

# Tópicos esp. POO

```
/**
 * This is about <code>ClassName</code>.
 * (@link com.yourCompany.aPackage.Interface)
 * @author author
 * @deprecated use <code>OtherClass</code>
 */
public class ClassName<E> implements InterfaceName<String> {
    enum Color { RED, GREEN, BLUE };
    /* This comment will span multiple lines. */
    static Object staticField;
    // This comment spans only one line.
    private E field;
    // TASK: refactor
    @SuppressWarnings("all")
    public int foo(Integer parameter) {
        abstractMethod();
        int local= 42*hashCode();
        staticMethod();
        return bar(local) + parameter;
    }
}
```

The Java logo, featuring a stylized blue and orange flame above the word "Java" in a bold, orange, sans-serif font.

# Ementa

Programação orientada a objetos com Java. Aplicações dos conceitos utilizando a linguagem de programação Java.



# Objetivos Específicos

1. Desenvolver programas orientados a objetos em Java.



2. Apresentar os conceitos sobre a programação.

3. Desenvolver aplicações com a plataforma Java.

# Pré-requisito

- ✘ Não possui disciplina como Pré-requisito
- ✘ Somente
  - ✘ Lógica de Programação
- ✘ Oportunidade
  - ✘ Dificuldades com programação



# Conteúdo da Disciplina

## 1. Programação Orientada a Objetos com Java

polimorfismo de classes)

- Interfaces gráficas de usuário

- O ambiente de programação Java
- Sintaxe geral da linguagem Java
- Bibliotecas de classe Java
- Tratamento de exceções
- Conceitos (revisão) de orientação a objetos em Java (herança, sobrecarga, sobrescrita,



# Metodologia de Trabalho

- Discussões em aula
- Trabalho em grupo (PIM)
- Apresentação de Resultados
- Aulas de laboratório
- Discussão de Soluções





# Critério Avaliação

- 🌐 Instrumentos de Avaliação
  - 🌐 1ª Prova: 0 a 10 pontos
  - 🌐 2ª Prova: 0 a 10 pontos
  - 🌐 Proj. Integrado Multidisciplinar (PIM)





🌈 A MS será:

$$((NP1) \times 4 + (NP2) \times 4 + (PIM) \times 2) / 10.$$

Para a aprovação, a MS deverá ser igual ou superior a 5,0;

E frequência mínima de 75%.

O desempenho do aluno é avaliado numa escala de 0 (zero) a 10 (dez).

$$\text{MédiaFinal} = ((NP1) \times 4 + (NP2) \times 4 + (PIM) \times 2) / 10.$$


🌈 Aprovação MédiaFinal  $\geq 5,0$

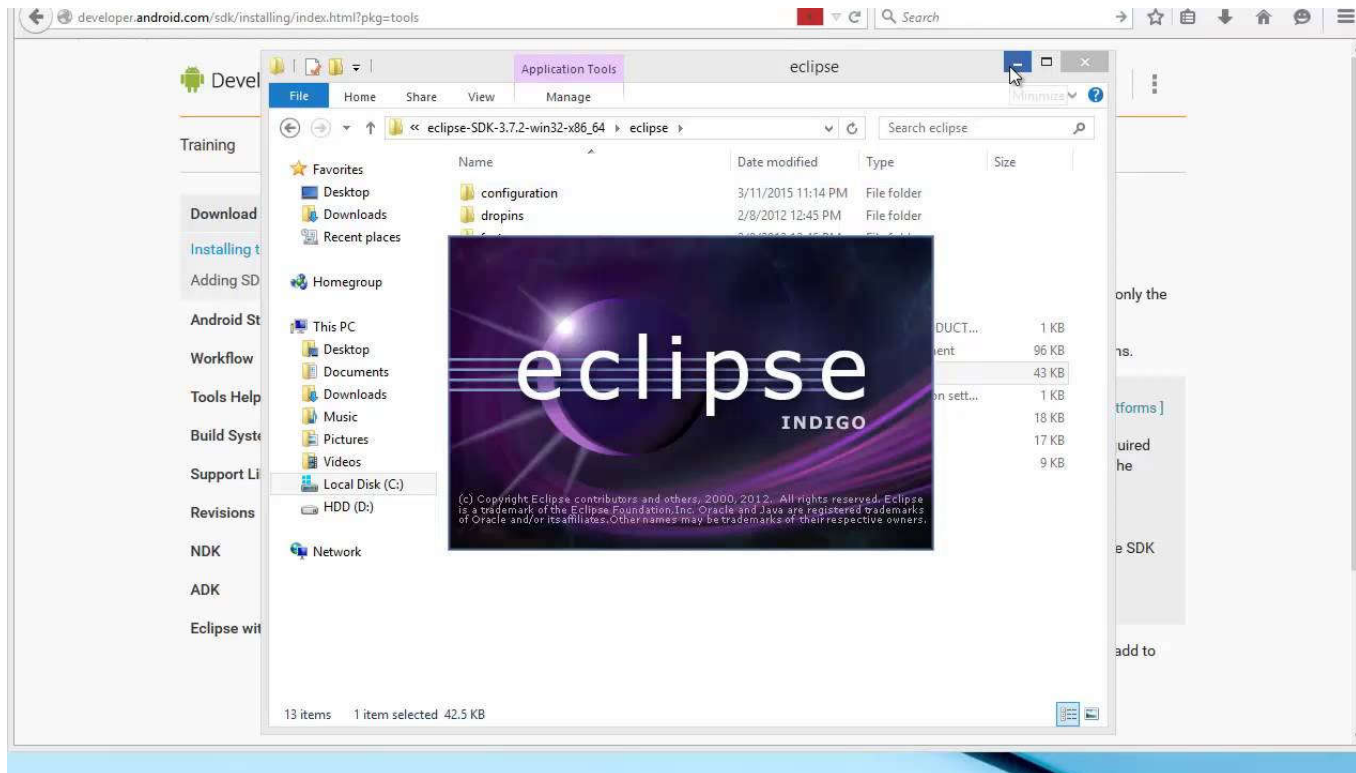




# Softwares Utilizados

 Eclipse

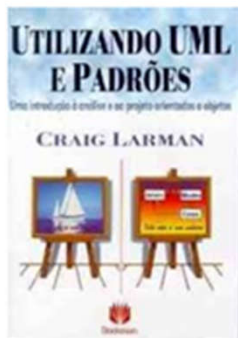
 Download : [www.eclipse.org/downloads/eclipse-packages](http://www.eclipse.org/downloads/eclipse-packages)



# Bibliografia

## Utilizando Uml e Padroes

Craig Larman



Tipo: seminovo/usado

Editora: [Bookman](#)

Ano: 2000

Estante: [Informática](#)

Peso: 1022g

ISBN: 9788573076516

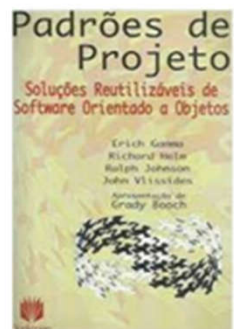
Idioma: Português

Cadastrado em: 04 de março de 2020

Descrição: Livro em bom estado, lateral levemente suja, marca de amassado na capa, sem grifos, escrita a caneta na primeira página, mancha de umidade nas primeiras páginas, 492 páginas...

## Padrões de Projeto

Erich Gamma/ Richard Helm/ Ralph Johnson



Tipo: seminovo/usado

Editora: [Bookman](#)

Ano: 2002

Estante: [Informática](#)

Peso: 768g

ISBN: 9788576862444

Idioma: Português

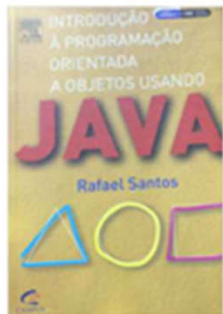
Cadastrado em: 04 de julho de 2020

Descrição: livro em bom estado para leitura, cortes com uma sigla K M A de canetinha e nome do antigo dono na primeira página de caneta, cortes amarelado, p 1

# Bibliografia

## Introdução à Programação Orientada a Objetos Usando Java

Rafael Santos



Tipo: seminovo/usado

Editora: [Campus](#)

Ano: 2003

Estante: [Engenharia](#)

Peso: 650g

ISBN: 0000000000

Idioma: Português

Cadastrado em: 30 de julho de 2020

Descrição: Miolo em perfeito estado de conservação e leitura, capa e contracapa com marcas de manuseio e ação do tempo, capa brochura, não possui grifos ou rasuras. OP10

# Bibliografia complementar

## Java: Como Programar - 6ª Ed + Cd

Harvey M. Deitel, Paul J. Deitel



Tipo: seminovo/usado

Editora: [Prentice-hall](#)

Ano: 2005

Estante: [Outros Assuntos](#)

Peso: 2550g

ISBN: 9788576050193

Idioma: Português

Cadastrado em: 10 de novembro de 2016

## Refatoração - Aperfeiçoando o Projeto de Código Existente

Autor: Fowler, Martin | Marca: Bookman





# Bibliografia complementar

[Voltar](#)



## **Orientação a Objetos com Java**

Código: 01122

Autor: Araújo, Everton Coimbra de

Editora: Visual Books

Páginas: 186

ISBN: 9788575022269

Ano de Lançamento: 2008

Este livro introduz o paradigma de Orientação a Objetos com implementação em linguagem Java de forma simples, fácil e eficiente.

O autor é professor universitário e elaborou esta obra com base nos feedbacks dos alunos, que durante suas aulas precisavam aprofundar vários conceitos aplicados.

Os princípios de Orientação a Objetos, como Abstração, Encapsulamento, Polimorfismo e Herança são apresentados através de analogias do dia a dia, tornando simples a compreensão e aplicação.

Durante a implementação dos exemplos, o leitor conhecerá ainda o IDE Eclipse, ferramenta mais conhecida e utilizada pelos desenvolvedores Java em todo o mundo.

Os princípios de Associação de Agregação e de Composição, Exceções, Collections, Sobrescrita e Sobrecarga de Métodos são apresentados de forma eficaz dentro de contextos reais, porém simples, para o rápido entendimento do leitor.

Ao final, é feita uma introdução básica ao Swing, motivando o leitor a conhecer esta API gráfica para o Java.

## Java 7 - Ensino Didático

Sergio Furgeri



Tipo: seminovo/usado

Editora: [Erica](#)

Ano: 2010

Estante: [Informática](#)

Peso: 580g

Idioma: Português

Cadastrado em: 09 de janeiro de 2016

**Descrição:** Livro em brochura, em bom estado, capa com uma marca pequena de vinco na aba inferior, sem orelhas, sem sinais de envelhecimento, sem sujidade. Interior sem marcações, sem grifos, sem manchas, sem cortes ou amassados. Perfeito estado para leitura. Contém 320 páginas.

Dúvidas?





# Paradigmas de Linguagens de Programação

- Antes de iniciarmos o estudo de uma linguagem precisamos entender em qual paradigma ela pertence
- Entretanto...

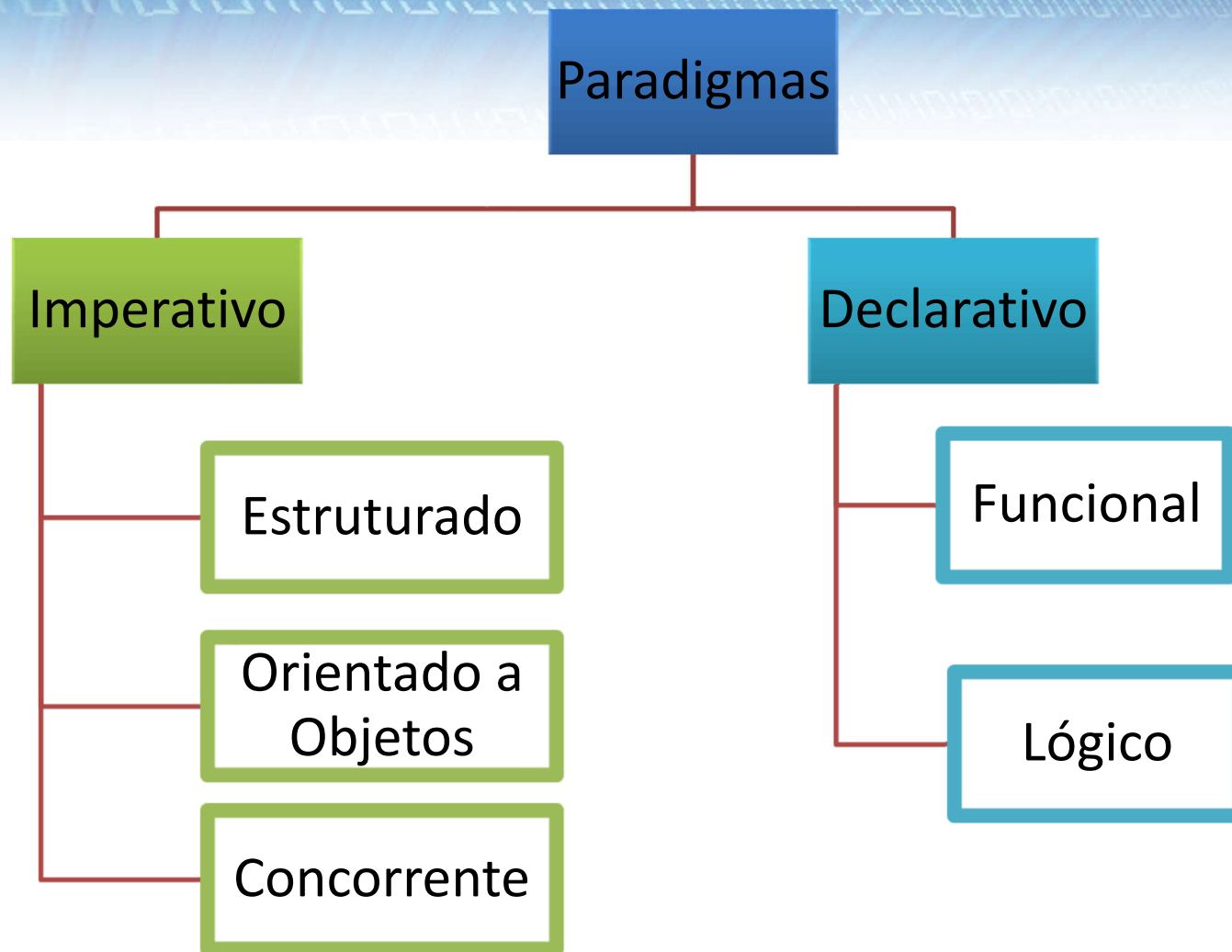
O que é um Paradigma?



# Paradigma de Linguagem de Programação

- Conjunto de Características que servem para categorizar um grupo de linguagens.
- Este conjunto de características irá influenciar a forma de programar.
- Existe um conjunto de paradigmas que podemos classificar as linguagens.

# Paradigmas



# Paradigma Imperativo

- Computação como um processo que realiza mudança de estados
  - Estado é uma configuração qualquer na memória do computador
- Foco
  - especificar como um programa dever ser feito no computador

# Exemplo de Linguagens Imperativas

- Pascal
- Fortran
- C
- C++
- C#
- Java
- Python
- Ruby
- ETC.



# Paradigma Declarativo

- Foco do programador
  - Descrever de forma abstrata a tarefa a ser realizada
- Programador não necessita se preocupar como o processamento será realizado ou sobre como o computador funciona
- Utiliza relações ou funções
- Não há o conceito de variável



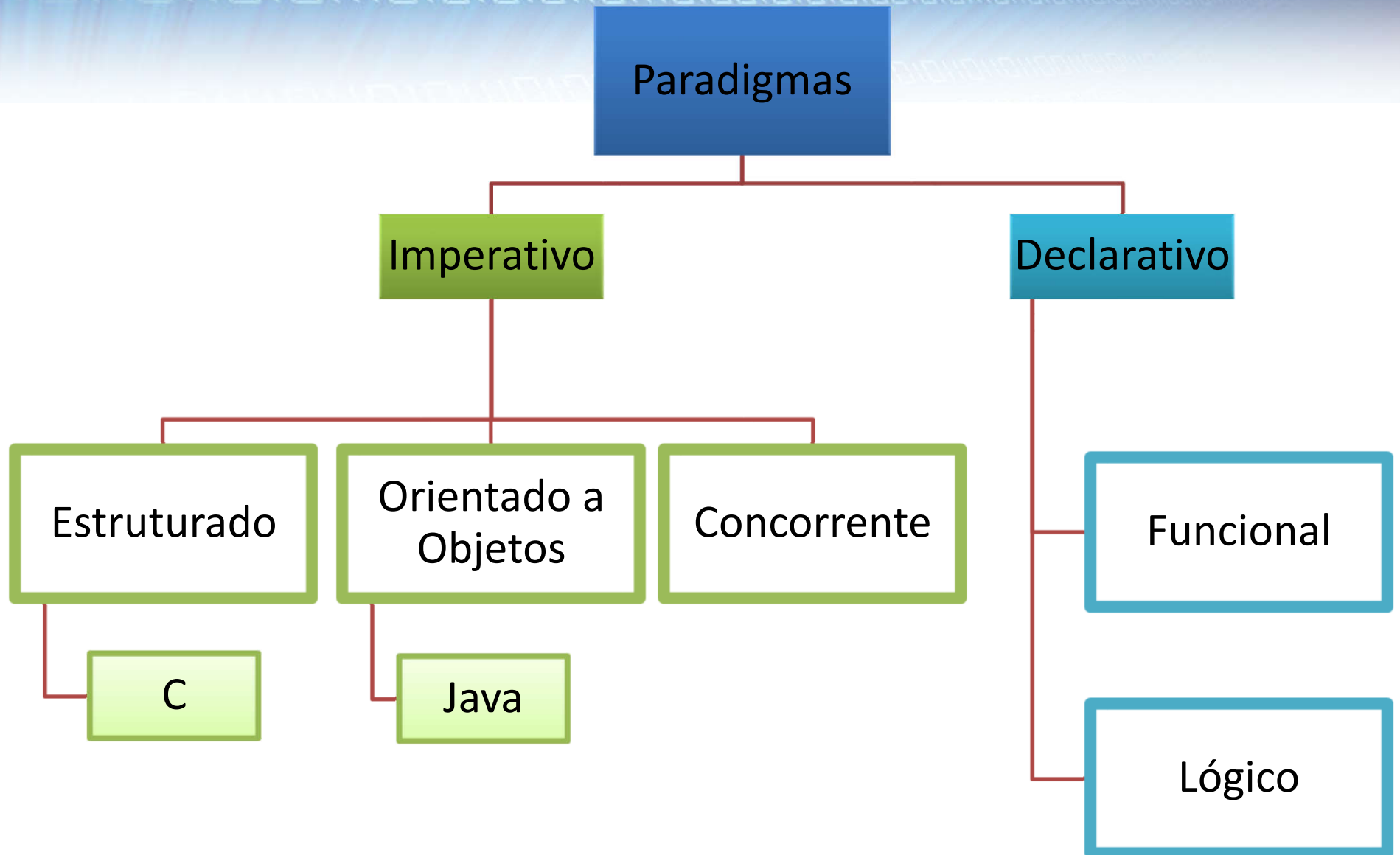
# Exemplo Linguagens Declarativas

- Lisp
- Prolog

# Programação funcional: LISP

- List Processing Language
  - Projetada no MIT por McCarthy
- Pesquisa de inteligência artificial (IA) precisava de uma linguagem para
  - Processar dados em listas (em vez de vetores)
  - Computação simbólica (em vez de numérica)
- Apenas dois tipos de dados: átomos e listas
- Sintaxe é baseada em *lambda calculus*

# Paradigmas





# Introdução a Linguagem de programação Java



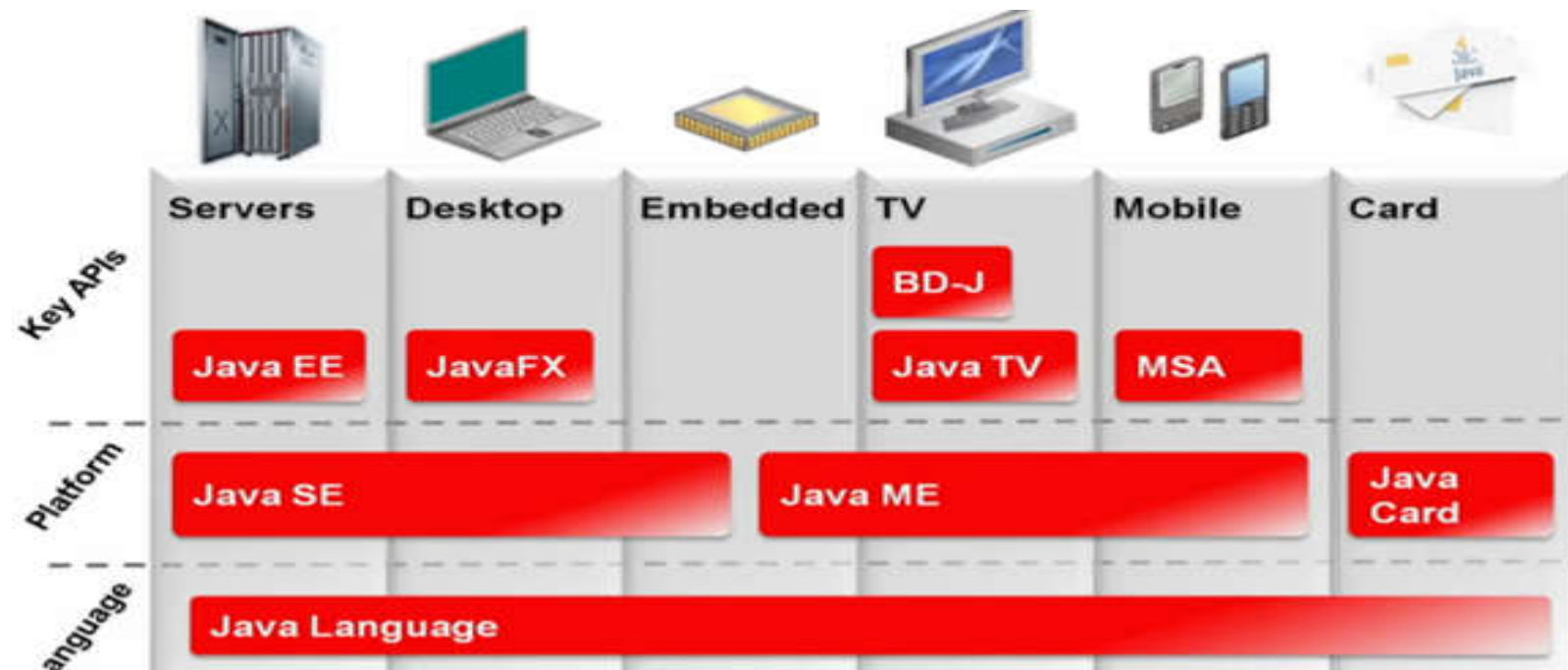
**UNIP**

UNIVERSIDADE PAULISTA

# CAPÍTULO 1

## Conceitos Gerais

# Java em todo lugar





# *Um Pouco de História*

**90/91**



## **Projeto Green – OAK**

Este sistema permitiria ao usuário controlar vários dispositivos (TV, videocassete, luzes, telefones, etc.)

# *Um Pouco de História*

**93/95**



Em 1993, depois da World Wide Web ter transformado a Internet em um ambiente rico em gráficos, a equipe Java percebeu que a linguagem que desenvolveram era perfeita para programação Web.

Surgiu então

- \* Applets
- \* HotJava

# *Um Pouco de História*

**95/97**



Netscape adicionou suporte a Java na versão 2.0 do seu navegador.



- Começa as Versões do Java.

# *Um Pouco de História*

**97**



Java 1.0

**98**



Java 1.1 - JCP



Código  
Aberto

**99**



Java 2.0



- 1ª evolução
- HotSpot / Jit

# *Um Pouco de História*

**2000**



J2SE 1.3

**2002**



J2SE 1.4

**2004**



J2SE 1.5



•JAVA 5

•2ª evolução

# Um Pouco de História

**Atual**



**JAVA 8 – JAVA 9**

**Java 8**

**Um novo método em todas as coleções: `forEach`**

A partir do Java 8 temos acesso a um novo método nessa nossa lista: o `forEach`. De onde ele vem? Veremos mais adiante. Iniciaremos por utilizá-lo. Podemos fazer algo como:

```
usuarios.forEach(...);
```



# Um Pouco de História

## Jshell

**Atual** Uma ferramenta para um ambiente interativo, na qual você pode digitar comandos Java em um terminal e obter retorno real-time, sem ter que criar uma classe, arquivo ou algo do tipo.



**JAVA 9**

## Projeto Jigsaw – Modularização do JDK

O projeto Jigsaw é um esforço da Oracle e da comunidade Java para a implementação da modularização da plataforma, tendo em mente os seguintes objetivos, de acordo com a página oficial do projeto:

- ✓ Tornar a plataforma Java SE e o JDK mais facilmente escaláveis para dispositivos pequenos;
- ✓ Melhorar a segurança e manutenibilidade da plataforma Java SE e do JDK;
- ✓ · Permitir um melhor desempenho das aplicações Java;
- ✓ · Tornar mais fácil aos desenvolvedores a construção e manutenção de bibliotecas e grandes aplicações nas plataformas Java.



# Plataformas Java

## → Plataforma

- Ambiente de hardware e/ou software no qual um programa é executado

## → A plataforma Java é somente de software

- Esta plataforma de execução funciona sobre outras plataformas de hardware e software

## → Plataformas Java

- JME – Java Platform, Micro Edition
  - Plataforma de desenvolvimento para dispositivos com suporte à Java como Palms, celulares, pagers, etc.
- JSE – Java Platform, Standard Edition
  - Contém serviços padrão para aplicações e applets.
- JEE – Java Platform, Enterprise Edition
  - Plataforma de desenvolvimento completa para aplicações empresariais robustas e escaláveis.

# J2EE × .NET

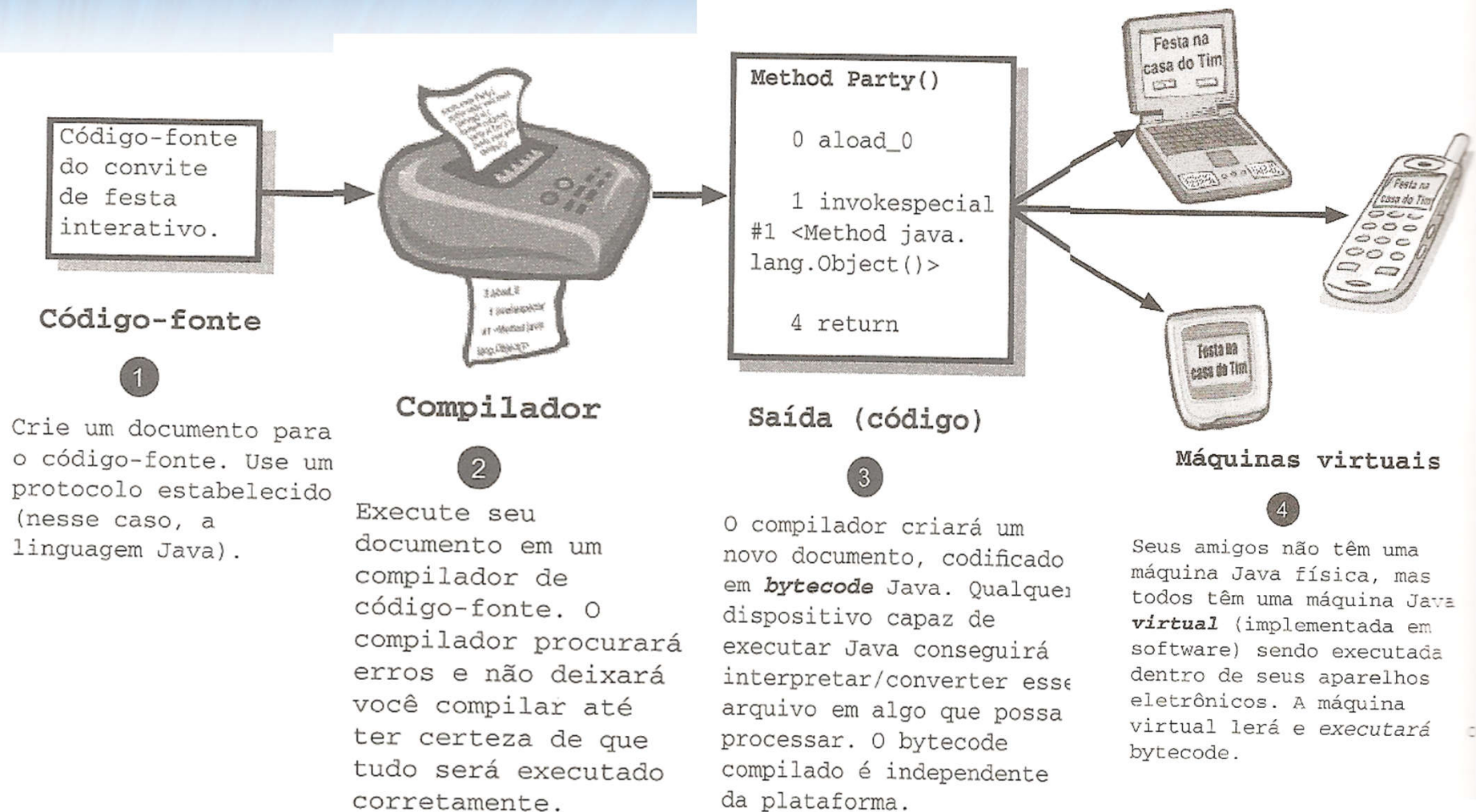
## J2EE

- ▮ Padrão Aberto
- ▮ Multiplataformas
- ▮ Treino Difícil
- ▮ Diversos IDEs
- ▮ Middleware Aberto
- ▮ Java, (Groovy)
- ▮ bytecode
- ▮ Suporta Web Services

## .NET

- ▮ Proprietário
- ▮ Windows
- ▮ Treino Difícil
- ▮ IDE Microsoft
- ▮ Middleware Proprietário
- ▮ C#, VB, C++, outras
- ▮ IL
- ▮ Suporta Web Services

# Como o Java funciona





# O que você fará em Java

```
import java.awt.*;
import java.awt.event.*;
class Party {
    public void buildInvite() {
        Frame f = new Frame();
        Label l = new
Label("Party at Tim's");
        Button b = new
Button("You bet");
        Button c = new
Button("Shoot me");
        Panel p = new Panel();
        p.add(l);
    } // mais código aqui...
}
```

## Código-fonte

1

Digite seu código-fonte.  
Salve como: **Party.java**

```
File Edit Window Help Plead
%javac Party.java
```

## Compilador

2

Compile o arquivo **Party.java** executando o javac (o aplicativo do compilador). Se não houver erros, você terá um segundo documento chamado **Party.class**. O arquivo **Party.class** gerado pelo compilador é composto de *bytecodes*.

```
Method Party()
0 aload_0
1 invokespecial #1 <Method java.
lang.Object()>
4 return
Method void buildInvite()
0 new #2 <Class java.awt.Frame>
3 dup
4 invokespecial #3 <Method java.
awt.Frame()>
```

## Saída (código)

3

Código compilado: **Party.class**

```
File Edit Window Help Swear
%java Party
```

## Máquinas virtuais

4

Execute o programa iniciando a Java Virtual Machine (JVM) com o arquivo **Party.class**. A JVM converterá o bytecode em algo que a plataforma subjacente entenda e executará seu programa.

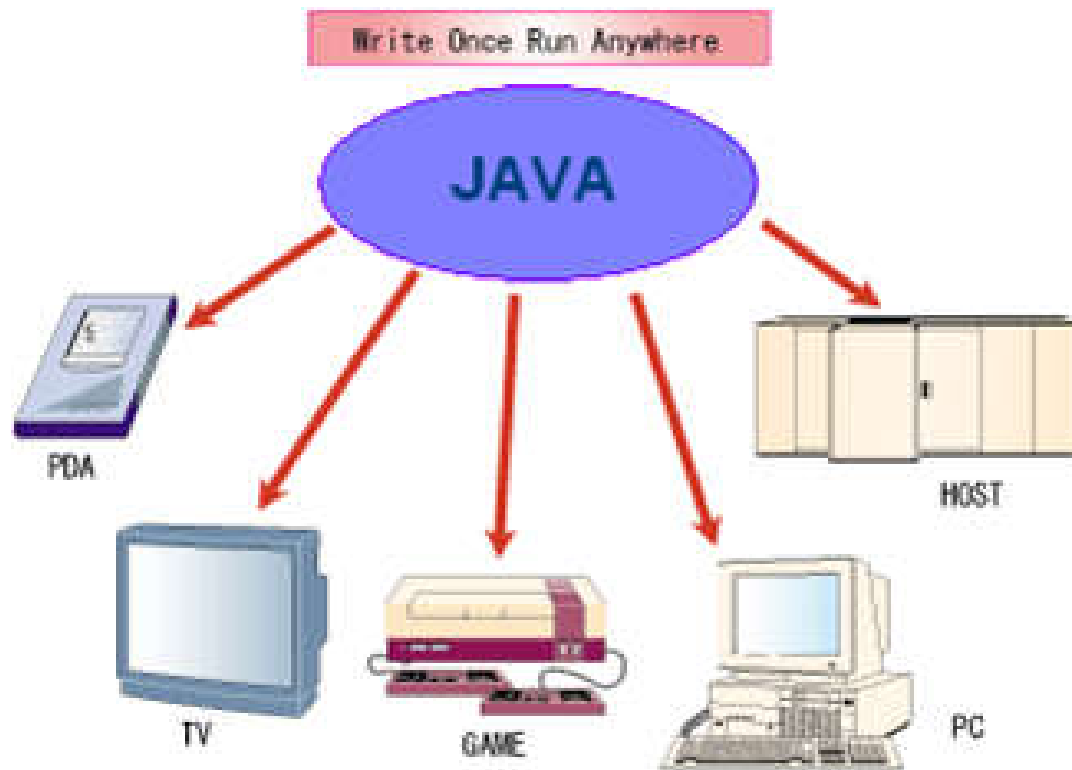




# O que é Máquina Virtual

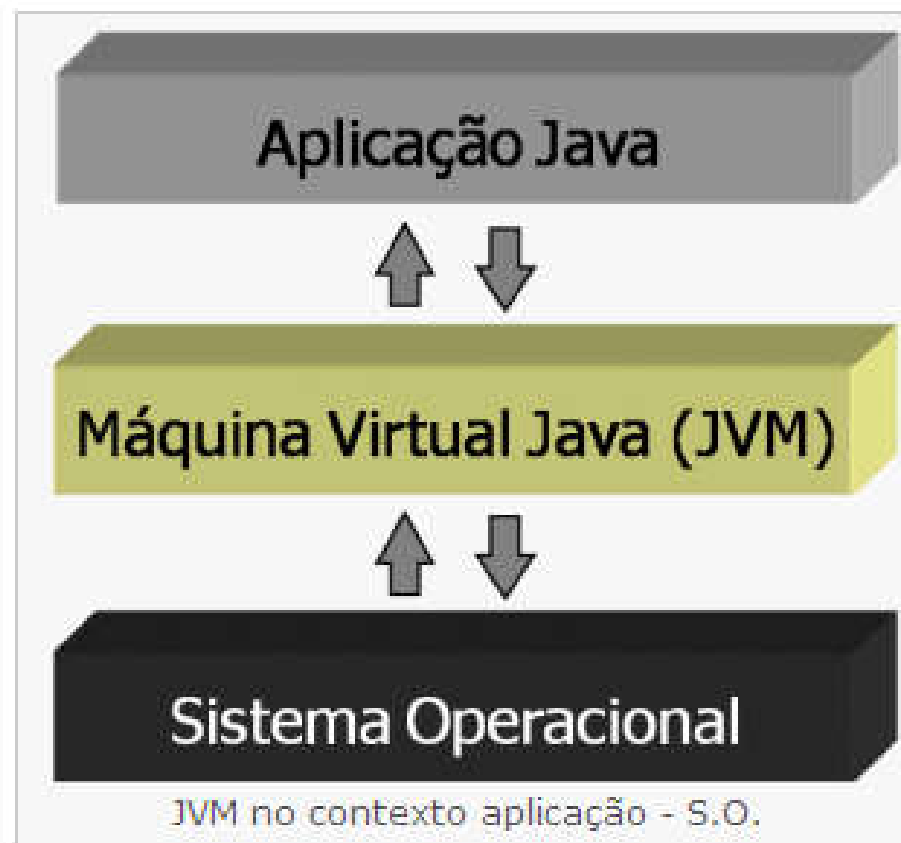
E uma das peças chaves para ser multiplataforma é o **JVM (Java Virtual Machine)**, necessário para executar qualquer aplicação escrita na linguagem de programação Java.

“Escreva uma vez. Execute em qualquer lugar”



O JVM nada mais é do que uma máquina virtual, que deixa o hardware e a plataforma transparentes a aplicação, que as seguintes funções:

- Interpreta o Bytecode Java para ser executado no S.O. (o arquivo tem contém o bytecode é o .class gerado na compilação do .java);
- Checagem de erros em tempo real;
- Gerenciamento automático de memória (Garbage Collector);



# Características do Java

Características

Princípios da linguagem

Estrutura básica de uma classe

Versões do Java

JDK e JRE

Ambientes de desenvolvimento

Compilando e executando

Path e JAVA\_HOME



# Características

Orientada a objetos

Interpretada e compilada

Multiplataforma

Robusta e confiável

Segura

Portável

Multithreaded



# Características

## Ciclo de vida de uma aplicação em C/C++

```
#include <stdio.h>
int main () {
    char[ ] nome = "Manuel da Silva";
    float salario = 800.0;
    char[ ] sql = "insert into usuario";
    -----
    -----
}
```

CÓDIGO FONTE



COMPILAÇÃO

```
1011101000110101101010001
1001010101100100100111110
0010101101001110101011001
```

CÓDIGO DE MÁQUINA



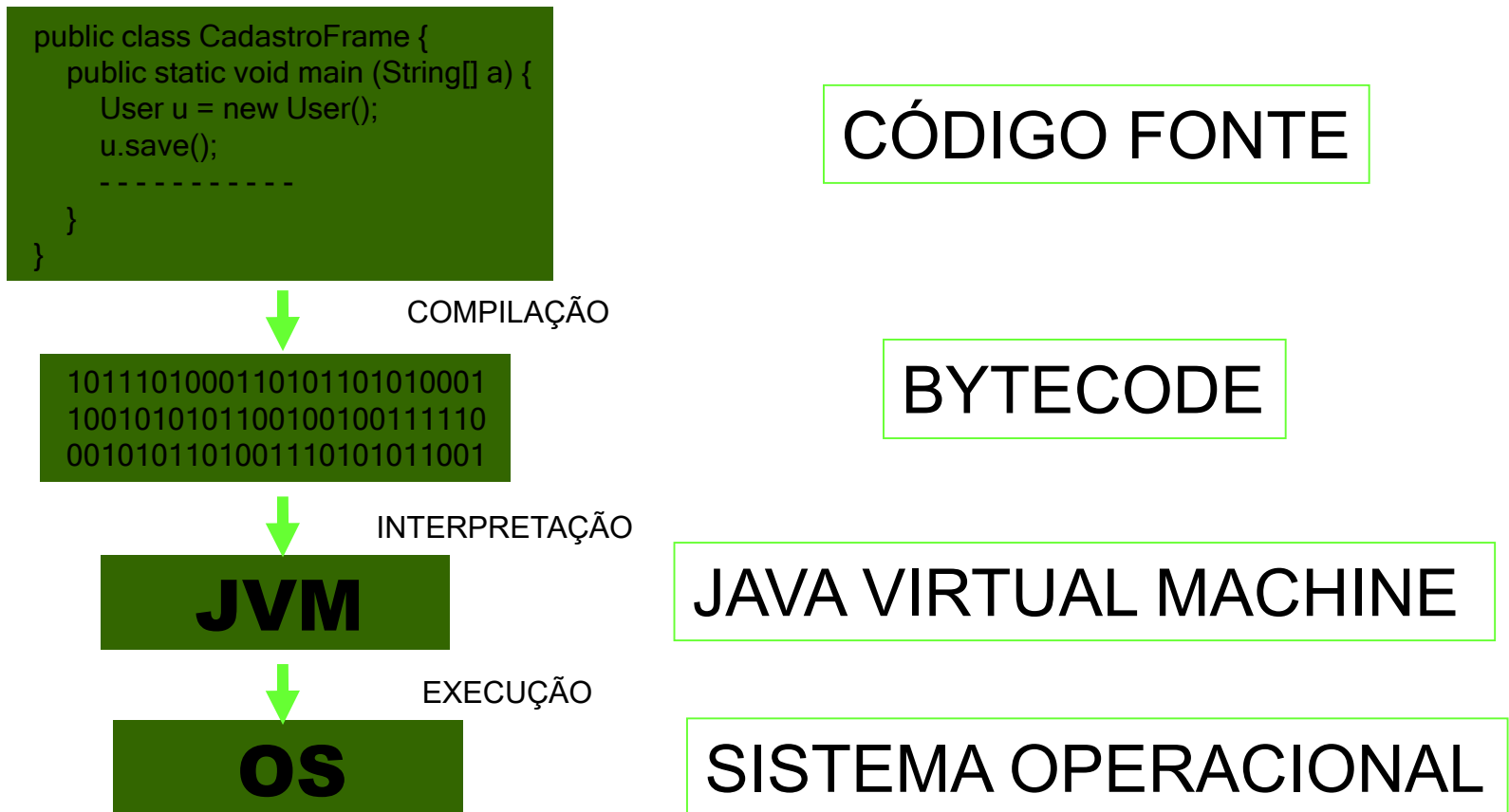
EXECUÇÃO

**OS**

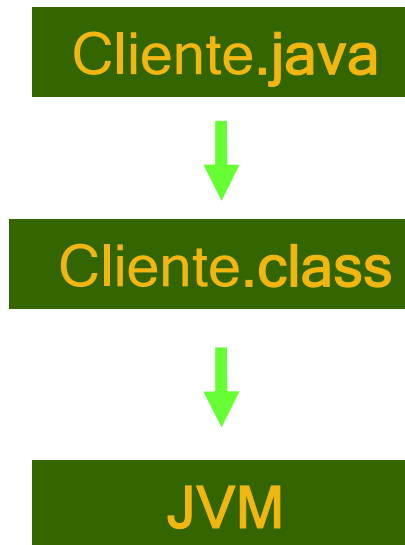
SISTEMA OPERACIONAL

# Características

## Ciclo de vida de uma aplicação em Java



# Princípios da Linguagem



# Princípios da Linguagem

`int idadeMinima;`

`int IdadeMinima;`

`int IDADEMINIMA;`

# Princípios da Linguagem

**String** telefone2;

**String** \_cont;

**String** nome\_completo;

**String** contador\$numerico;

**String** \$identificador;



# Princípios da Linguagem

```
public class ----- {  
    -----;  
    String findName() {  
        -----;  
        for (---;---;---) {  
            -----;  
            -----;  
        }  
    }  
}
```



# Comentários

Linhas de código ignoradas pelo compilador.

Podem ser utilizadas para:

- Descrever o significado do código a ser compilado;

- Marcar comandos que deseja-se ignorar temporariamente;

- Documentar blocos de código em seu programa (JAVADOC).

# Exemplos de Comentários

Comentário de linha:

```
- - - - -  
double saldoInicial = 815.5;  
- - - - -  
// Ignora o acréscimo de  
// saldoInicial += 300;  
- - - - -
```

# Exemplos de Comentários

Comentário de bloco:

-----

/\*

Calcula o percentual de comissão  
de venda do funcionário

\*/

valor = (venda \* perc) / 100;

-----

# Exemplos de Comentários

Comentário de documentação JAVADOC (antes da classe ou de um membro):

```
/**
 * Classe utilizada no cálculo do limite de crédito do cliente
 * @author sandrov
 */
public class CreditoUtil {
    /**
     * Calcula os juros do crédito
     * @param periodo Tempo de quitacao do emprestimo
     * @param valor Valor do emprestimo
     * @return Valor dos juros calculados
     */
    public double calculaJuros(int periodo, double valor) {
        -----
        -----
    }
}
```



# Identificadores reservados

abstract	assert	boolean	break	byte
case	catch	char	class	const*
continue	default	do	double	else
enum	extends	false	final	finally
float	for	goto*	if	implements
import	instanceof	int	interface	long
native	new	null	package	private
protected	public	return	short	static
strictfp	super	switch	synchronized	this
throw	throws	transient	true	try
void	volatile	while		

\*Não são utilizadas pela linguagem Java

# Distribuições do Java

## **JSE**

Principal pacote de distribuição.

Contém as principais bibliotecas para o desenvolvimento de aplicativos, como compilador, depurador, documentador além das API's básicas do Java.

## **JEE**

Pacote complementar ao JSE.

Contém API's utilizadas por aplicações distribuídas (Enterprise) tais como aplicações web, EJB's, Web Services.

## **JME**

Contém API's específicas para o desenvolvimento de aplicações para dispositivos portáteis como Palms, Handhelds e celulares.



# JDK x JRE

## JDK

Java Standard Developer Kit

Kit para desenvolvimento de aplicações em Java.

Contem todas as ferramentas necessárias para o programador Java.

## JRE

Java Runtime Environment

Núcleo do Virtual Machine

Contem somente o necessário para executar aplicações em Java.

# Ambiente de Desenvolvimento

Efetuar o download e instalação do JSDK a partir do site  
[www.oracle.com/technetwork/java/javase/downloads](http://www.oracle.com/technetwork/java/javase/downloads)

Criar a variável de ambiente **JAVA\_HOME** e alterar a variável **PATH** e conforme abaixo:

**JAVA\_HOME**=C:\Program Files\Java\jdk1.6.0\_22

**PATH**=%PATH%;%JAVA\_HOME%\bin

Se desejar, instale uma ferramenta auxiliar para digitar o código fonte de suas aplicações como o Eclipse, Netbeans ou outro.

# Primeiro Exemplo

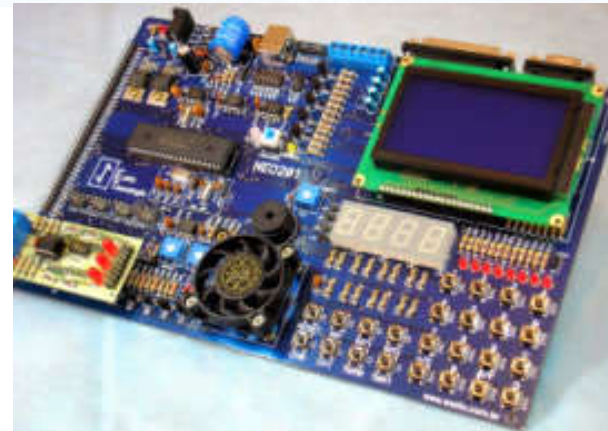
```
public class OlaMundo {  
    public static void main (String[ ] args) {  
        System.out.println("Olá, mundo!");  
    }  
}
```



# Utilização da Linguagem

- Desenvolvimento de softwares que envolvam uso de hardware
  - Sistemas Operacionais
  - Drivers de dispositivos
  - Robótica
  - Softwares Embarcado
  - Geração de Código de compiladores
- Softwares Aplicativos
  - Planilhas e Editores de Texto
  - Antivírus
  - Processadores de Imagens

# Exemplos de utilização



# Diferenças



- Programa principal é uma classe
- Lógica construída através da criação de objetos e dos relacionamentos entre eles
- Biblioteca de Classes
- Não há acesso direto à memória ou hardware
- Portabilidade
- Programa principal é uma função
- Lógica construída através da manipulação de informações por funções
- Bibliotecas de funções
- Acesso direto à memória e ao Hardware
- Performance

# Exemplos

```
import java.util.*;
public class Programa
{
    public static void main (string args[])
    {
        system.out.println ("Tecnicas de Programacao");
    }
}
```



```
#include <stdio.h>
int main()
{
    printf("Tecnicas de Programacao\n");
    return 0;
}
```





# Popularidade

Programming Language	2020	2015	2010	2005	2000	1995	1990	1985
Java	1	2	1	2	3	-	-	-
C	2	1	2	1	1	2	1	1
Python	3	7	6	6	22	20	-	-
C++	4	3	4	3	2	1	2	9
C#	5	5	5	9	9	-	-	-
JavaScript	6	8	8	10	6	-	-	-
PHP	7	6	3	5	21	-	-	-
SQL	8	-	-	-	-	-	-	-
Swift	9	16	-	-	-	-	-	-
R	10	12	54	-	-	-	-	-
Lisp	27	24	16	14	8	6	4	2
Fortran	31	25	23	15	16	4	3	5
Ada	33	29	24	17	14	5	6	3
Pascal	241	15	14	27	12	3	10	6



# Popularidade em projetos Open Source – 2018

## Language Types



Web

























Mobile



Enterprise





Embedded

Language Rank	Types	Spectrum Ranking
1. Python	  	100.0
2. C++	  	98.4
3. C	  	98.2
4. Java	  	97.5
5. C#	  	89.8
6. PHP		85.4
7. R		83.3
8. JavaScript	 	82.8
9. Go	 	76.7
10. Assembly		74.5

# Popularidade em projetos Open Source – 2020

Language Types

 Web  Mobile  Enterprise  Embedded

Rank	Language	Type	Score
1	Python▼	  	100.0
2	Java▼	  	95.3
3	C▼	  	94.6
4	C++▼	  	87.0
5	JavaScript▼		79.5
6	R▼		78.6
7	Arduino▼		73.2
8	Go▼	 	73.1
9	Swift▼	 	70.5
10	Matlab▼		68.4

# Popularidade (TIOBE)

Aug 2020	Aug 2019	Change	Programming Language	Ratings	Change
1	2	▲	C	16.98%	+1.83%
2	1	▼	Java	14.43%	-1.60%
3	3		Python	9.69%	-0.33%
4	4		C++	6.84%	+0.78%
5	5		C#	4.68%	+0.83%
6	6		Visual Basic	4.66%	+0.97%
7	7		JavaScript	2.87%	+0.62%
8	20	▲▲	R	2.79%	+1.97%
9	8	▼	PHP	2.24%	+0.17%
10	10		SQL	1.46%	-0.17%