



Fundamentos de Programação

Introdução à PROGRAMAÇÃO

Aula 02

Post It da Aula Passada

O Computador

- Ábaco
- Tear de Jacquard
- Máquina analítica
- Máquina de Turing
- Arquitetura de von Neumann
- Evolução dos computadores

Os Algoritmos

- Conjunto finito de regras perfeitamente definidas
- Livre de ambiguidade
- Diferentes...
 - Ordenamentos
 - Instruções

Projetando um Programa

Nosso cenário é: **desenvolvimento de software**

- Análise do sistema
- Escolha de metodologia
- Escolha da linguagem
- Determinação de bibliotecas viáveis
- ...

A programação demanda projeto e preparação!

Projetando um Programa

O primeiro passo é **DELIMITAR O PROBLEMA**

- Quais são as minhas entradas?
- Como se organiza a minha base de dados?
- Qual é a função de otimização que deve ser avaliada?
- De quanto em quanto tempo avaliar a função?
- Quais são as minhas saídas?

Projetando um Programa

O processo de abstrair um problema concreto em uma solução algorítmica é um dos mais difíceis de dominar no contexto da programação!

Pensar o mundo como uma sequência de **ações bem determinadas**

Reduzir um problema complexo em um **conjunto de problemas mais simples**

Projetando um Programa

O problema é:

“Faça um algoritmo que some dois números”

Quais são os dados envolvidos necessários (entradas)?

Quais são os processos que devem ser realizados?

Qual é a recorrência desses processos?

Quais são os resultados esperados (saídas)?

Projetando um Programa

O problema é:

“Faça um algoritmo que some dois números”

Entradas: dois números (operandos)

Processo: adição (operador)

Recorrência: não há

Saída: um número (resultado)

Algoritmo

1. Receba o primeiro número
2. Receba o segundo número
3. Execute uma adição entre os números recebidos
4. Receba o resultado
5. Indique o resultado

Projetando um Programa

O problema é:

“Faça um algoritmo que some dois números”

O problema analisado é, na verdade, bastante simples!

Apenas a saída está implícita no enunciado; as entradas e o processo são explicitamente indicados:

- “some” (processo)
- “dois números” (entradas)

Projetando um Programa

Vamos tentar algo um pouco mais difícil!

O problema é:

“Faça um algoritmo que calcule os zeros de uma equação de segundo grau ($ax^2 + bx + c$)”

Quais são os dados envolvidos necessários (entradas)?

Quais são os processos que devem ser realizados?

Qual é a recorrência desses processos?

Quais são os resultados esperados (saídas)?

Projetando um Programa

O problema é:

“Faça um algoritmo que calcule os zeros de uma equação de segundo grau ($ax^2 + bx + c$)”

Entradas: coeficientes (a, b, c)

Processo: fórmula de Bhaskara

Recorrência: não há

Saída: dois números (resultados)

Algoritmo

1. Receba o coeficiente a
2. Receba o coeficiente b
3. Receba o coeficiente c
4. Processe a fórmula de Bhaskara
5. Receba os resultados
6. Indique os resultados

Qual é o problema?

Projetando um Programa

O problema é:

“Faça um algoritmo que calcule os zeros de uma equação de segundo grau ($ax^2 + bx + c$)”

Processar a fórmula de Bhaskara não é uma operação atômica e nem trivial! Essa operação pode ser fragmentada em diversas operações mais simples.

Como podemos fazer isso?

$$x = (-b \pm \sqrt{b^2 - 4ac}) / (2a)$$

Projetando um Programa

O problema é:

“Faça um algoritmo que calcule os zeros de uma equação de segundo grau ($ax^2 + bx + c$)”

Entradas: coeficientes (a, b, c)

Processo: fórmula de Bhaskara

Recorrência: não há

Saída: dois números (resultados)

Algoritmo

4. Processe a fórmula de Bhaskara
 - a. Eleve b ao quadrado e subtraia deste a multiplicação entre 4, a e c
 - b. Processe a raiz quadrada do resultado parcial
 - c. Subtraia (some) b ao resultado parcial
 - d. Divida o resultado parcial pelo resultado da multiplicação entre 2 e a

Projetando um Programa

Note que as operações 4.c e 4.d serão executadas duas vezes, havendo uma modificação na operação 4.c a cada execução.

Por quê?

Como eu posso organizar meu algoritmo para que essa repetição modificada ocorra em 4.c e 4.d?

Algoritmo

4. Processe a fórmula de Bhaskara
 - a. Eleve b ao quadrado e subtraia deste a multiplicação entre 4 , a e c
 - b. Processe a raiz quadrada do resultado parcial
 - c. Subtraia (some) b ao resultado parcial
 - d. Divida o resultado parcial pelo resultado da multiplicação entre 2 e a

Mas... fica **mais**
complexo que isso?

Linguagens de Programação

Historicamente, os seres humanos enfrentam problemas complexos – e para solucionar os mesmos, diversas ferramentas são criadas!

Problema: **“Derrube uma árvore”**

- Solução #1: chutar o tronco
- Solução #2: cortar o tronco com uma pedra lascada
- Solução #3: cortar o tronco com um machado
- Solução #4: cortar o tronco com uma serra elétrica

Linguagens de Programação

Na programação, também usamos ferramentas para desenvolver programas, sendo a principal delas a...

LINGUAGEM DE PROGRAMAÇÃO

Porém, vale ressaltar: a linguagem de programação não é a programação em si!

A programação define um **processo**

A linguagem de programação estabelece um **meio**

Linguagens de Programação

Há uma infinidade de linguagens de programação no mundo:

- Existem cerca de 7000 idiomas no mundo (evolução idiomática de aproximadamente 4500 anos)
- Existem cerca de 2500 linguagens de programação no mundo (evolução formal de aproximadamente 200 anos – e de maneira mais acelerada nos últimos 65 anos)



Python: Uma Linguagem Amigável

“Currículo” do Python como linguagem de programação:

- Léxico em **inglês**
 - Sintaxe e semântica muito naturais para uma linguagem de programação
- Suporte a **diversos “modelos”** (paradigmas) de programação
- Linguagem **bastante abstraída** em relação a algumas outras
 - Fácil de manipular a memória
 - Muitos recursos de programação disponíveis

Python: Uma Linguagem Amigável

A linguagem Python não é nativamente disponibilizada em todos os sistemas operacionais.

<https://www.python.org/downloads/>

Para verificar se a linguagem está instalada, abrimos um terminal e...

`python --version`

Para iniciarmos o interpretador Python localmente na nossa máquina...

`pyhton`

Python: Uma Linguagem Amigável

Python 2 vs. Python 3

A linguagem Python tem duas versões principais: 2 e 3

Por um longo tempo, a versão padrão e mais amplamente utilizada da linguagem foi o Python 2. Porém, devido a questões como inconsistências sintáticas e melhorias de desempenho, a versão 3 assumiu o papel principal na evolução da linguagem

Nas nossas aulas, iremos utilizar o **Python 3** como a versão padrão da linguagem

Python: Uma Linguagem Amigável

Podemos programar em Python localmente, mantendo todos os arquivos e bibliotecas que serão utilizadas em nosso computador

Um bom editor: **Sublime Text**

Uma boa IDE: **PyCharm**

Também é possível utilizar sistemas disponibilizados na nuvem para programar em Python!

Uma boa alternativa: **Replit**

O Mito do *Hello World*

O **programa "Hello World!"** é conhecido e utilizado historicamente como um exemplo de código minimalista de uma linguagem de programação

Sua primeira implementação registrada para essa finalidade foi no livro "A Tutorial Introduction to the Language B" em 1996

Espontaneamente, criou-se o mito de que se o primeiro programa de um programador em uma linguagem for o "Hello World!", ele terá sorte no uso da mesma

O Mito do *Hello World*

Então, para não quebrar a tradição (e pela utilidade do comando necessário), vamos programar o “Hello World!” em Python

Este programa tem um objetivo muito simples: **mostrar a frase “Hello World!” na tela (terminal) do usuário**

Para fazer isso, vamos seguir uma sequência de passos...

O Mito do *Hello World*

Passo #1: criação de um arquivo (isolado), projeto (IDE – PyCharm) ou arquivo em nuvem (Replit) Python

- Arquivos de código da linguagem Python tem extensão “.py”
 - Porém, essa extensão não é obrigatória
- Projetos maiores em Python podem ser gerenciados por uma IDE
- Nesse caso, a própria IDE se responsabiliza pela criação e organização dos arquivos

O Mito do *Hello World*

Passo #2: Documentação (comentários em Python)

- Todo o bom algoritmo precisa de “documentação” no código
 - Comunicação entre programadores
 - Comunicação entre programadores e analista
 - Comunicação entre analista e clientes/usuários
- Podemos criar uma documentação simples em Python usando comentários
 - #: Comentários de linha
 - “””: Comentários de bloco

O Mito do *Hello World*

Passo #3: Comando de exibir na tela em Python

- Em Python, existe um comando específico para mostrar informações na tela (console) do usuário: **print**
 - O comando print é uma função (não precisamos entender esse conceito agora, apenas conhecer o fato)
- Toda a função em Python é executada através da escrita de seu nome e da inserção dos seus delimitadores de argumentos: **print()**

O Mito do *Hello World*

Passo #3: Comando de exibir na tela em Python

- Uma função pode ou não receber argumentos entre seus delimitadores
 - No caso da função *print*...
 - `print()`: mostra uma linha vazia na tela (terminal)
 - `print("alguma coisa")`: mostra a frase alguma coisa na tela (terminal)
 - Uma cadeia de caracteres é definida entre aspas (" ") ou entre aspas simples (' ')

O Mito do *Hello World*

Passo #3: Comando de exibir na tela em Python

- Um *print* constitui uma linha de código do programa, portanto a próxima linha de código pode estar...
 - no mínimo, uma linha abaixo
 - ao lado da linha atual, desde que utilizada a demarcação de fim de linha “;”

```
print("linha 1")  
print("linha 2")
```

FUNCIONA!

```
print("linha 1"); print("linha 2")
```

FUNCIONA!

```
print("linha 1") print("linha 2")
```

NÃO FUNCIONA!

O Mito do *Hello World*

Passo #3: Comando de exibir na tela em Python

- Para o nosso programa “Hello World!”, porém, basta que uma linha seja escrita na tela. Portanto:

```
print(“Hello World!”)
```

O Mito do *Hello World*

Passo #4: Executando o código em Python

- Existem duas possíveis maneiras de executar um código em Python através dos nossos recursos...
 - Execução via Terminal
 - Abra o console, navegue até a pasta do arquivo .py (por exemplo, programa.py) e digite “python programa.py”
 - Execução via IDE ou ReplIt
 - Abra o projeto, clique no botão de executar (*Run*)

O Mito do *Hello World*

Passo #5 (extra): Executando o “Hello World!” no Python Runtime

- O Python contém um módulo de Runtime para a execução direta de linhas de código
 - É um módulo muito útil para fazer testes
- Para entrar no Runtime, abra o terminal do computador e execute “python”
 - Pronto! Já podemos executar linhas de código
 - Faça o teste imprimindo “Hello World!” na tela (terminal)

Exercício #02

Arroz
Lentilha
Café (pó)
Café (solúvel)
Chocolate
Goma de tapioca
Presunto
Queijo
Amendoim

Considere a lista de mercado ao lado e faça um programa em Python que, quando executado, **mostre todos os itens na tela**

- a) Utilize o modelo de uma linha de código por linha do programa
- b) Utilize apenas uma linha no programa



Fundamentos de Programação

Aula 02

Obrigado e
ATÉ A
PRÓXIMA!