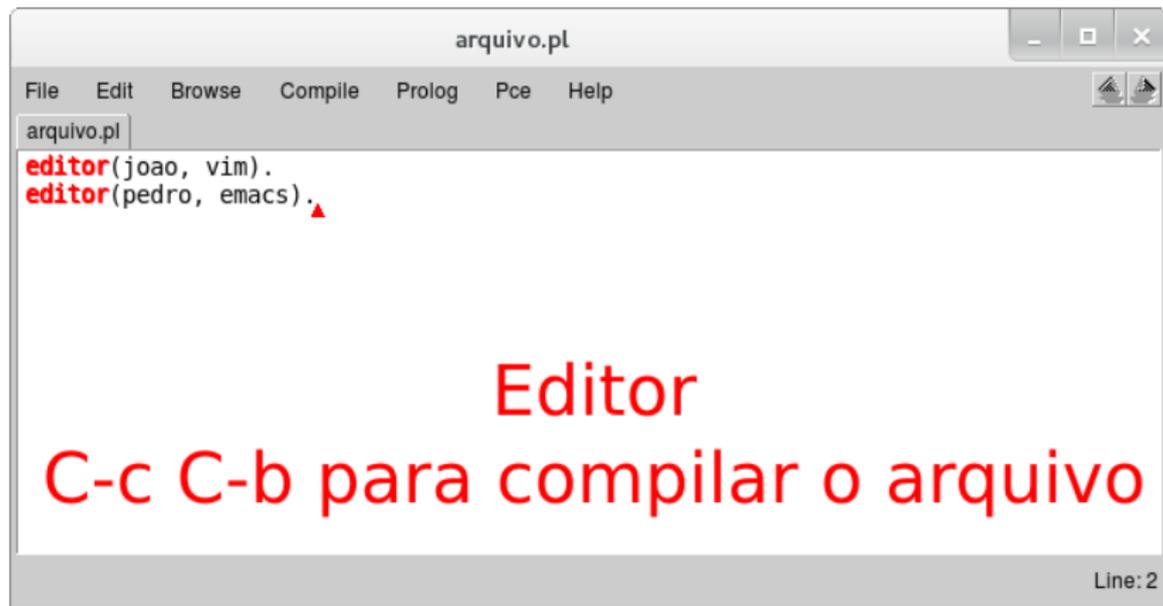
- ▶ Execução

```
$ swipl
```

Edição e consulta com editor integrado

```
malbarbo@drcopper:~  
Welcome to SWI-Prolog (Multi-threaded, 64 bits, Version 6.4.1)  
Copyright (c) 1990-2013 University of Amsterdam, VU Amsterdam  
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,  
and you are welcome to redistribute it under certain conditions.  
Please visit http://www.swi-prolog.org for details.  
  
For help, use ?- help(Topic). or ?- apropos(Word).  
?- emacs('arquivo.pl'). Editar arquivo.pl  
true.  
  
?- % /home/malbarbo/arquivo compiled 0.00 sec, 3 clauses  
?- editor(joao, E). Consulta  
E = vim. Resultado  
?- █
```

Editor integrado



Edição e consulta com outro editor

- ▶ Editar o arquivo usando o editor de sua preferência
- ▶ Ler o arquivo no swipl

```
?- consult('arquivo.pl').
```

- ▶ Fazer consultas

```
?- editor(joao, E).  
E = emacs.
```

- ▶ Depois de alterar o arquivo, ele deve ser lido novamente

Leitura recomendada

- ▶ The principal programming paradigms
- ▶ Declarative programming
- ▶ Logic programming
- ▶ Prolog