

Técnicas de Desenvolvimento de Algoritmos



- ✓ Operadores Relacionais e Lógicos
- ✓ Decisão simples, composta e aninhada

Na aula passada...

Métodos

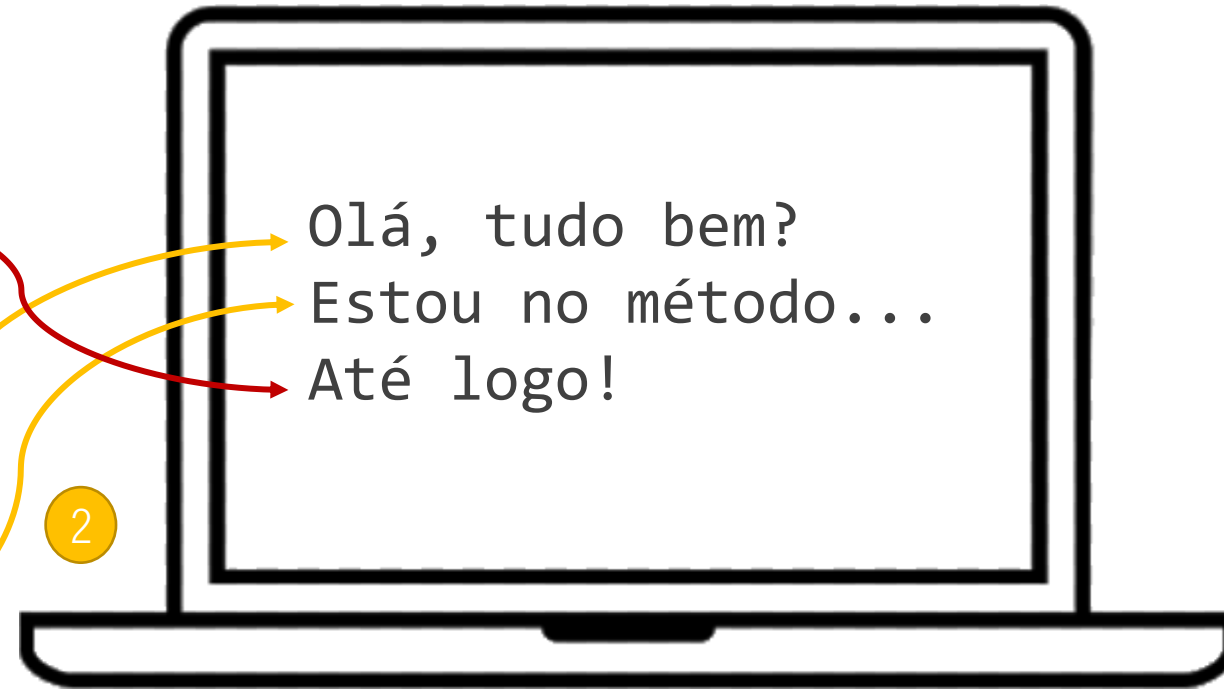
- ✔ Um algoritmo pode ser simplificado quando dividido em várias sub-rotinas (métodos). Os métodos podem ser classificados em: **procedimentos** (sem retorno de valor) e **funções** (com retorno de valor).
- ✔ Quando um método é chamado por um algoritmo, ele é executado e ao seu término o controle de processamento retorna automaticamente para a primeira linha de instrução após a linha que efetuou a chamada do método.



Exemplo – utilizando um método

```
algoritmo exemplo1
início
  0  exibirMensagens()
    escreva ("Até logo!")
fim
void exibirMensagens ()
  início
    escreva ("Olá, tudo bem?")
    escreva ("Estou no método...")
  fim
```

Na tela:



Este método é um procedimento, porque não retorna um valor.

Exemplo de aplicação

Construir um algoritmo que calcule e visualize na tela os valores de seno, cosseno e tangente de um ângulo fornecido em graus pelo usuário. Deve existir um método com retorno que converta o ângulo de graus para radianos antes de calcular o seno, cosseno e tangente.

Utilize um método que calcule e visualize estes três resultados.

Resposta: sen, cos, tan, utilizando procedimento

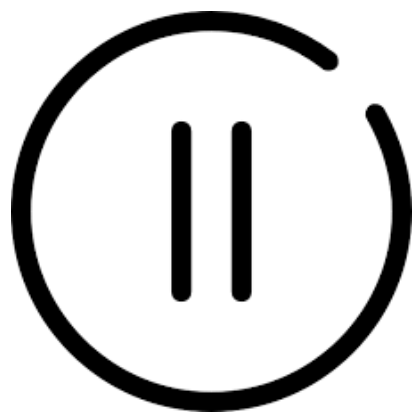
```
algoritmo Calculo_de_seno_cosseno_tangente
início
    real ang
    escreva ("Digite o valor do ângulo em graus: ")
    leia (ang)
    visualiza(convertGToR(ang))
fim
```

```
void visualiza (real angulo)
início
    escreva ("Seno: " + sen(angulo) )
    escreva ("Cosseno: " + cos(angulo) )
    escreva ("Tangente: " + tan(angulo) )
fim
```

```
real convertGToR(real angulo)
início
    retorne angulo*PI/180
fim
```

Antes de continuarmos...

Pausa para reflexão!!!



Introdução às decisões

Tomadas de decisão são importantes nas atividades profissionais, na vida, em programação...



www.peytonbolin.com



<https://www.sbcoaching.com.br/>



<https://consultormarketing.digital/>



Em programação, as decisões devem estar bem definidas: não podem ser ambíguas.

Introdução às decisões

Em programação, as decisões devem estar bem definidas: não podem ser ambíguas.



Operadores Relacionais

| Operador | Operador em Java | Exemplo | Significado |
|------------------|--------------------|------------------------|------------------------|
| Igualdade | <code>==</code> | <code>x == y</code> | x é igual a y |
| Diferença | <code>!=</code> | <code>x != y</code> | x é diferente de y |
| Maior que | <code>></code> | <code>x > y</code> | x é maior que y |
| Maior ou igual a | <code>>=</code> | <code>x >= y</code> | x é maior ou igual a y |
| Menor que | <code><</code> | <code>x < y</code> | x é menor que y |
| Menor ou igual a | <code><=</code> | <code>x <= y</code> | x é menor ou igual a y |

Comparações só podem ser feitas entre objetos de mesma natureza, isto é, variáveis do mesmo tipo de dado. O resultado de uma comparação será sempre um valor lógico.

Operadores Relacionais

| Operador | Significado | Exemplo | Resultado |
|----------|----------------|---------|------------|
| == | igual a | 5 == 5 | verdadeiro |
| | | 5 == 8 | falso |
| != | diferente de | 5 != 8 | verdadeiro |
| | | 5 != 5 | falso |
| > | maior que | 8 > 5 | verdadeiro |
| | | 5 > 8 | falso |
| < | menor que | 5 < 8 | verdadeiro |
| | | 8 < 5 | falso |
| >= | maior ou igual | 8 >= 5 | verdadeiro |
| | | 5 >= 8 | falso |
| <= | menor ou igual | 5 <= 8 | verdadeiro |
| | | 8 <= 5 | falso |

Comparações só podem ser feitas entre objetos de mesma natureza, isto é, variáveis do mesmo tipo de dado. O resultado de uma comparação será sempre um valor lógico.



Operadores Lógicos

| Operadores | Python | Java |
|------------|--------|------|
| ou | or | |
| e | and | && |
| não | not | ! |

| A | B | A or B | A and B | not A |
|---|---|--------|---------|-------|
| F | F | F | F | V |
| F | V | V | F | V |
| V | F | V | F | F |
| V | V | V | V | F |

- ✓ or → basta que um dos seus valores seja V para que o resultado seja V.
- ✓ and → é necessário que todos os valores sejam V para que o resultado seja V.



Operadores Lógicos e Relacionais

Exemplo: Considere $A = 10$, $B = 5$, $C = 7$, $D = 3$, $F = 5$

a) $A > C$ \longrightarrow Verdadeiro

b) $A < B$ \longrightarrow Falso

c) $A == D$ \longrightarrow Falso

d) $A != C$ \longrightarrow Verdadeiro

e) $B <= F$ \longrightarrow Verdadeiro

f) $A >= D$ \longrightarrow Verdadeiro

g) $A > B$ e $C > D$ \longrightarrow Verdadeiro

h) $C > B$ ou $A == D$ \longrightarrow Verdadeiro

i) $C == B$ ou $A == D$ \longrightarrow Falso

j) $!(A == D)$ \longrightarrow Verdadeiro

k) $!(B == F)$ \longrightarrow Falso

l) $C > B$ e $A == D$ \longrightarrow Falso

Prioridade dos grupos de operadores

| Operadores | Prioridade |
|-------------|----------------|
| Lógicos | 4 ^o |
| Relacionais | 3 ^o |
| Aritméticos | 2 ^o |
| Parênteses | 1 ^o |

menor



maior

Estrutura Sequencial

Nessa estrutura os comandos de um algoritmo são executados numa sequência pré-estabelecida. Cada comando é executado somente após o término do comando anterior. Veja o exemplo a seguir:

Exemplo.alg

```
1  algoritmo exibeIdade
2  inicio
3      inteiro idade
4      escreva("Digite a sua idade")
5      leia(idade)
6      escreva("A idade é: " + idade)
7  fim
8
```



Estruturas de decisão

Utilizadas para controlar o fluxo de execução dos programas, possibilitando que caminhos alternativos sejam seguidos de acordo com resultado de uma condição ou mais condições.

As estruturas de decisão existentes são:

- ✓ a estrutura **se (if)**
- ✓ a estrutura **se-senão (if-else)**
- ✓ a estrutura **escolha-caso (switch-case)**



Estrutura de decisão simples: SE

- ✔ Permite avaliar condições para que um determinado bloco de código seja executado ou não.
- ✔ Avalia uma expressão lógica e se o resultado da avaliação for verdadeiro (true) o bloco de código inserido nela será executado.

Exemplo.alg

```
1 se (nota >= 6)
2   escreva("Passou")
3
```

A **condição** é uma expressão lógica e, portanto, sempre resulta um valor lógico **F** ou **V**.

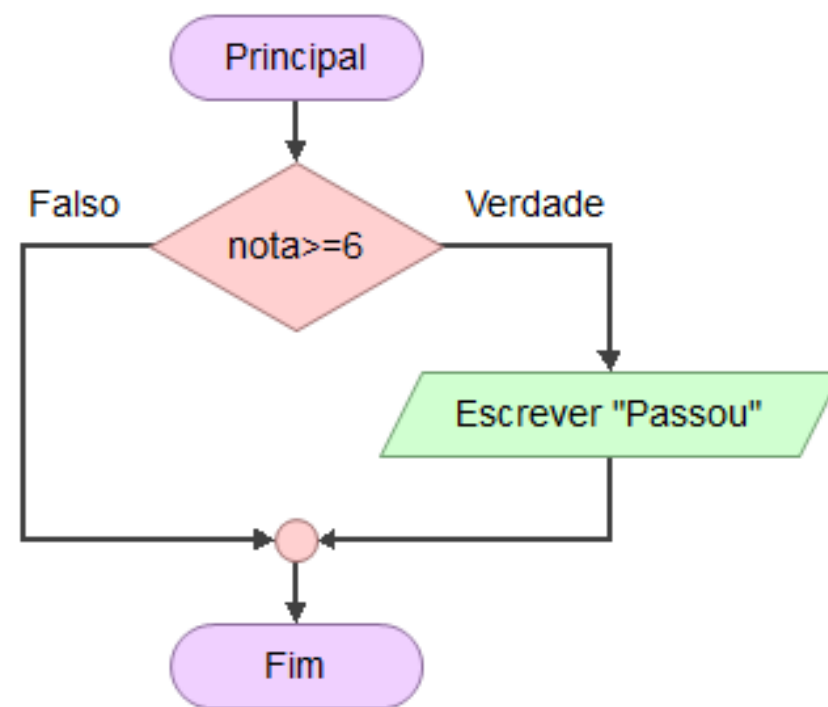
Executa as **<instruções>**
se a condição for verdadeira

Estrutura de decisão simples: SE

- ✔ Permite avaliar condições para que um determinado bloco de código seja executado ou não.
- ✔ Avalia uma expressão lógica e se o resultado da avaliação for verdadeiro (true) o bloco de código inserido nela será executado.

Exemplo.alg

```
1 se (nota >= 6)
2   escreva("Passou")
3
```



Exemplos de aplicação

1- Escreva um algoritmo que solicite um número inteiro ao usuário e mostre-o caso o mesmo seja par.

Exemplo1.alg

```
1  algoritmo par
2  inicio
3      inteiro num
4      escreva ("Digite um número inteiro")
5      leia (num)
6      se (num%2==0) {
7          escreva ("O número: " + num + "é par!")
8      }
9  fim
```



Pausa para programação

1- Escreva um algoritmo que solicite um número inteiro ao usuário e mostre-o caso o mesmo seja par.

```
# Exemplo1: programa que solicita um número  
# inteiro ao usuário e mostre-o caso  
# o mesmo seja par.  
  
num = int(input("Entre com um número inteiro"))  
if num % 2 == 0:  
    print("O número: ", num , " é par.")
```



Pausa para programação

1- Escreva um algoritmo que solicite um número inteiro ao usuário e mostre-o caso o mesmo seja par.

```
//Exemplo1: programa Java que verifica se o número é par.  
import java.util.Scanner;  
public class Exemplo1 {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Digite um número inteiro");  
        int x = sc.nextInt();  
        if(x % 2 == 0){  
            System.out.println("O número " + x + " é par");  
        }  
        sc.close();  
    }  
}
```



Estrutura de decisão composta: SE...SENÃO

- ✓ Permite avaliar condições para que um determinado bloco de código seja executado ou não.
- ✓ Avalia uma expressão lógica e se o resultado da avaliação for verdadeiro (true) será executado o primeiro bloco de código, caso contrário o bloco do senão.

```
Exemplo.alg
1 se (nota >= 6)
2   escreva("Passou")
3 senao
4   escreva("Reprovou")
```

Executa as <instruções 1>
se a condição for verdadeira

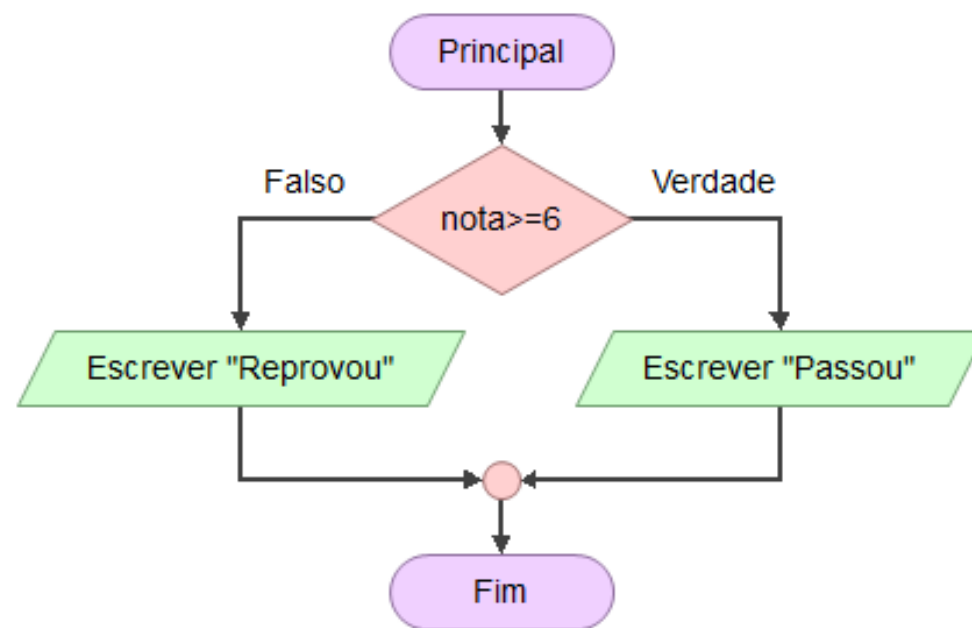
senão **não** avalia
condição!

e se for falsa executa
as <instruções 2>

Estrutura de decisão composta: SE...SENÃO

- ✓ Permite avaliar condições para que um determinado bloco de código seja executado ou não.
- ✓ Avalia uma expressão lógica e se o resultado da avaliação for verdadeiro (true) será executado o primeiro bloco de código, caso contrário o bloco do senão.

```
Exemplo.alg
1 se (nota >= 6)
2   escreva("Passou")
3 senao
4   escreva("Reprovou")
```



Exemplos de aplicação

2- Implementar um algoritmo que obtenha a idade e mostre se a pessoa é maior ou menor de idade

```
algoritmo idade_maior
início
    inteiro idade
    escreva ("Digite sua idade")
    leia (idade)
    se (idade >= 18) {
        escreva ("Maior de idade")
    }
    senão {
        escreva ("Menor de idade")
    }
fim
```

Bloco verdadeiro

Bloco falso



Pausa para programação

2- Implementar um algoritmo que obtenha a idade e mostre se a pessoa é maior ou menor de idade

```
# Exemplo3: programa que obtém a idade e mostra  
# se a pessoa é maior ou menor de idade  
  
idade = int(input("Digite sua idade: "))  
if idade >= 18:  
    print("Maior de idade")  
else:  
    print("Menor de idade")
```



Pausa para programação

2- Implementar um algoritmo que obtenha a idade e mostre se a pessoa é maior ou menor de idade

```
//Exemplo1: programa que obtém a idade e mostra
// se a pessoa é maior ou menor de idade
import java.util.Scanner;
public class Exemplo2 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Digite sua idade:");
        int x = sc.nextInt();
        if(x >=18){
            System.out.println("Maior de idade");
        } else {
            System.out.println("Menor de idade");
        }
    }
}
```



Observações importantes

- ✓ A diretiva **SE** deve ter suas expressões contidas entre parênteses.
- ✓ O único argumento válido para um **SE** é uma expressão lógica ou variável booleana (condição).
- ✓ Preste atenção nos sinais de comparação (==) dentro de um **SE**, pois eles podem ser confundidos com o operador de atribuição (=).
- ✓ As chaves não são obrigatórias para blocos **SE** que têm apenas uma instrução, mas tome cuidado com erros de endentação.

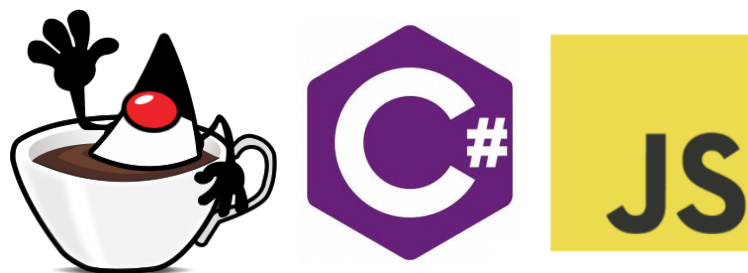


Tomada de decisão do tipo se... senão



No pseudocódigo

```
se (condição )  
    bloco de comandos  V  
senão  
    bloco de comandos  F
```



Em Java / Javascript / C#

```
if (condição) {  
    bloco de comandos V  
}  
else {  
    bloco de comandos F  
}
```



Em Python

```
if condição:  
    bloco de comandos  V  
else:  
    bloco de comandos  F
```

Exemplos de aplicação

3- Escreva um algoritmo que solicite duas notas de um aluno ao usuário, calcule a média e mostre se o mesmo está aprovado (média ≥ 6.0) ou reprovado caso contrário.

```
algoritmo media
    inicio
        real n1, n2, m
        escreva("Digite a nota 1")
        leia(n1)
        escreva("Digite a nota 2")
        leia(n2)
        m = (n1+n2)/2
        se(m >= 6){
            escreva("Aprovado")
        }
        senão{
            escreva("Reprovado")
        }
    fim
```



Estrutura se... senão aninhada

Usada quando várias condições devem ser testadas.

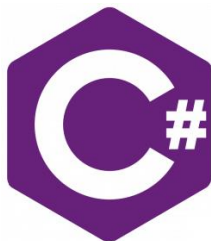
```
se(condição1){  
    instruções 1  
}  
senão{  
    se(condição2){  
        instruções 2  
    }  
    senão{  
        instruções 3  
    }  
}
```

Executa o <bloco de instruções 1>
se a **condição1** for **verdadeira**

e se for **falsa**, avalia a
condição2



Estrutura se... senão aninhada



Em Java / Javascript / C#

```
if (condição1) {  
    bloco de comandos 1  
}  
else {  
    if (condição2) {  
        bloco de comandos 2  
    }  
    else {  
        bloco de comandos 3  
    }  
}
```



Em Python

```
if condição1:  
    bloco de comandos 1  
else:  
    if condição2:  
        bloco de comandos 2  
    else:  
        bloco de comandos 3
```

Exemplos de aplicação

4- Escreva um algoritmo que solicite ao usuário a média de um aluno e o percentual de frequência e mostre a sua situação, conforme a tabela abaixo:

| Frequência | Media | Situação |
|------------|-------|---------------------|
| < 75 | | Reprovado por falta |
| >= 75 | < 6 | Reprovado por nota |
| | >= 6 | Aprovado |

Exemplo 4 (errado)

```
algoritmo media
inicio
  real m, f
  escreva("Digite a média")
  leia(m)
  escreva("Digite o percentual de frequência")
  leia(f)
  se(f < 75) {
    escreva("Reprovado por falta")
  }
  se(m < 6){
    escreva("Reprovado por nota")
  }
  senão{
    escreva("Aprovado")
  }
fim
```

O que será impresso se a média do aluno for igual a 7,0 e a frequência for 60%?



Exemplo 4.1

```
algoritmo media
inicio
    real n1, n2, m, f
    escreva("Digite a média")
    leia(m)
    escreva("Digite o percentual de frequência")
    leia(f)
    se(f < 75) {
        escreva("Reprovado por falta")
    }
    senão se(m < 6){
        escreva("Reprovado por nota")
    }
    senão{
        escreva("Aprovado")
    }
fim
```

O que será impresso se a média do aluno for igual a 7,0 e a frequência for 60%?



Exemplos de aplicação

5- Escreva um algoritmo, que faça o seguinte:

a) Obtenha do usuário seu peso e altura;

b) Calcule seu índice de massa corpórea: $IMC = \text{peso} / (\text{altura})^2$

c) Escreva a seguinte mensagem “Você está na categoria **<categoria>**”, obedecendo a tabela abaixo:

| Categoria | IMC |
|----------------|-------------------|
| Abaixo do peso | Abaixo de 20,0 |
| Peso normal | Entre 20,0 e 24,9 |
| Sobrepeso | Entre 25,0 e 29,9 |
| Obeso | Entre 30,0 e 39,9 |
| Obeso mórbido | 40,0 e acima |

Exemplo 5

```
algoritmo calculoIMC
inicio
    real peso, altura, imc
    escreva ("Digite o seu peso")
    leia (peso)
    escreva("Digite a sua altura")
    leia(altura)

    imc=peso/(altura*altura)
```



Exemplo 5 (errado)

```
se (imc < 20){  
    escreva("Você está na categoria: Abaixo do peso")  
}  
se (imc < 25){  
    escreva("Você está na categoria: Peso normal")  
}  
se (imc < 30){  
    escreva("Você está na categoria: Sobrepeso")  
}  
se (imc < 40){  
    escreva("Você está na categoria: Obeso")  
}  
senão{  
    escreva("Você está na categoria: Obeso Mórbido")  
}  
fim
```

O que será impresso se o imc for 24?



Exemplo 5.1

```
se (imc < 20){  
    escreva ("Você está na categoria: Abaixo do peso")  
}  
senão se (imc < 25){  
    escreva("Você está na categoria: Peso normal")  
}  
senão se (imc < 30){  
    escreva("Você está na categoria: Sobrepeso")  
}  
senão se (imc < 40){  
    escreva("Você está na categoria: Obeso")  
}  
senão{  
    escreva("Você está na categoria: Obeso Mórbido")  
}  
fim
```

O que será impresso se o imc for 24?



Exemplos de aplicação

6- Faça um algoritmo que solicite um código referente ao tipo da diária de hospedagem e também a quantidade de diárias desejada por um cliente. Calcule e mostre, usando estrutura condicional aninhada, o valor total a pagar pelo cliente, conforme a tabela abaixo:

| Tipo da diária | Quarto | Valor da diária |
|----------------|---------|-----------------|
| S | Simples | R\$ 255,50 |
| D | Duplo | R\$ 305,50 |
| T | Triplo | R\$ 360,50 |

Caso seja digitado um código diferente dos apresentados na tabela acima deve ser mostrada a mensagem “Tipo de diária inválido”.

Exemplo 6

```
algoritmo hospedagem
    inteiro qtDiarias
    caractere tipo
    inicio
        escreva ("Digite a quantidade de diárias: ")
        leia (qtDiarias)
        escreva ("Digite o tipo de hospedagem:\n[S]imples\n[D]upla\n[T]ripla")
        leia (tipo)
        se(tipo == 's' ou tipo == 'S')
            escreva("Valor a pagar: R$" + (qtDiarias * 255.50))
        senao se(tipo == 'd' ou tipo == 'D')
            escreva("Valor a pagar: R$" + (qtDiarias * 305.50))
        senao se(tipo == 't' ou tipo == 'T')
            escreva("Valor a pagar: R$" + (qtDiarias * 360.50))
        senao
            escreva("Tipo de hospedagem inválido")
    fim
```



Exemplo 6 - Python

main.py

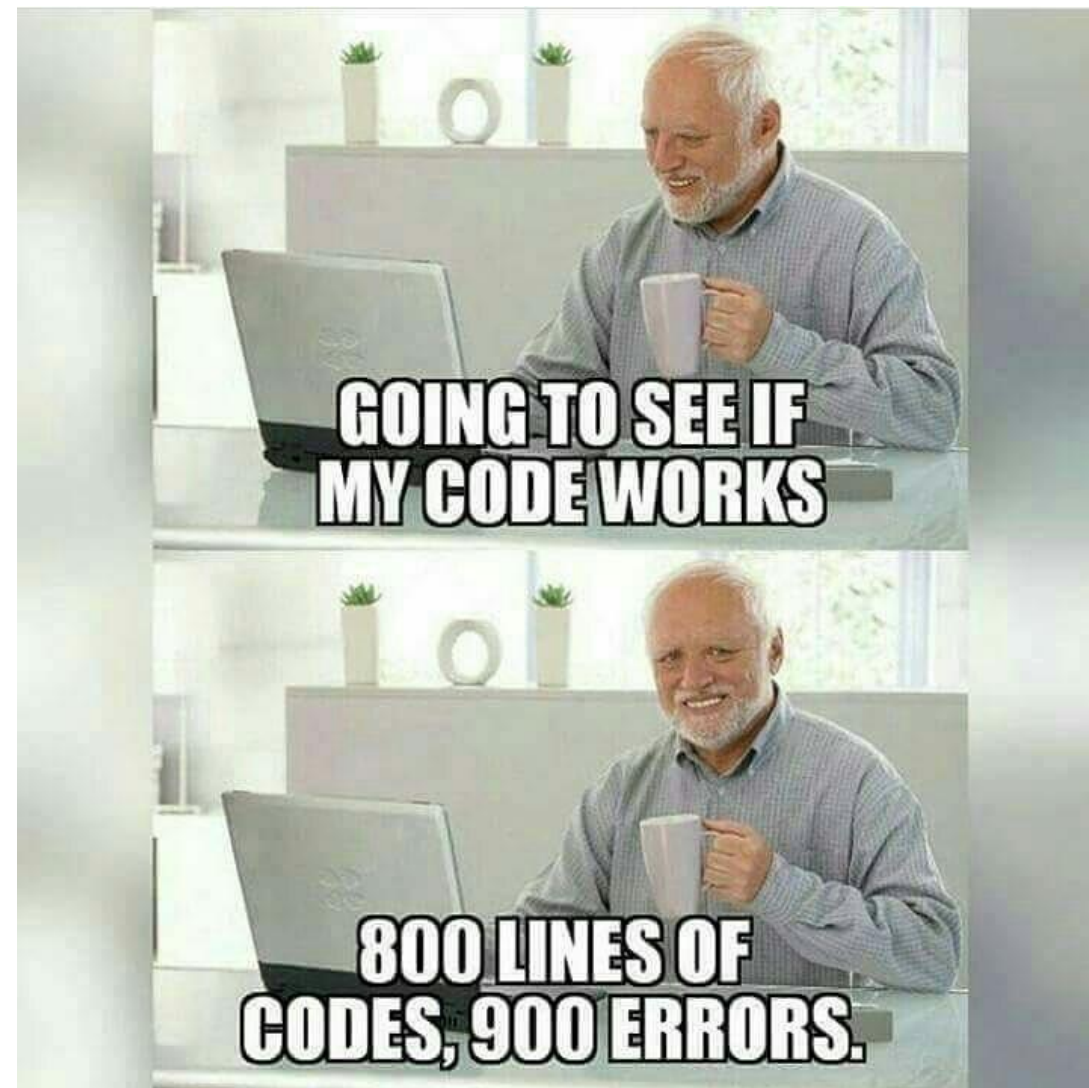
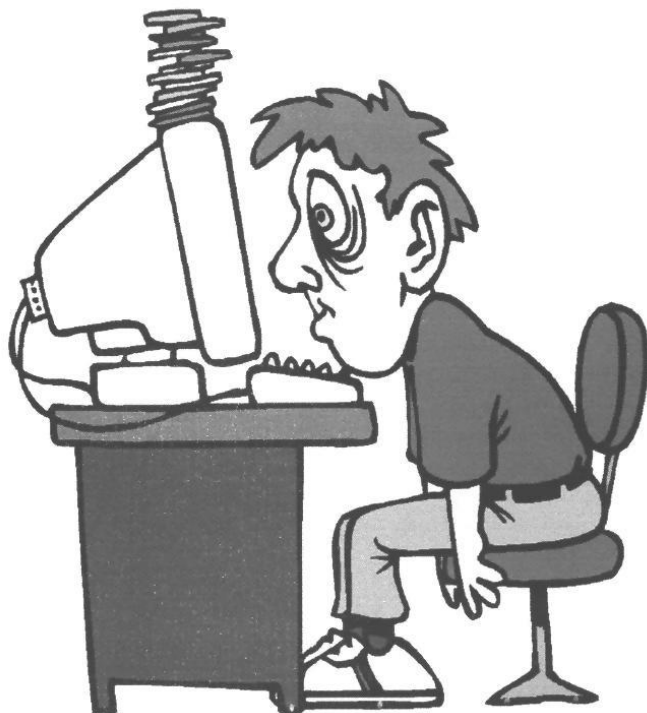
```
1 qtdDiarias = int(input("Digite a quantidade de diárias: "))
2 tipo = input("Digite o tipo de hospedagem: ")
3
4 if tipo=="s" or tipo=="S":
5     print("Valor a pagar R$ %.2f" %(qtdDiarias * 255.5))
6 elif tipo=="d" or tipo=="D":
7     print("Valor a pagar R$ %.2f" %(qtdDiarias * 305.5))
8 elif tipo=="t" or tipo=="T":
9     print("Valor a pagar R$ %.2f" %(qtdDiarias * 360.5))
10 else:
11     print("Tipo de hospedagem inválida!")
```



Alguma dúvida????



Vamos trabalhar um pouquinho?!!!



Observações sobre exercícios

- ✓ Todos os exercícios devem ser resolvidos em algoritmo (pseudocódigo).
- ✓ Cada exercício deverá ser salvo em um arquivo com a **extensão .alg** e os pseudocódigos podem ser feito no Notepad++ ou bloco de notas.
- ✓ **Não utilizar o Word ou qualquer outra ferramenta para edição de textos. Não salvar todos os exercícios em um único arquivo!!!!!!!!!!**
- ✓ Após finalizar todos os exercícios da aula, envie os arquivos e pastas pelo Blackboard (caso desejar, pode compactar tudo).



Exercícios de aplicação

- 1- Escreva um algoritmo que solicite um número ao usuário. Caso seja digitado um valor entre 0 e 9, mostre: “valor correto”, caso contrário mostre: “valor incorreto”.
- 2- Crie um algoritmo que solicite ao usuário o seu turno de trabalho e a quantidade de horas trabalhadas, calcule e mostre o valor do salário. Considere os valores de horas a seguir, de acordo com o turno de trabalho. Caso o turno seja igual a ‘N’ (utilize um caractere para representar) o valor da hora trabalhada é R\$ 45,00, caso contrário é R\$ 37,50

Exercícios de aplicação

3- Uma empresa está selecionando entre seus estagiários os que irão fazer um treinamento especial. O selecionado deve satisfazer ao mesmo tempo a dois critérios.

- ✔ O primeiro critério é que ele deve ter uma bolsa maior ou igual a R\$ 750,00 e menor ou igual a R\$ 950,00.
- ✔ O segundo critério leva em conta o tempo de estágio, este deve ser maior ou igual a 2 anos.

Escreva um algoritmo que solicite ao usuário o valor da bolsa e o tempo de estágio, e caso os critérios acima sejam satisfeitos, mostre a mensagem “Participará do treinamento”, caso contrário mostre “Não participará”.

Exercícios de aplicação

4- Elabore um algoritmo para calcular as raízes de uma equação de segundo grau, mas verifique nesta versão que delta deverá ser um valor positivo, para poder calcular e visualizar as raízes reais da equação. Caso contrário, visualize uma mensagem explicando que não serão calculadas as raízes.

5- Criar um algoritmo que leia a idade de uma pessoa e informe sua classe eleitoral:

- não-eleitor (abaixo de 16 anos)
- eleitor obrigatório (entre 18 e 65 anos)
- eleitor facultativo (entre 16 e 18 anos e maior de 65 anos)

Sugestão: Programação Java ou Python

Utilizando uma ferramenta on-line de sua preferência, procure implementar alguns dos exercícios propostos nesta aula em Java ou Python.

Algumas ferramentas on-line:

- ✓ <https://www.programiz.com/python-programming/online-compiler/>
- ✓ <https://repl.it/languages/>





That's all Folks!