

Linguagens de Programação 2020.2

Aluno: José Ribamar Silva Soares Filho
Matrícula: 472687

Questão 01:

Os métodos de implementação são:

- **Compilação:** O arquivo de texto (arquivo possuindo a extensão da linguagem, Ex. “.c”) contendo as instruções em dada linguagem são passados por um compilador, que os reúne em um único executável em uma linguagem conhecida como linguagem de máquina, essa linguagem por ser mais próxima ao computador, possui apenas comandos que o computador deve fazer para a execução do programa, isso torna linguagens compiladas muito rápidas em execução, porém o processo de compilar um programa pode demorar um pouco de acordo com o processamento do computador. Exemplos de linguagens compiladas são: C e C++.
- **Interpretação:** Nesse modo, o arquivo de texto (arquivo possuindo a extensão da linguagem, Ex. “.py”) não passa por nenhuma alteração e é executado diretamente do arquivo ao ser passado por um interpretador de tal linguagem, pelo interpretador ter a dificuldade de traduzir as informações contidas no arquivo durante a execução, ela tende a ser mais lenta do que a linguagem compilada e requer mais espaço de memória, pois há dois processos sendo executados, o de interpretar e o programa que está sendo interpretado. Exemplos de linguagens interpretadas são: Python, JavaScript ou ECMAScript e PHP.
- **Híbrido:** Nesse modo, o arquivo contendo uma linguagem de alto nível (arquivo possuindo a extensão da linguagem, Ex. “.cs”), passa por um compilador para virar uma linguagem de intermediária ou bytecode para então ser interpretada. Por serem compiladas para um nível médio de interpretação, é mais fácil o interpretador ler a linguagem, o que a torna mais rápida que uma linguagem totalmente interpretada. Exemplos de linguagens híbridas são: C# e Visual Basic.

Questão 02:

Os comentários como o nome já sugere, servem para comentar ou explicar sobre tal parte do código. Os comentários são utilizados por muitos programadores como forma de facilitar o código escrito para ele mesmo e outros programadores. Outro uso popular é o chamado “comentar parte de um código”, que serve para checar se a algum bug e testar a execução do programa sem aquela linha ou trecho de código comentado, já que o compilador/interpretador ignora a parte comentada.

Em questões de legibilidade, um comentário de uma linha é muito útil para explicar sobre aquela linha, porém em trechos onde muitas linhas precisam ser explicadas, se torna uma bagunça, onde é preferível um comentário de múltiplas linhas no início explicando aquele trecho inteiro.

Se tratando de facilidade de escrita, o comentário de uma linha é o mais fácil e rápido de se usar, onde ele pode rapidamente negar uma linha inteira ou comentar logo no final dela, já o comentário de múltiplas linhas apesar de ter uma sintaxe um pouco mais difícil dependendo da linguagem utilizada, pode negar trechos inteiros de código com facilidade.

Quanto à confiabilidade, os usos de comentários de ambos os tipos aumentam a segurança ao ler aquele trecho de código e entender o que ele faz, possibilitando assim uma possível mudança com mais segurança.

E sobre os custos, é importante notar que o como o comentário é ignorado pelo compilador, ele não pesa em nada para o programa executável, assim não adiciona em nada na memória, porém, no código base deve se ter seus cuidados, por exemplo, em um código open source, comentários são bem vindos para ajudar possíveis contribuintes, já em código de empresas, o recomendado é se gastar com uma documentação do código, o que gera alguns custos a mais.