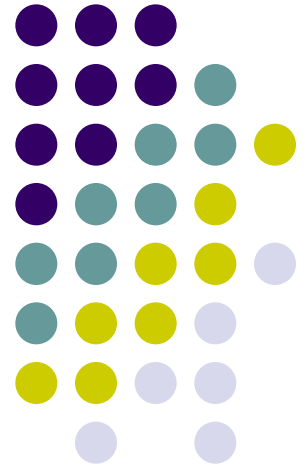


COM220

Aula 5: Introdução à Linguagem Java

Prof. Laércio Baldochi





Paradigmas de programação

- Paradigma de programação: **padrão conceitual** que orienta soluções de projeto e implementação
- Modelo OP: foco nos **procedimentos** (algoritmos) que alteram os dados
- Modelo OO: foco em **conceitos**, representados por dados, sobre os quais são definidas as operações
 - **classe** = modelo estático que permite especificar um conjunto de características que o conceito representa
 - **objeto** = entidade dinâmica criada a partir de uma classe, possuindo os dados sobre os quais são realizadas as operações disponíveis em sua classe

Modelos OP e OO



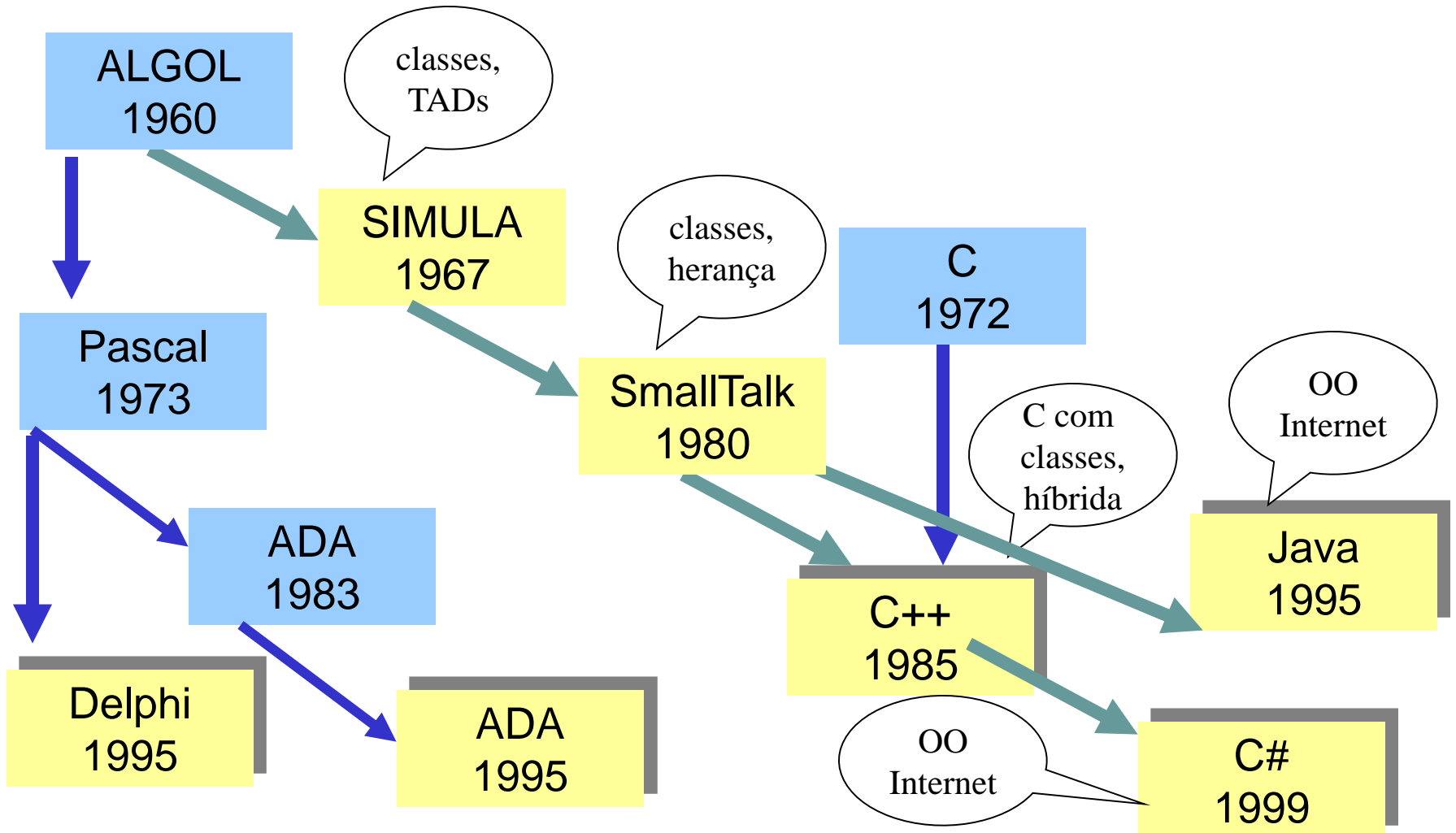
❖ Modelo procedimental

- ❑ projeto modular: decomposição funcional
- ❑ processos e dados separados
- ❑ reutilização: bibliotecas compostas de procedimentos e funções parametrizadas, geralmente em código objeto

❖ Modelo de objetos

- ❑ projeto de classes, seus relacionamentos, objetos e interações
- ❑ processos e dados encapsulados
- ❑ reutilização: bibliotecas compostas de classes/interfaces que podem ser adaptadas por extensão de propriedades

Modelo de Objetos: genealogia

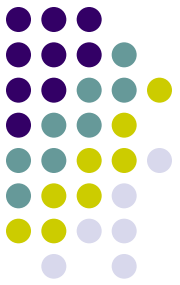




Linguagem Java

- O que é Java?
 - Linguagem de programação desenvolvida pela Sun Microsystems em 1991 como parte de um projeto de pesquisa que visava desenvolver software para produtos eletro-eletrônicos, como TVs, vídeos, torradeiras, etc.
 - Objetivos de Java em 1991:
 - Ser pequena, rápida, eficiente, e facilmente portátil para uma grande variedade de dispositivos de hardware.

Linguagem Java



- Portabilidade
 - Grande trunfo da linguagem Java
 - Possibilita a distribuição de programas executáveis através da World Wide Web (applets) e o desenvolvimento de aplicações "stand alone" que podem ser executadas em diferentes plataformas
 - Mais recentemente → servlets
 - Programas executados sob o controle do servidor
 - Carrega classes dinamicamente sob requisição de um cliente

Linguagem Java



- HotJava
 - A popularização da linguagem Java se iniciou com o desenvolvimento do **browser** HotJava em 1994
 - Primeiro browser a suportar a **execução de applets** Java
 - Desenvolvido inteiramente na linguagem Java (primeira **grande aplicação** desenvolvida em Java)



Vantagens da Linguagem Java

- Independência de plataforma
 - Capacidade de um programa de ser **movido de um sistema para outro** sem que seja necessário realizar alterações em seu código ou fazer uma nova compilação do mesmo.
 - A linguagem Java provê independência de plataforma tanto em termos de **código fonte** quanto em termos de **código binário**.



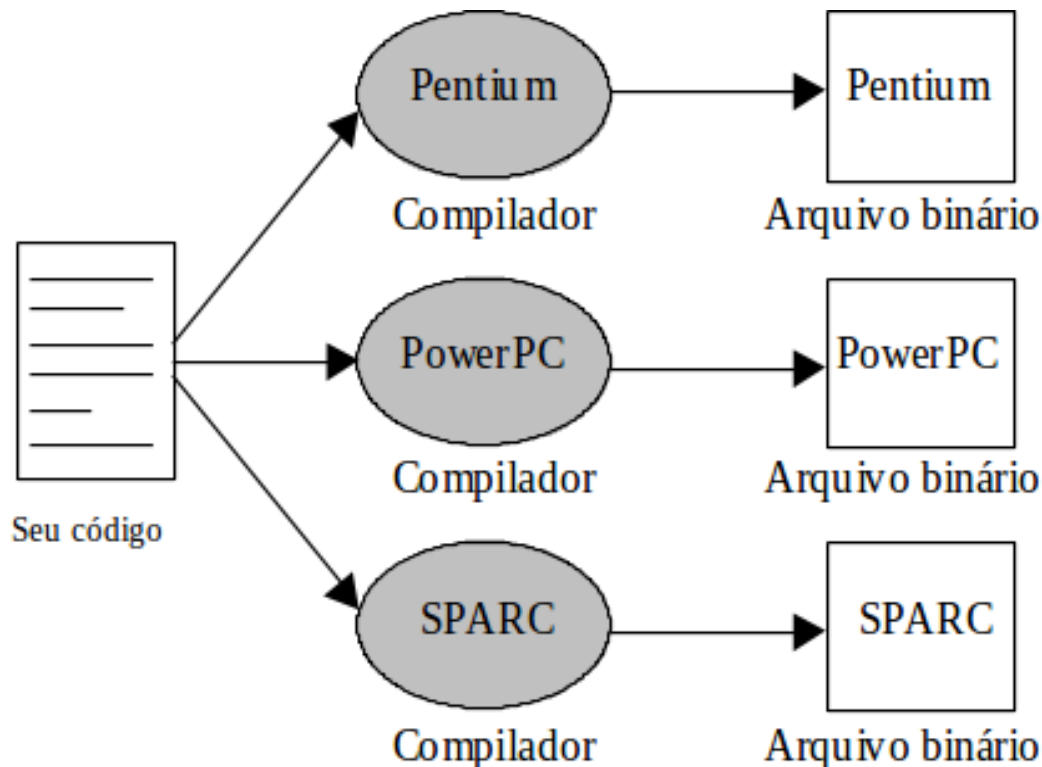
Vantagens da Linguagem Java

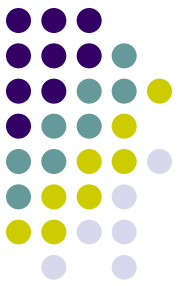
- Independência de plataforma (cont)
 - Em termos de código fonte
 - Tipos de dados primitivos (int, float, char) são consistentes entre diversas plataformas
- Em termos de código binário
 - O compilador Java gera um código intermediário chamado `bytecode`, que não é específico para o hardware de nenhuma máquina.



Vantagens da Linguagem Java

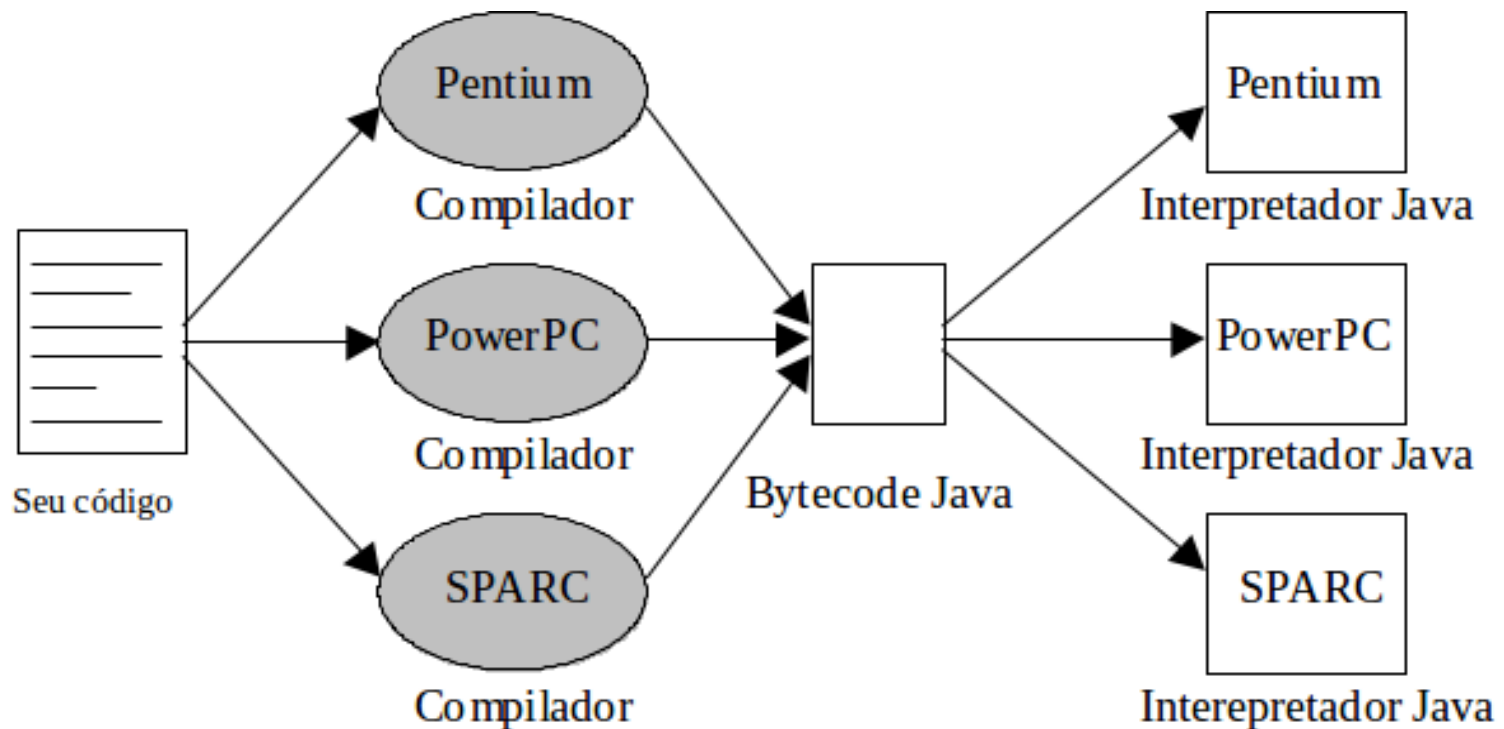
- Independência de plataforma (cont)
 - Funcionamento dos programas desenvolvidos em linguagens convencionais





Vantagens da Linguagem Java

- Independência de plataforma (cont)
 - Funcionamento dos programas Java





Vantagens da Linguagem Java

- Orientação a objetos
 - A linguagem Java foi **totalmente** desenvolvida de acordo com a metodologia de **orientação a objetos**, permitindo que desenvolvedores usem os princípios de orientação a objetos na criação de programas
 - Herança, polimorfismo, e outros conceitos de POO são diretamente aplicáveis na linguagem Java



Vantagens da Linguagem Java

- Facilidade de aprendizado
 - Um dos objetivos iniciais da linguagem Java era torná-la uma linguagem "pequena" e simples
 - Facilitar processo de escrita de código, compilação, depuração e, principalmente, aprendizado
- Java foi desenvolvida a partir de C e C++, utilizando a sintaxe e os princípios de orientação à objeto do C++



Vantagens da Linguagem Java

- Facilidade de aprendizado (cont)
 - Apesar da similaridade com C e C++, as partes mais complexas destas linguagens foram excluídas do Java
 - Não existem ponteiros em Java
 - Strings e arrays são implementados como objetos
 - O gerenciamento de memória é automático
 - Não existe herança múltipla



Programando em Java

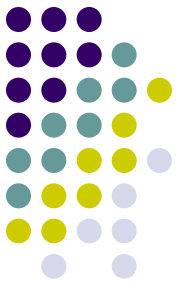
- Os programas Java se dividem em 3 tipos:
 - Applets
 - Aplicações
 - Servlets
- Applets
 - Programas Java associados à páginas HTML que podem ser "baixados" através da Web e executados em browsers clientes.

Programando em Java



- Aplicações
 - Programas convencionais, que poderiam ser desenvolvidos em qualquer linguagem de programação.
 - Possuem a vantagem da **portabilidade**, podendo ser executadas em quaisquer sistemas que possuam o Java Virtual Machine.

Programando em Java



- Servlets
 - Programas que executam no lado do servidor
 - Serão detalhadamente abordados em *COM222*

Java

Linguagem X Plataforma

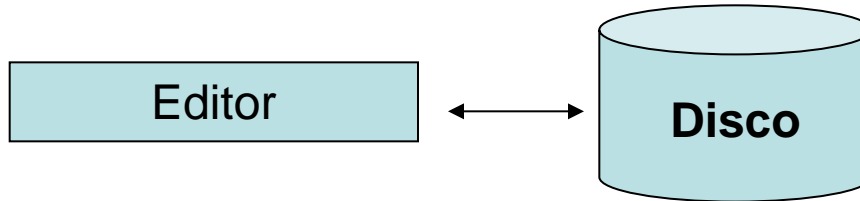


- Para executar em diferentes plataformas computacionais, a linguagem Java é disponibilizada como um **SDK**, que contém:
 - Ferramentas para desenvolver e testar programas
 - compilador, depurador
 - Ambiente de execução de programas
 - Máquina virtual, biblioteca de classes (APIs)
 - Classes de demonstração, arquivos utilitários, plug-ins para navegadores
 - Documentação de classes

Visão Geral

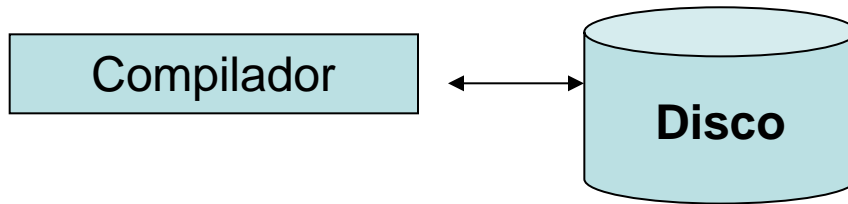


Fase 1



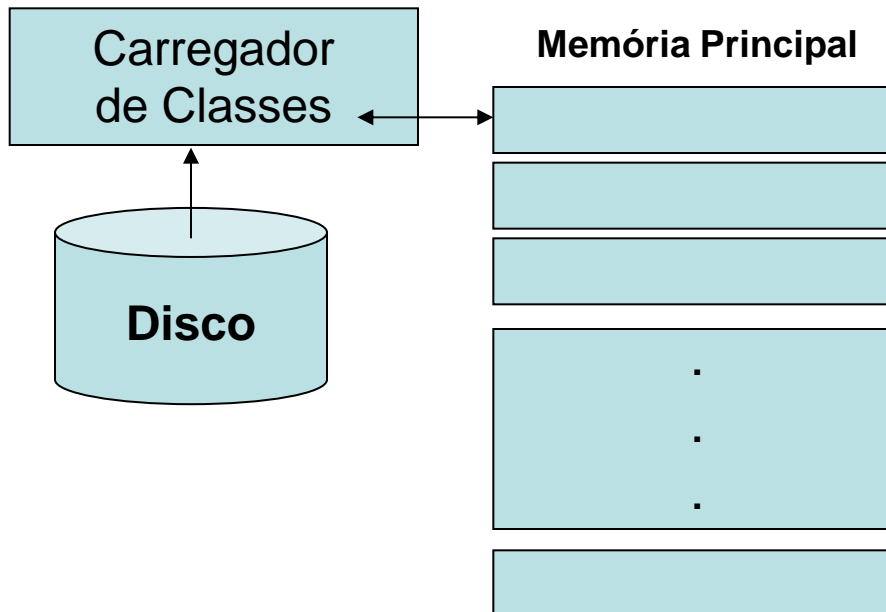
O programa é criado no editor e armazenado em disco.

Fase 2



O compilador cria bytecodes e os armazena em disco.

Fase 3

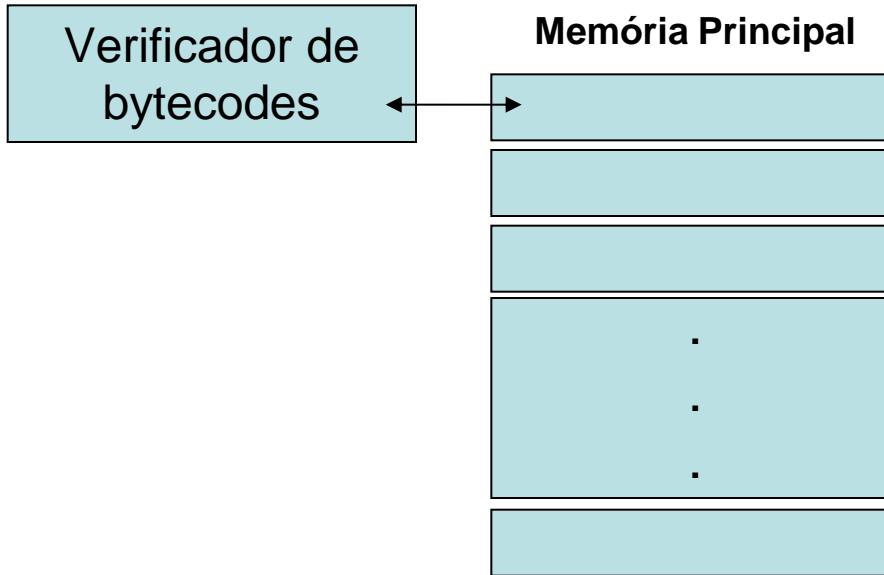


O carregador de classe coloca bytecodes na memória.

Visão Geral

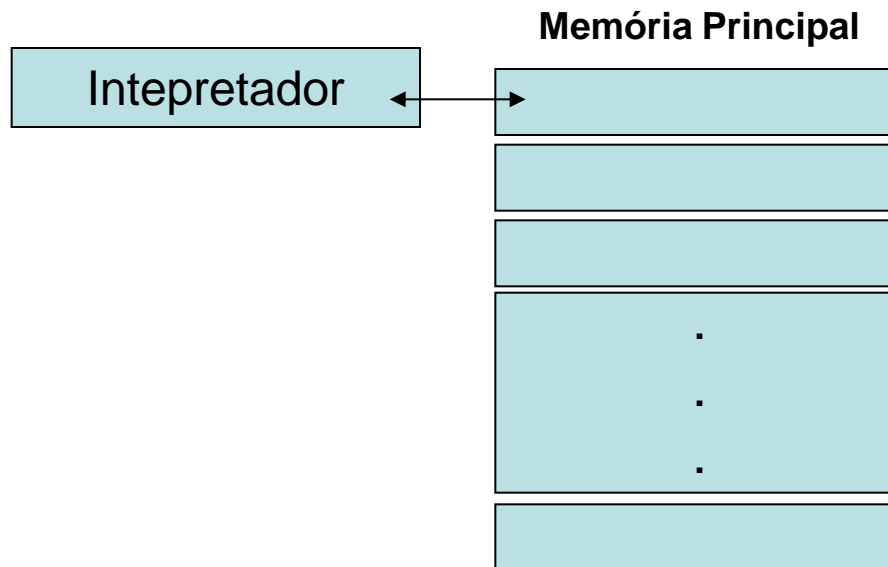


Fase 4



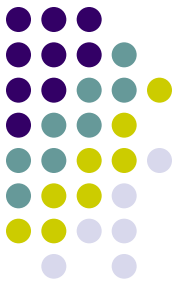
O verificador de bytecodes confirma que todos os bytecodes são válidos e não violam restrições de segurança do Java

Fase 5



O interpretador lê os bytecodes e os traduz para uma linguagem que o computador pode entender, possivelmente armazenando valores dos dados enquanto executa o programa.

Estrutura de um programa



Elementos da Programação Tradicional no Programa

```
// Nosso primeiro programa Java
// Conhecendo a estrutura de um programa Java
public class MeuPrimeiroPrograma {
    public static void main (String arg[]) {
        System.out.println("Olá Aluno de JAVA");
    } // fim do método main
} // fim da classe MeuPrimeiroPrograma
```

Função Principal. Assim como C e C++, programas Java buscam seu início pela função principal (main()).

Parâmetros. Parâmetros em funções permitem que essas iniciem com valores recebidos externamente, para variáveis que utilizarão internamente.

Estrutura de um programa



Elementos da Orientação a Objetos no Programa

```
// Nosso primeiro programa Java
// Conhecendo a estrutura de um programa Java
public class MeuPrimeiroPrograma {
    public static void main (String arg[]) {
        System.out.println("Olá Aluno de JAVA");
    } // fim do método main
} // fim da classe MeuPrimeiroPrograma
```

Classe. Como qualquer programa JAVA, esse programa exige uma classe (palavra reservada “class”). O fato de ser pública (palavra “public”) garante visibilidade em qualquer contexto de sua utilização

Método. A impressão da mensagem “Olá Aluno de Java” se deu pela execução do método “println” da classe “System”.

Biblioteca. A organização das classes JAVA se dá na forma de bibliotecas. Nesse programa utilizamos a biblioteca padrão da linguagem JAVA (biblioteca java.lang)

Objeto. Para imprimirmos a mensagem de saída de nosso programa precisamos de um objeto “out” da classe “System” da biblioteca padrão java.lang

Estrutura de um programa



```
package meupacote;
```

Package. Utilizado quando o código do programa deverá fazer parte de um pacote.

```
import java.lang.*;
```

Import. Seção de importação de bibliotecas.

```
/** Nosso primeiro programa Java  
    Conhecendo a estrutura de um  
    programa Java */
```

Comentários. Com sintaxe “// ... para comentários simples ou “/* */” e a mais recente “/** .. */” que permite geração de documentação automática (ferramenta javadoc)

```
public class MinhaClassePublica {  
    .....  
    .....  
    /** Comentário sobre o método */  
    public (private/protected) tipoRet  
    nomeMetodo(<parametros>) {  
        // código do método  
    } // fim da definição do método  
} // fim da classe
```

Método main(). Indica que a classe Java é um aplicativo que será interpretado pela máquina virtual.

Classes.
Declaração de classes, atributos e métodos do programa Java. A declaração e a definição dos métodos ocorre obrigatoriamente dentro do limite de declaração da classe.

Importante



// Comentários:

Use comentários para esclarecer conceitos utilizados no programa. Utilize:

// para comentários de linha única

/* */ para comentários de várias linhas

/***/ em ambos os casos e quando desejar incluir o texto na documentação de seu programa (javadoc).

Sempre inicie seus programas com comentário descrevendo o propósito do mesmo.

```
public class MeuPrimeiroPrograma { ..... }
```

- Classes são tipos de dados declarados com a palavra reservada **class**.
- Cada arquivo .java deve ter somente uma classe pública e essa deve ter o mesmo nome do arquivo

Importante



```
public static void main (String args[]) {...}
```

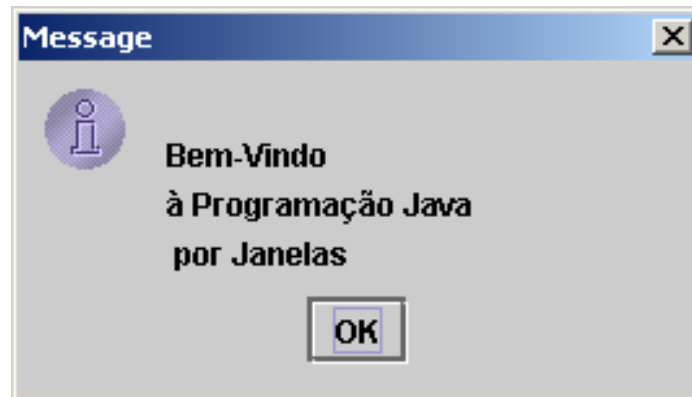
- O método “main()” faz parte de todo aplicativo Java
- Deve estar entre os métodos da classe pública e será sempre por onde o aplicativo se inicia.
- Os argumentos passados ao método “main()” são uma lista de objetos da classe String, separados por espaços em branco.

```
System.out.println (“parâmetros”) & System.out.print(....)
```

- System.out é o objeto de saída padrão em Java
- Permite exibir strings e outros tipos de informações na Janela de Comando (console do sistema operacional)
- System.out.println() exibe seus parâmetros e pula uma linha

```
import javax.swing.JOptionPane;
public class ProgramaSwing {
    // o método main inicia a execução do aplicativo Java
    public static void main (String arg[]) {
        JOptionPane.showMessageDialog(null, "\nBem-Vindo\nà Programação Java\n\t\t
por Janelas");

        System.exit(0); // termina programa
    } // fim do método main
} // fim da classe MeuPrimeiroPrograma
```





Alguns comentários relevantes

- A classe *JOptionPane* oferece caixas de diálogo pré-definidas que permitem aos programas exibir mensagens aos usuários
- A instrução ***import*** é usada para carregar classes utilizadas em um programa Java
 - Classes importadas podem pertencer a pacotes do núcleo da linguagem, extensões oficiais ou extensões fornecidas por terceiros
- As instruções ***import*** devem aparecer sempre antes da definição das classes

JOptionPane

Como usar (parte 1)



```
JOptionPane.showMessageDialog(null, "\nBem-Vindo\nà  
Programação Java\n\t\t por Janelas");
```

- Método ***showMessageDialog*** tem dois argumentos
 - O primeiro é utilizado para **posicionamento** da janela. Ao ser null é ignorado e a janela é apresentada no centro da tela
 - O segundo representa a **string** que será apresentada na janela
- **System.exit(0)**
 - **Termina** o aplicativo Java. Necessário em programas com interface gráfica
 - Classe System não precisa ser importada
 - Faz parte do pacote padrão **java.lang**