

LABORATÓRIO 0

1. Pesquise sobre as gerações das linguagens de programação e as diferenças entre:

- Linguagens não-estruturadas
- Linguagens estruturadas
- Linguagens orientadas a objeto

De acordo com os autores *Doris Appleby* e *Julius J. Vande Kopple* as linguagens de programação são divididas em quatro gerações:

- 1ª geração: linguagem de máquina (binário)
- 2ª geração: linguagens de montagem (assembly)
- 3ª geração: linguagens procedurais (não-estruturadas)
- 4ª geração: linguagens aplicativas (estruturadas, funcionais, orientadas a objeto etc.)

Linguagens não-estruturadas são caracterizadas principalmente pelo uso da estrutura de controle *goto*, que permite saltar o fluxo de execução para qualquer linha de código do programa.

Linguagens estruturadas introduzem estruturas de controle mais avançadas como laços *for* e *while*, blocos e sub-rotinas (funções ou métodos), permitindo agrupamento e reutilização de instruções.

Linguagens orientadas a objetos fazem a introdução de classes, que contém dados (atributos) e procedimentos (métodos) e podem ser instanciados em objetos. Também trazem novos conceitos como encapsulamento, herança, polimorfismo etc. Têm uma semântica mais intuitiva e permitem um melhor reaproveitamento de código.

2. Quais as linguagens de programação mais populares atualmente e em que tipos de aplicações elas são usadas?

JavaScript, Python, Java, PHP, C#, C++.

JavaScript e PHP são linguagens voltadas para web e juntas com Python e Java dominam a maior parte desse setor. Python e Java são utilizadas em várias áreas da computação por serem respectivamente fácil de aprender e compatível com várias plataformas. Enquanto C# e C++ são as principais linguagens quando o assunto é programar para Desktop.

3. Qual a diferença entre uma linguagem de programação compilada e uma interpretada? Existe diferença de desempenho entre elas? Dê exemplos de linguagens dos dois tipos.

Linguagem compilada é uma linguagem de programação que passa pelo processo de compilação e, depois, seu resultado (em código de máquina) é executado diretamente pelo sistema operacional ou processador. Exemplos são: C, C++, Pascal.

Uma linguagem interpretada, por outro lado, é processada por um interpretador enquanto está em execução, sendo convertida de forma mais rápida e menos eficiente. Exemplos: JavaScript, Python, PHP.

4. Quais os editores de código mais usados atualmente? Como o Visual Studio está posicionado nesse ranking?

De acordo com a Empresa Software.com, o Visual Studio está em segundo lugar, perdendo apenas para o Visual Studio Code, e na frente de: Notepad++, Sublime Text, Vim, IntelliJ etc.

5. Qual a diferença entre um editor de código e um ambiente de desenvolvimento? Cite exemplos de ambos. O Visual Studio está classificado em qual categoria?

Um editor de código é como um editor de texto especializado, com recursos que auxiliam a escrita de códigos de linguagem de programação.

Um ambiente de desenvolvimento integrado (ou IDE) é um programa que possui um editor de código integrado com várias ferramentas de apoio ao desenvolvimento de software, como um gerenciador de arquivos especializado, um compilador, um depurador, geradores de código etc. É nessa categoria que o Visual Studio está inserido.

6. Quais os principais ambientes de desenvolvimento para criação de código C++?

Visual Studio é considerada a IDE mais potente para desenvolvimento de código C++, mas está disponível apenas para Windows. Outras IDEs como Eclipse, NetBeans e CodeBlocks também estão no topo e são multiplataformas.

7. Quais os principais compiladores para a linguagem C++?

MinGW / GCC, Borland C++, Dev C++, Embracadero C++ builder, Clang, Visual C++, Intel C++, Code block. Muitos desses compiladores são integrados em IDEs de mesmo nome ou nome similar.