

Métodos de implantação de LP's



Prof.: Claudio Junior (claudiojns@ufba.br)

Paradigmas de Linguagens de Programação (MATA56)

2023.1

Na aula anterior...

- O estudo de linguagens de programação é valido por uma série de motivos:
 - Aumenta nossa capacidade de usar diferentes construções para escrever programas;
 - Nos habilita a escolher linguagens de forma mais inteligente;
 - Nos ajuda a aprender novas linguagens mais facilmente
- Os critérios mais importantes para avaliação de linguagens são:
 - Legibilidade, Capacidade de escrita, confiabilidade e custo;

Agenda

- Métodos de implementação
- Compilação
- Interpretação Pura
- Sistemas de Implementação Híbridos

Custo/benefício no projeto da linguagem

Confiabilidade x Custo

- Java requer que todas as referências a vetores sejam checadas para garantir que os índices estejam dentro dos limites, mas isso aumenta o custo de execução

Legibilidade x Capacidade de escrita

- A LP provê muitos operadores poderosos (e uma grande quantidade de novos símbolos), permitindo que computações complexas sejam escritas em programas compactos, porém isso dificulta a leitura

Capacidade de escrita x Confiabilidade

- Ponteiros em C++ são poderosos e muito flexíveis

Outros critérios de avaliação

Portabilidade

- Quão facilmente um programa pode ser movido de uma **implementação** para outra.

Generalidade

- Seu uso em diversas aplicações (domínios).

Boa definição

- A precisão e a completeza da definição oficial da linguagem.



O que significa implementar uma linguagem de programação?

Implementadores

- Diferentes tipos de *software* que convertem o código escrito em um programa executável;
- Implementam uma linguagem de programação no sentido de torná-la utilizável para que programadores possam escrever e gerar programas.

Métodos de implementação

Compilação

- Programas são traduzidos para linguagem de máquina

Interpretação pura

- Programas são interpretados por outro programa conhecido como interpretador

Sistemas de Implementação Híbridos

- Um meio-termo entre compiladores e interpretadores puros

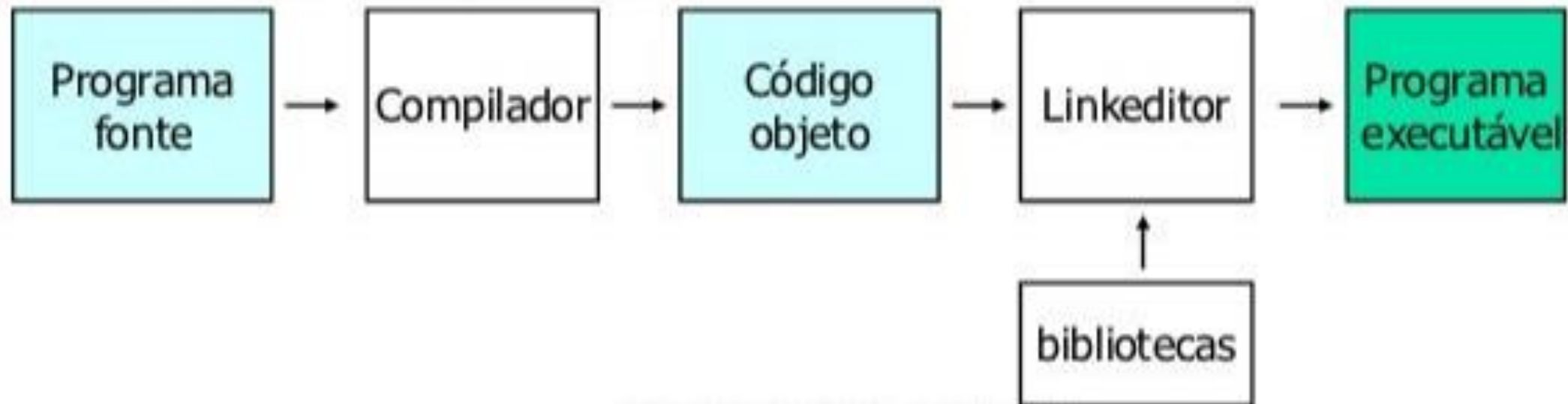
Implementadores

- Um implementador de linguagens de programação pode ser um **compilador**, um **interpretador** ou possuir as duas implementações (**híbrido**).

Compilação

- Traduz programas em alto-nível (linguagem fonte/código) em código de máquina (linguagem de máquina), ou seja, um formato executável pelo computador, gerando um novo *software*;
- Tradução lenta, execução rápida;
- O processo de compilação possui várias fases:
 - Análise léxica → Converte caracteres de um programa fonte em unidades léxicas;
 - Análise sintática → Transforma unidades léxicas em parse trees, as quais representam a estrutura sintática do programa;
 - Análise semântica → Gera código intermediário;
 - Geração de código → Código de máquina é gerado.

Processo de compilação



Integrated Development Environment (IDE)

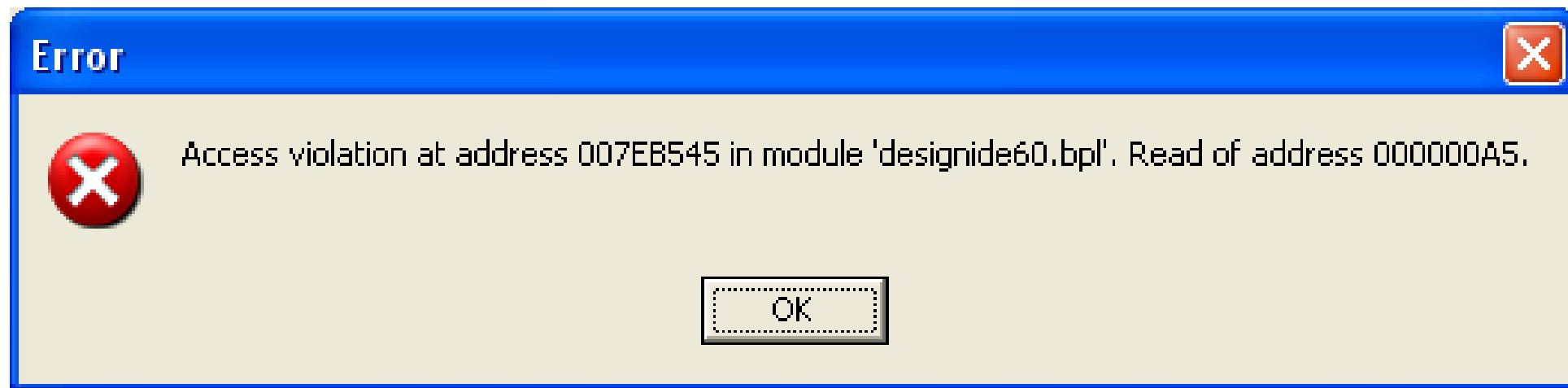
- Ferramentas de software que possuem editor de código, compilador e outros recursos;
- Eclipse, mais utilizada ???;
 - Gratuito;
 - Aceita plugins;
 - Implementa diferentes linguagens.
- E o Visual Studio?

Tipos de erros



- E se ocorrer erro na compilação ou **erro em tempo de compilação?**
 - Erro relacionado com a sintaxe da linguagem;
 - Fácil de encontrar;
 - Erro de digitação, falta de atenção,
- **Erro em tempo de execução:**
 - Compilação com sucesso;
 - Ocorre durante a execução do programa gerado;
 - Às vezes, mais difícil de encontrar;
 - Erro de lógica, conceitual, estouro de memória, violação de acesso de memória...

Tipos de erros



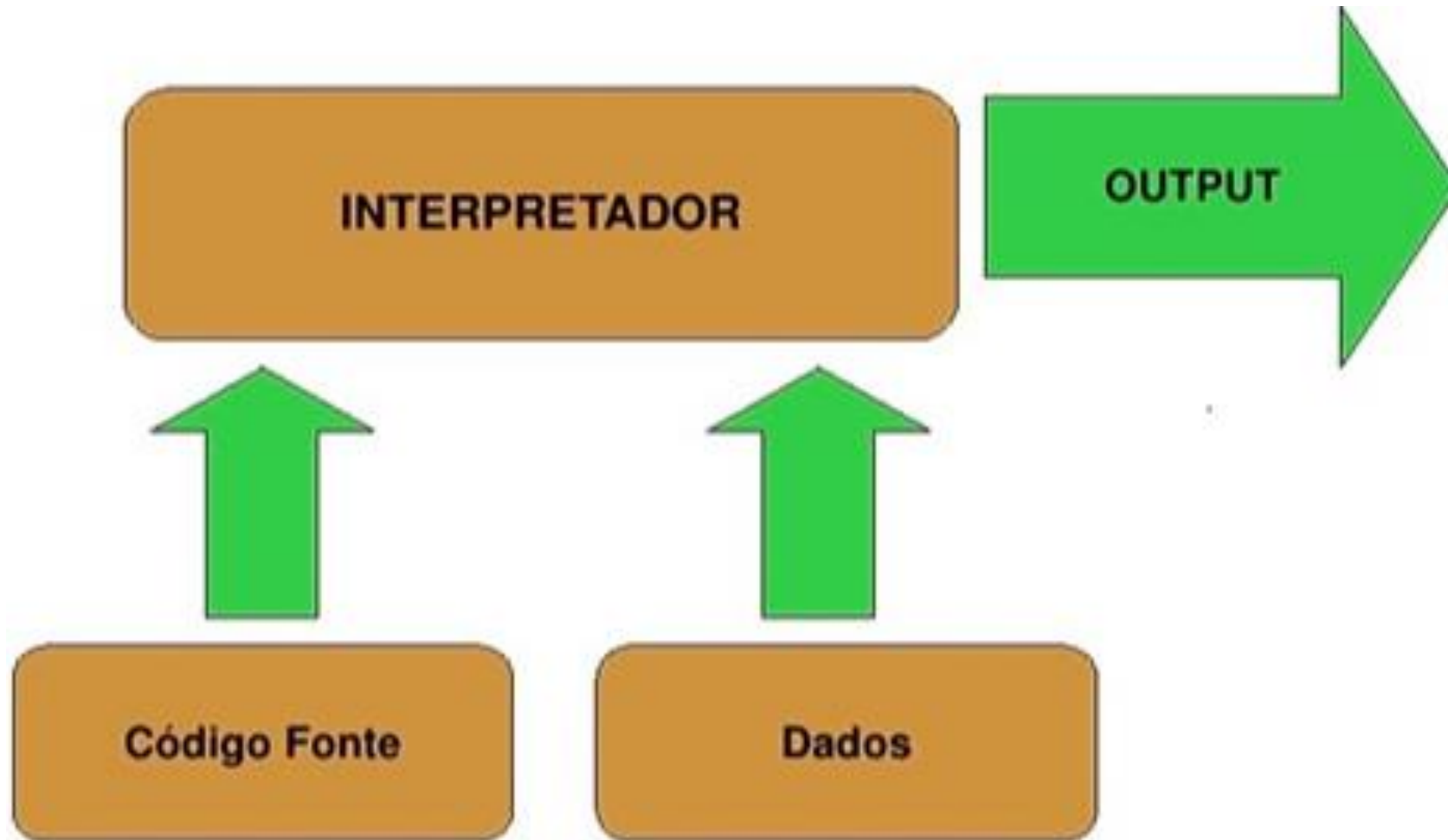
Linguagens Compiladas

- Linguagens tipicamente processadas por compiladores, embora, teoricamente qualquer linguagem pode ser compilada ou interpretada;
- C, C++, C#, Objective-C, Fortran, Basic, Cobol, Pascal, Ada, ALGOL, SMALL, Common Lisp, Delphi, Visual Basic, Visual Foxpro...

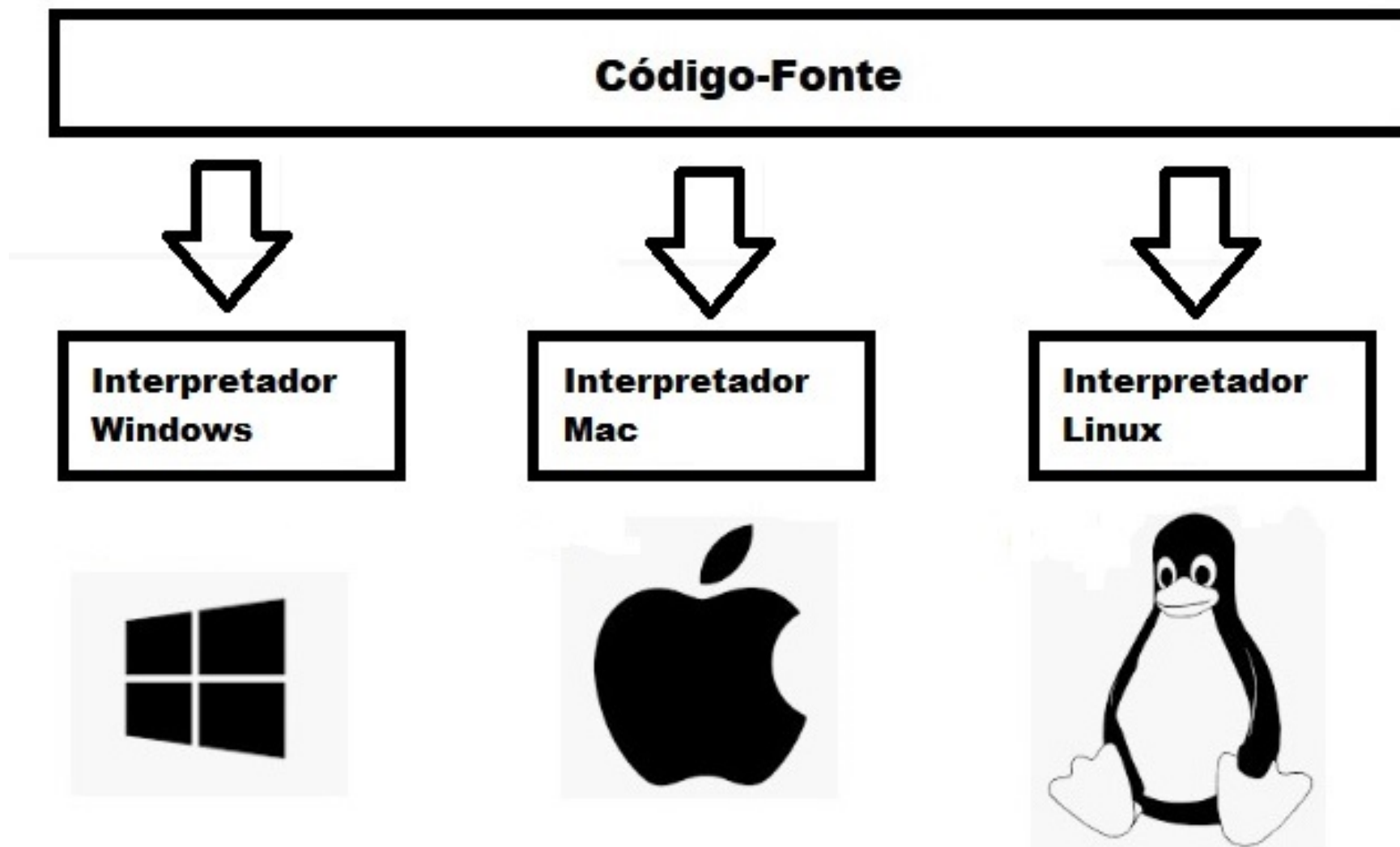
Interpretação Pura

- O texto do programa é executado à medida que vai sendo traduzido, num processo de tradução de trechos seguidos de sua execução imediata:
 - Programa foi interpretado;
 - Mecanismo utilizado para a tradução é um interpretador.
- Fácil implementação de programas (erros de execução podem ser fáceis e rapidamente mostrados);
- Execução lenta (de 10 a 100 vezes mais lenta do que programas compilados);
- Geralmente requer mais espaço;
- Cada vez mais raro em linguagens de alto-nível.

Interpretação



Interpretação



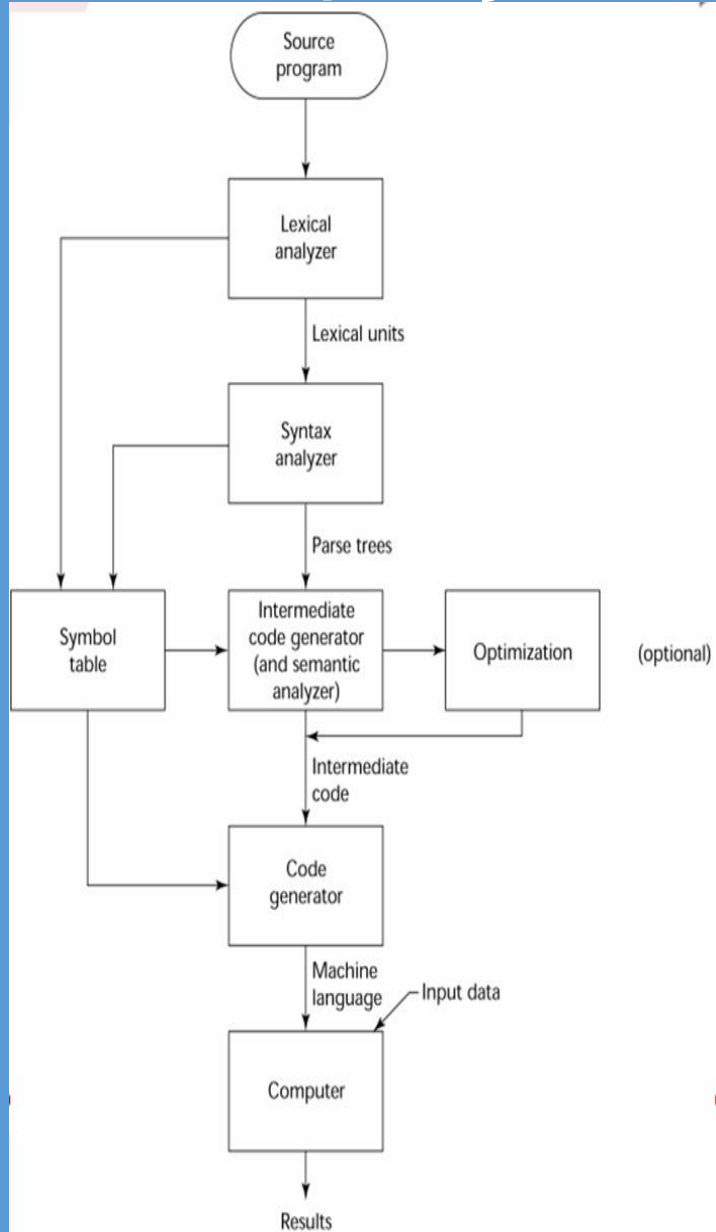
Interpretação

- Nem sempre é trivial distinguir entre linguagens interpretadas e compiladas;
- Há outras formas de interpretar, *Just in Time Compilation* (JIT):
 - Os códigos fontes, no lugar de serem interpretados linha-a-linha, têm blocos “compilados” para a memória, de acordo com as necessidades;
 - Aumenta a performance dos programas quando os mesmos módulos são chamados várias vezes (ex: função chamada 1x, interpreta; n vezes, compila).
- Fases da tradução:
 - Análise léxica → a análise sintática (ou *parsing*) → a geração de código → a otimização.
 - Em compiladores também é comum a geração de código intermediário.

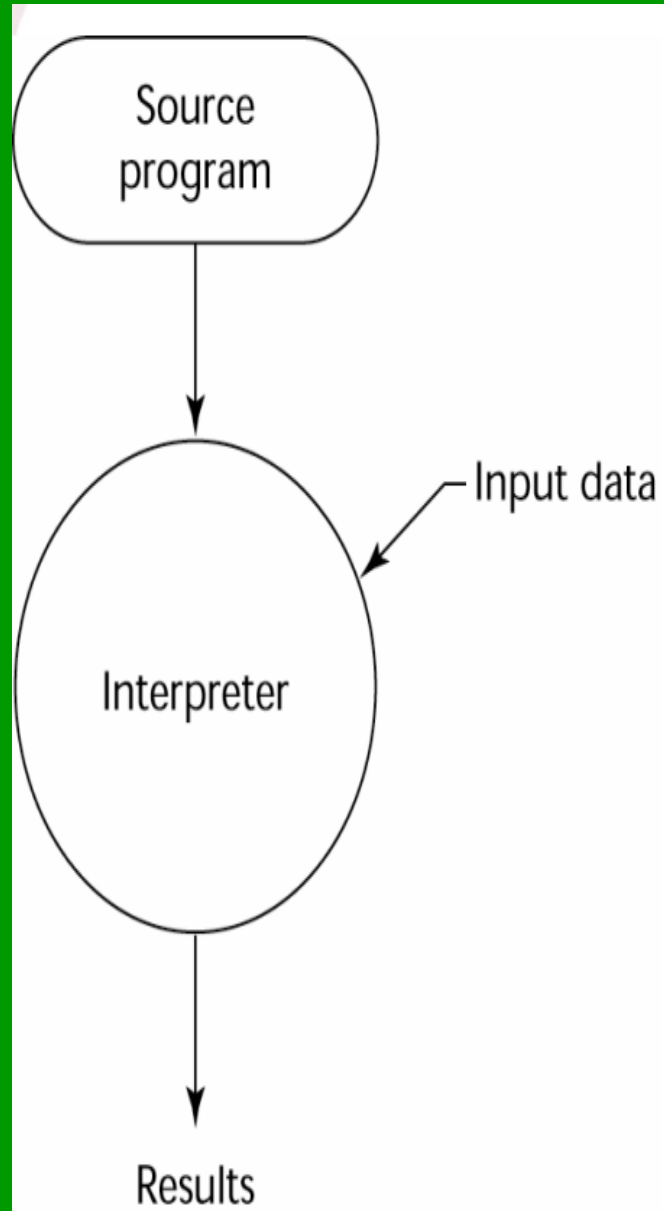
Sistemas de Implementação Híbridos

- Um meio-termo entre compilador e interpretador puro;
- Um programa em uma linguagem de alto-nível é traduzido para uma linguagem intermediária que permite fácil interpretação;
- Mais rápido do que interpretação pura:
 - Exemplos:
 - Programas em Perl são parcialmente compilados para detectar erros antes da interpretação

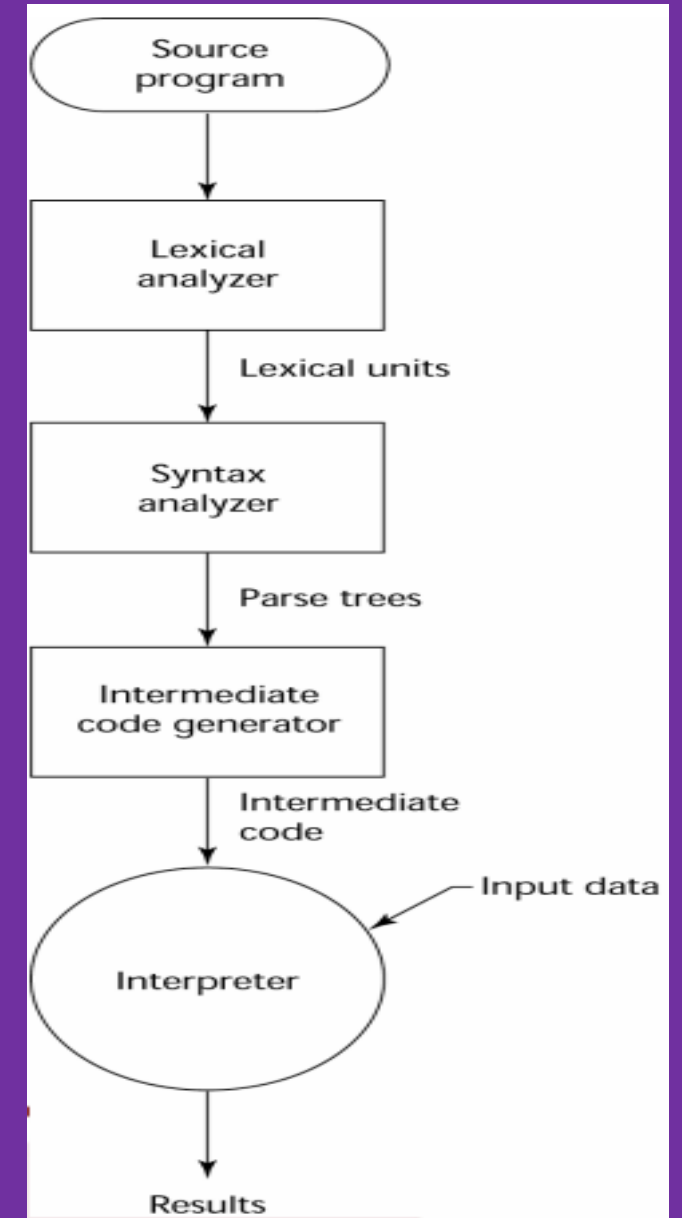
Compilação



Interpretação pura



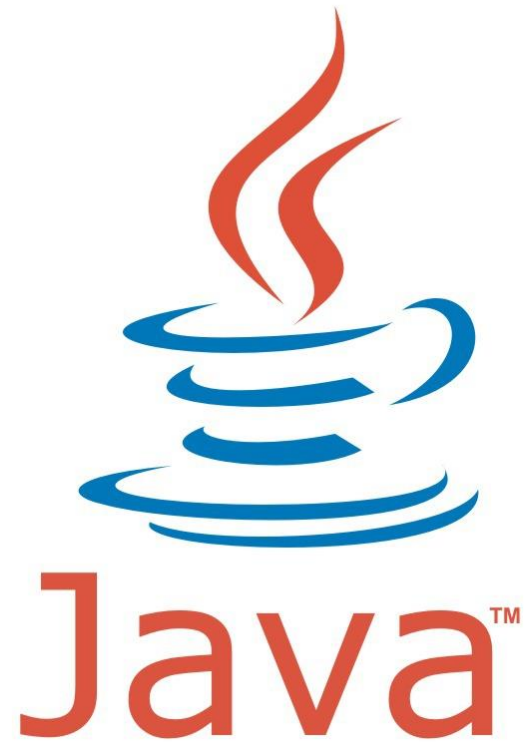
Sistema híbrido



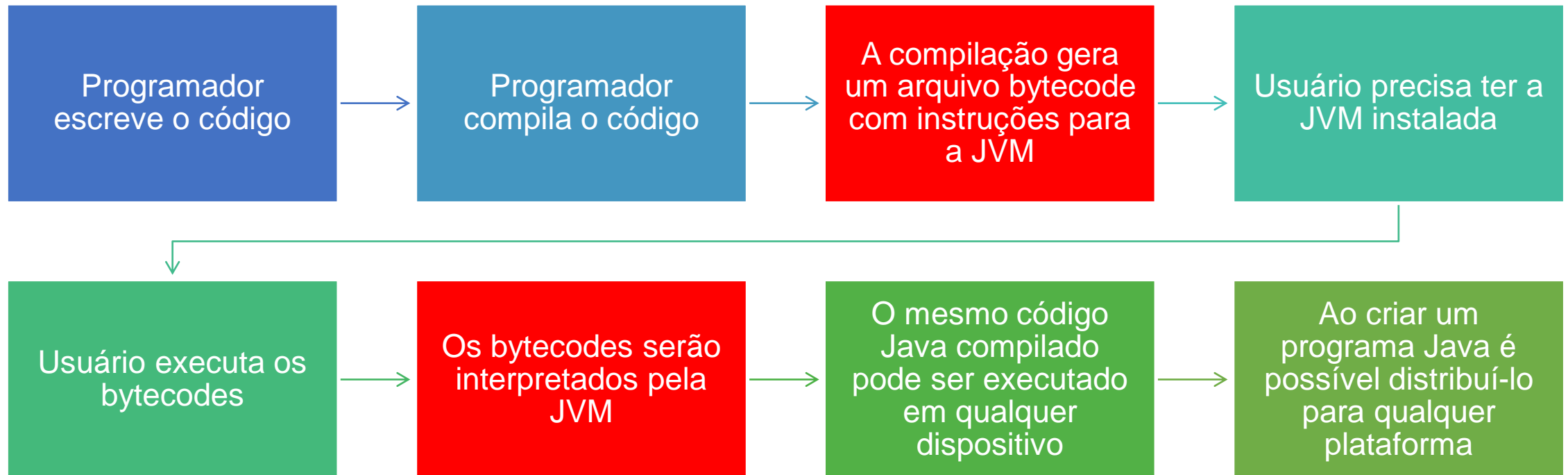
Vantagens x Desvantagens

Implementação	Vantagens	Desvantagens
Compiladores	Execução mais rápida	Várias etapas de tradução
	Permite estruturas de programação mais completas para sua execução	Programação final é maior necessitando de mais memória
	Permite otimização do código fonte	Processo de correção de erros e depuração é mais demorado
Interpretadores	Depuração do programa é mais simples	Execução do programa é mais lenta
	Consome menos memória	Estrutura de dados demasiadamente simples
	Resultado imediato do programa ou rotina desenvolvida	Necessário fornecer o programa fonte ao usuário

Java é uma linguagem compilada ou interpretada?



Java é uma linguagem compilada ou interpretada?



E o Ruby?

- Pode ser interpretado usando um interpretador;
- Pode ser compilado usando um compilador;

Interpretador/Compilador	Descrição
MRI	<i>Matz Ruby Interpreter (mais popular)</i>
mruby	Implementação leve de Ruby
jRuby	Implementação que permite interpretar Ruby usando JVM
ironRuby	Implementação em .Net (interpretar em ambiente Windows)
RubyMotion	Implementação comercial para iOS, macOS e Android
Rubinius	Compilador de Ruby escrito em Ruby