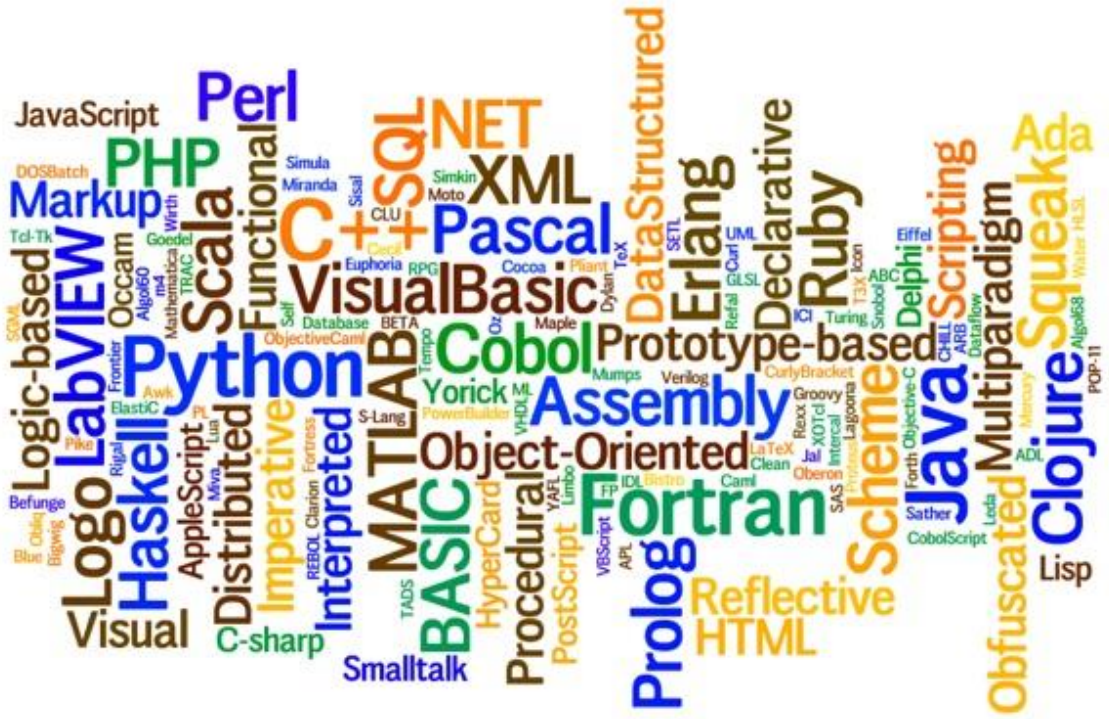
Introdução à Programação #3

Linguagem de programação

O conceito linguagem de programação ainda não foi abordado. Antes de continuar, é fundamental que compreenda o que realmente é uma linguagem de programação.



Uma **linguagem de programação** é uma forma padronizada para comunicar instruções a um computador ou seja, através de diversas regras semânticas e sintáticas, podemos instruir o computador a fazer algo.

Paradigmas de Programação

Todas as linguagens de programação têm várias características dentro delas os **paradigmas de programação**. Paradigmas são modelos ou padrões logo, paradigmas de programação são formas de estruturar o código.

Cada linguagem de programação pode adotar um ou mais paradigmas de programação. As que adotam mais do que um paradigma chamam-se **multi-paradigma**.

Paradigma imperativo

O paradigma imperativo concentra-se num **estado** (que são as variáveis) e **ações** (comandos) que **modelam** o estado.

Pode ser comparado ao modo imperativo da linguagem humana visto que é criado para ordenar a realização de ações.

Exemplos (de linguagens de programação imperativas): C, Java, C#, Pascal.

Paradigma procedimental

O paradigma procedimental permite a reutilização de código sem o copiar através da utilização de funções e procedimentos (que falaremos mais à frente na saga). De certa forma, com este paradigma, podemos "reciclar" código. A maioria das linguagens de programação são procedimentais.

Paradigma estruturado

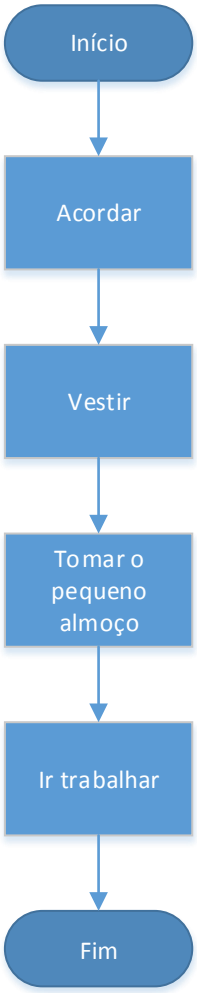
Uma linguagem de programação estruturada é aquela em que todos os programas podem ser reduzidos a três estruturas: sequência, decisão e repetição (iteração).

Sequência

Nesta estrutura as tarefas são executadas de forma linear, ou seja, uma após a outra. Exemplo:

```
1. Acordar;
2. Vestir;
3. Tomar o pequeno-almoço;
4. Ir trabalhar;
```

Fluxograma correspondente:



Este é um exemplo de uma sequência. Em muitas linguagens de programação, os comandos/ações terminam com ponto e vírgula pois estas permitem que os comandos sejam colocados em linha da seguinte forma:

```
1.  Acordar; Vestir; Tomar o pequeno-almoço; Ir trabalhar;
```

A utilização do ponto e vírgula é, normalmente, obrigatória quando existem várias instruções numa única linha. Linguagens de programação como Java, C# e C obrigam ao uso do ponto e vírgula em todas as instruções independentemente se estão em linha ou não.

Decisão

Neste tipo de estrutura, um determinado trecho de código é executado ou não dependendo do resultado de um teste lógico. Abaixo encontra vários exemplos práticos.

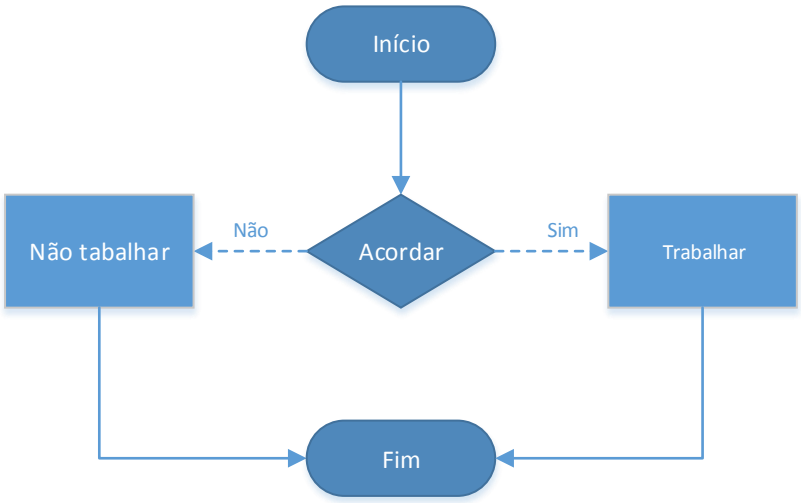
Exemplo 1:

O exemplo abaixo descreve a condição/decisão “se Acordar, vou Trabalhar, caso contrário, não vou Trabalhar” através de pseudocódigo.

```
1.  if "Acordar" then
2.    "Trabalhar"
3.  else
```

```
4.    "Não trabalhar"
5.  endif
```

Fluxograma correspondente:



O pseudocódigo acima já não está em português e já se assemelha ao que irá visualizar nas linguagens de programação. Sempre que aparecerem termos em inglês no código relacionado com a sintaxe, encontrará a sua tradução mais abaixo. Neste caso:

- **if** → se
- **then** → então
- **else** → caso contrário
- **endif (end if)** → fim do if

Agora, e novamente relacionado com a decisão, repare que o trecho de código "Trabalhar" só será executado **se** o indivíduo "Acordar". **Caso contrário**, o trecho "Não trabalhar" será executado. Abaixo pode visualizar mais exemplos.

Exemplo 2:

Dói-me a cabeça. Se doer muito pouco, vou trabalhar. Se doer pouco, tomo um comprimido e vou trabalhar. Se doer muito, vou ao médico e falto ao trabalho.

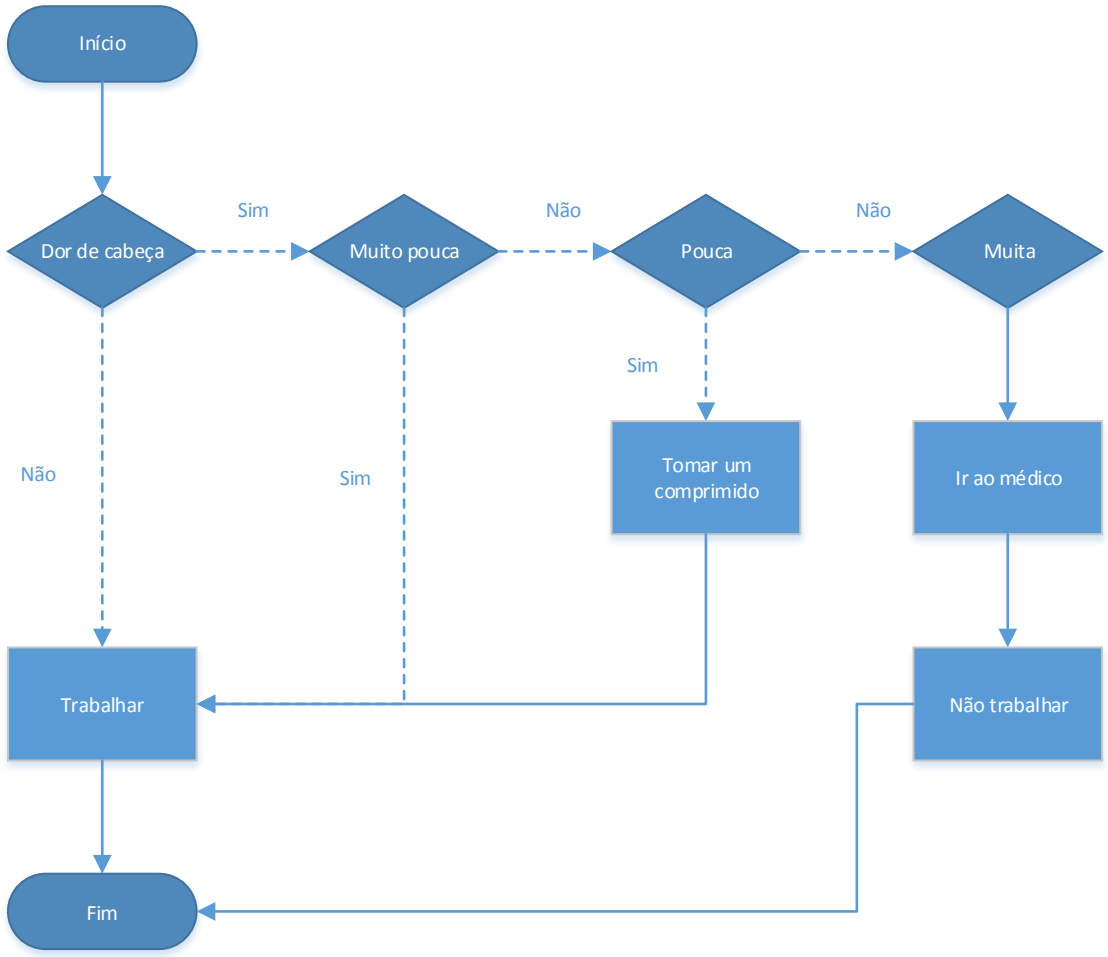
```
1.  case "Dor de cabeça"
2.    when "muito pouco" then "trabalhar"
3.    when "pouco" then "tomar comprimido"; "trabalhar"
4.    when "muito" then "ir ao médico"; "não trabalhar"
```

Este é mais um exemplo mas utilizando diferentes comandos. Este trecho poderia ser também escrito através de comandos if/else da seguinte forma:

```
1.  if "Dor de cabeça"
2.    if "muito pouco" then
3.      "trabalhar";
4.    elseif "pouco" then
```

```
5.      "tomar comprimido";
6.      "trabalhar";
7.     else if "muito" then
8.      "ir ao médico";
9.      "não trabalhar";
10.    endif
11. endif
```

Em fluxograma:



Novos termos:

- **case** → caso
- **when** → quando
- **else if** → caso contrário se

Iteração

Neste tipo de estrutura, também conhecido como repetição, um trecho de código será repetido um número finito de vezes dependendo do resultado de um teste lógico.

Exemplo 1:

Abaixo encontra a repetição “não saio de casa enquanto não estiver vestido” em formato de pseudocódigo para melhor análise:

```
1.  do {
2.    "não sair de casa";
3.  } while ( "não estou vestido" )
```

Ou seja, pode ler da seguinte forma: fazer x enquanto y.

Novo termo:

- **do** → fazer

Exemplo 2

Enquanto estiver a Dormir, não me vou Vestir.

```
1.  while ( "Durmo" )
2.    "Não me visto";
```

Ou seja, enquanto acontece algo, faço outra coisa.

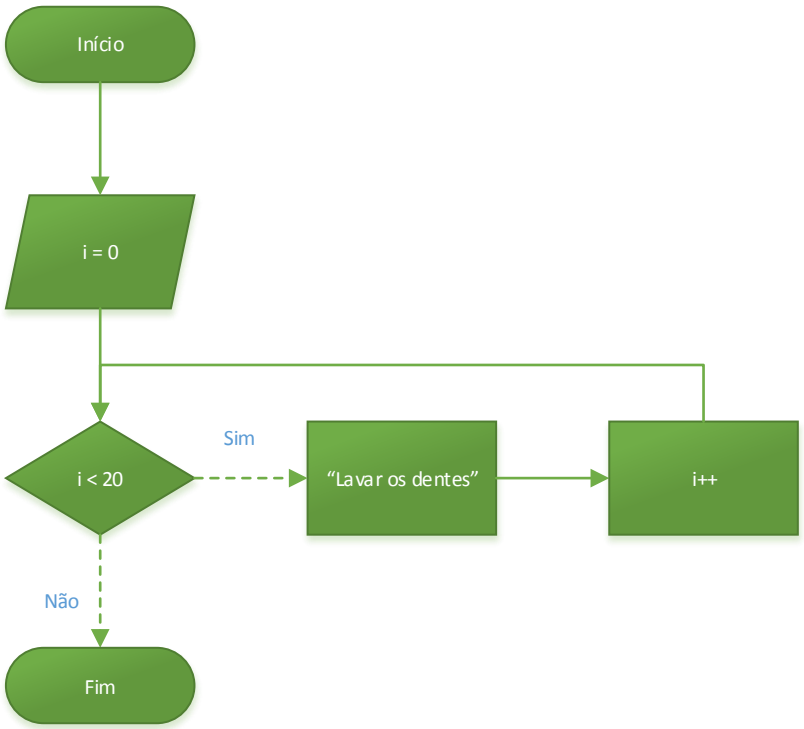
Exemplo 3

Lavo os dentes 20 vezes.

```
1.  for ( i = 0; i < 20; i++ )
2.    "Lavar os dentes"
```

Ou seja, enquanto não acontece qualquer coisa, faço qualquer coisa.

Em fluxograma:



Novo termo:

- **for** → para/enquanto

Exemplo 4

Para cada dente, lavo-o muito bem.

```
1.  for each "dente" in "boca"
2.    "Lavar muito bem"
```

Ou seja, para cada item do conjunto, fazer qualquer coisa. Novos termos:

- **each** → cada
- **in** → em

Paradigma Declarativo

O Paradigma Declarativo contrasta com o Imperativo pois é capaz de expressar a lógica sem descrever como o fluxo de comandos funciona, ou seja, apenas diz ao computador **o que** fazer e não **como** fazer.

Um excelente **exemplo** de uma linguagem que utiliza este paradigma é **Prolog**, muito utilizado na área de inteligência artificial.

Paradigma Funcional

O Paradigma Funcional engloba todas as linguagens de programação que utilizam funções matemáticas. Estas linguagens de programação são muito utilizadas no campo da matemática.

Exemplos: Matlab, Wolfram Language/Mathmatica/M, B.

Paradigma Orientado a Objetos

A Programação Orientada a Objetos permite a criação de **objetos** com base em **classes**. Estes objetos são instâncias dessas classes e possuem todos os atributos e funções presentes nas classes em questão.

Este paradigma é muito extenso e tem muita informação que mais à frente irá ser abordada. Atualmente, existem muitas linguagens que utilizam este paradigma.

Exemplos: Java, C++, C#, PHP. Os paradigmas de programação não se limitam aos 6 (seis) apresentados pois existem inúmeros outros.

Estes são os mais utilizados. Existem ainda paradigmas que são baseados noutros, contrastando com outros, etc.

Agradeço à Ana Narciso pela disponibilização de alguns exemplos utilizados na seção "Paradigmas de Programação".

Artigo original:

http://pplware.sapo.pt/tutoriais/programacao/vamos-programar-introducao-a-programacao-3/


pplware
no comments