

Pseudocódigo

Pseudocódigo é uma linguagem de programação artificial que permite aos programadores de computador para desenvolver a lógica antes de executar um código particular. Pode ser definido como o plano de um código, que detalha uma etapa de ações (ou o processo) e a ordem em que os passos devem ser executados. Ele pode ser pensado como uma representação textual de um fluxograma, que reflecte os conceitos estruturais de código.

Background Fotografia de Pseudocódigo é um tipo de Inglês estruturado, que define os algoritmos do programa. Ele permite que os programas para se concentrar em um núcleo lógica algoritmos antes de defini-lo na sintaxe de programação formal. Pseudocde não é executável num computador. Em vez disso, ele ajuda a um código de projeto programador antes de sua implementação real.

Finalidade

O principal objetivo do pseudocódigo é descrever claramente o que um código (ou programa) é suposto fazer. Ele quebra um código em etapas e destaca sua seqüência de ação. Pseudocódigo permite que programadores e programadores para secar a executar um código para verificar se há erros e inconsistências antes de escrevê-lo em uma linguagem de programação.

Regras

Pseudocódigo é composto de frases, orações e palavras. Ele segue um padrão estruturado, com depoimentos escritos em uma ordem seqüencial . Pseudocódigo usa instruções de seleção (if, then e if, then , else), ea palavra-chave inicial em cada declaração precisa ser capitalizado . Ela exige uma declaração por sentença, que precisam ser recuados para mostrar hierarquia código.

Exemplo

$M = 10$ $N = 5$ (onde M e N são variáveis) IF ($M > N$) Então, de acordo com este exemplo , a condição de $M > N$ é verdadeira (uma vez que M é 10 e n é 5) , e uma instrução será executada.

Pseudocódigo facilita a implementação de um programa. É o passo intermediário entre um código de linguagem de programação formal e linguagem humana, e explica a finalidade de um código especial em termos leigos . É fácil de escrever e requer um simples editor de texto (Microsoft Word ou o Bloco de Notas) para criar declarações pseudocódigo . Pseudocódigo exige conhecimento mínimo de programação , e tem poucas regras formais de sintaxe. Ele economiza tempo que é outra forma gasto em códigos de depuração e testes, uma vez que permite que os programadores de detectar e corrigir erros.

Desvantagens

Pseudocódigo pode ser demorado e complicado quando a codificação um problema complexo. Não existem regras padronizadas para escrever pseudocódigo e isso varia de empresa para empresa e indivíduo para indivíduo. Pseudocde não oferece nenhuma representação visual do código, o que torna difícil determinar a validade de certas afirmações.

A nossa língua recorre a diversos afixos para formar palavras. Um dos termos mais habituais é pseudo, que permite fazer referência a que algo não é original, uma vez que é falso ou uma imitação.

Um código, por sua vez, é uma série de símbolos que fazem parte de um sistema e que têm um determinado **valor** já atribuído. Os símbolos incluídos num código são combinados, respeitando regras e permitindo transmitir uma mensagem.

Estas definições permitem-nos compreender o que é um pseudocódigo, uma noção que é usada no âmbito da informática.

Trata-se de uma falsa linguagem, tendo em conta que recorrer às normas de estrutura de uma linguagem de programação ainda é desenvolvida para que possa **ser** lido por um ser humano e não interpretado por uma máquina.

O pseudocódigo, neste sentido, é considerado como uma descrição de um algoritmo que é independente de outras linguagens de programação. Para que uma pessoa possa ler e interpretar o código em questão, são excluídos diversos dados que não são chave para sua compreensão.

Um pseudocódigo, por conseguinte, é algo que é usado quando se pretende descrever um algoritmo sem a necessidade de difundir quais são os seus princípios básicos. Desta forma, um ser humano encontrará maiores facilidades para compreender a mensagem, ao contrário daquilo que aconteceria se estivesse diante de uma linguagem de programação real.

Na hora de levar a **cabo** a criação de um pseudocódigo, é necessário que a sua estrutura seja composta por seguintes partes:

- Um cabeçalho, que, por sua vez, deve ser composto por cinco áreas diferenciadas, entre os quais o programa, o módulo, os tipos de dados, as constantes e as variáveis.

- O **corpo**, que se dividirá em início, instruções e fim.

Para além de tudo o que é aqui referido, é importante estabelecer outra série de dados de interesse relativos a qualquer pseudocódigo:

- Deve-se **poder** executar em qualquer ordenador. -Não tem nada que ver com a linguagem de programação que se vai poder usar depois, isto é, que é independente relativamente ao mesmo.

- Tem que ser simples de usar e também de manipular.

- Deve permitir que se possa fazer a descrição de diversos tipos de instruções, tais como de processo, de controlo, de descrição, primitivas ou compostas.

- Na hora de poder realizar a criação desse pseudocódigo deve-se ter em conta que serão utilizados diversos tipos de estruturas de controlo. Neste **caso**, podemos dizer que são de três classes: selectivas, sequenciais e interactivas.

Estas características fazem com que os pseudocódigos sejam utilizados em obras científicas e educativas e nas etapas anteriores ao desenvolvimento de um software, em jeito de rascunho antes de proceder à programação.

A lógica de programação é a capacidade que todo programador precisa ter para resolver os problemas que aparecem no dia-a-dia.

A capacidade de dividir o problema em partes menores é uma etapa essencial da lógica de programação e precisa ser levada em consideração quando nos deparamos com qualquer exercício/desafio. É nesse ponto que entra o conceito de algoritmo, descrito, geralmente, como uma sequência lógica de ações capaz de resolver um problema.

É válido ressaltar, no entanto, que o conceito de algoritmo vai muito além da programação. Uma receita de bolo, por exemplo, é um exemplo simples de algoritmo. Até mesmo algo como “mascar um chiclete” pode ser descrito como um algoritmo, como mostrado na Código 1.

11 – pegar o chiclete

22 – retirar o papel

33 – mascar o chiclete

44 – jogar o papel no lixo

55 – jogar o chiclete no lixo

Código 1. Algoritmo para mascar chiclete.

A maioria dos algoritmos, obviamente, será mais complexa do que simplesmente “mascar um chiclete” ou “fazer um bolo”.

No caso da computação, uma soma de dois números é o exemplo mais clássico de um **algoritmo simples**. Algo como o mostrado na Código 2 é bastante comum para observarmos que temos uma sequência lógica de ações que envolvem os três elementos que comentamos anteriormente: entrada de dados, processamento e saída de dados.

11 – Inserir o primeiro número

22 – Inserir o segundo número

33 – Somar os dois valores

44 – Mostrar o resultado

Código 2. Algoritmo para soma de dois números.

Ações 1 e 2: entrada de dados;

Ação 3: processamento das informações inseridas;

Ação 4: saída de dados.

Introdução Ao Pseudocódigo

A representação de **algoritmos** pode ser feita de várias formas; a que mostramos na Código 2 é uma delas. Entretanto, essa opção não possui nenhum tipo de regra mais formal, o que faz com que esteja um pouco distante da programação propriamente dita. Para aproximar um pouco mais os dois conceitos, o pseudocódigo foi criado (um dos exemplos de pseudocódigo mais conhecidos no Brasil é o Portugal). Além dele, podemos encontrar várias opções. O objetivo de todos, no entanto, é um só: criar um código que se aproxime do que será visto em linguagens de programação como C, C# ou Java.

Utilizando o Portugal como exemplo, nosso algoritmo da soma de dois números seria algo como o mostrado na Código 3. Nessa “linguagem de programação”, é importante observar que temos um outro conceito, conhecido como regiões; nesse código, temos uma região de variáveis e outra de processamento. Esse conceito também é algo que será levado para as linguagens de programação e sempre deve ser considerado durante o desenvolvimento de software.

1início

2inteiro a, b, resultado

3

4ler (a)

5ler (b)

6resultado <- a + b

7escrever (resultado)

8fim

Código 3. Soma de dois números em pseudocódigo (Portugal).

Linha 01: Indicação do início do algoritmo em Portugal;

Linha 02: Região de definição de variáveis. As variáveis inteiras “a” e “b” são utilizadas para guardar os valores inseridos pelo usuário, enquanto o “resultado”, como o nome sugere, irá receber a soma deles;

Linhas 04 e 05: Lê os valores inseridos pelo usuário e os guarda em “a” e “b”. Note que, no Portugal, as ações são sempre definidas por verbos no infinitivo. Ler serve para indicar a leitura de informações do usuário (entrada de dados), e Escrever, para mostrar o resultado final para o mesmo (saída de dados);

Linha 06: Utilizamos o operador de atribuição <-. Esse operador atribui a soma de “a” e “b” à variável “resultado”. Essa é a fase de processamento dos dados de entrada;

Linha 07: Para finalizar o algoritmo, temos a amostragem dos dados para o usuário. Nesse caso, utilizamos o verbo Escrever para mostrar essa ação;

Linha 08: Indicação do fim do algoritmo em Portugol.

Com isso, podemos observar que o Portugol possui regras simples e bastante flexíveis. Ao começar a programar com Java ou C#, por exemplo, você irá notar que a proposta de se aproximar de linguagens como elas é obedecida. Se compararmos o código da Código 3 com um em C#, por exemplo, notaremos que não existe uma grande diferença. É claro que as ações (ler e escrever) e os operadores (atribuição e operador aritmético "+") serão substituídos por seus respectivos na linguagem, mas a estrutura será semelhante à apresentada no exemplo.

Existe uma linguagem "básica" para quem quer aprender a programar, ensinada nos cursos de lógica da programação, o pseudocódigo. Ele não é uma linguagem "de verdade", mas sim uma maneira mais simples para aprender os fundamentos usados em todas as linguagens de programação.

Podemos começar com um exemplo simples. Vamos fazer um programa capaz de tomar uma decisão fácil. Ele pergunta a nota do aluno e diz se ele passou ou não. Para um ser humano isso seria um problema muito elementar, mas para o computador as coisas não são assim tão simples. Lembre-se de que ele é burro e precisa ser orientado passo a passo. Nosso programinha em pseudocódigo poderia ficar assim:

Escreva: "Qual é a nota do aluno?" leia nota se nota maior ou igual a sete então: escreva "Ele passou" senão: escreva: "Ele foi reprovado" fim do se fim do programa

Este programinha perguntaria a nota e, com base no número que for digitado, avisaria se o aluno passou ou não.

Ele poderia ser escrito em qualquer linguagem, mas a lógica seria a mesma. De acordo com os recursos oferecidos pela linguagem escolhida, ele poderia ter uma interface simples em modo texto, uma interface gráfica mais trabalhada, aparecer no meio de uma página web e assim por diante. O mesmo programinha escrito em shell script, a linguagem mais elementar que temos no Linux, onde utilizamos comandos de terminal, poderia ficar assim:

```
echo "Qual é a nota do aluno" read nota if [ "$nota" -ge "7" ] then echo "Ele passou" else echo "Ele foi reprovado" fi
```

As linguagens de programação são conjuntos de padrões e comandos que você pode usar para dar ordens para nossos amigos burros. Assim como nas línguas faladas, existem diferenças de sintaxe, gramática e existem linguagens mais simples ou mais complicadas de aprender e linguagens mais adequadas para cada tipo de tarefa a realizar.
