

UNIP

UNIVERSIDADE PAULISTA

Linguagem de Programação Orientada a Objeto



Introdução a Orientação a Objetos
Professora Sheila Cáceres

Introdução a Orientação a Objetos

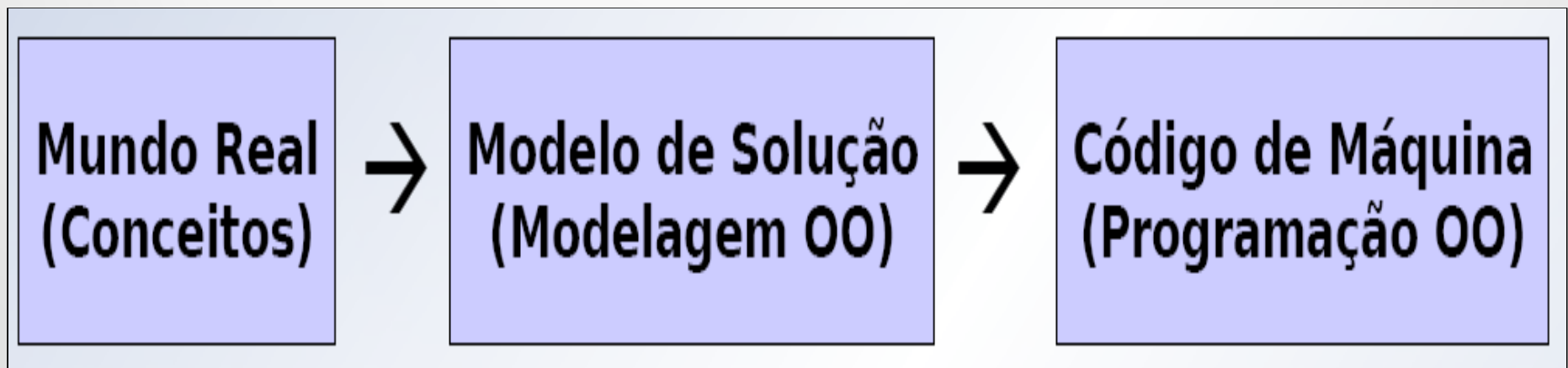
- No mundo real, tudo é objeto!;
- Os objetos se relacionam entre si de diversas maneiras;
- Um programa orientado a objetos é estruturado como uma comunidade de agentes que interagem entre si, denominados **objetos.**;
- Cada objeto tem um papel a cumprir;
- Cada objeto oferece um serviço ou realiza uma ação que é usada por outros membros da comunidade;

Exemplo real: montagem de um computador



Introdução a Orientação a Objetos

- Cada componente é independente dos demais.
- Para quem está montando, interessa apenas como os componentes interagem entre si:
 - a placa de vídeo encaixa no slot ?
 - O monitor funciona com essa placa ?
 - A CPU é compatível com a placa-mãe ?

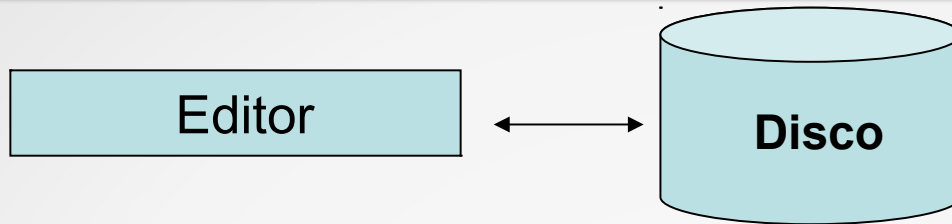




O que você entende por essas figuras???

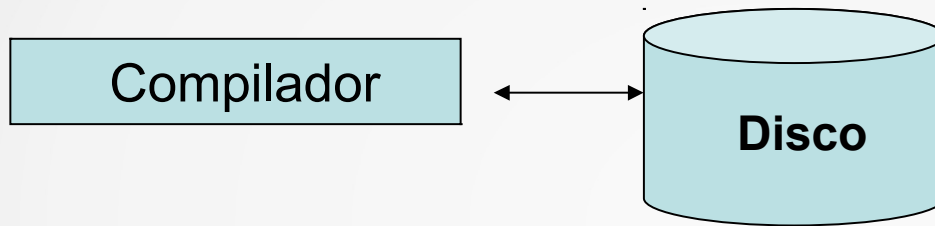
Se você usar a OO, o que você entende por essas figuras??

Fase 1



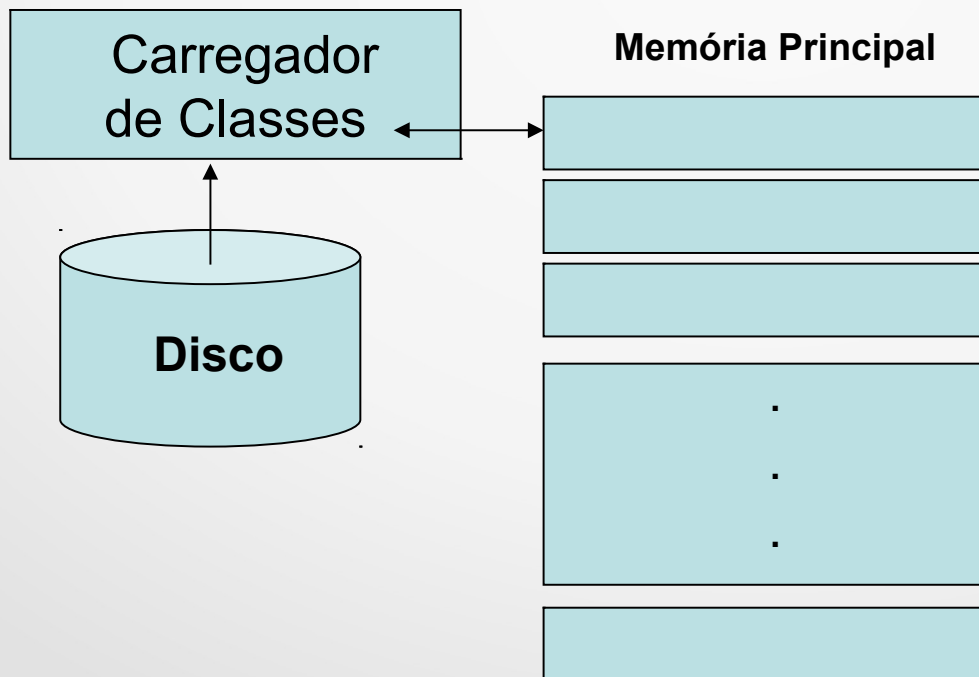
O programa é criado no editor e armazenado em disco.

Fase 2



O compilador cria bytecodes e os armazena em disco.

Fase 3

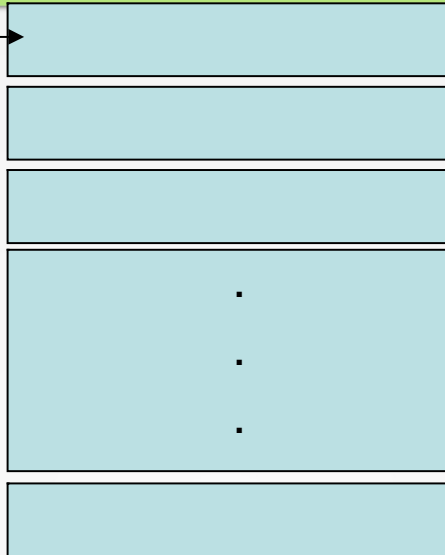


O carregador de classe coloca bytecodes na memória.

Fase 4

Verificador de
bytecodes

Memória Principal

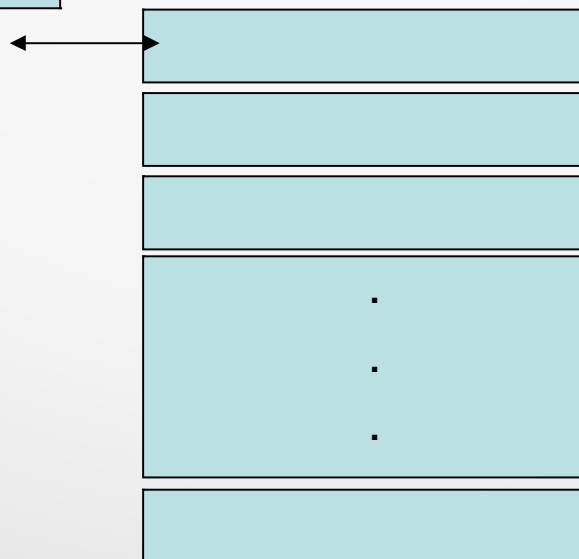


O verificador de bytecodes confirma que todos os bytecodes são válidos e não violam restrições de segurança do Java

Fase 5

Intepretador

Memória Principal



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.

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. Programas em Linguagem C e C++ buscam seu início pela função principal (main()).

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

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 Java

```
package meupacote;
```

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

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

```
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.

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

Import. Seção de importação de bibliotecas.

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

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.

// 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

```
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

javac *MeuPrimeiroPrograma.java*

- Compila o arquivo .java, gerando o arquivo .class, que contém o bytecode da classe.

java *MeuPrimeiroPrograma*

- Executa o programa Java

start java *MeuPrimeiroPrograma*

- Executa o programa Java em uma nova janela

set CLASSPATH=C:\dir1;C:\dir2;.

- Define os locais onde a JVM vai procurar por pacotes e classes Java

Histórico da POO

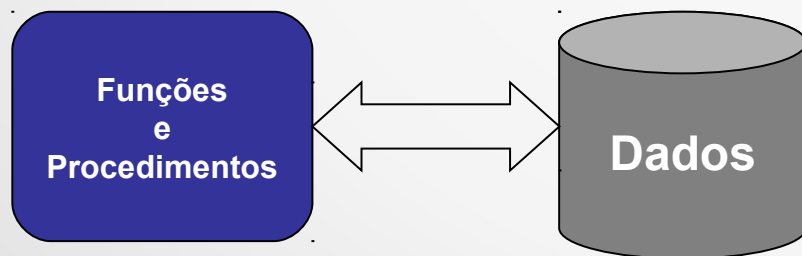
- Grandes mudanças ao longo dos anos => acomodar o aumento da complexidade dos programas;
- FORTRAN: primeira linguagem de alto nível difundida;
- Anos 60: Programação Estruturada (C e Pascal):
 - **No Início:** razoavelmente fácil escrever programas complexos;
 - **Projetos grandes:** difícil e muito custoso efetuar manutenção e fazer modificações;
 - **Solução:** criação da POO;
- POO: aproveitou as melhores idéias da programação estruturada e combinou-as com novos conceitos permitindo que um problema seja mais facilmente decomposto em subgrupos relacionados;

Histórico da POO

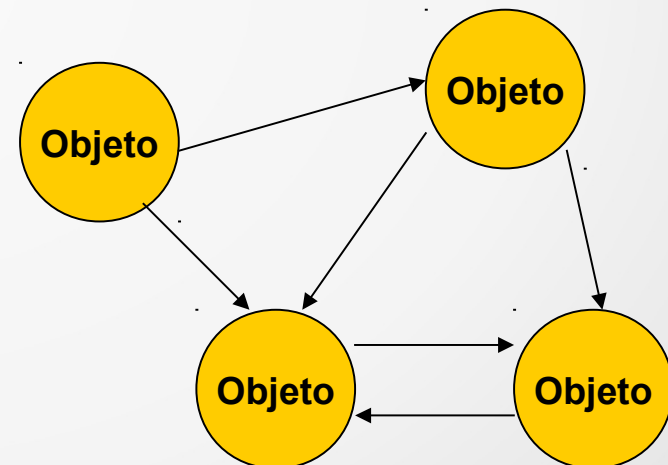
- 1966: linguagem Simula67 (Noruega) – primeira linguagem OO;
- Década de 70: SmallTalk (Centro de Pesquisas da Xerox);
- Início década 80: AT&T lança a Linguagem C++;
- 1995: Java;
- POO: ênfase está na simulação de modelos de sistemas ;
- Atualmente:
 - Maioria das linguagens incorpora características de OO;
 - Conceitos de OO em SO (Windows 2000);
 - Conceitos de OO em Banco de Dados relacionais (Oracle);
 - Banco de Dados totalmente OO (ObjectStore);

Paradigma Estrutural X Paradigma OO

- **Programação Estruturada:** Aplicação separada em duas partes: dados e funcionalidade;
- **Programação OO:** Aplicação definida como uma coleção de objetos que se interagem;



Aplicação Estruturada



Aplicação Orientada a Objetos

Paradigma Estrutural X Paradigma OO

Modelagem estruturada

- ❑ Foco: algoritmos
- ❑ Estruturas de dados e processamento
- ❑ Funções e procedimentos
- ❑ Diagramas de fluxos de dados, bolhas
- ❑ Decomposição funcional
- ❑ Análise e desenho estruturados

Modelagem OO

- ❑ Foco: objetos
- ❑ Dados (atributos) e comportamento (métodos)
- ❑ Classes e relacionamentos
- ❑ Diagramas de classes, objetos, colaboração
- ❑ Divisão de responsabilidades
- ❑ Análise e desenho orientados a objetos

Definição da POO

- É a programação implementada pelo envio de mensagens a objetos. Cada objeto irá responder às mensagens conhecidas por este, e cada objeto poderá enviar mensagens a outros, para que sejam atendidas, de maneira que ao final do programa, todas as mensagens enviadas foram respondidas, atingindo-se o objetivo do programa;
- ❖ Obs: ***“A utilização de uma linguagem que tem como paradigma a orientação a objetos, não gera automaticamente um sistema ou programa orientado a objetos.”***

POO X POE

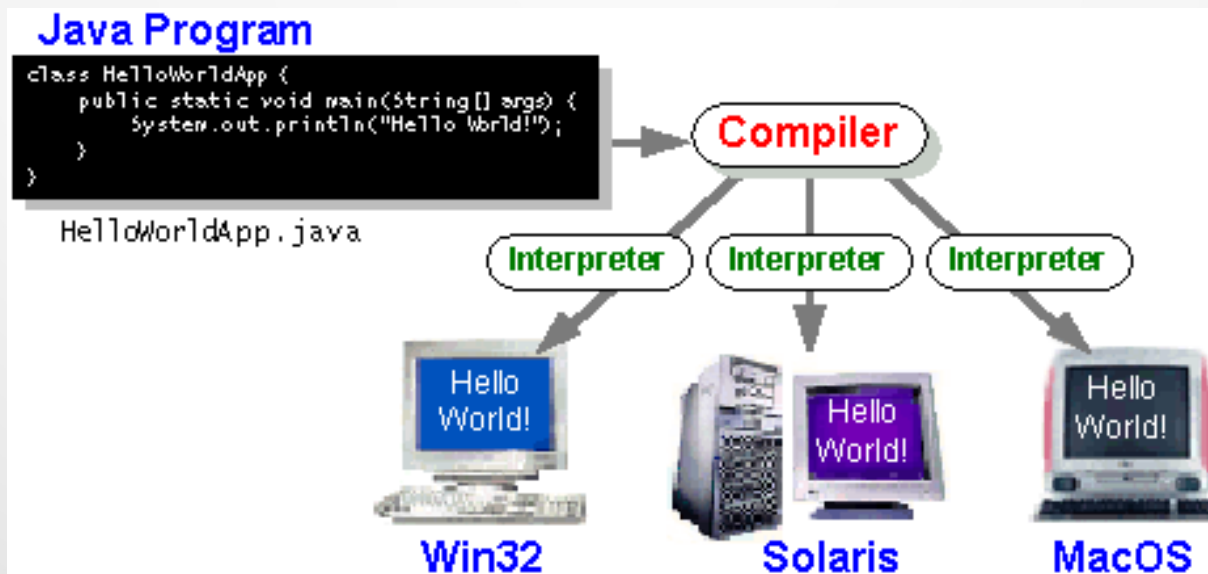
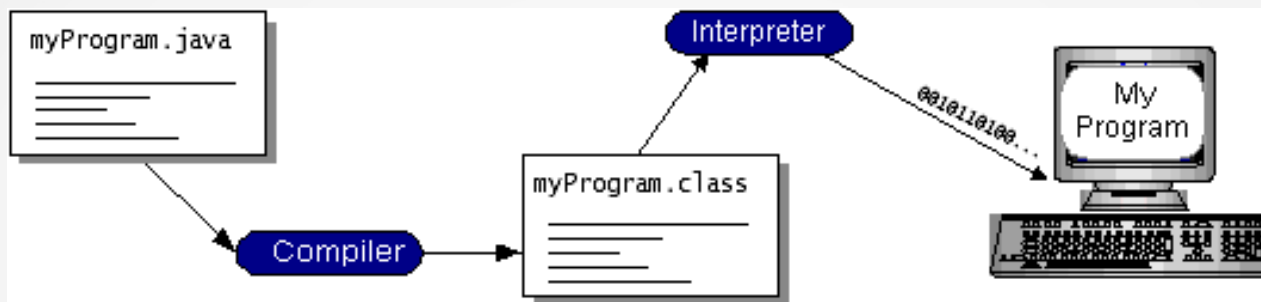
- POE - Programação Orientada a Eventos;
- POO contém a POE mas a POE não contém a POO;
 - Java é uma linguagem de POO, que utiliza eventos na implementação de interfaces gráficas.
 - Delphi é uma linguagem de POE.
- Eventos são associados a objetos;

Sobre a Tecnologia Java

- A Tecnologia Java é tanto uma linguagem de programação quanto uma plataforma.
- Palavras-chaves de Java:
 - ✓ Simples
 - ✓ Orientada a Objetos
 - ✓ Distribuída
 - ✓ Interpretada
 - ✓ Robusta
 - ✓ Segura
 - ✓ Indiferente da Arquitetura
 - ✓ Portável
 - ✓ Alta Performance
 - ✓ Suporta Concorrência
 - ✓ Dinâmica
- **Nota:** Cada uma das palavras-chave acima, são explicadas com detalhes em [The Java Language Environment](#).

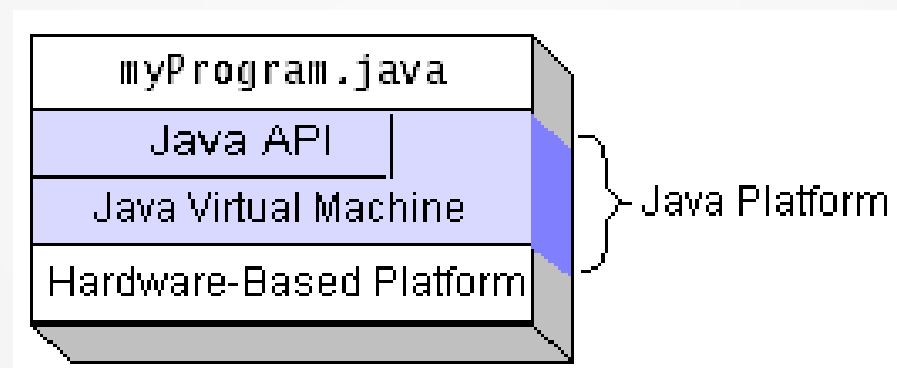
Sobre a Tecnologia Java

- Como a linguagem trabalha:



Sobre a Tecnologia Java

- A plataforma Java:
 - The *Java Virtual Machine* (Java VM)
 - The *Java Application Programming Interface* (Java API)



- **Nota:** A figura acima mostra um programa rodando sobre a plataforma Java. Como a figura mostra a API Java e a máquina virtual isola o programa do hardware.

O que a Tecnologia Java pode fazer?

- **Essencial:** Objetos, strings, threads, números, entrada e saída, estrutura de dados, propriedades do sistema, data e hora, e outras.
- **Applets:** Conjunto de convenções usadas pelos applets.
- **Networking:** URLs, TCP (Transmission Control Protocol), UDP (User Datagram Protocol) sockets, e endereçamento IP (Internet Protocol).
- **Internacionalização:** Suporte para escrever programas que podem ser “localizados” para usuários no mundo todo. Programas podem automaticamente se adaptar a locais específicos e serem apresentados na língua apropriada.
- **Segurança:** Tanto baixo nível quanto alto nível, incluindo assinaturas digitais, gerenciamento de chaves privadas e públicas, controle de acesso e certificados.
- **Componentes de Software:** Conhecidos como JavaBeans™, podem ser “plugados” em uma arquitetura de componentes existente.
- **Serialização de Objetos:** Permite “persistência leve” e comunicação via Remote Method Invocation (RMI).
- **Conectividade com banco de dados (JDBC™):** Provê acesso uniforme a uma grande variedade de bancos de dados relacionais.

Principais vantagens

- Escrever menos código;
- Escrever um código melhor;
- Desenvolver programas de forma mais rápida;
- Evitar dependência de plataforma;
- “Write once, run anywhere”;
- Desenvolver softwares distribuídos de forma mais fácil;

Referências

- Filho, Renato Rodrigues; Desenvolva Aplicativos com Java 2 / Renato Rodrigues Filho. –1. ed. – São Paulo: Érica, 2005.
- Santos, Rafael; Introdução a Programação Orientada a Objetos Usando Java / Rafael Santos – Rio de Janeiro: Campus, 2003. Material no site editora.
- Deitel, H.M; Java Como Programar. Ed. Bookman, 2005.
- The Java Phenomenon (The JavaTutorial):
<http://java.sun.com/docs/books/tutorial/getStarted/intro/index.html>
- Nota: O material da apresentação foi extraído de algumas das fontes aqui apresentadas