

Algoritmos - AULA 1

Sumário

1	Definições.....	1
1.1	Algoritmos.....	1
1.2	Linguagem de Programação	2
1.2.1	Exemplos de codificação	2
1.3	Compilador.....	3
1.4	IDE.....	3
1.5	Diretivas de Compilação	3
2	Variáveis	3
2.1	Tipos de dados	3
2.2	Declaração de variáveis	4
2.3	Regras para nomear uma variável.....	4
2.4	Strings.....	4
2.5	Atribuição	5
3	Saída padrão	5
3.1	Formato geral	5
3.2	Mostrado o valor de uma variável	5
4	Um programa completo em C.....	6
5	Exercícios	6

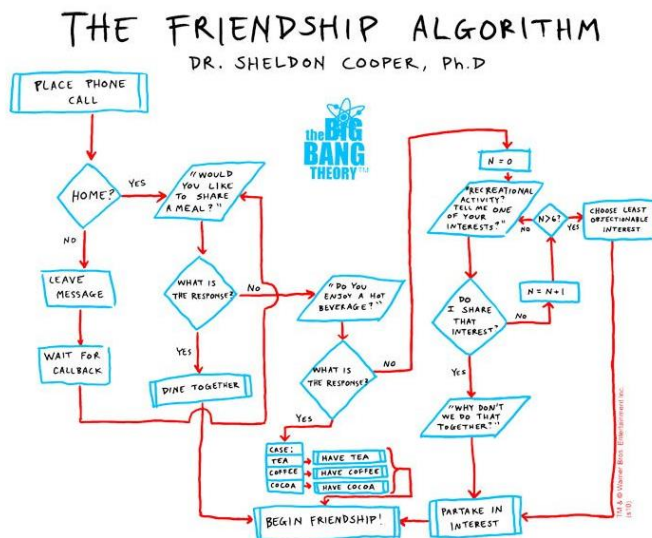
1 Definições

1.1 Algoritmos

Um algoritmo é uma sequência finita de instruções bem definidas e não ambíguas, cada uma das quais pode ser executada mecanicamente num período de tempo finito e com uma quantidade de esforço finita.

O conceito de algoritmo é frequentemente ilustrado pelo exemplo de uma receita culinária, embora muitos algoritmos sejam mais complexos. Eles podem repetir passos (fazer iterações) ou necessitar de decisões (tais como comparações ou lógica) até que a tarefa seja completada.

Um algoritmo corretamente executado não irá resolver um problema se estiver implementado incorretamente ou se não for apropriado ao problema.



1.2 Linguagem de Programação

Uma linguagem de programação é um método padronizado para comunicar instruções para um computador. É um conjunto de regras sintáticas e semânticas usadas para definir um programa de computador. Permite que um programador especifique precisamente sobre quais dados um computador vai atuar, como estes dados serão armazenados ou transmitidos e quais ações devem ser tomadas sob várias circunstâncias. Linguagens de programação podem ser usadas para expressar algoritmos com precisão.

Existem várias linguagens de programação, as mais populares são:

- C
- Java
- Objective-C
- C++
- PHP
- C#
- Visual Basic
- Python
- Perl
- Ruby
- Lisp
- Pascal
- Pascal
- Ada

1.2.1 Exemplos de codificação

Em C:

```
#include <stdio.h>
int main(void)
{
    puts("Olá, Mundo!");
    return 0;
}
```

Em Java:

```
public class OlaMundo {
    public static void main(String[] args) {
        System.out.println("Olá, Mundo!");
    }
}
```

1.3 Compilador

O nome "compilador" é usado principalmente para os programas que traduzem o código fonte de uma linguagem de programação de alto nível para uma linguagem de programação de baixo nível (por exemplo, Assembly ou código de máquina).

1.4 IDE

IDE, do inglês Integrated Development Environment ou Ambiente Integrado de Desenvolvimento, é um programa de computador que reúne características e ferramentas de apoio ao desenvolvimento de software com o objetivo de agilizar este processo.

As principais IDE's para desenvolvimento em C, no contexto do nosso curso, são as seguintes:

- Dev C++
- NetBeans
- Eclipse
- Xcode
- C Builder

1.5 Diretivas de Compilação

O pré-processador C é um programa que examina o programa fonte escrito em C e executa certas modificações nele, baseado nas Diretivas de Compilação. As diretivas de compilação são comandos que não são compilados, sendo dirigidos ao pré-processador, que é executado pelo compilador antes da execução do processo de compilação propriamente dito.

As diretivas do C são identificadas por começarem por #. A diretiva #include diz ao compilador que ele deve incluir um arquivo externo na hora da compilação.

2 Variáveis

Na programação, uma variável é um objeto (uma posição, frequentemente localizada na memória) capaz de reter e representar um valor ou expressão. Enquanto as variáveis só "existem" em tempo de execução, elas são associadas a "nomes", chamados identificadores, durante o tempo de desenvolvimento.

Quando nos referimos à variável, do ponto de vista da programação de computadores, estamos tratando de uma "região de memória (do computador) previamente identificada cuja finalidade é armazenar os dados ou informações de um programa por um determinado espaço de tempo". A memória do computador se organiza tal qual um armário com várias divisões. Sendo cada divisão identificada por um endereço diferente em uma linguagem que o computador entende.

O computador armazena os dados nessas divisões, sendo que em cada divisão só é possível armazenar um dado e toda vez que o computador armazenar um dado em uma dessas divisões, o dado que antes estava armazenado é eliminado. O conteúdo pode ser alterado, mas somente um dado por vez pode ser armazenado naquela divisão.

2.1 Tipos de dados

Os principais tipos de dados em C são os seguintes:

- char: quando se referir a caracteres;
- int: quando se referir a números inteiro;
- float: quando se referir a números com casas decimais (ponto flutuante).

2.2 Declaração de variáveis

As variáveis no C devem ser declaradas antes de serem usadas. A forma geral da declaração de variáveis é:

tipo nome;

Onde:

- Tipo é o tipo de dado que será armazenado nesta variável. Usaremos neste momento um dos tipos apresentados no item anterior;
- Nome: é o nome da variável, que o desenvolvedor define com base em algumas regras.
- O ponto e vírgula indica o fim de uma instrução.

Exemplos:

```
int idade;  
float minhaAltura;  
char resposta;
```

Observação:

Você pode declarar mais de uma variável do mesmo tipo na mesma instrução. Para isso basta separar cada uma das variáveis por vírgula.

Exemplo:

```
float salario,altura,media;
```

2.3 Regras para nomear uma variável

Você pode utilizar qualquer nome para uma variável, basta apenas seguir estas regras:

- Não começar com número

```
int 10valores; (ERRADO)  
int dezValores; (CORRETO)
```
- Não utilizar uma palavra reservada. Exemplos: auto, break, case, char, const, do, else,...

```
int break; (ERRADO)
```
- Não utilizar caracteres especiais (acentos, cedilha, ...), salvo o underline.

```
float salário; (ERRADO)
```
- Não utilizar espaço em branco no caso de um nome composto

```
int dez Valores; (ERRADO)
```

Observações:

- A linguagem C é case sensitive, em outras palavras a linguagem C faz diferença entre letras minúsculas e maiúsculas.
- Use nomes de variáveis que sejam autoexplicativos, ou seja, se for armazenar a idade chame a variável de idade ou idadeUsuario, algo assim.

Imagine a seguinte situação: você precisa armazenar o seu nome ou o seu endereço. Nesse caso, para armazenar caracteres em uma variável a mesma deveria ser uma string.

2.4 Strings

Uma string é uma série de caracteres, e um caractere é o mesmo que um byte. Em programação, cada caractere armazenado na memória de uma string é representado por um valor numérico.

Para representar uma string em C você deve fazer o seguinte:

char nomeDaVariável[tamanho];

Onde:

- char indica que se trata de armazenamento de caracteres;
- nomeDaVariável é o nome identificador que o desenvolvedor definiu;
- [tamanho] define o número de máximo de caracteres a serem armazenados na string.

2.5 Atribuição

Você pode guardar algum dado numa variável previamente declarada. Essa ação se chama atribuição. O sinal de atribuição é o =.

Exemplo:

```
int x;  
x = 2;
```

Observações:

- A variável que recebe o dado deve estar sempre à esquerda do sinal de atribuição;
- Para marcar a casa decimal utilize o ponto e não a vírgula.

3 Saída padrão

Para mostrar alguma informação na tela do computador você pode utilizar uma das funções existentes para isso. Uma delas é a uma função printf(). Para utilizá-la você deve incluir o arquivo stdio.h usando a diretiva include.

3.1 Formato geral

```
printf(" texto");
```

Exemplo:

```
printf("Bem vindo ao C\n");
```

Observação:

O \n insere uma quebra de linha na informação que é mostrada na tela do computador.

3.2 Mostrado o valor de uma variável

Se quiser mostra o valor de uma variável no meio de um texto, deverá indicar um %tipo dentro das aspas duplas, seguido pelo nome da variável onde a informação será buscada.

Os principais %tipo são os seguintes:

- %d ou %i para inteiros
- %f para float
- %c para 1 caractere
- %s para strings

Exemplo:

```
int x;  
x= 2;  
print("valor = %d",x);
```

4 Um programa completo em C

```
#include<stdio.h>
#include<stdlib.h>
int main( )
{
    printf("Bem vindo ao C\n");
    system("pause");
    return 0;
}
```

5 Exercícios

- 1) Implemente no computador o exemplo completo acima.
- 2) Crie uma variável para armazenar a sua idade. Atribua um valor a essa variável e mostre o valor armazenado.
- 3) Crie uma variável para armazenar o salário de alguém. Atribua um valor a essa variável e mostre o valor armazenado.