Algoritmos e Programação

Comandos simples e estruturas de controle

Prof. Dr. Joaquim V. C. Assunção

CENTRO DE TECNOLOGIA
UFSM
2025

Comandos

Comandos são usados para expressar alguma ação que deve ser desempenhada pelo algoritmo. Exemplos:

```
cin >>   ler()

         escrever()

cout <<  imprimir()
```

Comandos

Também podemos usar comandos mais complexos...

→ É importante lembrar que um algoritmo não pode ser executado por um computador. Ele deve ser transcrito em alguma linguagem de programação antes de ser executado pelo computador.

Comandos

Comandos simples devem ser facilmente compreendidos pelo programador. Comandos como `ler()` e `escrever()` são facilmente mapeados para funções que as linguagens de programação disponibilizam.

Por outro lado, um simples comando `verificar_face()`, pode requerer milhares de linhas de código para ser implementada, isso se for possível.

Comandos

Existem várias formas de usar, por exemplo, o comando `ler()`, como visto no seguinte exemplo. As linguagens de programação oferecem alguns meios de realizar a leitura de dados. Em alguns exemplos foi utilizado um operador de atribuição.

```
Var: minhaVariavel

minhaVariavel ← ler() //ou ler(minhaVariavel)
```

Comandos

Com o comando `escrever()` pode-se exibir informações geradas pelo algoritmo. Pode-se escrever dados de variáveis, mensagens ou combinação de ambos, como mostrado nos seguintes exemplos. Deve-se cuidar para não confundir a impressão de mensagens (entre “ ”) com a impressão de valores.

```
int: a;
literal: b;
a ← 100;
b ← “alo mundo”;
```

```
escrever ( a ); //100
escrever ( “a” ); //a
escrever ( b ); // alo mundo
Escrever(‘b’); //b
escrever (“o valor de a vale ” + a)
o valor de a vale 100
```

Comandos

```
…
int main() {
    int a;
    string b;
    a = 100;
    b = "alo mundo";

    cout << (a) << "\n";
    cout << ("a") << "\n";
    cout << (b) << "\n";
    cout << ('b') << "\n";
    cout << "O valor de a vale " << a;
}
```

Comandos

Caso tenhamos construído uma função, podemos usar o comando `retorne` para retornar um valor.

...

Exercícios

1. Crie um algoritmo que leia o nome e a idade de um individuo e imprima os dois juntos, mas com um “-” no meio.
2. Crie um algoritmo que efetue a leitura de dois valores numéricos inteiros. Processe a operação de adição dos dois valores e apresente na sequência a soma obtida com a operação. Exemplo de saída: “2 + 2 = 4”
3. Crie um algoritmo que calcule a área de uma circunferência e apresente a medida da área calculada.
4. Crie um algoritmo que calcule o salario liquido de um funcionário. O programa deve ler o valor hora e o desconto recebido mensalmente (considerar 40h/semana e 4 semanas/mês).

Definição

- Uma estrutura de controle permite a escolha de um grupo de ações (bloco) a ser executado quando determinadas condições, representadas por expressões lógicas ou relacionais, são ou não satisfeitas.

Seleção simples

- Quando precisamos testar uma certa condição antes de executar uma ação, usamos uma seleção simples, que segue o seguinte modelo:

```
if (something){
        //ação!
    }
```

- `something` é uma expressão lógica que, quando inspecionada, pode gerar um resultado falso ou verdadeiro.

Seleção simples – fluxo alternativo

- Se tivermos situações em que duas alternativas dependem de uma mesma condição (uma da condição verdadeira e outra da falsa) usamos a estrutura de seleção composta…

Seleção simples – fluxo alternativo

- Supondo serem N1, N2 as duas notas semestrais de um aluno, podemos avaliar sua situação quanto à aprovação, obtida atingindo-se média superior ou igual a 7

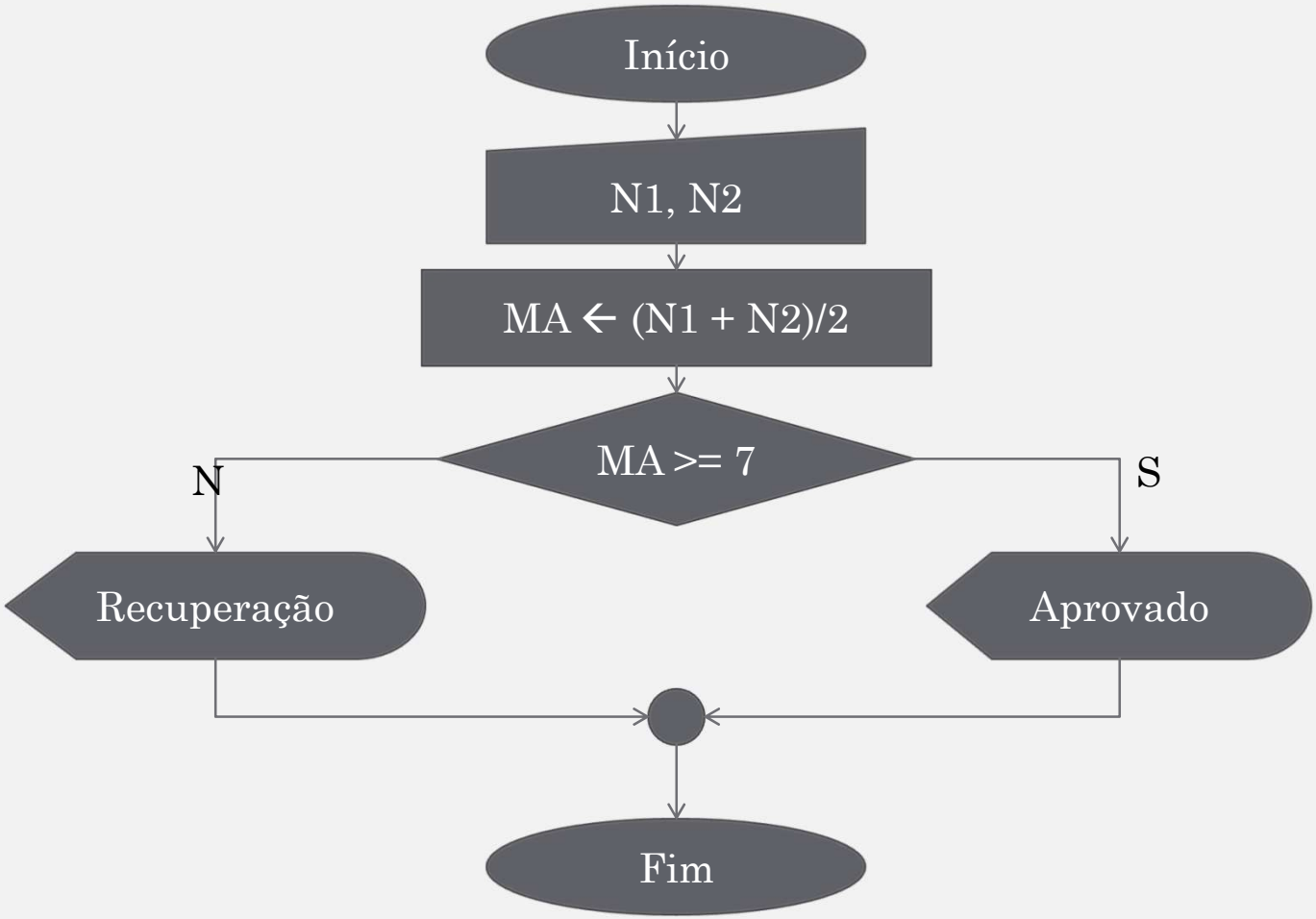
```
início
        real: N1, N2, MA;
        leia(N1, N2);
        MA <- (N1 + N2 )/2;
        escreva(MA);
        se (MA >= 7) então
                escreva("Aluno aprovado");
        senao
                escreva("Aluno em recuperação");
        fim_se

fim
```

Seleção simples – em C++

- Supondo serem N1, N2 as duas notas semestrais de um aluno, podemos avaliar sua situação quanto à aprovação, obtida atingindo-se média superior ou igual a 7

```
#include <iostream>
using namespace std;

int main() {
    float N1, N2, MA;
    cout << "Digite as duas notas: ";
    cin >> N1 >> N2;

    MA = (N1 + N2) / 2;

    cout << "Média: " << MA << "\n";

    if (MA >= 7) {
        cout << "Aluno aprovado" << "\n";
    } else {
        cout << "Aluno em recuperação" << "\n";
    }

    return 0;
}
```

Seleção simples – fluxo alternativo



Exercícios

5. Ler um número e se ele for maior do que 20 e menor que 30, então imprima a metade do número.

6. Ler um número e se ele for positivo, imprimir seu inverso (1/numero); caso contrário, imprimir o valor absoluto do número. //abs();
7. Leia uma idade, caso ela seja menor que 12 ou maior que 70, imprima: “Paga meia entrada”.
8. Ler duas respostas em que o usuário só pode responder verdadeiro ou falso. Imprimir na tela de acordo com as respostas: “Tudo verdade”, “Parcial”, “Tudo falso”.