

The background features a collection of 3D blocks in various colors (purple, blue, yellow, green, orange, pink, grey) arranged in a staggered, overlapping pattern. Some blocks are solid, while others have a square cutout in the center. Thin, curved lines in blue, red, and green sweep across the scene, adding a sense of movement and depth.

## Leitura complementar

# Processamento predefinido

- ✓ Processamento predefinido;
- ✓ Construção de processamento predefinido.

# Introdução

**Processamento Predefinido** é um programa que pode ser usado em outro programa. No contexto de linguagens de programação, um subprograma, sub-rotina, função ou procedimento consiste numa parte do programa que resolve um problema específico.

O conceito de **função** difere de **procedimento** porque ela retorna um valor, sendo que, em algumas linguagens, esta distinção não existe.

Uma **sub-rotina** pode ser usada em várias partes do programa ou sistema e assim podemos ter aproveitamento de uma determinada ação em diversos momentos diferentes, com a vantagem do gerenciamento da lógica dessa ação estar centralizado. Esta sub-rotina também pode ser reaproveitada em outros programas ou sistemas.

Os parâmetros são os argumentos do processamento predefinido. Eles são a comunicação da sub-rotina com os demais programas que a chamarão em algum momento e, através deles, a sub-rotina pode receber e retornar valores que serão processados para um objetivo final.

Sendo assim, no momento de se declarar um processamento predefinido, é necessário declarar de que tipo e quantos serão os parâmetros de entrada e de saída para que ela funcione corretamente.

Concluindo, quando notamos que, em um sistema que vamos desenvolver, será necessário usar várias vezes uma mesma rotina, criamos um programa só com ela e, quando necessário, o programa principal chama essa rotina, que será executada e depois retornará ao lugar de onde partiu com o resultado obtido e, então, o programa principal continua a partir da próxima linha de programação.

# Construindo um processamento predefinido

Para construir um processamento predefinido, considera-se um trecho de programa. Vejamos um exemplo de um algoritmo para:

- Receber 2 números nas variáveis **A** e **B**;
- Trocar os conteúdos das 2 variáveis, ou seja, o conteúdo da variável **A** deve ser transferido para a variável **B**, e o da variável **B**, para a variável **A**;
- Mostrar na tela os valores de **A** e **B**, para conferir se os conteúdos foram realmente trocados.

PROGRAMA1

← Programa principal

Declara A, B, C numéricas

A = 0

B = 0

C = 0

Ler A

Ler B

TROCAR(A,C)

TROCAR(B,A)

TROCAR(C,B)

Exibir A, B

FIM

TROCAR(X,Y numéricas)

← Processamento predefinido

Y=X

SAÍDA

Vejamos os detalhes do trecho de programa anterior:

- **TROCAR(X,Y numéricas)**: Aqui está o nome da rotina e entre os parênteses está sendo definido 2 variáveis numéricas (**X,Y**) e, portanto, quando algum programa principal chamar essa rotina, deverá enviar 2 valores para que ela seja executada;
- **Y = X**: A variável **Y** está recebendo o mesmo valor da variável **X**;
- **SAÍDA →** Saída da rotina. Retorna a execução para a próxima linha do programa que a chamou.

No programa principal, as variáveis **A**, **B** e **C** foram definidas inicialmente com valor igual a zero (0), depois as variáveis **A** e **B** receberam os valores que foram fornecidos pelo usuário e depois a rotina **TROCAR( )** foi chamada levando esses valores.

Para entendimento, vamos supor que os valores digitados pelo usuário foram **1** e **2**, ou seja, **A=1** e **B=2**.

Na rotina **TROCAR( )**, as variáveis **X** e **Y** recebem os valores das variáveis na ordem em que foram enviadas, portanto:

- **TROCAR(A,C)**: Chama a rotina e faz com que **X** receba o valor de **A**, e **Y** receba o valor de **C**:

**X=1**  
**Y=0**

Depois é executado o cálculo existente:

**Y=X**  
**Y=1**

Ao retornar ao programa principal, **A** continua com **1**, **B** continua com **2**, e **C**, que recebeu o valor de **Y**, fica com **1** após o cálculo.

- **TROCAR(B,A)**: Chama a rotina e faz com que **X** receba o valor de **B**, e **Y** receba o valor de **A**:

**X=2**  
**Y=1**

Depois é executado o cálculo existente:

```
Y=X  
Y=2
```

Ao retornar ao programa principal, **B** continua com 2, **C** continua com 1, e **A**, que recebeu o valor de **Y**, fica com 2 após o cálculo.

- **TROCAR(C,B)**: Chama a rotina e faz com que **X** receba o valor de **C**, e **Y** receba o valor de **B**:

```
X=1  
Y=2
```

Depois é executado o cálculo existente:

```
Y=X  
Y=1
```

Ao retornar ao programa principal, **C** continua com 1, **A** continua com 2, e **B**, que recebeu o valor de **Y**, fica com 1 após o cálculo.

Após a execução do programa, o valor atual de **A** é o valor antigo de **B** e vice-versa. A variável **C** foi utilizada somente para auxiliar a operação.

Sendo assim, durante o decorrer do programa, as variáveis receberam os seguintes valores:

**A:** 0, 1 e 2

**B:** 0, 2 e 1

**C:** 0 e 1

- **Exibir A, B**: Exibe na tela os valores atuais de **A** e **B** (2 e 1), mostrando que os conteúdos de **A** e **B** foram realmente trocados.

O resultado mostrado representa o **teste de mesa**, ou seja, a simulação, passo a passo, da execução de um algoritmo.

A seguir, temos o fluxograma do programa principal e da rotina:

