

Engenharia de Computação

Lógica de Programação

Estruturas Elementares de Algoritmos

Prof. Anderson Luiz Fernandes Perez

Universidade Federal de Santa Catarina
Campus Araranguá

Email: anderson.perez@ufsc.br

Conteúdo

- Introdução
- Tipos de Dados
- Variáveis
- Constantes
- Operadores
- Entrada e Saída de Dados

Introdução

- Um **algoritmo** é **composto de várias partes** que vão desde a representação dos **dados de entrada**, passando pelo **processamento** mais complexo desses dados, até a representação dos **dados (informações)** de **saída**.
- É importante conhecer e saber aplicar cada uma das partes que constituem um algoritmo.
- Os **tipos de dados**, os **operadores lógicos, aritméticos, relacionais** e os procedimentos de **entrada e saída de dados** são elementos fundamentais em qualquer algoritmo.

Tipos de Dados

- Um dado pode ser considerado como um símbolo que pode representar alguma informação.
- Exemplos de dados:
 - Dia
 - Mês
 - Ano
 - Preço do quilo de pão francês
 - Valor da multa por entrega de livros atrasados na biblioteca
 - ...

Tipos de Dados

- A partir de um ou mais dados é possível extrair **informações**.
- Por exemplo:
 - Dia 21 de abril (**dado**)
 - *Dia da inconfidência mineira – Tirandentes* (**informação**)
 - Ayrton Senna da Silva (**dado**)
 - *Piloto brasileiro de Fórmula 1 que faleceu em 1 de maio em Bolonha, Itália* (**informação**)

Tipos de Dados

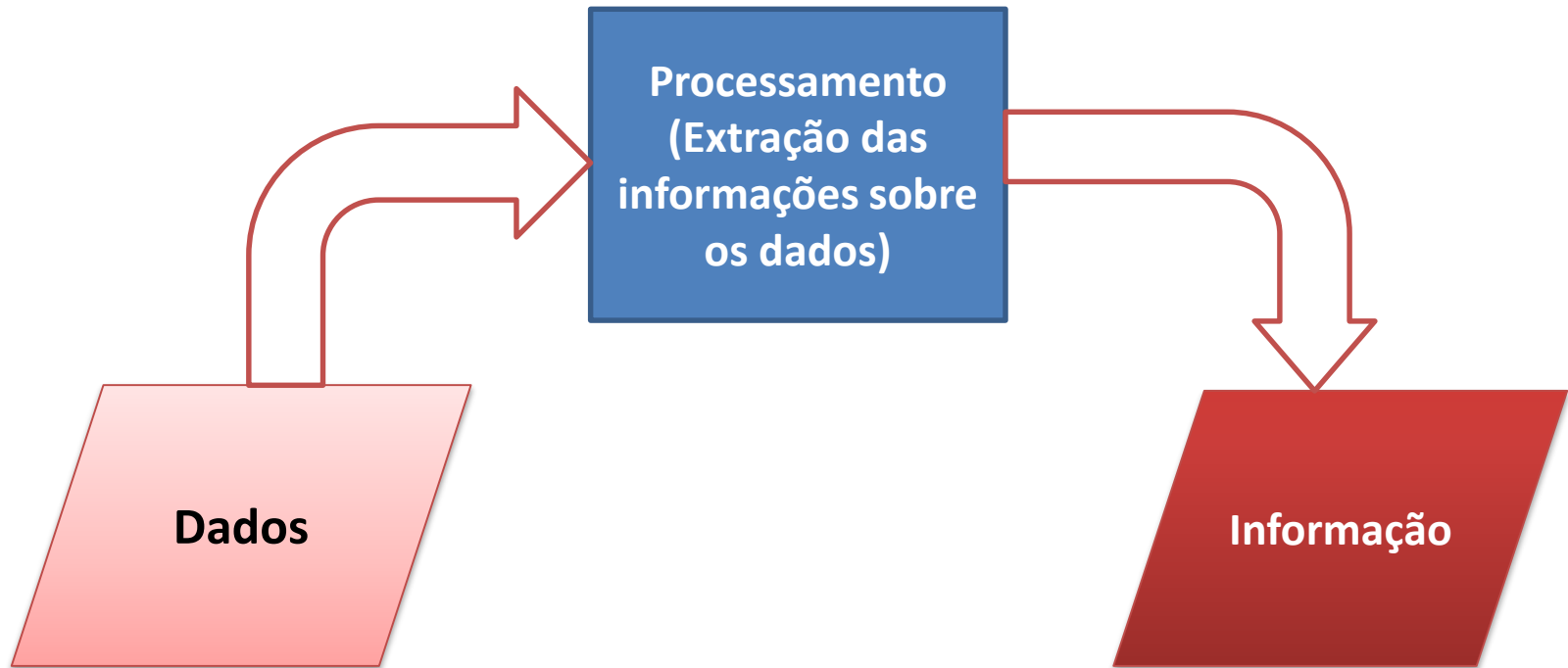
- Resumindo ...

Um **dado** constitui a **matéria prima da informação**, ou seja, é uma informação mas que ainda não foi tratada (processada).

Uma **informação** representa o **resultado do processamento ou tratamento de um ou mais dados**.
Representam significado sobre algo ou alguém, inclusive para a tomada de decisões.

Tipos de Dados

- Logo ...



Tipos de Dados

- Para que um algoritmo possa manipular os **dados** de entrada para gerar informação, é necessário que estes estejam **armazenados em algum lugar**.
- Os dados precisam ser armazenados em lugares na **memória do computador**, ou seja, em porções da memória RAM (*Random Access Memory*).

Tipos de Dados

- Como o dado representa alguma informação, é necessário tipificá-lo, ou seja, determinar qual o tipo de informação que o dado é capaz de representar.
- Um dado pode ser do seguinte tipo:
 - **Inteiro** – toda a qualquer informação numérica que pertença ao conjunto dos números inteiros.
 - **Real** – toda e qualquer informação numérica em que pertença ao conjunto dos números reais.
 - **Caracter** – toda e qualquer informação composta por apenas um caractere.
 - **Cadeia de caracteres** – toda e qualquer informação composta por mais de um caracter (conjunto de caracteres).
 - **Lógico** – toda e qualquer informação que pode assumir apenas duas informações (verdadeiro ou falso).

Tipos de Dados

- Exemplos de Dados do Tipo Inteiro
 - Idade
 - *João tem 12 anos de idade*
 - Quantidade
 - *Comprei 15 pães francêss*
 - Opção
 - *Eu prefiro a opção 7*

Tipos de Dados

- Exemplos de Dados do Tipo Real
 - Altura
 - Marcos tem *1,87* metros
 - Valor
 - O valor do passe do Restaurante Universitário é R\$ *1,50*
 - Peso
 - A massa de ar vale *1,3* grama/litro

Tipos de Dados

- Exemplos de Dados do Tipo Caracter

- Opção

- *A melhor opção é a “C”*

- Sexo

- *Todas as pessoas do sexo MASCULINO devem preencher o formulário com a letra “M” e todas as do sexo FEMININO devem preencher com a letra “F”*

Tipos de Dados

- Exemplos de Dados do Tipo Cadeia de Caracteres
 - Nome
 - *O candidato se chama Miguel*
 - Endereço
 - *A Universidade Federal de Santa Catarina em Araranguá está localizada em dois campi, no Jardim das Avenidas e no Mato Alto*
 - Resumo
 - *Engenheiro de Computação tem sua formação fortemente baseada na Ciência da Computação, nas Ciências para Engenharia, na Matemática, na Elétrica e na Eletrônica Digital.*

Tipos de Dados

- Exemplos de Dados do Tipo Lógico

- Resposta

- Responda *sim (1)* para as questões corretas e *não (2)* para as questões incorretas

- Comparação

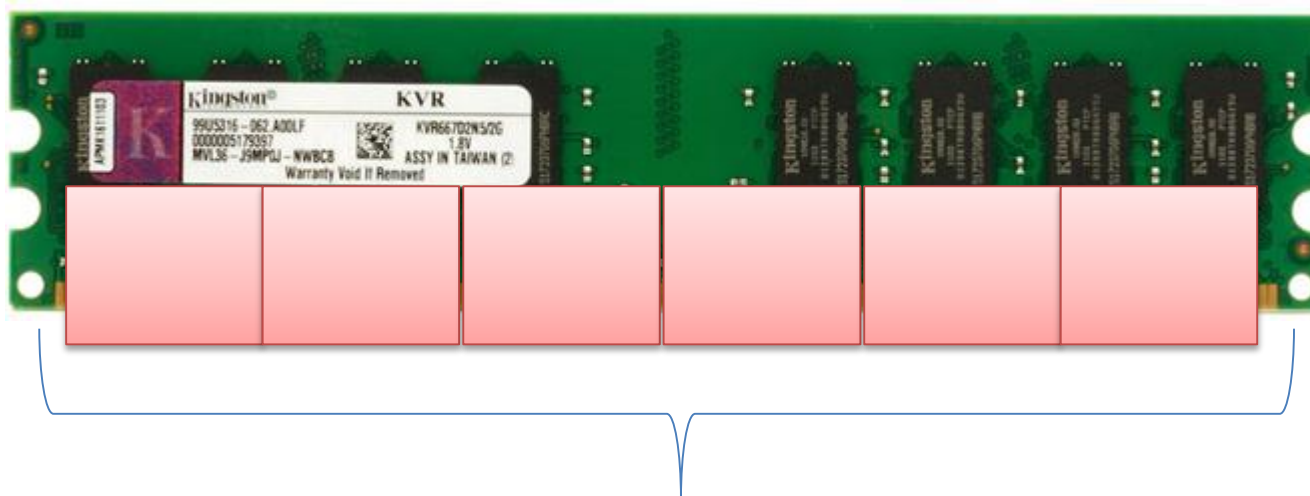
- *25 é maior do que 30?*

Tipos de Dados

- Dado Alocado na Memória
 - Todos os dados **devem** ser alocados na memória principal do computador.
 - Para tanto, a **memória é segmentada em compartimentos**, onde **cada compartimento pode conter apenas um dado**.
 - O que determina a quantidade de compartimentos **alocados** ou reservados para um determinado dado é o seu tipo.

Tipos de Dados

- Dado Alocado na Memória
 - Supondo um **computador hipotético** com a seguinte organização da memória principal.



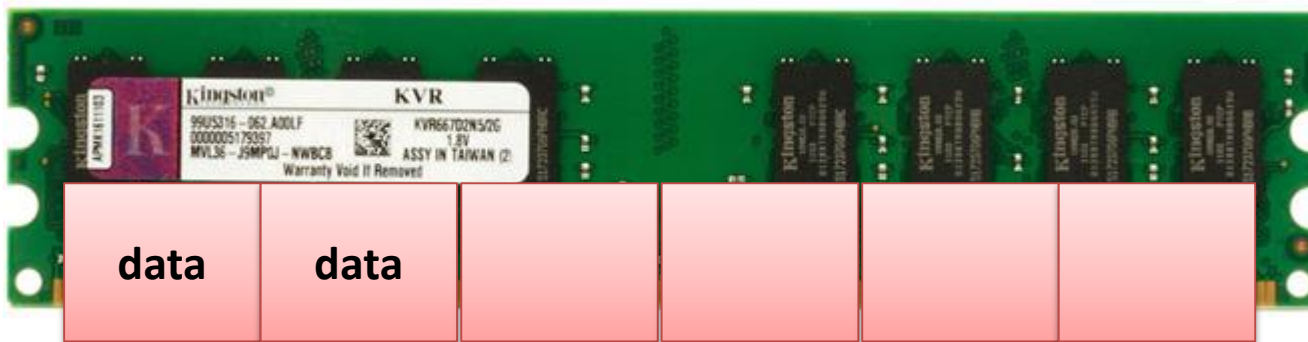
Memória dividida em seis segmentos

Tipos de Dados

- Dado Alocado na Memória
 - Considerando o **computador hipotético** e a **memória descrita anteriormente**, considere que os tipos de dados ocuparão as seguintes quantidades de memória:
 - **Inteiro** – 2 segmentos
 - **Real** – 2 segmentos
 - **Caracter** – 1 segmento
 - **Cadeia de caracter** – x segmentos (depende da quantidade de caracteres presentes na cadeia)
 - **Lógico** – 1 segmento

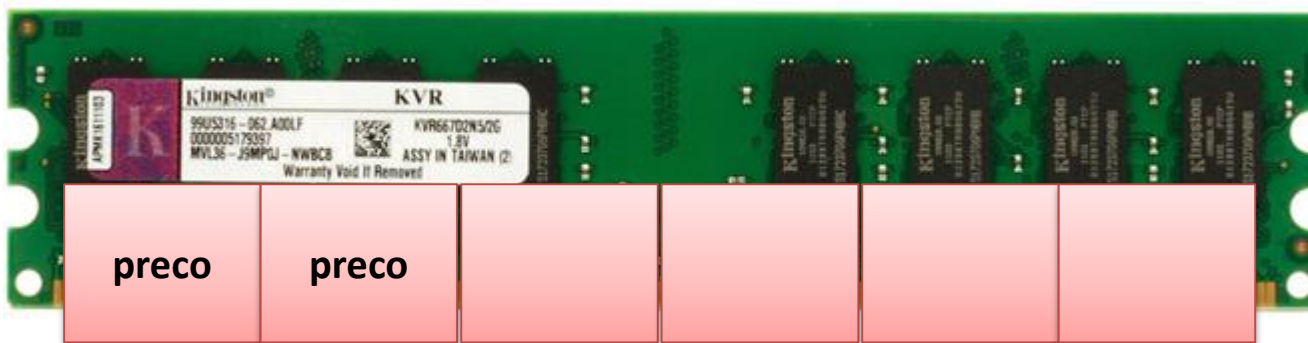
Tipos de Dados

- Dado Alocado na Memória
 - Exemplo: alocando um dado do tipo inteiro
 - *inteiro* data



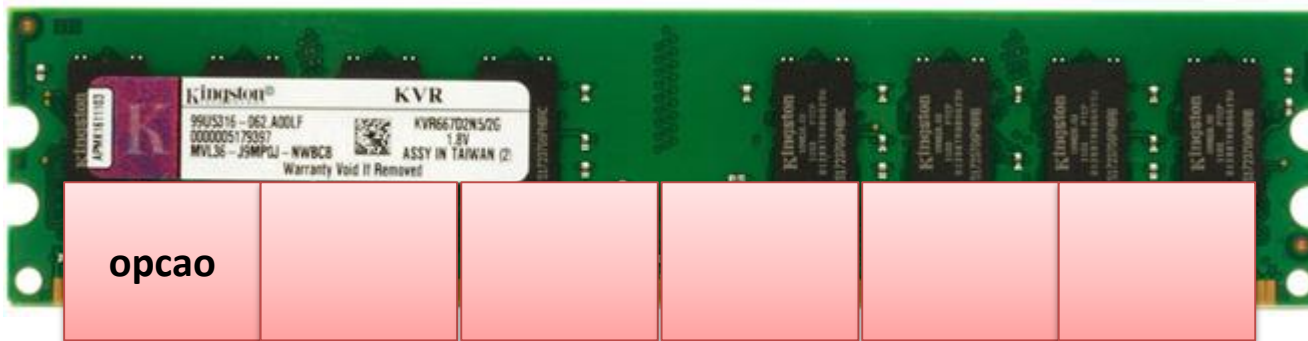
Tipos de Dados

- Dado Alocado na Memória
 - Exemplo: alocando um dado do tipo inteiro
 - *real* preço



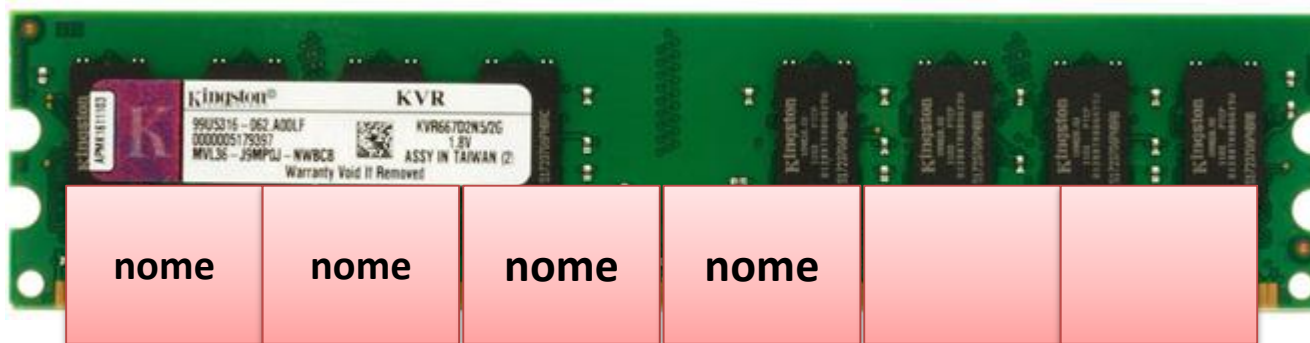
Tipos de Dados

- Dado Alocado na Memória
 - Exemplo: alocando um dado do tipo inteiro
 - *caracter* opcao



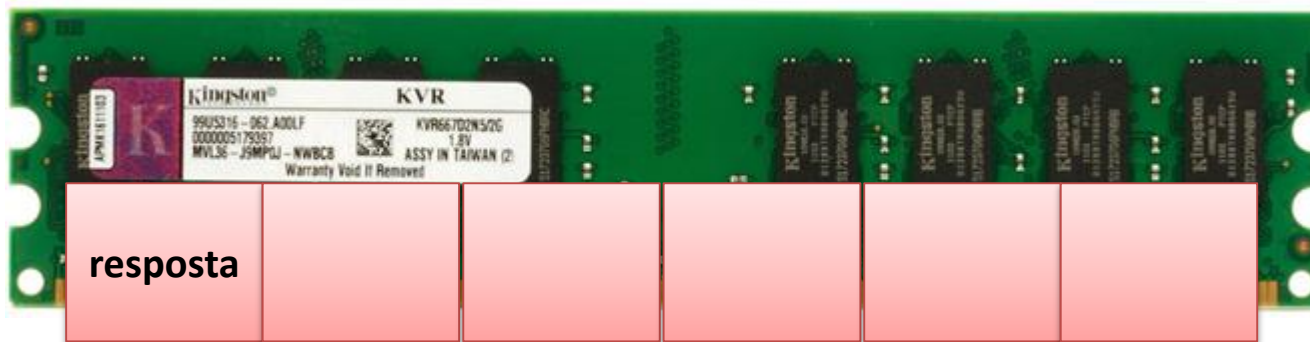
Tipos de Dados

- Dado Alocado na Memória
 - Exemplo: alocando um dado do tipo inteiro
 - *cadeia* nome
 - Supondo que nome contenha os seguintes caracteres: “UFSC”



Tipos de Dados

- Dado Alocado na Memória
 - Exemplo: alocando um dado do tipo inteiro
 - *lógico* resposta



Tipos de Dados

- Os dados armazenadas na memória principal do computador podem ser de dois tipos:
 1. **Variáveis** – seu conteúdo pode sofrer alterações durante a execução do algoritmo (programa).
 2. **Constantes** – seu conteúdo não se altera ao longo de toda a execução do algoritmo (programa).

Variáveis

- Uma variável é um nome (identificador) dado pelo programador para representar um dado qualquer.
- Uma variável deve ser declarada de maneira explícita.
- A declaração de variáveis deve ser feita de seguinte maneira:
 - **tipo de dado** *nome da variável*

Variáveis

- Declaração de Variáveis
 - Exemplos:
 - inteiro dia, mes, ano*
 - real preco*
 - caracter escolha*
 - ***Dia, mes, ano, preco*** e ***escolha*** são variáveis com seus respectivos tipos, ou seja, ocupam porções distintas de memória.
 - O nome dado a uma variável não pode conter caracteres especiais, tais como \$, %, ! etc e nem iniciar com números.

Variáveis

- Atribuição de Valores a Variáveis
 - A atribuição de valores a variáveis deve ser feito usando o comando de atribuição.
 - Exemplo:
***inteiro** dia, mes, ano*
dia = 14
mes = 4
ano = 2015
 - O caractere **=** significa atribuição de dados.

Em pseudocódigo:

inicio

dia, mês, ano : inteiro

dia <- 14

mes <- 4

ano <- 2015

fim

Variáveis

- O conteúdo de uma variável é passível de alteração durante a execução do algoritmo (programa).
- Exemplo:
inteiro a, b
 $a = 10$
 $b = 5$
 $a = a + b$
- O novo conteúdo da variável ***a*** será o valor inteiro 15.

Constantes

- Um dado é constante quando **não** sofre nenhuma **variação** no decorrer do tempo.
- Do início ao fim do programa o valor permanece **inalterado**.
- Exemplos:

10

“Bata antes de entrar!”

-0,58

PI = 3.141592653589793

Constantes

- Declaração de Constantes
 - A regra para declaração de constantes é a mesma usada na declaração de variáveis. Entretanto, antes do tipo deve-se usar a palavra reservada ***const***.
 - Exemplos:
 - const inteiro x***
 - const real preco***
 - const character opcao***

Constantes

- Atribuição de Valores a Constantes
 - A atribuição de valores a constantes é realizada no momento da declaração.
 - Não é permitido atribuir um conteúdo a uma constante em qualquer outro momento que não seja o da declaração.
 - Exemplos:
 - const inteiro* valor = 100
 - const inteiro* quantidade

quantidade = 500  

Operadores

- Em uma linguagem de programação existem vários **operadores** que permitem operações do tipo:
 - Aritmética
 - Relacional
 - Lógica
 - Composta

Operadores

- Aritméticos

- Os operadores aritméticos permitem realizar **operações matemáticas** sobre constantes e variáveis.
- Esses operadores permitem somar, dividir, multiplicar e subtrair valores com **conteúdo de variáveis e/ou constantes, bem como o conteúdo de variáveis com variáveis.**

Operadores

- Aritméticos

Símbolo	Função
+	Adição
-	Subtração
*	Multiplicação
/	Divisão
%	Módulo (resto da divisão inteira)
Pot	Potenciação
Rad	Radiciação

Operadores

- Aritméticos

- Exemplos:

inteiro x, y

$x = 100$

$y = 200$

$x = y$

$x = y + 3$

Qual o conteúdo da variável x após a execução das linhas do algoritmo descrito acima?

Operadores

- Aritméticos
 - Precedência entre Operadores Aritméticos

Prioridade	Operadores
1ª	Parênteses mais internos
2ª	Potenciação e radiciação
3ª	$*$, $/$, $\%$
4ª	$+$, $-$

Operadores

- Lógicos
 - Os operadores lógicos são úteis para se criar **novas proposições lógicas** compostas.
 - Existem diversas situações onde **o algoritmo precisa prosseguir para uma determinada ação** (processamento) caso uma **situação** seja **verdadeira** e outra **falsa** ou **vice-versa**. Ou ainda se duas situações foram falsas ou verdadeiras.

Operadores

- Lógicos

Símbolo	Função
&&	E (and)
	OU (or)
!	Não (not)

Operadores

- Lógicos
 - Tabela Verdade

A	B	A (E) B	A (OU) B	NÃO A
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

Operadores

- Lógicos
 - Exemplos:
 - logico sim, nao*
 - sim = verdadeiro*
 - nao = falso*
 - sim e nao*
 - sim ou nao*

Operadores

- Lógicos
 - Precedência entre Operadores Lógicos

Prioridade	Operadores
1ª	Não
2ª	E
3ª	OU

Operadores

- Relacionais
 - Os operadores relacionais **permitem fazer relações entre dois ou mais valores** (variáveis/constantes).
 - O **resultado** de um operador relacional **é sempre um valor lógico**, ou seja, *verdadeiro* ou *falso*.
 - Os operadores relacionais são muito utilizados para fazer **desvios condicionais** no algoritmo, ou seja, dependendo do resultado de uma comparação, o algoritmo executa uma ou outra ação(es).

Operadores

- Relacionais

Símbolo	Função
>	Maior
<	Menor
>=	Maior ou igual
<=	Menor ou igual
==	Igual
!=	Diferente

Operadores

- Relacionais

- Exemplos:

***inteiro** valor_a, valor_b*

valor_a = 230

Valor_b = 0

valor_a > valor_b

valor_a == valor_b



Os resultados são
valores lógicos
(verdadeiro ou falso).

Operadores

- Compostos
 - Os operadores compostos são uma mistura dos operadores aritméticos com o comando de atribuição de valores a variáveis.
 - Permitem fazer operações sobre variáveis e atribuir esses valores a variável na mesma instrução.

Operadores

- Compostos

Símbolo	Função
++	Incremento
--	Decremento
+=	Adição com atribuição
-=	Subtração com atribuição
*=	Multiplicação com atribuição
/=	Divisão com atribuição

Operadores

- Compostos

- Exemplos:

inteiro x, y

$x = 100$

$y = 200$

$x = y$

$x += y$

$x++$

Qual o conteúdo da variável x após a execução das linhas do algoritmo descrito acima?

Operadores

- Precedência entre todos os Operadores

Prioridade	Operadores
1ª	Parênteses mais internos
2ª	Operadores aritméticos
3ª	Operadores relacionais
4ª	Operadores lógicos

Entrada e Saída de Dados

- Um sistema computacional deve permitir entrada e saída de dados.
- Na descrição de um algoritmo, seja em pseudocódigo ou graficamente, é necessário representar as entradas e as saídas de dados.
- As entradas de dados podem ser provenientes do teclado ou mesmo de arquivos.
- As saídas de dados podem ser direcionadas para o vídeo (monitor) ou para arquivos.

Entrada e Saída de Dados

- Entrada
 - Exemplo em pseudocódigo:

inicio

inteiro quantidade, idade

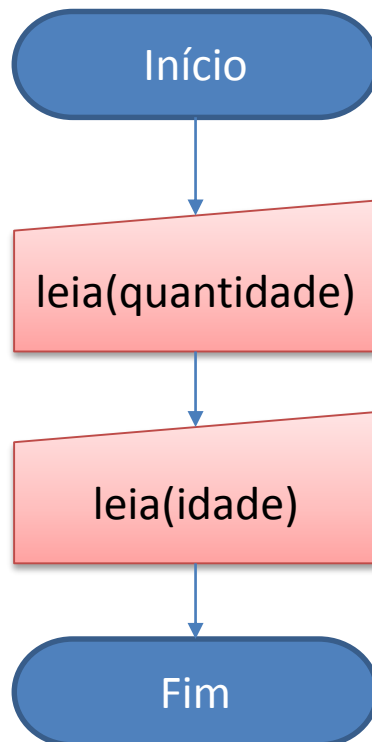
leia(quantidade)

leia(idade)

fim

Entrada e Saída de Dados

- Entrada
 - Exemplo em fluxograma:



Entrada e Saída de Dados

- Saída
 - Exemplo em pseudocódigo:

inicio

inteiro quantidade, idade

leia(quantidade)

leia(idade)

escreva(quantidade)

escreva(idade)

fim

Entrada e Saída de Dados

- Saída
 - Exemplo em fluxograma:

