

Lista de Execício

Pedro Correa

11718563 - `pedro.figueiredo563@al.faj.br`

30 de agosto de 2020

1 De forma sucinta, explique com suas palavras o que é o compilador

O compilador é responsável por transformar a linguagem de programação que foi realizada em linguagem humana para a linguagem de máquina. Com o passar dos tempos, o compilador também foi tendo mais funcionalidades com ele, uma parte também é auxiliar o desenvolvedor a seguir algumas regras para fazer seu trabalho da melhor forma e também para mostrar-lhe alguns erros que podem existir antes do programa ser traduzido para linguagem de máquina, assim o desenvolvedor não precisa executar o seu código para ver se o que ele fez está ocorrendo certo, o compilador lhe auxilia nessa tarefa.

2 Com o objetivo de identificar as vantagens e desvantagens entre compiladores e interpretadores, comente as características dos métodos.

Um das maiores vantagens que o compilador possui sobre o interpretador é que o trabalho dele será transformar o código realizado em binário, com isso o desenvolvedor pode executar a parte do programa mais tarde sem precisar passar por todo o processamento de validação. Outra vantagem é a performance, pois não é necessário fazer uma ‘tradução’ da linguagem para outra linguagem e compilar como é o caso das linguagens interpretadas.

A vantagem do interpretador é que ele é utilizado em linguagens de alto nível, que no caso são mais próximos a linguagem humana e o desenvolvedor não precisa se preocupar com tantas coisas como em linguagens de baixo nível necessitam (exemplo seria o gerenciamento de memória). Outra vantagem é o feedback mais rápido de se algo não está correto na linguagem de programação que o desenvolvedor está trabalhando, agilizando assim o trabalho do programador.

3 Explique o que é o processo just-in time utilizado em alguns interpretadores.

Esse método é utilizado para fazer a compilação da parte de um programa que necessita ser executado naquela hora, sem precisar compilar todo o código fonte e que provavelmente não será utilizado naquele momento. Isto é muito bom para aproveitamento de memória RAM e rapidez para testar na hora do desenvolvimento.

4 É possível afirmar que uma GLC é suficiente para atender os requisitos das linguagens de programação?

Não, não é possível fazer todas as validações por uma GLC, ainda faltam a parte de validação da análise sintática.

Mas temos alguns marcadores que podem se beneficiar somente dessa etapa, o HTML é um deles.

5 Qual a diferença entre lexema e token?

Lexema: menor parte reconhecível do sistema semântico de uma língua.

Token: segmento de texto ou símbolo que pode ser manipulado por um analisador sintático.

6 O que é otimização de código?

A otimização de código é um método para diminuir a complexidade do código, complexidade se refere a quantas vezes algum pedaço de código será executado. Existe a medida da notação O , que faz a referência a complexidade do código,

Exemplo: $O(n)$, seria um código com complexidade linear, quanto mais dados terá que processar, mais demorado será a execução dele.

7 Dê dois exemplos de erros sintáticos (em uma linguagem de sua escolha)

Utilizado na linguagem de Javascript:

```
console.log('hello world  
console.log(hello world)
```

8 Dê dois exemplos de erros semânticos (em uma linguagem de sua escolha)

Utilizado na linguagem de Javascript:

```
const dividedByZero = 10 / 0  
const dividedByString = 10 / 'dez'
```