



**NÃO ME
PERGUNTE
SE SOU CAPAZ
APENAS ME DÊ
A MISSÃO**

VICTOR GREEN



Desenvolvimento de Aplicações Desktop

Conceitos Básicos de Programação

Professor: Charles Leite

O Desenvolvimento de Programas

- ▶ A programação consiste em indicar como o computador (hardware) deve trabalhar ou realizar (executar) tarefas
 - ▶ Enviar mensagens eletrônicas, armazenar e recuperar dados, realizar exames e cirurgias, jogar e muitas outras coisas
- ▶ Porém, para que um computador consiga realizar esses serviços, deve-se fazer o que?

Escrever Programas

O Desenvolvimento de Programas

- ▶ Um programa consiste numa série de **INSTRUÇÕES** que indicam como o computador irá realizar seus serviços
- ▶ Um programa deve definir a ordem em que as instruções devem ser executadas pelo computador
- ▶ Geralmente, um computador possui muitos programas que podem ser executados “ao mesmo tempo”
- ▶ Programas constituem o grupo de softwares de um computador

O Desenvolvimento de Programas

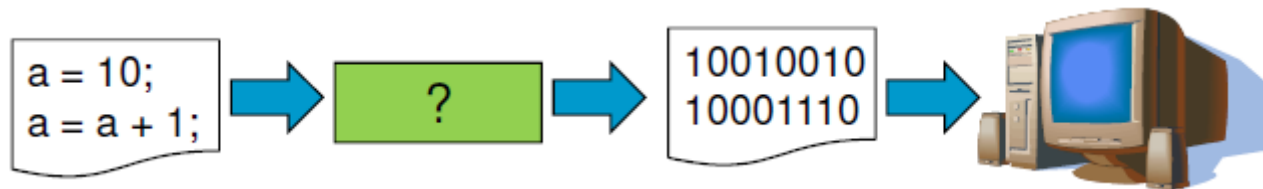
- ▶ Um programa contém instruções a serem executadas por um computador
- ▶ Quando solicitado, um computador executa as instruções do programa
 - ▶ O computador executa o programa
- ▶ Um programa pode ser executado mais de uma vez
- ▶ Um programa pode se comportar de maneira diferente nas várias execuções
 - ▶ Depende do comportamento do usuário, de outros programas, de hardware, etc.

O Desenvolvimento de Programas

- ▶ Um computador pode executar vários programas ao mesmo tempo
- ▶ Cópias de um mesmo programa podem ser executadas em vários computadores ao mesmo tempo
- ▶ Para executar um programa, um computador tem que ter uma cópia do programa
 - ▶ O programa tem que ser instalado ou carregado no computador
- ▶ Um programa define a ordem de execução das suas instruções pelo computador

Linguagens de Programação

- ▶ Para escrever programas, precisamos de uma linguagem de programação
 - ▶ Por exemplo: Java, C, C++, ...
- ▶ Esses programas, por sua vez, devem ser traduzidos para uma forma que um computador entenda e possa executá-los
 - ▶ Ou seja, uma linguagem que possa ser entendida pelo computador



Linguagens de Programação

- ▶ Eventualmente, erros podem ser gerados durante esse processo e devem ser corrigidos
 - ▶ Veremos em breve!
- ▶ Diversas ferramentas de software podem ser usadas para auxiliar em todo processo de construção de programas
 - ▶ Veremos em breve!

Linguagens de Programação

- ▶ Para ilustrar melhor o processo de construção de programas, vamos supor que você esteja dando orientações de viagem a um amigo
 - ▶ Você pode expressar essas orientações em qualquer linguagem (inglês, português, italiano, ...)
 - ▶ As orientações são as mesmas, independente da linguagem usada
 - ▶ Porém, a forma de expressar essas orientações é diferente em cada linguagem
 - ▶ Ou seja, para que essas informações sejam compreendidas, seu amigo deve ser capaz de entender a linguagem que você está falando

Linguagens de Programação

- ▶ Similarmente, um problema pode ser resolvido pela escrita de um programa, através de várias linguagens de programação (Java, C, C++, C#, Pascal etc.)
 - ▶ O propósito do programa é o mesmo, independente da linguagem que está sendo usada
 - ▶ Porém, as declarações usadas para expressar as instruções e a própria estrutura de organização dessas instruções variam de linguagem para linguagem
 - ▶ Para que o programa seja executado, um computador deve ser capaz de entender as instruções para que possa executá-las

Linguagens de Programação

- ▶ Nesse sentido, as linguagens de programação podem ser classificadas em quatro grupos:
 - ▶ Linguagens de máquina
 - ▶ Linguagens *assembly*
 - ▶ Linguagens de alto nível
 - ▶ Linguagens de 4ª geração

Linguagens de Programação

▶ Linguagem de máquina

- ▶ Para que um programa seja executado num computador, ele deve ser expresso na linguagem de máquina desse computador
 - ▶ Cada CPU tem a sua própria linguagem de máquina
 - ▶ Um programa escrito para uma estação de trabalho da Sun (processador Sparc) não pode ser executado em um computador da Dell (processador Intel)
- ▶ O código de um programa em linguagem de máquina é expresso por uma série de dígitos binários, sendo extremamente difícil de entender (ler e escrever) para os humanos
- ▶ A tarefa de escrever em linguagem de máquina consome muito tempo e é propensa a geração de muitos erros

Linguagens de Programação

- ▶ Linguagem *assembly*
 - ▶ Diante da dificuldade de se escrever em linguagem de máquina, surgiram as linguagens *assembly*, onde os dígitos binários foram substituídos por mnemônicos – palavras pequenas em inglês para representar comandos e dados
 - ▶ Mesmo assim, um código em *assembly* deve ser traduzido para linguagem de máquina a fim de ser executado
 - ▶ Mesmo sendo um avanço em relação às linguagens de máquina, escrever em *assembly* é ainda muito tedioso
- ▶ Linguagens de máquina e *assembly* são consideradas linguagens de baixo nível

Linguagens de Programação

- ▶ Linguagens de alto nível
 - ▶ Uma linguagem de alto nível é muito semelhante com uma linguagem natural, sendo de fácil compreensão pelos humanos
 - ▶ As linguagens de alto nível permitem que um programador ignore os detalhes específicos de uma linguagem de máquina
 - ▶ Mesmo assim, um código escrito em uma linguagem de alto nível deve ser traduzido para linguagem de máquina a fim de ser executado
 - ▶ Exemplos
 - ▶ Java, C, C++, C#, Pascal etc.

Linguagens de Programação

▶ Linguagens de 4ª geração

- ▶ Algumas linguagens de programação operam num nível ainda mais alto que as linguagens de alto nível
- ▶ Elas incluem facilidades especiais para geração de relatórios ou interação com banco de dados
- ▶ São as chamadas linguagens de 4ª geração, pelo fato delas terem sucedido as três primeiras gerações de linguagens
- ▶ Exemplo
 - ▶ SQL

Java

Sintaxe e Semântica de LP

- ▶ Uma linguagem de programação define as palavras e símbolos que podem ser usados para escrever um programa
- ▶ Assim como uma linguagem natural, uma linguagem de programação é definida por um conjunto de regras sintáticas e semânticas

Sintaxe e Semântica de LP

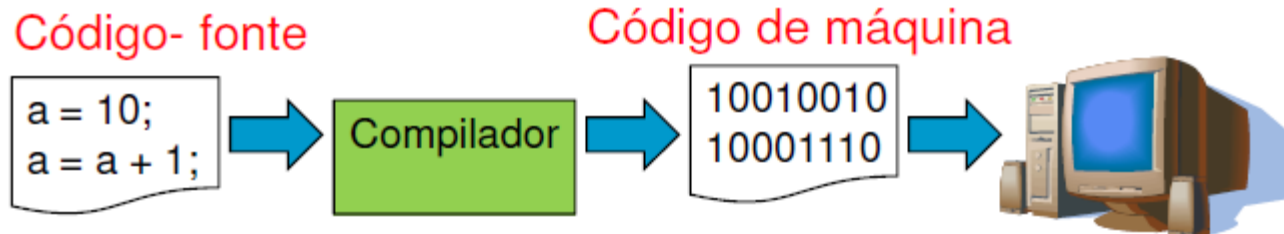
- ▶ A sintaxe define como as palavras e símbolos podem ser agrupados de maneira a formar instruções válidas em um programa
- ▶ A semântica de um programa define o significado das instruções declaradas nesse programa
 - ▶ Ou seja, o seu propósito ou função num programa

Tradutores

- ▶ Cada tipo de CPU executa programas apenas na sua própria linguagem de máquina
- ▶ Portanto, deve-se traduzir um programa, escrito numa linguagem de programação, para um programa equivalente na linguagem de máquina apropriada
- ▶ Portanto, precisamos de tradutores:
 - ▶ Compiladores
 - ▶ Interpretadores

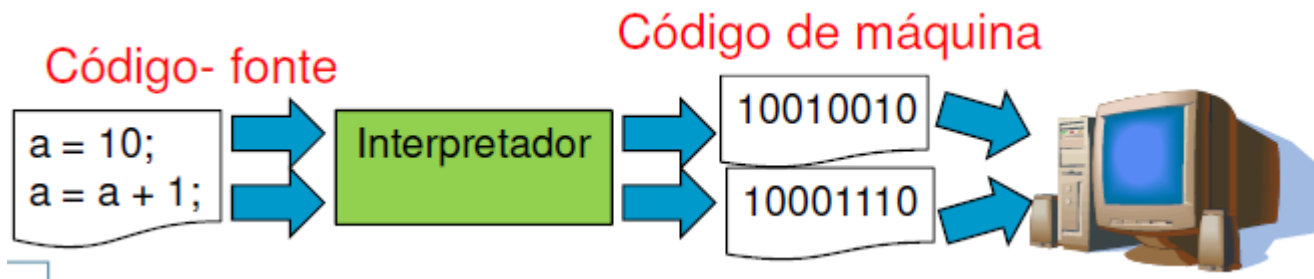
Tradutores

- ▶ Um **COMPILADOR** é um programa que traduz o código de um programa (código fonte) em uma linguagem num código equivalente numa outra linguagem
- ▶ O resultado desse processo é um código pronto para ser executado numa máquina



Tradutores

- ▶ Um INTERPRETADOR é similar a um compilador, porém com uma diferença importante
 - ▶ Um interpretador mistura as atividades de tradução e execução, ao mesmo tempo
 - ▶ Uma pequena parte do código fonte (uma declaração, por exemplo) é traduzida e executada
 - ▶ Em seguida, uma outra declaração é traduzida e executada, e assim por diante
 - ▶ Ou seja, um interpretador traduz instrução por instrução em linguagem de máquina e imediatamente



Tradutores

- ▶ Existem vários exemplos de linguagens tanto compiladas quanto interpretadas
- ▶ Java é uma linguagem que utiliza um processo híbrido de tradução (compilação + interpretação)
 - ▶ Veremos em breve!

Java

Editores

- ▶ Através de um editor, o código de um programa pode ser digitado no computador e salvo num arquivo
- ▶ Após editar e salvar um programa, ele deve ser traduzido da linguagem de alto nível em que ele foi criado para uma forma que possa ser executado numa máquina
 - ▶ Essa tradução pode gerar erros
 - ▶ Nesse caso, deve-se retornar ao editor a fim de fazer as modificações necessárias no código do programa
 - ▶ Uma vez que a tradução ocorra com sucesso, o programa pode ser executado e seus resultados avaliados
 - ▶ Se esses resultados não são o esperado ou se pretende melhorar o programa, novamente, o editor deve ser usado para alterar o código

Ambientes Integrados de Desenvolvimento

- ▶ Ambientes Integrados de Desenvolvimento – IDEs (*Integrated Development Environment*) – são ferramentas usadas no processo de desenvolvimento de programas, incluindo:
 - ▶ Editor de código, compilador, interpretador, depurador para encontrar e corrigir erros, gerador de documentação do código, ferramentas de arquivamento do código, ferramentas de visualização da estrutura do código

Erros em Programação

- ▶ Diversos tipos de problemas (erros) podem ocorrer com software, desde o seu processo de desenvolvimento até o seu uso
 - ▶ Erros em tempo de compilação
 - ▶ Erros em tempo de execução
 - ▶ Erros lógicos

Erros em Programação

▶ Erros de compilação

- ▶ O compilador verifica se o programa está em conformidade com a sintaxe (regras sintáticas) da linguagem
 - ▶ Qualquer declaração que não esteja de acordo com a sintaxe, o compilador gera um **ERRO DE SINTAXE**
- ▶ O compilador também tenta encontrar outros problemas como, por exemplo, o uso de tipos de dados incompatíveis
 - ▶ Nesses casos, a sintaxe pode até estar correta, mas houve uma tentativa de se fazer algo que a linguagem **SEMANTICAMENTE** não permite – **ERRO DE SEMÂNTICA**
- ▶ Enfim, qualquer erro identificado pelo compilador é considerado um **ERRO DE COMPILAÇÃO**
- ▶ Erros de compilação não permitem gerar uma versão executável de um programa

Erros em Programação

▶ Erros de execução

- ▶ Ocorre durante a execução do programa, causando a sua interrupção
- ▶ Nesse caso, o programa é compilado (ou seja, livres de erros de compilação)
- ▶ Porém, ao ser executado, um erro é encontrado
 - ▶ Por exemplo, uma tentativa de dividir um número por zero
- ▶ Nesse caso, o sistema interrompe o processamento do restante do programa

Erros em Programação

▶ Erros lógicos

- ▶ Nesse caso, o software compila e executa sem problemas
- ▶ Porém, produz resultados incorretos
- ▶ Por exemplo, um valor que é calculado incorretamente ou um botão gráfico que não aparece no lugar correto

Erros em Programação

O processo de encontrar e corrigir erros é chamado de
DEPURAÇÃO
(*debugging*)

DÚVIDAS ...



**CERTEZA QUE NÃO TEM MAIS
DÚVIDAS ?**

