Listagem de linguagens

- ***Binário***
- ***Assembly***
- ***Linguagem embarcada***
- ***Linguagem compiladas***
- ***Linguagens interpeladas***
- ***Linguagens híbridas***
- ***Low code***
- ***Linguagens nativas***
- ***Linguagens dinâmicas***
- ***Linguagens estáticas***

Binário

Características:

A informação com que os computadores trabalham é baseada no código binário, de valência dois (número de símbolos distintos que constituem o alfabeto desse código). Este sistema é composto por dois dígitos (1 e 0). Cada dígito, chamado um bit, é representado por estados elétricos: ON (com corrente) - representa o número 1 e OFF (sem corrente) - representa o número 0.
Assim, a informação é representada por uma sequência de bits (impulsos elétricos). Os bits agrupam-se em conjuntos de oito - os bytes.

Diferenças:

É a linguagem de programação de baixo nível. Ele só pode ser representado por 0s e 1s. Anteriormente, quando temos que criar imagens ou mostrar dados na tela do computador, é muito difícil desenhar usando apenas dígitos binários (0s e 1s). Por exemplo: Para escrever 120 no sistema de computador sua representação é 1111000. Então, é muito difícil de aprender. Para superar esse problema, a linguagem assembly é inventada.

Assembly

Características:

Assembly é uma linguagem de programação de baixo nível que se comunica diretamente com o hardware do computador, utilizando instruções compreensíveis pelo processador. Ela oferece um controle detalhado sobre a CPU e a memória, permitindo aos programadores realizar operações precisas e eficientes. Embora seja poderosa para otimizações de desempenho e programação de sistemas embarcados, seu uso é menos comum devido à sua complexidade e à necessidade de conhecimento profundo da arquitetura do hardware.

Diferenças:

A linguagem Assembly é na realidade uma versão legível da linguagem de máquina. Ela utiliza palavras abreviadas, chamadas mnemônicos, indicando a operação a ser realizada pelo processador.
A linguagem assembly só é compreensível para seres humanos, não para computadores.
A linguagem assembly é fácil de entender pelo ser humano em comparação com a linguagem de máquina.
A execução é lenta em comparação com a linguagem de máquina.

Linguagem embarcado

Características:

Os sistemas embarcados (embedded systems) são sistemas computacionais baseados em microprocessador, projetados para realizar tarefas em tempo real, dentro de um sistema maior.

Vale dizer, em complemento, que esses sistemas também atuam de forma independente, ou seja, sem a necessidade de participarem de um sistema maior.

Eles **são programáveis** ou podem também ter funcionalidades fixas, e essa versatilidade é uma de suas principais vantagens.

Apesar de sua natureza computacional, alguns sistemas podem não ter **interface de usuário**, enquanto outros apresentam interfaces gráficas complexas, com elementos como botões, LEDs **.**

Diferenças:

Na programação embarcada, você escreve software para ser executado em sistemas embarcados, como microcontroladores, microprocessadores ou sistemas embarcados em geral.

Os sistemas embarcados são dispositivos autônomos dedicados a uma única tarefa ou conjunto de tarefas específicas.

As linguagens de programação comuns para desenvolvimento de sistemas embarcados incluem C, C++, Assembly e outras linguagens de baixo nível.

A programação embarcada geralmente envolve restrições de recursos, como memória limitada, poder de processamento limitado e espaço de armazenamento limitado.

Linguagem compilado

Características:

No processo de compilação, o código fonte em uma linguagem de alto nível é transformado em código objeto em linguagem de baixo nível. A análise (front end) envolve entender o código e gerar uma representação intermediária. A síntese (back end) constrói o código objeto. Em linguagens híbridas como Java, o compilador gera bytecode, uma forma de código de baixo nível, executável em diferentes plataformas pela Máquina Virtual Java.

Diferenças:

O compilador gera um arquivo executável que pode ser executado diretamente no sistema operacional.
Um compilador é um tradutor de linguagens de programação casuais para linguagens de programação do nível da máquina. Isso é, um programa que a partir do código de uma linguagem qualquer, realiza algumas etapas como a validação e, por fim, gera um ou mais arquivo(s) que na maioria das vezes é binário.

Linguagem interpretada

Características:

As linguagens interpretadas passam por um processo similar. Sua maior diferença é de que o programa utilizado para traduzir se chama “interpretador”, que ao invés de fazer a total conversão de uma vez só, ele fará a conversão analisando os códigos linha por linha, em um processo minucioso e lento, ao contrário do que acontece com o compilador. Porém, atualmente temos o desenvolvimento da compilação JIT, Just in time, que veio para aperfeiçoar a produção, mesclando esses dois conceitos.

Diferenças:

Utilizado para traduzir se chama “interpretador”, que ao invés de fazer a total conversão de uma vez só, ele fará a conversão analisando os códigos linha por linha, em um processo minucioso e lento, ao contrário do que acontece com o compilador.

Linguagem híbrida

Características:

O método híbrido, representa um meio termo entre os compiladores e os interpretadores. Ele traduz (compila) os programas em linguagem de alto nível para uma linguagem intermediária projetada para facilitar a interpretação. Depois, usa um interpretador.

Diferenças:

-Suporta múltiplos paradigmas de programação, como orientação a objetos, programação procedural e programação funcional.

-Exemplos: C++, Scala.

Low code

Características:

O low-code é uma abordagem de desenvolvimento de software que requer pouco ou nenhum código para construir aplicações e processos. Uma plataforma de desenvolvimento de low-code utiliza interfaces visuais com funcionalidades simples de lógica de arrastar e largar em vez de extensas linguagens de programação.

Diferenças:

A aplicação criada em Low Code exige algum nível de personalização, um recurso que os elementos existentes não oferecem, por exemplo, é possível por meio de programação acrescentar ou modificar o resultado final.

Linguagem nativas

Características:

Geralmente se refere a linguagens de programação que são compiladas para código de máquina específico da arquitetura do processador alvo.
As linguagens de programação nativas são compiladas diretamente para código de máquina, o que significa que o código resultante é executado diretamente pelo processador do computador, sem a necessidade de uma máquina virtual ou interpretador.

Diferenças:

O aplicativo nativo possui seu armazenamento em dispositivo smartphone ou tablet, sendo possível fazer o download do sistema operacional Android, através do Google Play, por exemplo. Esse app foi desenvolvido com a linguagem nativa das plataformas, o que explica a origem do seu nome.

Linguagem dinâmica

Características:

Linguagem dinâmica, também conhecida como linguagem de scripting, ainda é algo obscuro para a grande maioria dos desenvolvedores. Entender e aplicar seus conceitos é algo que pode causar espanto e muitas dúvidas. Porém, é importante conhecer as vantagens e destantagens deste tipo de linguagem, saber quando é válido ou não a sua utilização no desenvolvimento de sistemas.

Diferenças:

-Facilitam o desenvolvimento rápido, mas geralmente tornam a detecção de erros mais difícil, uma vez que as falhas só são percebidas quando o programa está em execução, o que é particularmente preocupante em projetos muito grandes**.**

Linguagem estáticas

Características:

As linguagens de programação podem ser classificadas como de tipo de dados estáticos (tripagem estática) ou de tipos de dados dinâmicos (tipagem dinâmica), dependendo da forma como tratam a verificação de tipos em tempo de execução.

Em linguagens de tripagem estática, os tipos de dados são verificados em tempo de compilação, o que significa que o compilador verifica se as operações usadas no código são compatíveis com os tipos de dados definidos.

Diferenças:

-Os desenvolvedores podem encontrar erros antes da efetiva execução do código, com a possível desvantagem de linguagens como esta não serem tão práticas quanto às mais atuais.