

Programação I

Jordana S. Salamon

jssalamon@inf.ufes.br

jordanasalamon@gmail.com

DEPARTAMENTO DE INFORMÁTICA
CENTRO TECNOLÓGICO
UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

Introdução

▶ Etapas de programação:

- ▶ entendimento do problema a ser resolvido,
- ▶ planejamento de uma solução,
- ▶ formalização da solução usando uma **linguagem de programação**,
- ▶ verificação da conformidade da solução obtida com o problema proposto.



Introdução

- ▶ Para que serve programação?
 - ▶ Resolver algum problema computacionalmente.
 - ▶ Desenvolver um programa de computador (solução) cuja execução leve a um resultado (aceitável) de um dado problema.
 - ▶ * Nem todos os problemas têm solução computacional viável
- ▶ Entender a natureza do problema e estar atento aos requisitos da solução desejada ****
 - ▶ Escolha do *paradigma de programação*;
 - ▶ Escolha da *linguagem de programação*;

Paradigmas de Programação

▶ Visões sobre a estruturação da solução do problema.

- ▶ **Prog. Funcional:** consiste em representar a solução em forma de funções (matemáticas).
- ▶ **Prog. Procedural:** consiste em representar a solução em forma de uma série de ações/procedimentos que são executados sequencialmente para produzir a solução.
- ▶ **Prog. Orientada a Objetos:** consiste em representar a solução em forma de objetos que enviam e recebem mensagens entre si e assim produzem a solução.
- ▶ **Prog. em Lógica:** consiste em representar o problema em forma de lógica simbólica e usar um processo de inferência para produzir resultados.

Funcional X Prog. Procedural

- ▶ Desvinculada da arquitetura de Von Neumann
 - ▶ Ligada à representação matemática do problema, particularmente ao uso de funções e composição de funções.
- ▶ Ligada à arquitetura de Von Neumann: execução sequencial de instruções



Funcional X

- ▶ Descreve *o que* está sendo computado.

```
fatorial :: Int → Int
fatorial 0 = 1
fatorial n = n*fatorial(n-1)
```

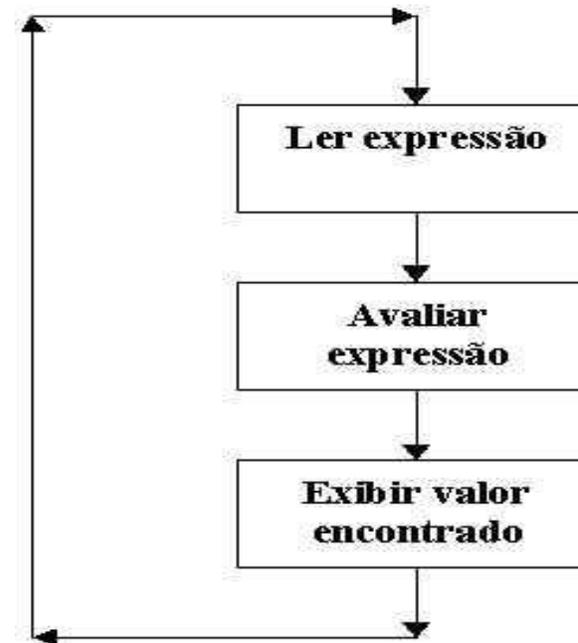
Prog. Procedural

- ▶ Descreve *como* computar (manipulando variáveis)

```
Fatorial(n):
begin
  a := 1;
  while n>0 do
  begin a := a*n;
        n := n-1;
  end;
  return n;
end;
```

Programação Funcional

<i>usuário:</i>	$6 + 4 / 2$
<i>Máquina funcional:</i>	8
<i>usuário:</i>	$f(x, y) = (x + y) / 2$
<i>Máquina funcional:</i>	definição de f foi aceita
<i>usuário:</i>	$f(6, 4) + f(10, 40)$
<i>Máquina funcional:</i>	30



Funções

- ▶ Uma associação entre elementos de dois conjuntos A e B de tal forma que para cada elemento de A existe apenas um elemento de B associado
- ▶ Uma função que associa os elementos de um conjunto A aos elementos de um conjunto consiste em um conjunto de pares ordenados onde o primeiro elemento do par pertence a A o segundo a B
- ▶ Seja a função T que associa as vogais do alfabeto com os cinco primeiros inteiros positivos.
 - ▶ $T = \{(a,1), (e,2), (i,3), (o,4), (u,5)\}$
- ▶ Seja a função Q, que associa a cada número natural o seu quadrado.
 - ▶ $Q = \{(0,0), (1,1), (2,4), (3,9), (4,16), \dots\}$

Descrição Funcional

- ▶ Extensional: os elementos são explicitados
 - ▶ $Q = \{(0,0), (1,1), (2,4), (3,9), (4,16), \dots\}$
- ▶ Intensional: os elementos são descritos através de um critério de pertinência
 - ▶ $Q = \{(x, y) \mid x \text{ é natural e } y = x.x\}$

Linguagem Python

▶ Histórico

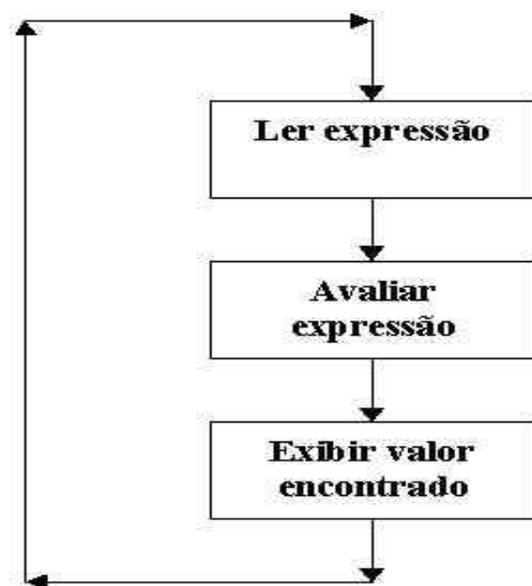
- ▶ Python é uma linguagem de programação de alto nível criada em 1991 Guido van Rossum.
- ▶ Atualmente possui um modelo de desenvolvimento comunitário, aberto e gerenciado por uma organização sem fins lucrativos.

▶ Características Principais

- ▶ Multiparadigmada
- ▶ Prioriza legibilidade do código sobre a velocidade ou expressividade
- ▶ Combina uma sintaxe concisa e clara com os recursos poderosos



Linguagem Python e ambiente IDLE



- Utilizaremos a o ambiente IDLE como avaliador/interpretador de programas computacionais escritos na linguagem Python segundo o paradigma funcional!

Linguagem Python e ambiente IDLE

- ▶ Podemos usar o IDLE também como uma calculadora qualquer, à qual submetemos expressões que ela avalia e nos informa o valor resultante.
- ▶ Exemplo:

```
>>> 3 * 5
15
>>> 6 + 4/2
8
>>> (6 + 4) / 2
5
>>>
```



Linguagem Python e ambiente IDLE

- ▶ Expressões são aplicações de funções que manipulam valores para produzir um resultado.
 - ▶ Exemplo: a expressão $3 * 5$ é a aplicação da função de multiplicação para manipular os valores 3 e 5 e produzir o resultado 15.
- ▶ As operações aritméticas são funções matemáticas. A linguagem Python oferece operações aritméticas básicas utilizando a notação usual, ou seja, infixada (o símbolo da operação fica entre os operandos). Elas são ditas funções primitivas, ou seja, funções já oferecidas pelo ambiente.
- ▶ Além de usarmos tal ambiente para escrever expressões e solicitar ao sistema que as avalie, podemos ousar mais e usá-lo também para descrever novas funções.



Descrição de Funções em Python

► *Notação Matemática*

$espaco(v, t) = v \times t$

Notação de Python

```
def espaco(v, t): return v * t
```

	INTERFACE DA FUNÇÃO			CORPO DA DEFINIÇÃO
	nome da função	parâmetros		expressão aritmética que define a relação que há entre os parâmetros
def	espaco	(v, t)	: return	v * t

Descrição de Funções em Python

- ▶ Funções paramétricas

- ▶ `def espaco(v, t): return v * t`

- ▶ Funções não-paramétricas (constantes)

- ▶ `def pi(): return 3.1416`

Descrição de Funções em Python

▶ Funções globais

- ▶ `def quad(x): return x * x`
- ▶ `def pi(): return 3.1416`
- ▶ `def areaCirc(r): return pi() * quad(r)`

▶ Funções locais

- ▶ `def areaCirc2(r):`
 `def pi2(): return 3.1614`
 `def quad2(): return r * r`
 `return pi2() * quad2()`

No ambiente IDLE, deve-se teclar 'Enter' duas vezes ao terminar a definição de uma função.

Descrição de Funções em Python

```
▶ def f(x, y):  
    def h():  
        def g(x): return x/2  
        return g(x) + g(y)  
    return h()
```

Observe que a **identação** em Python tem significado, pois é assim que se determina o que está dentro do escopo de cada função!!

Modelos de Avaliação de Expressões

▶ $4 + 6 / 2 \Rightarrow 4 + 3 \Rightarrow 7$ ou $(4 + 6) / 2 \Rightarrow 10 / 2 \Rightarrow 5$

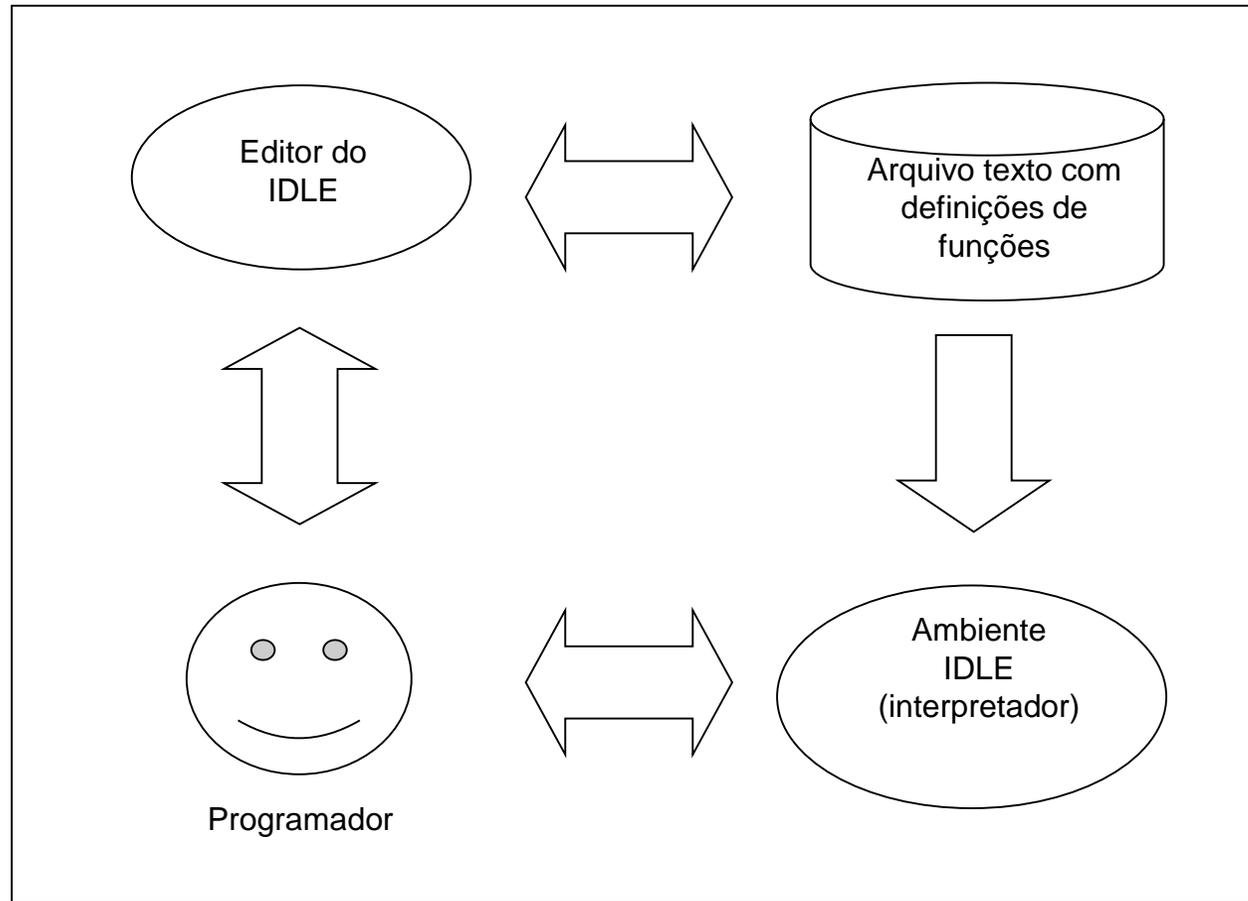
ordem	expressão	redução a ser aplicada
1	<code>areaCirc(3) + areaCirc(5)</code>	definição de <code>areaCirc</code>
2	<code>(pi() * quad(3)) + areaCirc(5)</code>	definição de <code>PI</code>
3	<code>(3.1416 * quad(3)) + areaCirc(5)</code>	definição de <code>quad</code>
4	<code>(3.1416 * 9) + areaCirc(5)</code>	definição de <code>areaCirc</code>
5	<code>(3.1416 * 9) + (pi() * quad(5))</code>	definição de <code>pi</code>
6	<code>(3.1416 * 9) + (3.1416 * quad(5))</code>	definição de <code>quad</code>
7	<code>(3.1416 * 9) + (3.1416 * 25)</code>	operação *
8	<code>28.4526 + (3.1416 * 25)</code>	operação *
9	<code>28.4526 + 79.035</code>	operação +
10	<code>107.4876</code>	

Arquivo-fonte ou Script Python

- ▶ As definições de funções realizadas no ambiente IDLE são temporárias, ou seja, se ele for reiniciado as definições serão perdidas.
- ▶ Assim, uma alternativa a isto é a criação de um arquivo-fonte ou script com as definições, de forma que essas sejam persistidas e possam ser reutilizadas sempre que necessário. Cada arquivo pode ter uma ou mais definições de funções.



Arquivo-fonte ou Script Python



Bibliotecas ou módulos

- ▶ Algumas funções prontas, inclusive matemáticas, estão disponíveis em bibliotecas ou módulos que podem ser importados, ou seja, inseridos no ambiente IDLE, conforme a necessidade.
- ▶ Por exemplo, o módulo **math** oferece acesso às funções da biblioteca C para matemática.



Bibliotecas ou módulos

- ▶ `import math`
- ▶ `math.sqrt(4)`
- ▶ `2`

- ▶ `from math import *`
- ▶ `sqrt(4)`
- ▶ `2`

- ▶ `from math import sqrt, sin`
- ▶ `sqrt(4)`
- ▶ `2`

Exercícios

1. Avalie as expressões abaixo e apresente a sequência de reduções necessárias para a obtenção do termo irredutível (resultado final):
 - ▶ $15 \% 2$
 - ▶ $15 \% 2 + 6 / 3$
 - ▶ $\text{sqrt}(15 - 2*3) / (17 - 12)$
2. Escreva um arquivo texto contendo a definição das funções abaixo. Use, quando for adequado, definições locais:
 - ▶ Determinação da área de um retângulo de lados a e b
 - ▶ Determinação da área de um círculo de raio r
 - ▶ Determinação da média aritmética de três números a, b e c

That's all Folks!



nemo