

Paradigmas de Programação

Paulo Torrens

paulotorrens@gnu.org

Departamento de Ciência da Computação
Centro de Ciências e Tecnologias
Universidade do Estado de Santa Catarina

5 de Agosto de 2019

- O que é uma *linguagem de programação*?
 - Uma linguagem de programação é uma linguagem formal usada para representar uma computação
 - Através de linguagens de programação podemos instruir computadores a executar um algoritmo

Introdução

- O que é uma *linguagem de programação*?
 - Uma linguagem de programação é uma linguagem formal usada para representar uma computação
 - Através de linguagens de programação podemos instruir computadores a executar um algoritmo

- O que é uma *linguagem de programação*?
 - Uma linguagem de programação é uma linguagem formal usada para representar uma computação
 - Através de linguagens de programação podemos instruir computadores a executar um algoritmo

“Linguagem é tudo igual, só muda a sintaxe!”

Não, linguagem não é tudo igual.

- O que uma linguagem de programação *precisa* ter?
 - Estruturas de repetição (for, while, etc)? Estruturas de controle (if)? Atribuição de variáveis? Recursão?
 - Linguagens diferentes possuem paradigmas diferentes, abstrações diferente, e, conseqüentemente, formas diferentes de resolver o mesmo problema
- O que são paradigmas?
 - Ao longo da história, formas totalmente diferentes de se programar foram descobertas
 - Paradigmas são uma forma de classificar linguagens de programação por suas características e abstrações
 - Os paradigmas que uma linguagem suporta influenciam diretamente na forma de pensar do programador

- O que uma linguagem de programação *precisa* ter?
 - Estruturas de repetição (for, while, etc)? Estruturas de controle (if)? Atribuição de variáveis? Recursão?
 - Linguagens diferentes possuem paradigmas diferentes, abstrações diferente, e, conseqüentemente, formas diferentes de resolver o mesmo problema
- O que são paradigmas?
 - Ao longo da história, formas totalmente diferentes de se programar foram descobertas
 - Paradigmas são uma forma de classificar linguagens de programação por suas características e abstrações
 - Os paradigmas que uma linguagem suporta influenciam diretamente na forma de pensar do programador

Introdução

- O que uma linguagem de programação *precisa* ter?
 - Estruturas de repetição (for, while, etc)? Estruturas de controle (if)? Atribuição de variáveis? Recursão?
 - Linguagens diferentes possuem paradigmas diferentes, abstrações diferente, e, conseqüentemente, formas diferentes de resolver o mesmo problema
- O que são paradigmas?
 - Ao longo da história, formas totalmente diferentes de se programar foram descobertas
 - Paradigmas são uma forma de classificar linguagens de programação por suas características e abstrações
 - Os paradigmas que uma linguagem suporta influenciam diretamente na forma de pensar do programador

- O que uma linguagem de programação *precisa* ter?
 - Estruturas de repetição (for, while, etc)? Estruturas de controle (if)? Atribuição de variáveis? Recursão?
 - Linguagens diferentes possuem paradigmas diferentes, abstrações diferente, e, conseqüentemente, formas diferentes de resolver o mesmo problema
- O que são paradigmas?
 - Ao longo da história, formas totalmente diferentes de se programar foram descobertas
 - Paradigmas são uma forma de classificar linguagens de programação por suas características e abstrações
 - Os paradigmas que uma linguagem suporta influenciam diretamente na forma de pensar do programador

- O que uma linguagem de programação *precisa* ter?
 - Estruturas de repetição (for, while, etc)? Estruturas de controle (if)? Atribuição de variáveis? Recursão?
 - Linguagens diferentes possuem paradigmas diferentes, abstrações diferente, e, conseqüentemente, formas diferentes de resolver o mesmo problema
- O que são paradigmas?
 - Ao longo da história, formas totalmente diferentes de se programar foram descobertas
 - Paradigmas são uma forma de classificar linguagens de programação por suas características e abstrações
 - Os paradigmas que uma linguagem suporta influenciam diretamente na forma de pensar do programador

- O que uma linguagem de programação *precisa* ter?
 - Estruturas de repetição (for, while, etc)? Estruturas de controle (if)? Atribuição de variáveis? Recursão?
 - Linguagens diferentes possuem paradigmas diferentes, abstrações diferente, e, conseqüentemente, formas diferentes de resolver o mesmo problema
- O que são paradigmas?
 - Ao longo da história, formas totalmente diferentes de se programar foram descobertas
 - Paradigmas são uma forma de classificar linguagens de programação por suas características e abstrações
 - Os paradigmas que uma linguagem suporta influenciam diretamente na forma de pensar do programador

- O que uma linguagem de programação *precisa* ter?
 - Estruturas de repetição (for, while, etc)? Estruturas de controle (if)? Atribuição de variáveis? Recursão?
 - Linguagens diferentes possuem paradigmas diferentes, abstrações diferente, e, conseqüentemente, formas diferentes de resolver o mesmo problema
- O que são paradigmas?
 - Ao longo da história, formas totalmente diferentes de se programar foram descobertas
 - Paradigmas são uma forma de classificar linguagens de programação por suas características e abstrações
 - Os paradigmas que uma linguagem suporta influenciam diretamente na forma de pensar do programador

- Conforme Van Roy [1], podemos dividir as linguagens de programação pelo seu controle de estado: linguagens imperativas e linguagens declarativas
- Imperativas
 - Linguagens imperativas usam comandos para alterar o estado do programa sequencialmente e explicitamente
 - Dentre elas, podemos citar:
 - Linguagens procedurais
 - Linguagens orientadas a objetos
- Declarativas
 - Linguagens declarativas expressam a lógica de uma computação sem serem explícitas quanto ao fluxo de controle
 - Dentre elas, podemos citar:
 - Linguagens lógicas
 - Linguagens reativas
 - Linguagens funcionais e puramente funcionais

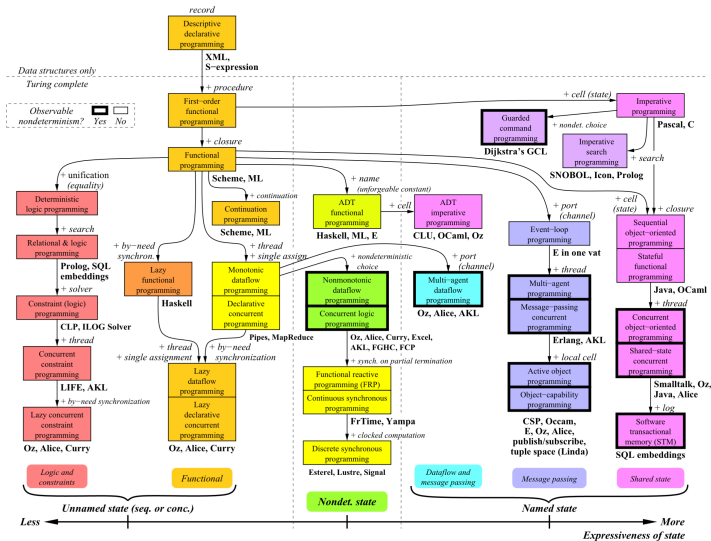
- Conforme Van Roy [1], podemos dividir as linguagens de programação pelo seu controle de estado: linguagens imperativas e linguagens declarativas
- Imperativas
 - Linguagens imperativas usam comandos para alterar o estado do programa sequencialmente e explicitamente
 - Dentre elas, podemos citar:
 - Linguagens procedurais
 - Linguagens orientadas a objetos
- Declarativas
 - Linguagens declarativas expressam a lógica de uma computação sem serem explícitas quanto ao fluxo de controle
 - Dentre elas, podemos citar:
 - Linguagens lógicas
 - Linguagens reativas
 - Linguagens funcionais e puramente funcionais

- Conforme Van Roy [1], podemos dividir as linguagens de programação pelo seu controle de estado: linguagens imperativas e linguagens declarativas
- Imperativas
 - Linguagens imperativas usam comandos para alterar o estado do programa sequencialmente e explicitamente
 - Dentre elas, podemos citar:
 - Linguagens procedurais
 - Linguagens orientadas a objetos
- Declarativas
 - Linguagens declarativas expressam a lógica de uma computação sem serem explícitas quanto ao fluxo de controle
 - Dentre elas, podemos citar:
 - Linguagens lógicas
 - Linguagens reativas
 - Linguagens funcionais e puramente funcionais

- Conforme Van Roy [1], podemos dividir as linguagens de programação pelo seu controle de estado: linguagens imperativas e linguagens declarativas
- Imperativas
 - Linguagens imperativas usam comandos para alterar o estado do programa sequencialmente e explicitamente
 - Dentre elas, podemos citar:
 - Linguagens procedurais
 - Linguagens orientadas a objetos
- Declarativas
 - Linguagens declarativas expressam a lógica de uma computação sem serem explícitas quanto ao fluxo de controle
 - Dentre elas, podemos citar:
 - Linguagens lógicas
 - Linguagens reativas
 - Linguagens funcionais e puramente funcionais

- Conforme Van Roy [1], podemos dividir as linguagens de programação pelo seu controle de estado: linguagens imperativas e linguagens declarativas
- Imperativas
 - Linguagens imperativas usam comandos para alterar o estado do programa sequencialmente e explicitamente
 - Dentre elas, podemos citar:
 - Linguagens procedurais
 - Linguagens orientadas a objetos
- Declarativas
 - Linguagens declarativas expressam a lógica de uma computação sem serem explícitas quanto ao fluxo de controle
 - Dentre elas, podemos citar:
 - Linguagens lógicas
 - Linguagens reativas
 - Linguagens funcionais e puramente funcionais

Paradigmas



Objetivos da matéria

- O objetivo principal da matéria é uma visão geral de linguagens de programação
 - Sistemas de tipos
 - Abstrações comuns
 - Uma visão comparativa dos principais paradigmas
 - Compreensão de como o mesmo problema pode ser resolvido com estilos de programação diferentes
- Foco maior em programação puramente funcional

Objetivos da matéria

- O objetivo principal da matéria é uma visão geral de linguagens de programação
 - Sistemas de tipos
 - Abstrações comuns
 - Uma visão comparativa dos principais paradigmas
 - Compreensão de como o mesmo problema pode ser resolvido com estilos de programação diferentes
- Foco maior em programação puramente funcional

Objetivos da matéria

- O objetivo principal da matéria é uma visão geral de linguagens de programação
 - Sistemas de tipos
 - Abstrações comuns
 - Uma visão comparativa dos principais paradigmas
 - Compreensão de como o mesmo problema pode ser resolvido com estilos de programação diferentes
- Foco maior em programação puramente funcional

Objetivos da matéria

- O objetivo principal da matéria é uma visão geral de linguagens de programação
 - Sistemas de tipos
 - Abstrações comuns
 - Uma visão comparativa dos principais paradigmas
 - Compreensão de como o mesmo problema pode ser resolvido com estilos de programação diferentes
- Foco maior em programação puramente funcional

Objetivos da matéria

- O objetivo principal da matéria é uma visão geral de linguagens de programação
 - Sistemas de tipos
 - Abstrações comuns
 - Uma visão comparativa dos principais paradigmas
 - Compreensão de como o mesmo problema pode ser resolvido com estilos de programação diferentes
- Foco maior em programação puramente funcional

Objetivos da matéria

- O objetivo principal da matéria é uma visão geral de linguagens de programação
 - Sistemas de tipos
 - Abstrações comuns
 - Uma visão comparativa dos principais paradigmas
 - Compreensão de como o mesmo problema pode ser resolvido com estilos de programação diferentes
- Foco maior em programação puramente funcional

Roy, P. V.

Programming paradigms for dummies: What every programmer should know, 2009.

