

Java

1. Introdução

2. JSDK

**3. Sintaxe
Básica**

**4. Classes e
Objetos:
Atributos e
Métodos**

**5. Tipos de
Referência,
Strings e
Arrays**

**6. Herança,
Poliformismo e
Ligação
Dinâmica**

**7. Modelagem
em Camadas**

**8. Classes
Abstratas e
Interfaces**

9. Exceções

10. Pacotes

Conceitos básicos da linguagem

Fundamentos da linguagem

Operadores

Modificadores

Controle de fluxo

Conversão de tipos

Ambiente de desenvolvimento Java

J2SE SDK (Java 2 Standard Edition Software Development Kit)

IDE (Integrated Development Environment)

Conceitos de OO

Classes, objetos, atributos e métodos

Herança

Classes abstratas e interfaces

Exceções

Criação

Levantamento

Tratamento

Pacotes

Nomes

Importação

Pacotes e diretórios

APIs

Como consultar a documentação (API)

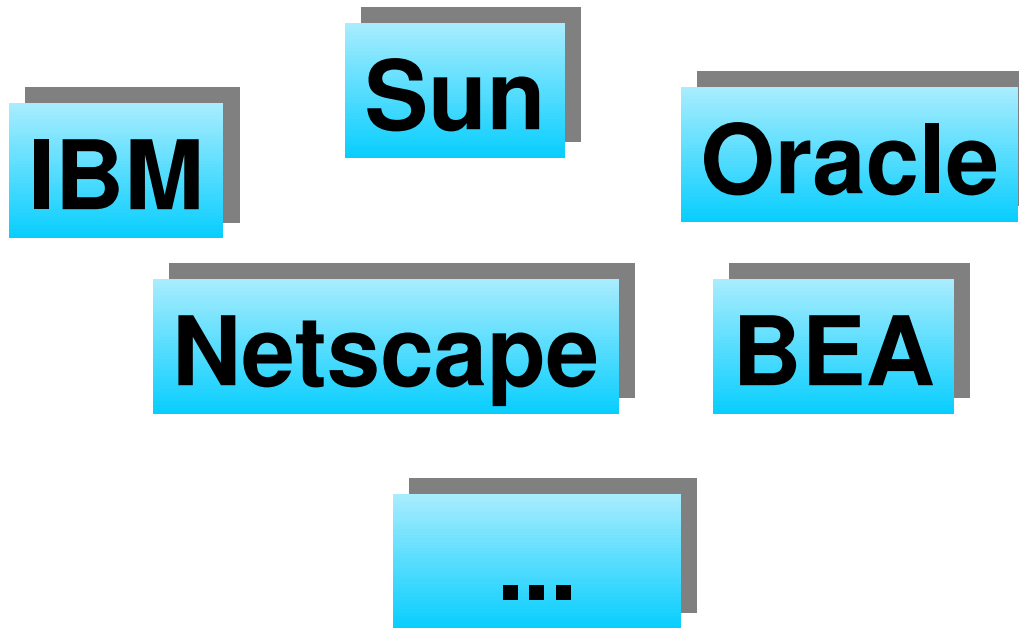
Pacote `java.lang`

Pacote `java.util`

- Origem de Java
 - Java surgiu do projeto Green, em 1990, na Sun Microsystems, que tinha como objetivo desenvolver softwares que fizessem diferentes dispositivos comunicarem entre si.
- A primeira versão da linguagem foi lançada em 1996
- O uso de Java cresceu rapidamente com o crescimento da Internet
- Mais de 5,000,000 de pessoas possuem o software de desenvolvimento Java

- O número de desenvolvedores Java passou de **3 milhões**
- Já há mais desenvolvedores Java que desenvolvedores C++
- Há um forte movimento *OpenSource* com Java
 - Bibliotecas completas
 - Aplicações e ferramentas gratuitas
 - Padrões abertos
- Java está se transformando em uma linguagem largamente utilizada para o desenvolvimento de **aplicações web e aplicações para dispositivos móveis**

- Empresas que apóiam fortemente Java



- O que é a plataforma Java?
 - É uma infra-estrutura para programação baseada no poder das redes de computadores e na idéia de que uma mesma aplicação possa executar em diferentes máquinas, aparelhos e dispositivos eletrônicos.

“Write once, run anywhere.”

- Simplicidade
- Portabilidade
- Distribuição
- Robustez
- Concorrência
- Performance
- Paradigma de Orientação a Objetos

- Sintaxe familiar a vários programadores (baseada em C e C++)
- Elimina várias redundâncias de C++
- Não é necessário gerenciar memória explicitamente
 - Destruição de objetos em memória
- Simples para muitas aplicações, desde que se conheça algumas APIs

- Redução de custos com migração, instalação, treinamento, etc.
- Algumas vezes ainda é necessário depurar programas antes de migrar para outra plataforma
- Java tem se tornado cada vez mais portátil: aplicações desenvolvidas e testadas uma única vez têm rodado em dezenas de plataformas sem alteração

- Oferece suporte de alto nível para acesso à Internet/Web
- Objetos distribuídos com RMI, CORBA e EJB
- Suporte para acesso a arquivos remotos, banco de dados, etc.

- Ausência de ponteiros
- Fortemente tipada
- Coleta de lixo automática (garbage collection)
- Exceções tipadas
- Variáveis (*member variables*) são inicializadas automaticamente

- Facilidade em criar aplicações com concorrência
- Métodos sincronizados
- Monitores (construções essenciais para o controle de concorrência)

- Código no **cliente** pode ser interpretado
 - Compiladores JIT aumentam performance
 - Cliente universal
 - Código móvel, carregado pela rede, evitando instalações, conflito entre versões, etc.
- Código no **servidor** pode ser compilado para código de máquina
 - Performance similar a C++
- A performance de Java vem aumentando rapidamente!

- Objetos e classes
- Encapsulamento de dados e operações
- Herança e Subtipos
- Polimorfismo
- Ligações dinâmicas (*dynamic binding*)
- Criação e remoção dinâmica de objetos
- Exceções

J2ME

Micro Edition

Palms

Celulares

Dispositivos inteligentes

J2SE

Standard Edition

Aplicações convencionais

Applets

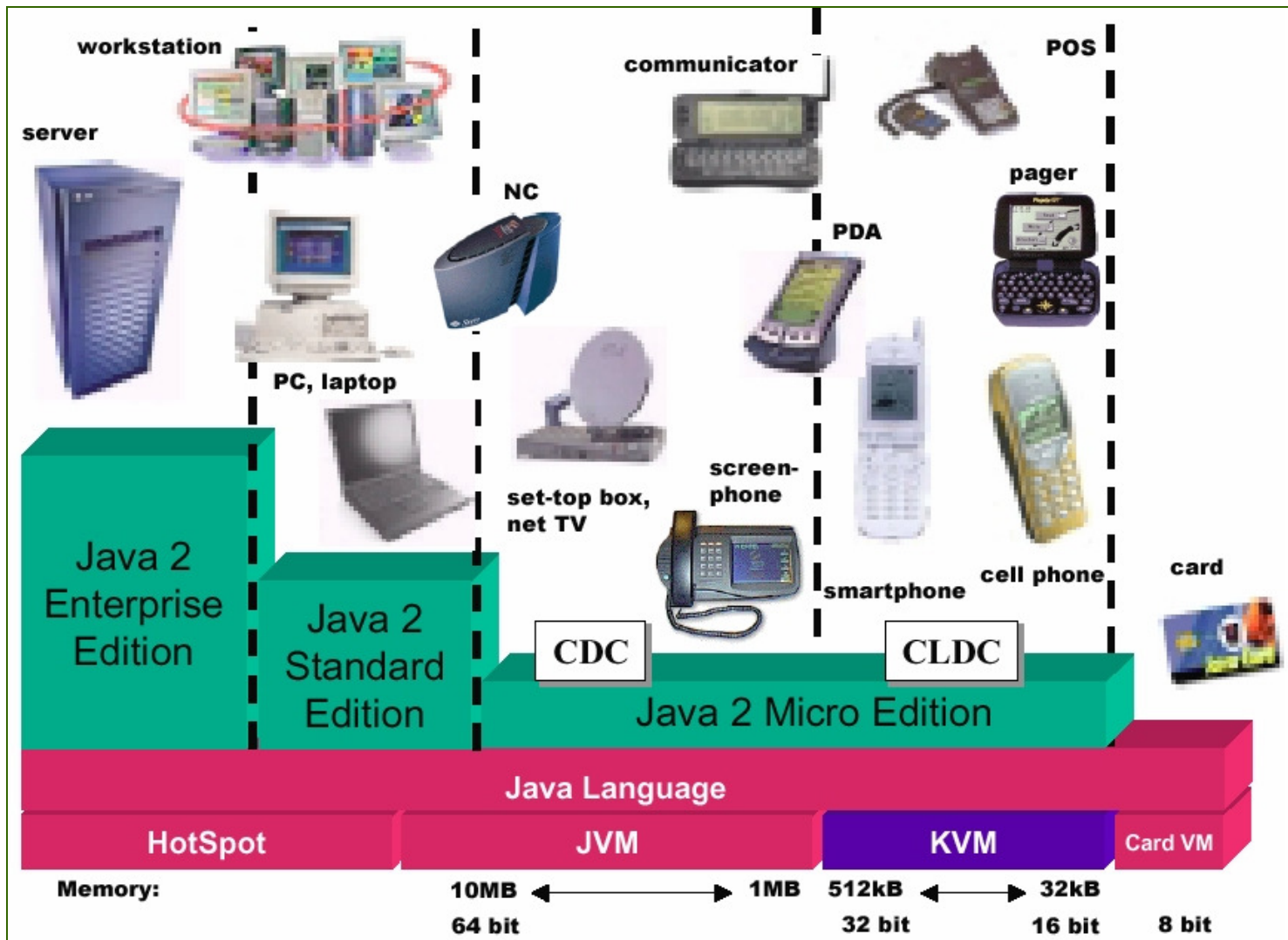
J2EE

Enterprise Edition

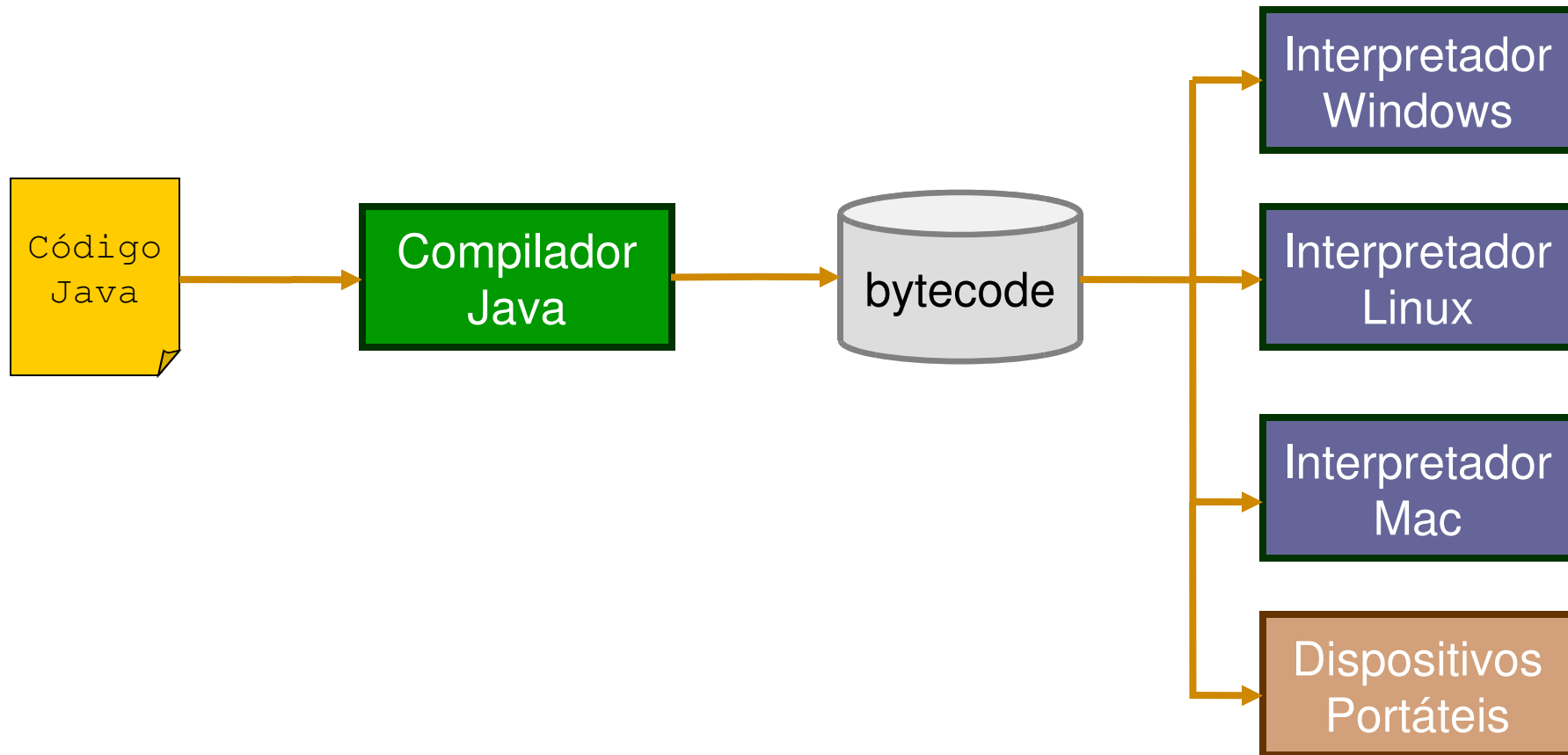
Aplicações Web

Aplicações Distribuídas

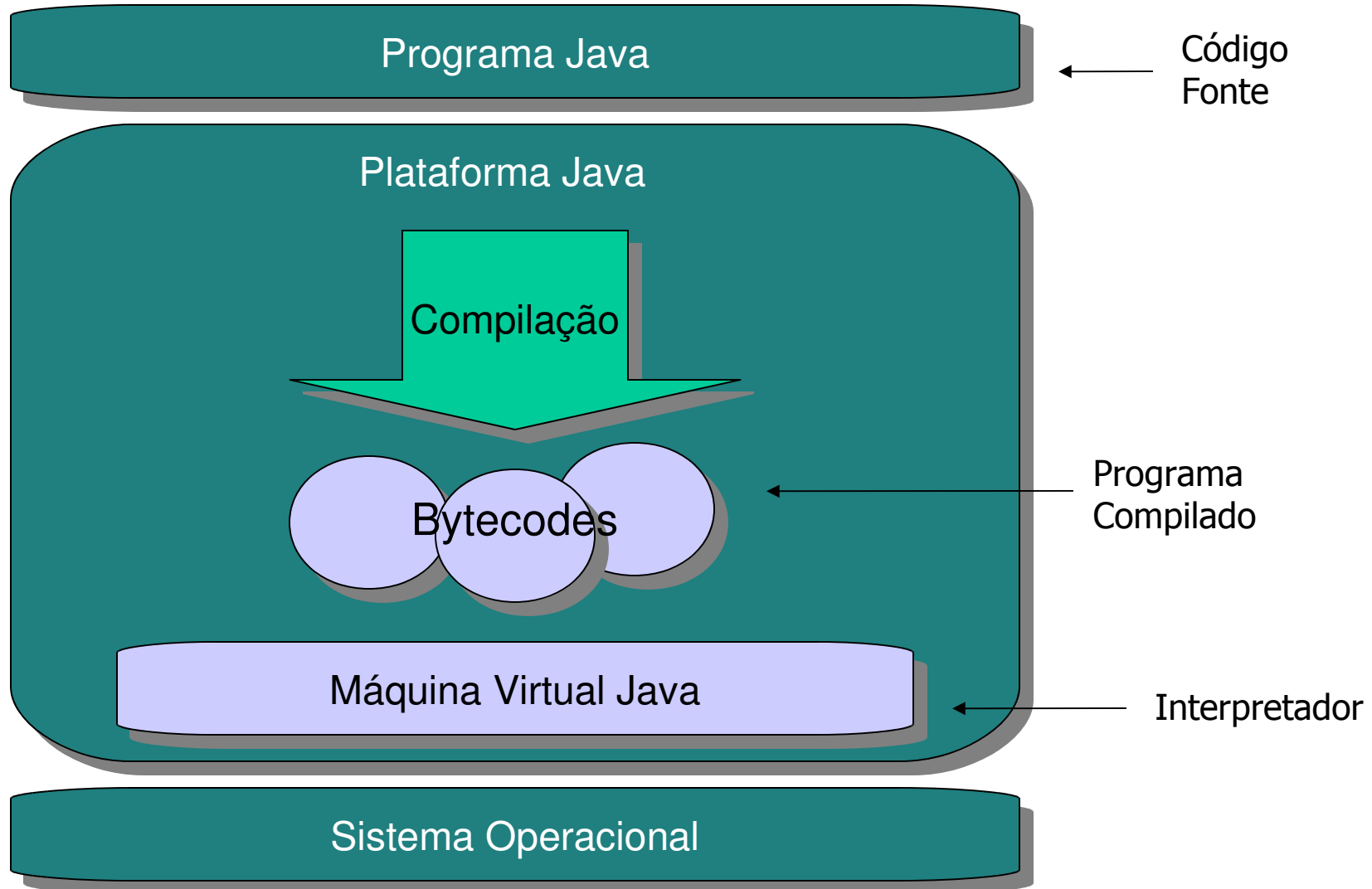
Aplicações Transacionais

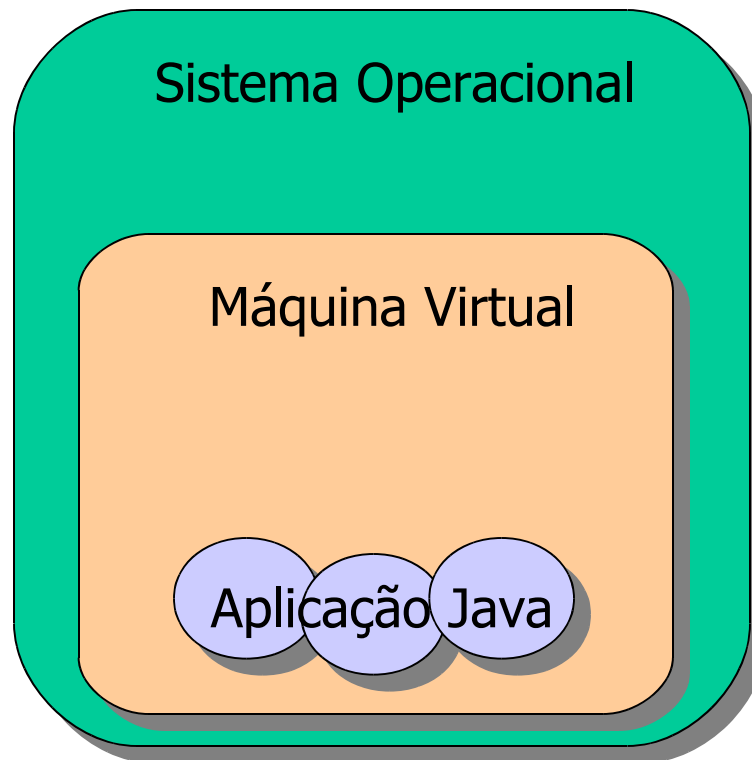


- Java Virtual Machine (JVM)
 - Componente da plataforma Java que assegura a independência das aplicações entre diferentes plataformas
 - É uma espécie de tradutor existente nos dispositivos para traduzir comandos da linguagem Java para a linguagem da máquina nativa.
- Plataformas compatíveis com Java
 - Implementam o tradutor da linguagem Java para sua linguagem de máquina



- O que são bytecodes?
 - Instruções de código de máquina específicas para a máquina virtual Java
- A máquina virtual transforma os bytecodes em instruções da máquina que está executando o programa





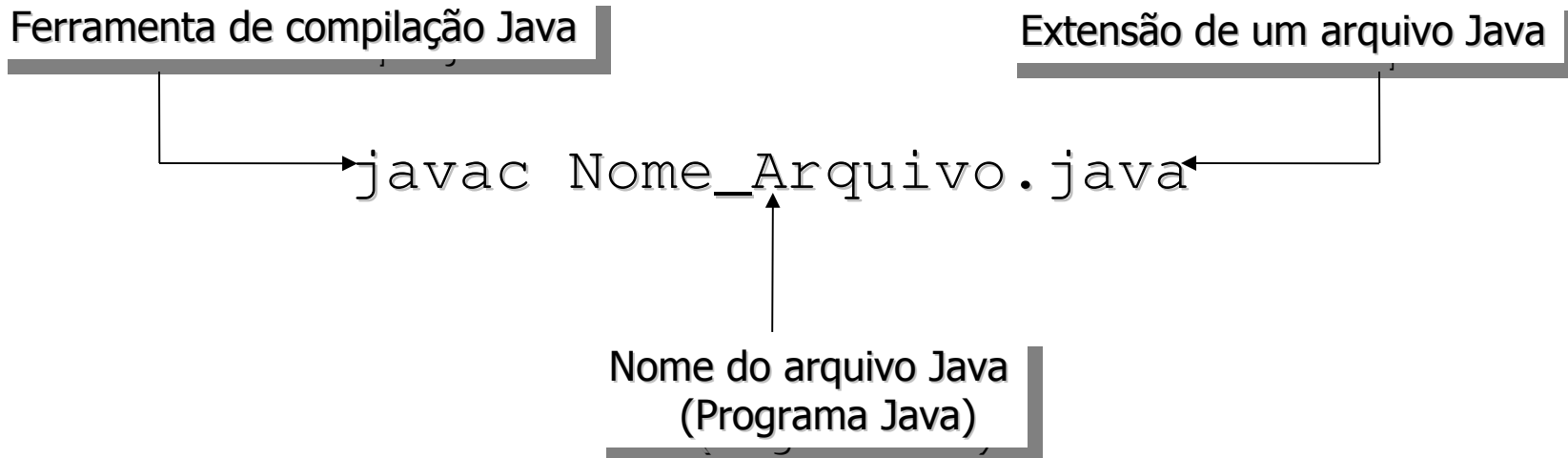
Java Software Development Kit

- Conjunto de ferramentas, bibliotecas e exemplos para o desenvolvimento em Java
- Ferramentas para desenvolvimento de aplicações Java usam o JSDK como base
 - Eclipse, Net beans, IntelliJ IDEA, Borland JBuilder, WSAD, entre outros

- Ferramentas para compilação, execução e depuração de programas Java
- Bibliotecas compiladas
- Código fonte completo das bibliotecas
- Exemplos em Java

- Essenciais
 - **javac**: compila programas Java
 - **java**: executa programas Java
- Importantes
 - **javadoc**: gera documentação automática
 - **jar**: manipula arquivos “Java Archive” (JAR)
- Outras
 - **javap**
 - **appletviewer**

- Compila arquivos **.java** transformando-os em **.class** (bytecodes)



- Compilação de um arquivo Java específico

`javac Conta.java`

- Compilação de todos os arquivos Java, em um mesmo diretório

`javac *.java`

- Compilação de mais de um arquivo Java

```
javac Conta.java Cliente.java
```

- Especificando o diretório no qual os arquivos .class devem ser gerados

```
javac *.java -d c:\MinhasClasses
```


- Interpretador Java
- Utilizado para:
 - Executar arquivos **.class** (gerados após a compilação, a partir dos arquivos **.java**)
 - Executar arquivos **.jar**
- Um arquivo Java precisa conter um método **main** para poder ser executado:

```
public static void main (String[] args) {  
  
    /*código Java que deve ser executado */  
  
}
```

Interpretador Java

java Nome_Arquivo

Nome do arquivo
(a extensão **.class**
não deve ser fornecida)

java Conta

java -jar bancoDatosus.jar

- Extrai toda a documentação no estilo “javadoc” (`/** ... */`)
- Gera um site completo com a documentação

Ferramenta para geração de documentação html

→ `javadoc Nome_Arquivo.java`

Nome do arquivo com a extensão .java

- Gera a documentação para todos os arquivos Java, no mesmo diretório onde estes estão localizados

javadoc *.java

- Gera a documentação em um diretório diferente da localização dos arquivos .java

javadoc -d nome_do_diretorio *.java

Para criar um arquivo .jar

```
jar cf arquivo.jar *.class  
jar cf arquivo.jar *.java
```

Para listar o conteúdo de um arquivo .jar

```
jar tf arquivo.jar
```

Para extrair o conteúdo de um arquivo .jar

```
jar xf arquivo.jar
```

- *Java class disassembler* (decompilador)
- Analisa um arquivo .class e retorna um código Java correspondente
- Pode também retornar os bytecodes para os métodos da classe

```
javap MinhaClasse
```

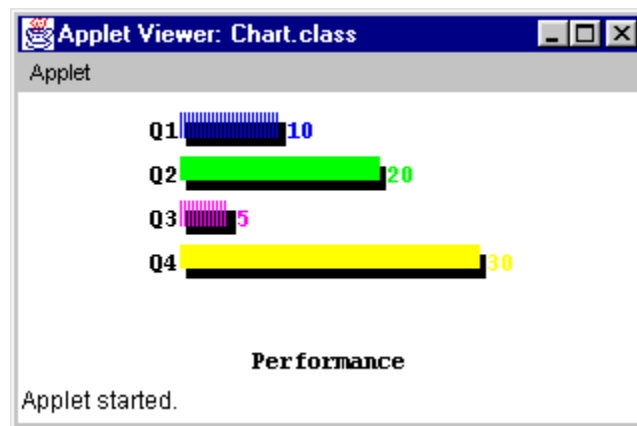
mostra definição
dos métodos

```
javap -c MinhaClasse
```

mostra definição
e bytecodes

- Usado para testar Applets sem a necessidade de um browser
- Permite maior controle sobre a execução de Applets do que em um browser

```
appletviewer pagina.html
```



- Java 2 Plataform, Standard Edition v1.5
- Java 2 Plataform, Standard Edition v1.4
- Java 2 Plataform, Standard Edition v1.3
- Java 2 Plataform, Standard Edition v1.2
- Java 2 Plataform, Standard Edition v1.1
- Java 2 Plataform, Standard Edition v1.0

