C# Fundamentals

Início

Deal

Apresentação:

- Conteúdo: Fundamentos de C#.
- Teremos uma abordagem teórica e prática.
- Será cedido slides contendo exemplos teóricos e links para documentações oficiais.
- Será cedido exemplos de código fonte.

Linguagens de programação:

- É uma linguagem formal que, através de uma série de instruções, permite que um programador escreva um conjunto de ordens, ações consecutivas, dados e algoritmos para criar programas que controlam o comportamento físico e lógico de uma máquina.
- Programador e máquina se comunicam por meio dessa linguagem, permitindo especificar, com precisão, aspectos como:

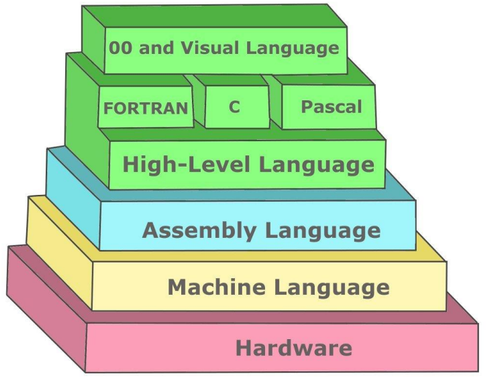
Linguagens de programação:

- ✓ Quais dados um software deve operar;
- ✓ Como esses dados devem ser armazenados ou transmitidos;
- ✓ Quais ações o software deve executar, de acordo com cada circunstância variável.
- ✓ Para explicar melhor, a linguagem de programação é um sistema de comunicação estruturado, composto por conjuntos de símbolos, palavras-chave, regras semânticas e sintáticas que permitem o entendimento entre um programador e uma máquina.

Linguagens de programação:

Que tipos de linguagem de programação existem?

- **A linguagem de programação é a base para a construção de todos os aplicativos digitais usados no dia a dia e são classificados em dois tipos principais:**
- **Linguagem de baixo e alto nível.**



Linguagem de programação de baixo nível:

- São linguagens totalmente orientadas à máquina. Esse idioma serve como uma interface e cria um link inseparável entre hardware e software.
- Além disso, exerce controle direto sobre o equipamento e sua estrutura física. Para aplicá-la adequadamente, é necessário que o programador conheça muito bem o hardware. Essa categoria pode ser subdividido em dois tipos:
- Linguagem de máquina e Linguagem Assembly.

Linguagem de programação de baixo nível:

- **Linguagem de máquina:**
- **É o mais primitivo dos idiomas e é uma coleção de dígitos ou bits binários (0 e 1) que o computador lê e interpreta e é o único idioma que os computadores entendem. Exemplo: 10110000 01100001**
- **Não dá para entender muito bem o que ele diz, certo? Portanto, a linguagem Assembly nos permite entender melhor a que se refere esse código.**

Linguagem de programação de baixo nível:

- **Linguagem Assembly:**
- **A linguagem Assembly é a primeira tentativa de substituir a linguagem de máquina por uma mais próxima da usada por seres humanos.**
- **Um programa escrito nessa linguagem é armazenado como texto (como nos programas de alto nível) e consiste em uma série de instruções que correspondem ao fluxo de pedidos executáveis por um microprocessador.**

Linguagem de programação de baixo nível:

- **Linguagem Assembly:**
- **No entanto, essas máquinas não entendem a linguagem Assembly. Portanto, devem ser convertidas em linguagem de máquina por meio de um programa chamado Assembler.**
- **Ele gera códigos compactos, rápidos e eficientes criados pelo programador que tem controle total da máquina.**

Linguagem de programação de alto nível:

- Sua sintaxe é voltada para o entendimento humano. Isso significa que esse tipo de linguagem é muito mais intuitivo, amigável e mais fácil de aprender.
- Elas abstraem conceitos voltados para a máquina e sintetizam comandos. Por exemplo: “print” e “delete” são instruções comuns nessas linguagens que, como os termos sugerem, servem para mostrar uma informação na tela ou deletar algum dado, respectivamente.
- Exemplo: JavaScript, Java, Ruby, Python, C#; Swift, Php

Linguagem compilada:

- **Linguagem compilada é uma linguagem de programação em que o código fonte, é executado diretamente pelo sistema operacional ou pelo processador, após ser traduzido por meio de um processo chamado compilação, usando um programa de computador chamado compilador, para uma linguagem de baixo nível. Exemplo: C, C++, COBOL, Visual Basic, etc...**

Linguagem interpretada:

- **Linguagem interpretada é uma linguagem de programação em que o código fonte nessa linguagem é executado por um programa de computador chamado interpretador, que em seguida é executado pelo sistema operacional ou processador. Mesmo que um código em uma linguagem passe pelo processo de compilação, a linguagem pode ser considerada interpretada se o programa resultante não for executado diretamente pelo sistema operacional ou processador.**
- **Exemplo: C#, Java, Ruby, etc...**

Tipagem de dados:

- **Linguagens fortemente tipadas:**
- **Linguagens fortemente tipadas são aquelas em que todas as variáveis tem um tipo específico e seus tipos são importantes para a linguagem. Normalmente as linguagens fortemente tipadas também são estaticamente tipadas, o que significa que uma vez que a variável foi declarada com um tipo ela será até o seu fim do mesmo tipo. Também normalmente possuem declaração explicita de tipo onde o tipo da variável deve ser especificado. Exemplo: C#, Java, etc...**

Tipagem de dados:

- **Linguagens fracamente tipadas:**
- **Linguagens fracamente tipadas são aquelas que não se importam com o tipo de dados contido em uma variável. Permitem que o programador não tenha de fazer conversões de tipos (cast). Neste caso o exemplo abaixo funcionaria sem erros mesmo sendo de tipos diferentes.**
- **Exemplo: JavaScript, etc...**

Tipagem de dados:

- **Linguagens não tipadas:**
- **Linguagens não tipadas, são aquelas em que existe apenas um tipo genérico ou mesmo nenhum tipo de dados.**
- **Exemplos: Linguagem de montagem ou Assembly, BCPL, Perl, etc...**


Dúvidas?