



Introdução à Programação

Pedro Ribeiro Queiroz de Faria

Departamento de Economia da UFPE
(Decon/UFPE)

Estrutura da apresentação

- 1 Fundamentos
- 2 Estruturas de Controle
- 3 Instalando o R

Lógica de Programação

O que será aprendido?

- Pensamento lógico: organizado, passo a passo.
- Programação: dar instruções a um computador.
- Lógica de Programação: base para qualquer linguagem.

Algoritmos:

Definição

Sequência finita e ordenada de passos para resolver um problema.

Características:

- Finito
- Bem definido
- Eficaz
- Apresenta entrada e saída

Exemplo: Trocar uma lâmpada, fazer uma média de dois números.

Variáveis e Tipos de Dados

Conceito

Quando definimos uma variável e seu tipo, criamos uma "caixinha" na memória do computador onde ele guardará a variável.

Regras:

- Não declare variáveis que não irá utilizar
- Coloque um nome de fácil identificação
- Sempre sem espaços

Tipos Primitivos

- **Inteiro:** Números sem casas decimais (idade, quantidade)
- **Real:** Números com casas decimais (altura, preço, média)
- **Caractere:** Letras, palavras, frases (nome, endereço)
- **Lógico:** Verdadeiro ou falso (decisão)

Declaração e atribuição:

```
int x
x = 10 ou x < -10
real y = 7.5
```

Operadores

- **Aritméticos:** $+$, $-$, $*$, $/$
- **Relacionais:** $>$, $<$, $>=$, $<=$
- **Outros Relacionais:** $=$, $==$, $<>$, $!=$ (Resultam no tipo lógico)
- **Lógicos:** $E ()$, $OU(—)$, NAO (expressões lógicas)

Entrada e Saída

Conceito

A interação do programa com o usuário.

- **Saída:** Mostrar informações na tela (Escreva, Read)
- **Entrada:** Pedir informações ao usuário (Leia, Input, Readline)

Exemplo 1

- Crie um programa que calcule uma média simples entre duas notas. Valores que você desejar.

Resolução

```
programa{  
  
  funcao inicio(){  
  
    real nota1  
    real nota2  
    real media  
  
    nota1 = 10  
    nota2 = 8  
  
    media = (nota1 + nota2)/2  
  
    escreva("A média foi:", media)  
  
  }  
  
}
```

Figure: Caption

Estruturas Condicionais

Conceito

Permitir que o programa tome decisões e siga caminhos diferentes diante de condições

- If/else
- Switch/case

If/else

- **Simples:** Se (condição) então {bloco de comandos}
- **Composto:** Se (condição) então {bloco1} Senão se {bloco2} Senão {bloco3}
- **Aninhado:** Um se dentro de outro se

Exemplo 2

- Faça um programa que peça ao usuário duas notas e, se a nota for maior que 7, responda que ele foi aprovado. Se for menor que 7 e maior que 5, diga que ele foi aprovado por pena. Se menor que 5, diga que foi reprovado.

Resolução 2

```
programa{  
  
  funcao inicio(){  
  
    real nota1, nota2, media  
  
    escreva("Coloque sua primeira nota:\n")  
    leia(nota1)  
  
    escreva("\n Coloque sua segunda nota\n")  
    leia(nota2)  
  
    media = (nota1+nota2)/2  
  
    se(media >= 7) {  
      | escreva("\nAprovado")  
    }  
  
    senao se(media >=5 e media < 7){  
      | escreva("\n Passou no choro")  
    }  
  
    senao{  
      | escreva("\nReprovado")  
    }  
  
  }  
}
```

Switch/case

- **Uso:** Quando há multiplas escolhas baseadas em um valor.
- **Sintaxe:** Escolha (variável1) caso <valor1> <comandos>
<pare> caso <valor2> <comandos> <pare> caso contrario
<comandos>
- Bem menos utilizado do que o if/else

Exemplo 3

Peça ao usuário informar se ele é maior de idade ou não. Caso sim, escreva que ele pode tirar carteira de habilitação. Caso contrário, que ele não pode.

Resolução 3

```
programa {  
  funcao inicio() {  
  
    inteiro confirmacao  
  
    escreva("Você é maior de idade? Digite 1 caso sim ou 2 caso nao\n")  
    leia(confirmacao)  
  
    escolha(confirmacao){  
      caso 1:  
        escreva("Você pode tirar carteira de habilitação")  
        pare  
  
      caso 2:  
        escreva("Você não pode tirar carteira de habilitação")  
        pare  
  
      caso contrario:  
        escreva("Tã de sacanagem")  
        pare  
    }  
  }  
}
```

Estruturas de Repetição (Laços/Loops)

Conceito

Executar um bloco de comando várias vezes até ser atingida certa condição.

Componentes:

- Condição de parada
- Contador (na maioria das vezes)
- Corpo do laço

Tipos:

- While
- Do/while
- For

Do (Enquanto))

- Testa a condição antes de executar. Se for verdadeira, executa e testa de novo.
- **Sintaxe:** Enquanto (Condição) {Comandos}

Do/While (Faça/Enquanto)

- Executa ao menos uma vez e testa a condição depois
- **Sintaxe:** Faça {Comandos} enquanto (condição)

For (Para)

- Outra forma de se usar as repetições
- **Sintaxe:** para(variável, condição de parada, mudança de variável) {Comandos}

Exemplo 4

Faça um algoritmo que some os números de 1 a 100 usando a estrutura de repetição de sua preferência.

Resolução 4

```
1  programa {
2      funcao inicio() {
3
4          inteiro n = 1
5
6          faca{
7              escreva(n, " ")
8              n += 1
9          } enquanto(n <= 100)
10
11      }
12  }
13
```

Passo a Passo da instalação do R

- 1 Abra este link: Clique aqui
- 2 Clique em Download and Install R
- 3 Clique no download que tem o seu sistema operacional
- 4 Clique em Install R for the first time
- 5 Execute o programa que baixou. Será um executável.
- 6 Prossiga com a instalação normalmente.
- 7 Seu R está baixado.

Passo a passo do R Studio

- 1 Volte no mesmo site do link acima
- 2 Clique em Download R STUDIO DESKTOP FOR WINDOWS (ou MAC ou LINUX)
- 3 Abra o arquivo baixado
- 4 Prossiga com a instalação normalmente
- 5 Aceite tudo, dê ok em tudo.
- 6 Abra o R Studio