

INSTITUTO FEDERAL
SISTEMA DE ENSINO SUPERIOR

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

Lógica de Programação

- **Gestão da Tecnologia da Informação**
- **1º /2017**
- Profª. Drª. Teresinha Magalhães

INSTITUTO FEDERAL
SISTEMA DE ENSINO SUPERIOR

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

Lógica de programação

- Lógica é a forma correta de **organizar** os **pensamentos** e demonstrar o **raciocínio** de maneira correta.
- A utilização da lógica é a melhor forma de **solucionar problemas** e **atingir objetivos**.
- Sempre que se quer pensar, falar ou escrever corretamente, deve-se colocar **os pensamentos em ordem**.

INSTITUTO FEDERAL
SISTEMA DE ENSINO SUPERIOR

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

ALGORITMOS

Introdução e Conceitos Básicos


INSTITUTO FEDERAL
SISTEMA DE ENSINO SUPERIOR

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

Lógica de programação

- A Lógica está presente em nosso dia-a-dia. Veja exemplo abaixo:

A gaveta está fechada.



A Bala está dentro da gaveta

- **Lógica:**
 - Preciso primeiro abrir a gaveta, para depois pegar a bala

INSTITUTO FEDERAL
SISTEMA DE ENSINO SUPERIOR

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

Lógica de programação

- A lógica faz parte de nosso cotidiano, do nosso pensar, fazer, decidir.
- Entende-se por lógico o que não permite questionamento; o que tem coerência, é óbvio, certo.
- A lógica de programação é a maneira pela qual se representa em linguagem convencional (compreensível), instruções que compõem um programa a ser executado por um computador.
- É o **raciocínio lógico** do programador que vai influenciar diretamente o seu produto final, o **programa**.

INSTITUTO FEDERAL
SISTEMA DE ENSINO SUPERIOR

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

Lógica de programação

Terminologias aplicadas em Lógica de Programação

- **Lógica de programação**
 - É a técnica de encadear pensamentos para atingir determinado objetivo.
- **Sequência Lógica**
 - São passos executados até atingir um objetivo ou solução de um problema.
- **Instruções**
 - Em informática, são ações elementares (ordens) que devem ser executadas por um sistema computacional.



Lógica de programação

- Convém ressaltar que a ordem das instruções influencia no resultado do processo. Ou seja, um resultado pode ser válido ou inválido, pois não houve lógica na resolução do problema.
- Para que isso não ocorra é necessário um conjunto de **instruções** colocadas em ordem, **sequencial lógica**.
- Veja um Exemplo: trocar uma lâmpada.
 1. pegue a escada;
 2. posicione-a embaixo da lâmpada;
 3. suba na escada;
 4. retire a lâmpada velha;
 5. coloque a lâmpada nova.



Algoritmos

- Um Algoritmo é uma **sequência de instruções** ordenadas e sem ambiguidades de forma lógica para a resolução e uma determinada tarefa ou problema.
- Algoritmo é um caminho para a solução de um problema.



Solucionando os problemas

- Para solucionar com precisão os problemas de programação devemos ter em mente os seguintes passos:
- Interpretar o problema proposto;
- Identificar os dados disponíveis para a sua resolução;
- Dividi-lo em partes menores, mediante a sua complexidade;
- Definir seu objetivo. Pergunte-se: “Qual o resultado que se deseja alcançar?”;
- Se mesmo assim não encontrara solução, reveja os passos anteriores.



Algoritmos

- A parte mais importante da tarefa de programar é a construção de algoritmos.
- Segundo Niklaus Wirth:
“Programação é a arte de construir e formular algoritmos de uma forma sistemática”
- Dado um problema, as principais **funções de um programador** são:
 - Entender perfeitamente o problema
 - Escolher métodos para sua solução
 - Desenvolver um algoritmo baseado nos métodos
 - Codificar o algoritmo na linguagem de programação disponível.



Algoritmos

- Embora a palavra **ALGORITMO** possua designação estranha, fazemos uso constantemente de algoritmos em nosso cotidiano. Exemplo:
 - Maneira como uma pessoa toma banho.
 - Instruções para se utilizar um aparelho eletrodoméstico;
 - Uma receita para preparo de algum prato.



Como estudar algoritmos?

O aprendizado de algoritmos não se consegue a não ser através de muitos exercícios.

- “Algoritmos não se aprendem:
 - Copiando algoritmos
 - Estudando algoritmos
- Algoritmos só se aprendem:
 - Construindo algoritmos
 - Testando algoritmos”

Algoritmos

- **Ação:** é um evento que ocorre num período de tempo finito, estabelecendo um efeito intencionado e bem definido.

Ex.: “Colocar o livro em cima da mesa”;
“Atribuir o valor 3,1416 em uma variável”;

Obs.:

- Toda ação deve ser executável em um tempo finito.

Algoritmos

- **Padrão de Comportamento:** em todo o evento pode-se reconhecer um padrão de comportamento, isto é, cada vez que o padrão de comportamento é seguido, o evento ocorre.

Ex.: seja a seguinte descrição:

“Uma dona-de-casa descasca as batatas para o jantar”

“traz a cesta com batatas do porão”;

“traz a panela do armário”;

“descasca as batatas”;

“devolve a cesta ao porão”;

Algoritmos

- **Estado:** O Estado de um dado sistema de objetos é o conjunto de propriedades desses objetos que são relevantes em uma situação considerada.

Ex.: “Livro na estante ou sobre a mesa”;

“Conjunto de valores das variáveis do programa num certo instante da execução”;

Algoritmos

- Essa descrição pode ser usada para descrever eventos distintos (dias diferentes, batatas diferentes etc).

- Isso só é possível porque os eventos possuem o mesmo **padrão de comportamento**.

- O efeito de um evento fica totalmente determinado pelo padrão de comportamento e eventualmente pelo estado inicial.

Algoritmos

- **Processo:** é um evento considerado como uma sequência temporal de (sub) ações, cujo efeito total é igual ao efeito acumulado dessas (sub) ações.

- Em outras palavras, se há interesse, uma ação pode ser geralmente detalhada em um processo.

Algoritmos

- **Algoritmo:** é a descrição de um **padrão de comportamento**, expressado em termos de um repertório bem definido e finito de **ações** primitivas que, com certeza, podem ser executadas.

- Um algoritmo possui caráter imperativo, razão pela qual uma **ação** em um algoritmo é chamada de **comando**.

Ex.: algoritmo para descascar batatas para o jantar:

“traga a cesta com batatas do porão”;

“traga a panela do armário”;

“descasque as batatas”;

“devolva a cesta ao porão”;

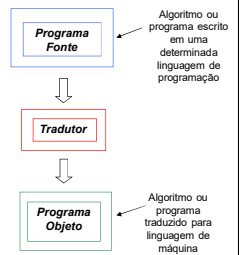
Comandos

Algoritmos

- A generalização do algoritmo para descascar batatas para o jantar pode ser:
 - “traga a cesta com batatas do porão”;
 - “traga a panela do armário”;
 - se “saia é clara” **então** “coloque avental”;
 - enquanto** “número de batatas é insuficiente” **faça** “descasque uma batata”;
 - “devolva a cesta ao porão”;
- Um algoritmo deve ser determinístico, isto é, dadas as mesmas condições iniciais, deve produzir em sua execução, os mesmos resultados.
- Só interessam os algoritmos executáveis em tempo finito.

Tradutor

- Os computadores só podem executar **diretamente** os algoritmos expressos em linguagem de máquina (que é um conjunto de instruções capazes de ativar diretamente os dispositivos eletrônicos do computador).
- Um tradutor é um programa que traduz um algoritmo que está escrito em uma determinada linguagem de programação em linguagem de máquina.



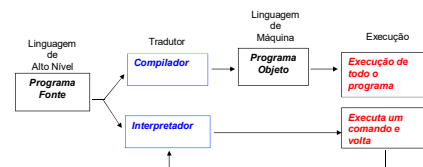
Algoritmo: Características importantes

- ENTRADAS:** Deve ter uma ou mais entradas, isto é, dados que são fornecidos para que sejam processados.
 - Ex: Calcular a soma de dois números
 - A Entrada são os dois números
- SAIDAS:** Deve ter uma ou mais saídas, isto é, resultados que tem uma relação específica com as entradas.
 - Ex: A saída será o valor da soma dos dois números

Processo de Tradução

O processo de tradução pode ser feito por:

- Compilação:** Lê, analisa e traduz todos os comandos do programa fonte, criando o programa objeto.
- Interpretação:** Traduz ou interpreta cada comando ao executá-lo.



Algoritmos - Tradução

- Para que um algoritmo possa ser executado pelo computador, é necessário que ele seja programado, isto é, que ele seja transcrito para uma linguagem que o computador possa “entender”, direta ou indiretamente.

Conceito de algoritmo

- Outras definições consideram a visão da Ciência da computação:
 - * Algoritmo é a forma de ensinar o computador a realizar as tarefas desejadas.
 - * Algoritmo é a descrição lógica de um programa de computador que poderá ser codificado posteriormente em linguagens de programação.

Descrição Narrativa

- Nesta forma de representação os algoritmos são expressos diretamente em **linguagem natural**.
- Como por exemplo, têm-se os algoritmos seguintes:

Troca de um pneu furado:

- Afrouxar ligeiramente as porcas
- Suspender o carro
- Retirar as porcas e o pneu
- Colocar o pneu reserva
- Apertar as porcas
- Abaixar o carro
- Dar o aperto final nas porcas

Conceito de algoritmo

Algoritmo 1 – Calcular a soma de três números digitados e exibir o resultado

- Este algoritmo está escrito em uma descrição narrativa.
- Ela é muito útil para ajudar a entender e organizar as etapas necessárias para solucionar um problema.
- Porém, ela não é compreendida pelo computador, pois infelizmente o computador não compreende a nossa linguagem humana (ainda).
- Por isso, para podermos ensinar programas ao computador, temos que escrever os algoritmos de uma forma padronizada que possa ser facilmente transformada em uma linguagem que o computador entenda, a **linguagem de programação**.

28

Descrição Narrativa

Cálculo da média de um aluno:

- Obter as notas da primeira e da segunda prova
- Calcular a média aritmética entre as duas
- Se a média for maior ou igual a 6, o aluno foi aprovado, senão ele foi reprovado.

Obs: Esta representação é pouco usada na prática porque o uso de linguagem natural muitas vezes dá oportunidade a más interpretações, ambigüidades e imprecisões.

Como o computador entende e executa um programa

O computador é uma máquina que realiza uma sequência programada de instruções, ou seja, ela executa um programa.

Para que a máquina possa realizar estas instruções elas devem ser representadas em uma linguagem que a máquina compreende, a eletricidade, ou seja zeros e uns.

29

Conceito de algoritmo

Algoritmo 1 – Calcular a soma de três números digitados e exibir o resultado

Passo 1 – solicitar a digitação do primeiro número

Passo 2 – armazenar o primeiro número

Passo 3 – solicitar a digitação do segundo número

Passo 4 – armazenar o segundo número

Passo 5 – solicitar a digitação do terceiro número

Passo 6 – armazenar o terceiro número

Passo 7 – somar os três números

Passo 8 – armazenar o resultado da soma

Passo 9 – exibir na tela o resultado da soma

27

Como o computador entende e executa um programa

Podemos entender que o ZERO representa desligado e o UM representa ligado. Portanto o computador entende uma sequência de 01010110011 como sendo um código pré-definido para realizar alguma operação, este código é chamado de código binário.

Para que nós possamos programar as tarefas a serem realizadas pelo computador, devemos criar uma sequência de zeros e uns representando as instruções que desejamos.

30

Como o computador entende e executa um programa

Escritos nas linguagens de programação, existem outros programas que realizam a conversão da linguagem de alto nível em linguagem de máquina. Estes programas são chamados de compiladores e montadores.

Um compilador transforma uma linguagem de programação em uma representação intermediária chamada de linguagem assembly. Cada linguagem de programação possui um compilador específico para fazer esta conversão.

31

Ferramentas para representação de algoritmos

Para escrever um programa ou um algoritmo de uma forma que o computador entenda temos que respeitar um conjunto de regras.

Nas linguagens de programação, estas regras são chamadas de **sintaxe**. Para que um programa funcione ele precisa respeitar a sintaxe da linguagem.

34

Como o computador entende e executa um programa

O montador transforma a linguagem assembly em zeros e uns (chamado de código objeto) e a partir daí o programa pode ser compreendido pela máquina

A seguir temos um exemplo de um pequeno programa em C e suas representações em assembler e em binário.

32

Conceito de algoritmo

Além da descrição Narrativa, existem outras 3 formas de escrever um algoritmo:

* **Pseudo-código ou Portugol**: utiliza palavras escolhidas e regras bem definidas.

* **Fluxograma**: utiliza símbolos gráficos combinados em sequência.

* **Linguagem de Programação**: utiliza os comandos e instruções próprios de uma linguagem de programação específica.

A seguir seguem exemplos de cada um deles para o problema apresentado no Algoritmo 1.

35

Como o computador entende e executa um programa

Linguagem C	Assembly	Binário
	...	0001000010
	<code>_main proc near</code>	0011100010
	<code>push bp</code>	0001101010
	<code>mov bp,sp</code>	0011101000
	<code>dec sp</code>	0100101010
	<code>mov word ptr [bp-2],1</code>	0001000000
<code>void main () {</code>	<code>mov sp,bp</code>	0011100010
<code>int a;</code>	<code>pop bp</code>	0001000010
<code>a=1;</code>	<code>ret</code>	0001000010
<code>}</code>	<code>_main endp</code>	0011101000
	<code>public _main</code>	0100101010
	<code>end</code>	0001000010
	...	0100101010

Após ser compilado se tornará assembly Após ser montado se tornará binário Está pronto para ser compreendido pelo computador

33

Conceito de algoritmo

PORTUGOL	FLUXOGRAMA	LINGUAGEM C
<pre> algoritmo Exemplo2 inteiro a, b, c, soma declarações inteiro a, b, c, soma início escreva("Digite 3 valores") leia(a, b, c) soma <- a + b + c escreva(soma) fim </pre>	<pre> graph TD Inicio([início]) --> Digite{ "Digite 3 valores" } Digite --> Leia[a, b, c] Leia --> SomaCalc[soma <- a + b + c] SomaCalc --> SomaExib{ soma } SomaExib --> Fim([fim]) </pre>	<pre> #include <stdio.h> #include <stdlib.h> int main() { int a, b, c, soma; printf("Digite 3 valores!\n"); scanf("%d", &a); scanf("%d", &b); scanf("%d", &c); soma = a + b + c; printf("Soma = %d", soma); return 0; } </pre>

36

Representação de algoritmos

Português Estruturado (Portugol)

O Portugol é uma representação que se assemelha bastante com as linguagens de programação, porém é escrito em português.

As regras para escrever um algoritmo em portugol são as seguintes:

```
programa <nome_do_programa>
declarações
  <definições_de_constantes>
  <declarações_de_tipos_compostos>
  <declarações_de_variáveis>
início
  <instruções_do_programa>
fim
```

37

Pseudolinguagem

A seguir é mostrado a representação do algoritmo que solicita o nome do usuário e imprime na tela o nome digitado, na forma de um pseudocódigo

```
algoritmo "semnome"
// Função :
// Autor :
// Data : 05/09/2011
// Seção de Declarações
var
início
// Seção de Comandos
finalgoritmo
```

```
algoritmo "seunome"
// Função :
// Autor :
// Data : 05/09/2011
// Seção de Declarações
var
nome: caracter
início
escreva("Digite seu nome")
leia(nome)
escreva("O nome digitado é:", nome)
finalgoritmo
```

Você podem testá-lo usando o compilador de portugol - VisuAlg

Ferramentas para representação de algoritmos

Exibe a mensagem "Ola !"

```
programa ola
início
  escreva ("Ola !")
fim
```

Calcula e exibe a soma de 2 + 2

```
programa soma
declarações
  inteiro soma
início
  soma <- 2 + 2
  escreva (soma)
fim
```

38

Fluxograma

- Representação gráfica de algoritmos onde formas geométricas diferentes implicam ações (instruções, comandos) distintos.
- Tal propriedade facilita o entendimento das idéias contidas nos algoritmos.






Pseudolinguagem

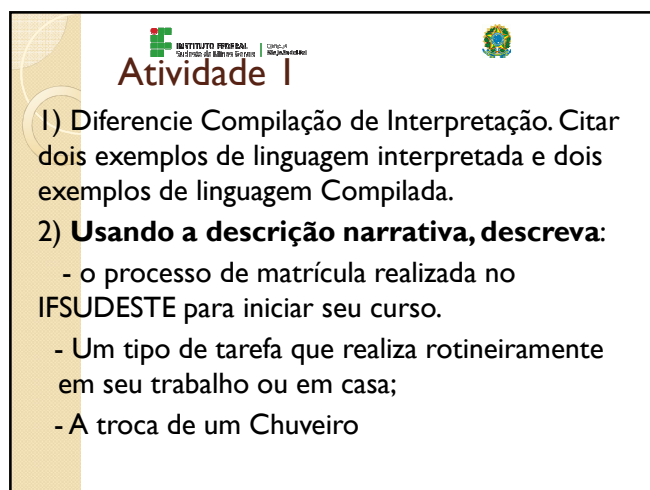
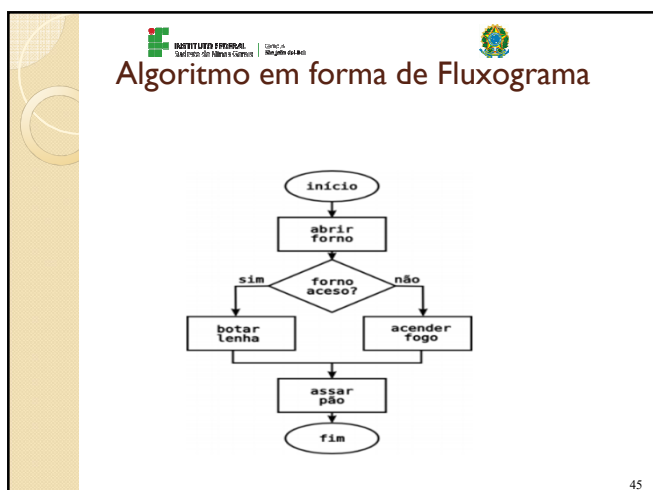
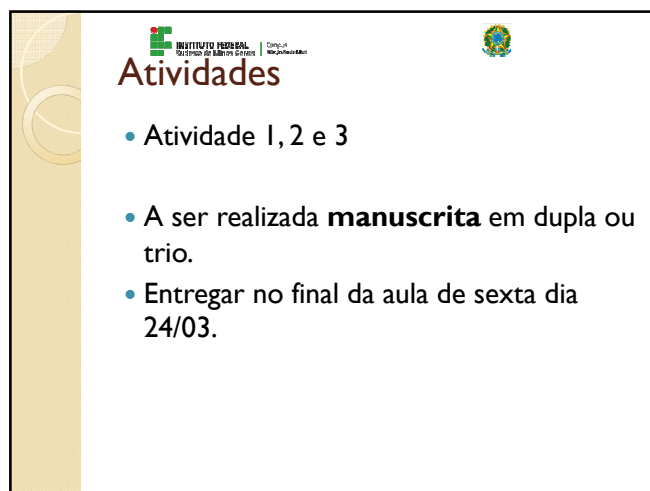
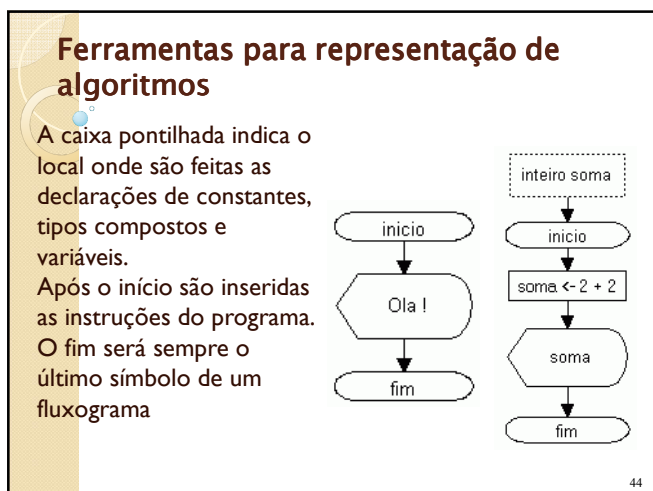
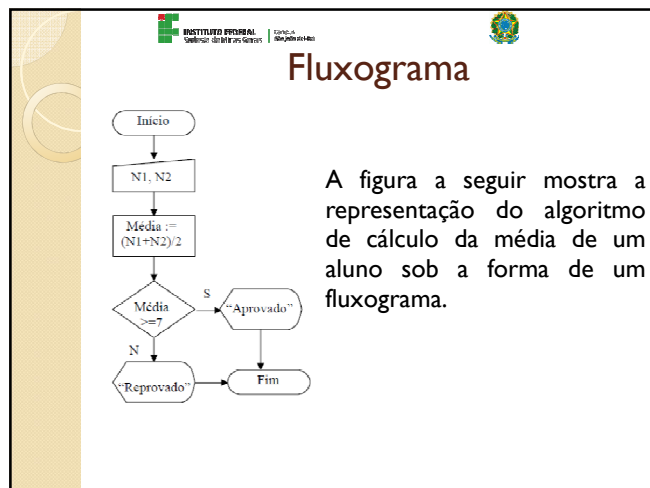
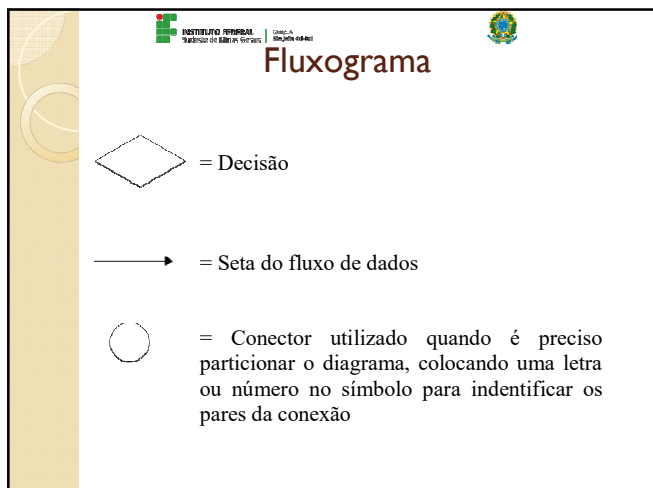
Representação de um Algoritmo na Forma de Pseudolinguagem:

```
início
  <declaração_de_variáveis>
  <comandos>
finalgoritmo.
```

Fluxograma ou Diagrama de Blocos

Principais formas geométricas usadas em fluxogramas.

-  = Início e final do fluxograma .
-  = Operação de entrada de dados
-  = Operação de saída de dados em impressora
-  = Operação de saída de dados em vídeo
-  = Operações de atribuição





Representar através de **fluxograma** os seguintes algoritmos

1. Ler a idade de um aluno e informar se o mesmo pode ou não ser eleitor;
2. Ler 4 notas e calcular a média das notas. Se MÉDIA maior ou igual a 6, mostrar aprovado, senão Reprovado;
3. Ler um número e mostrar se é par; **OBS: O número é par se dividido por 2, sobrar resto 0.**
4. Calcular o aumento que será dado a um funcionário, obtendo do usuário as seguintes informações: salário atual e a porcentagem de aumento. Apresentar o novo valor do salário e o valor do aumento.
5. Converter uma quantidade de horas digitadas pelo usuário em minutos. Informe o resultado em minutos.
6. Ler uma temperatura em graus Celsius e apresentá-la convertida em graus Fahrenheit. A fórmula de conversão: $F \leftarrow (9 * C + 160) / 5$



Representar através de **português estruturado** os seguintes algoritmos

1. Ler a idade de um aluno e informar se o mesmo pode ou não ser eleitor;
2. Ler 4 notas e calcular a média das notas. Se MÉDIA maior ou igual a 6, mostrar aprovado, senão Reprovado;
3. Ler um número e mostrar se é par; **OBS: O número é par se dividido por 2, sobrar resto 0.**
4. Calcular o aumento que será dado a um funcionário, obtendo do usuário as seguintes informações: salário atual e a porcentagem de aumento. Apresentar o novo valor do salário e o valor do aumento.
5. Converter uma quantidade de horas digitadas pelo usuário em minutos. Informe o resultado em minutos.
6. Ler uma temperatura em graus Celsius e apresentá-la convertida em graus Fahrenheit. A fórmula de conversão: $F \leftarrow (9 * C + 160) / 5$



- Conceito de tipos de dados;
- Variáveis;
- Operadores;
- Comandos de entrada e saída...