Unidade 1

Seção 3

Algoritmos e Técnicas de Programação

iStock 2018

Webaula 3

Componentes e elementos de linguagem de programação

Após termos visto os algoritmos, as famílias das linguagens de programação e as possibilidades profissionais de um programador, nesta webaula, vamos estudar os componentes de um programa de computador, as variáveis e constantes utilizadas em uma linguagem de programação, as operações e as atribuições, assim como, as estruturas dos algoritmos e programas.

iStock 2018

Um **programa** é uma sequência de código organizada de tal forma que permita resolver um determinado problema.

Para Criar um programa

Para Executar um programa

As sintaxes (instruções) devem ser escritas e armazenadas na memória do computador na mesma ordem em que se espera serem executadas, ou seja, ela pode ser linear (executada sequencialmente) e não linear (executada de forma a serem redirecionadas, isto é, uma instrução de bifurcação).

De maneira geral, a sequência de instruções para criação de um programa de computador ficaria assim:

1. Início do programa.

2. Definição das variáveis e de possíveis atribuições.

3. Instrução de leitura dos dados.

4. Instrução do formato de escrita.

5. Demais instruções e funções.

6. Fim do programa.

Algoritmo

```
Nome_Programa
      Declaração das variáveis
Início
       Ações
      .
      .
Fim.
```

Clique aqui para saber mais

Linguagem C

```
Definição das bibliotecas
Início do programa (main)
Início das funções
    Declaração das variáveis
    Instrução
        .
        .
Termino das funções.
```

Linguagem C

Quando iniciamos um programa na linguagem de programação C, as primeiras linhas de programação são definidas pelas bibliotecas, também conhecida como arquivos de cabeçalho.

Para inserir as bibliotecas no programa é necessário colocar:

`#include` - inclusão de um arquivo no programa fonteentre os símbolos de menor “<” e maior “>” (quando se usa < e > o arquivo é procurado na pasta include) o nome da biblioteca.

Explore a galeria e veja as principais bibliotecas em linguagem de programação C.

stdio

Essa biblioteca é responsável pelas funções de entradas e saídas, como é o caso da função printf e scanf que vamos aprender mais à frente.

Exemplo:

```
#include <stdio.h>
```

Função `main()`

Após a definição das bibliotecas, o programa é inicializado pela função (main). Exemplo:

```
main()
{
}
```

A "{" indica o início de uma função em C e a "}" indica o término das funções e do programa. Elas também podem indicar o início e o término de alguns contextos na programação.

Quando usamos a int antes de main() significa que retornará um número do tipo inteiro.

```
int main()
{
}
```

Também pode ser utilizada a função void, esta é uma função sem retorno, ou seja, não recebe nenhum argumento.

```
void main
()
{
}
```

Variáveis

As variáveis são locais reservados na memória para armazenamento dos dados, cada uma possui um nome próprio para sua identificação. Clique e conheça os tipos mais usados:

Inteiro

Real

Caractere

Constantes

Constante é algo que não se altera. Para Schildt (2005), as constantes são consideradas modificadores de tipo de acesso, ou seja, não podem ser alteradas, elas podem ser representadas pelo comando “const”.

Exemplos:

```
const int fixo=500

const int fixo= -100

const letra x= ‘a’

const texto t= “Conversão da temperatura de graus centígrados
para Fahrenheit”.
```

Operadores

Clique nos tipos para conhecer os principais operadores utilizados em algoritmos e em Linguagem C.

- Operadores Aritméticos
- Operadores de Incremento e Decremento
- Operadores Relacionais
- Operadores Lógicos
- Operadores de Atribuição

Comentários

Para fazer comentários em qualquer lugar do seu programa, basta utilizar barras duplas “//”.

Exemplo:

```
#include <stdio.h> // biblioteca para entrada e saída de dados
   int main() // comando de início e o mais importante do
programa
   { // início de uma função
   printf("Meu primeiro programa"); // comando para saída de
dados na tela return 0; // indica que o processo está voltando
para o Sistema Operacional
} // fim de uma função ou de um programa
```

Função `printf()`

É um comando de saída, o qual possui um vínculo com a biblioteca `stdio.h`. É utilizada quando se pretende obter uma resposta na tela do computador.

A sua síntese é definida por:

```
printf (“expressão de controle”, listas de argumentos);
```

Exemplo:

```
printf (“O valor encontrado foi  %d”, valor1);
```

Saiba mais sobre formatações utilizadas na função `printf()`

Em **algoritmos**, a função utilizada para saída de dados é a palavra "**escreva**", veja o exemplo a seguir:

```
Valor_program;
var
real: valor;
inicio
        escreva("Digite um valor"); // saída da informação na
tela do computador
        .
        .
Fim.
```

Função `scanf()`

É um **comando de entrada**, isto é, são informações que possibilitam a entrada de dados pelo teclado, assim, a informação será armazenada em um determinado espaço da memória, como o nome e tipo específico da variável. A sintaxe é definida por uma expressão de controle (sempre entre aspas duplas) e pela lista de argumento.

Exemplo:

```
scanf ("%d",
&valor);
```

No exemplo, o computador entrará com um valor decimal e retornará o valor da variável "valor".

O "&" é utilizado na função scanf() na lista de argumentos, sua função é retornar o conteúdo da variável, ou seja, retorna o endereço do operando.

A sintaxe da função scanf() é definida por:

```
scanf("expressão de controle", lista de argumentos);
```

Saiba mais sobre formatações utilizadas na função `scanf()`

Linguagem C

```
main()
{
int valor;
printf("Digite um número: ");
scanf("%d",&valor);
printf("\n o número é
%d",valor);
printf("\no endereço e
%u",&valor);
}
```

Algoritmo

Para realizar a entrada de dados em algoritmos utiliza-se a palavra “leia”

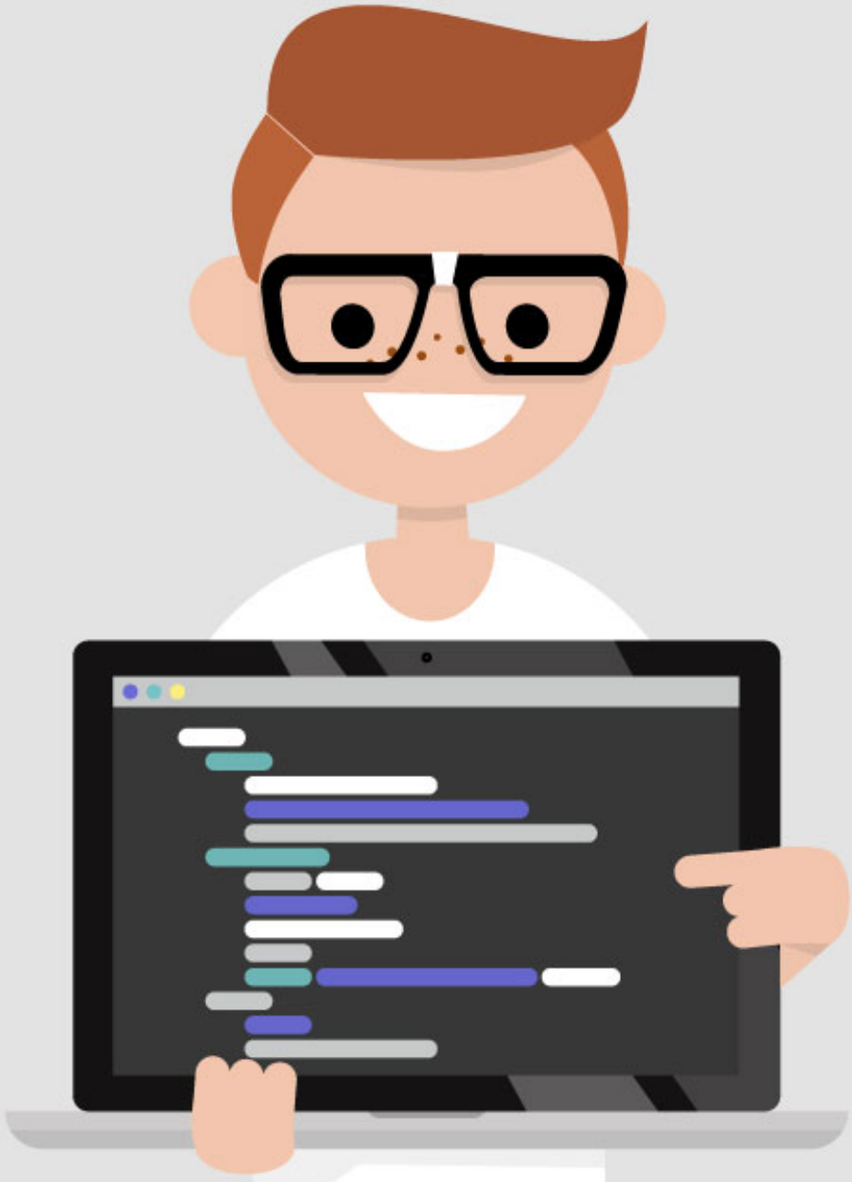
```
valor_program;
var
real: valor;
inicio
escreva("Digite um valor");
leia(valor); // valor de
entrada, será armazenado na
memória do computador
escreva("O valor digitado
foi:”, valor);
fim.
```

Instrução Return

Para um programa retornar ao sistema operacional, é necessário utilizar a instrução retorna zero “ `return 0` ”, assim como a instrução `system("pause")` tem a função de pausar a execução do programa, para que o resultado seja visualizado.

Veja o exemplo a seguir:

```
# include <stdio .h>
int main ( )
{
   int idade;
   printf ( "Digite a idade do
candidato" ) ;
   scanf ( "%d" , &id ) ;
   printf ( "O candidato tem  %d
anos !\n" , id);
   system (“pause”);
   return 0 ;
}
```



Para cada linguagem, existe uma sintaxe a ser seguida (os comandos usados na criação dos programas), assim, para um melhor entendimento, vimos os componentes de linguagem de programação envolvendo algoritmos e linguagem de programação C.

iStock 2018

Vídeo de encerramento

Você já conhece o Saber?

Aqui você tem na palma da sua mão a **biblioteca digital** para sua **formação profissional**.

Estude no celular, tablet ou PC em qualquer hora e lugar sem pagar mais nada por isso.

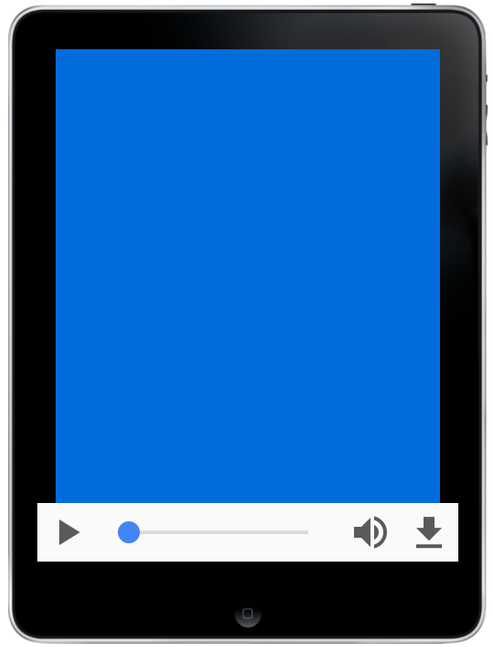
Mais de 450 livros com interatividade, vídeos, animações e jogos para você.

DISPONÍVEL NA Google play

Android:
https://goo.gl/yAL2Mv

Baixar na App Store

iPhone e iPad - IOS:
https://goo.gl/OFWqcq



Bons estudos!