

POO - PROGRAMAÇÃO ORIENTADA A OBJETOS

Plataforma JAVA

Rodrigo R Silva

Instituto Federal de Educação, Ciência e Tecnologia Sul-Rio-Grandense
Campus Bagé

- Java é uma linguagem de programação e uma plataforma. Java é alto nível, robusta, linguagem de programação segura e orientada a objeto.
- Java foi desenvolvido pela Sun Microsystems (que agora é filial da Oracle) no ano de 1995. James Gosling é conhecido como o pai do java. Antes java, tinha como nome Oak. Visto que Oak já era usado por uma companhia registrada, portanto James Gosling e seu time mudaram o nome Oak para Java.
- **Plataforma:** Qualquer hardware ou ambiente de software em que um programa rode, é conhecido como uma plataforma. Já que java tem um ambiente de tempo de execução (JRE) e API, é chamado de plataforma.

Exemplo Código JAVA

```
class Simples {  
    public static void main(String args[]) {  
        System.out.println("Olá Java");  
    }  
}
```

Aplicação

De acordo com a Sun, 3 bilhões de dispositivos rodam java. Existem muitos dispositivos onde java é usado atualmente. Alguns deles são os seguintes:

- 1 Aplicações desktop tais como acrobat reader, media player, antivírus, etc.
- 2 Aplicações web tais como irctc.co.in, javatpoint.com, etc.
- 3 Aplicações enterprise tais como aplicações bancárias.
- 4 Mobile.
- 5 Sistema embarcado.
- 6 Smart card.
- 7 Robótica.
- 8 Jogos, etc.

Tipos de Aplicações

Existem principalmente quatro tipos de aplicações que podem ser criadas usando programação java:

- **Aplicação Standalone:** Aplicações Standalone são também conhecidas como aplicações desktop, ou aplicações baseadas em janelas. Existem softwares tradicionais que nós precisamos instalar em cada máquina. Exemplos de aplicações standalone são Media player, antivírus, etc. AWT e Swing são usados em java para criar aplicações standalone.
- **Aplicações Web:** Uma aplicação que roda no lado do servidor e cria uma página dinâmica é chamada de aplicação web. Atualmente, Servlet, JSP, Struts, Spring, Hibernate, JSF, etc. São tecnologias usadas para criar aplicações web em java.

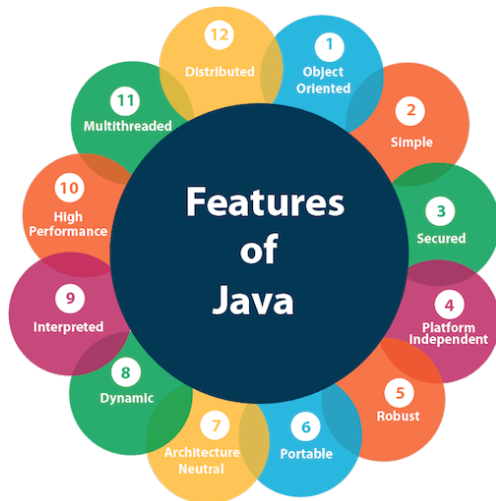
Tipos de Aplicações

- Aplicações Enterprise: Uma aplicação que é distribuída, tais como aplicações bancárias, etc. É chamada aplicação enterprise. Possui vantagens no alto nível de segurança, balanceamento de carga (load balancing), e agrupamento (clustering). Em java, EJB é usado para criar aplicações enterprise.
- Aplicações Mobile: Uma aplicação que é criada para dispositivos móveis é chamada de aplicação mobile. Atualmente, Android e Java ME são usados para criar aplicações mobile.

Plataformas JAVA

Existem quatro plataformas ou edições de java:

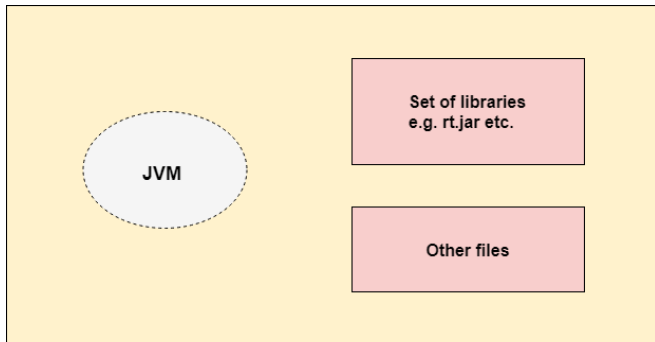
- Java SE (Java Standard Edition): Esta é uma plataforma de programação java. Inclui APIs de programação java tais como `java.lang`, `java.io`, `java.net`, `java.util`, `java.sql`, `java.math` etc. Inclui principais tópicos como OOPs, String, Expressões Regulares (Regex), Exceções (Exception), Classes internas (Inner classes), Multithreading, I/O Stream, Networking, AWT, Swing, Reflection, Collection, etc.
- Java EE (Java Enterprise Edition): Esta é uma plataforma enterprise que é principalmente usada para desenvolver aplicações web e enterprise. É construída no topo da plataforma Java SE. Inclui tópicos como Servlet, JSP, Web Services, EJB, JPA, etc.
- Java ME (Java Micro Edition): É uma micro plataforma que é principalmente usada para desenvolver aplicações mobile.
- JavaFX: É usada para desenvolver aplicações ricas. Usa uma API de interface de usuário leve.



- JVM (Java Virtual Machine) é uma máquina abstrata. É chamada de máquina virtual porque não existe fisicamente. É uma especificação que fornece um ambiente de tempo de execução no qual o bytecode do java pode ser executado. Ele também pode executar os programas que são escritos em outras linguagens e compilados no bytecode do ava.
- JVMs estão disponíveis para muitas plataformas de hardware e software. JVM, JRE e JDK dependem da plataforma porque a configuração de cada sistema operacional é diferente uma da outra. No entanto, Java é independente de plataforma. Existem três noções da JVM: especificação, implementação e instância.
- A JVM executa as seguintes tarefas principais:
 - Carrega código
 - Verifica código
 - Executa código
 - Fornece ambiente de tempo de execução

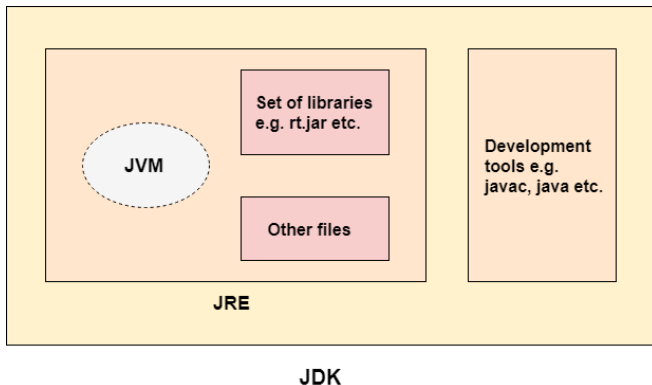
- JRE é um acrônimo para Java Runtime Environment. Também é escrito como Java RTE. O Java Runtime Environment é um conjunto de ferramentas de software usadas para o desenvolvimento de aplicativos Java. É usado para fornecer o ambiente de tempo de execução. É a implementação da JVM. Existe fisicamente. Ele contém um conjunto de bibliotecas mais outros arquivos que a JVM usa em tempo de execução.
- A implementação da JVM também é ativamente lançada por outras empresas além da Sun Micro Systems.

JRE



JRE

- JDK é um acrônimo para Java Development Kit. O Java Development Kit (JDK) é um ambiente de desenvolvimento de software usado para desenvolver aplicativos e applets Java. Existe fisicamente. Ele contém JRE mais ferramentas de desenvolvimento.
- O JDK é uma implementação de qualquer uma das plataformas Java abaixo fornecidas pela Oracle Corporation:
 - Plataforma Java Edição Standard
 - Plataforma Java Enterprise Edition
 - Plataforma Java Micro Edition
- O JDK contém uma Java Virtual Machine (JVM) privada e alguns outros recursos, como um Interpretador / carregador (java), um compilador (javac), um arquivador (jar), um gerador de documentação (Javadoc) etc. para concluir o desenvolvimento de uma aplicação Java.



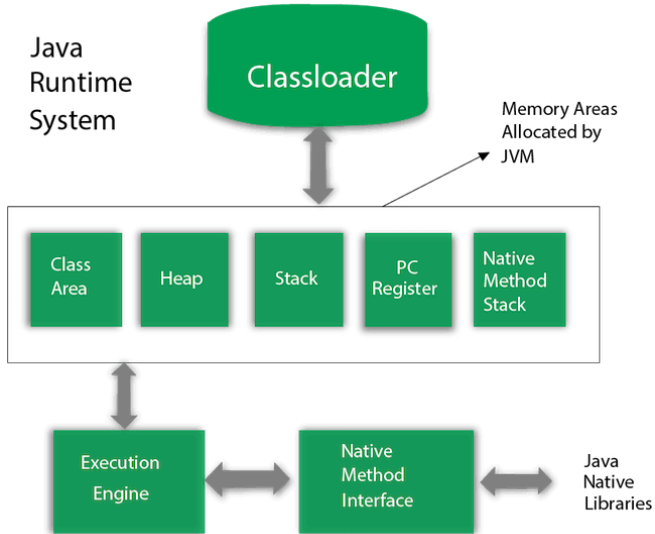
Arquitetura JVM

- Uma especificação em que o trabalho da Java Virtual Machine é especificado. Mas o provedor de implementação é independente para escolher o algoritmo. Sua implementação foi fornecida pela Oracle e outras empresas.
- Uma implementação Sua implementação é conhecida como JRE (Java Runtime Environment).
- Instância de tempo de execução Sempre que você grava o comando java no prompt de comando para executar a classe java, uma instância da JVM é criada.

Arquitetura JVM

- A JVM executa a seguinte operação:
 - Carrega código
 - Verifica código
 - Executa código
 - Fornece ambiente de tempo de execução
- A JVM fornece definições para:
 - Área de memória
 - Formato de arquivo de classe
 - Conjunto de registros
 - Pilha de garbage collector
 - Relatório de erros fatais etc.

Arquitetura JVM

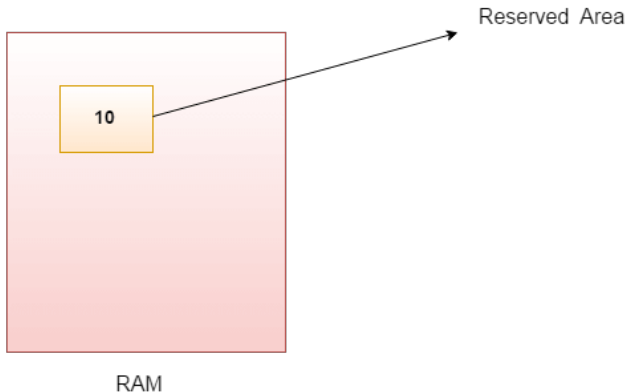


Variáveis JAVA

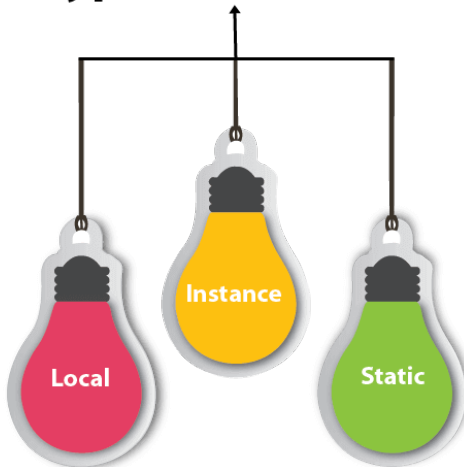
- Uma variável é um contêiner que armazena o valor enquanto o programa java é executado. Uma variável é assinada com um tipo de data.
- Variável é um nome de alocação de memória. Existem três tipos de variáveis em java: local, global e estática.
- Existem dois tipos de data em java: primitiva e não primitiva.

Variável

- Variável é o nome de uma área reservada de alocação de memória. Em outras palavras, é um nome de memória alocada. É uma combinação de "vary + able" isso significa que seu valor pode ser modificado.



Types of Variables

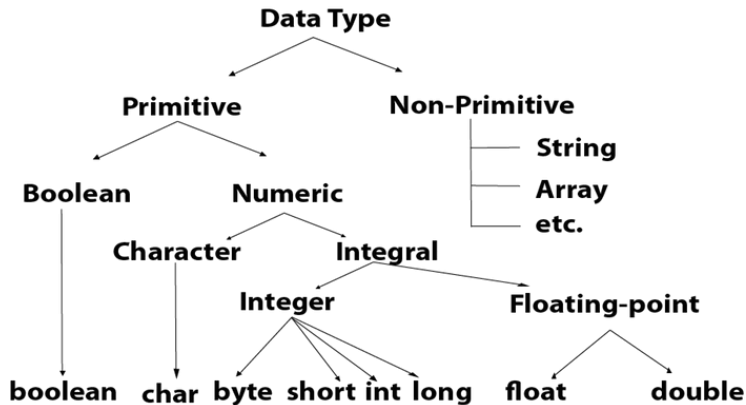


Tipo de Dado em JAVA

Tipos de dados são especificados de diferentes tamanhos e valores que podem ser armazenados na variável. Existem dois tipos de dados em java:

- Tipo de dado primitivo: Incluem boolean, char, byte, short, int, long, float, e double.
- Tipo de dado não primitivo: Incluem classes, interfaces e arrays.

Tipo de Dado em JAVA



Tipos de Dado Primitivo em JAVA

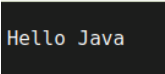
Data Type	Default Value	Default size
boolean	false	1 bit
char	'\u0000'	2 byte
byte	0	1 byte
short	0	2 byte
int	0	4 byte
long	0L	8 byte
float	0.0f	4 byte
double	0.0d	8 byte

Primeiro Programa em JAVA | Exemplo de Hello World

```
class Simple {  
    public static void main(String args[]) {  
        System.out.println("Hello Java");  
    }  
}
```

Salve este arquivo como Simple.java

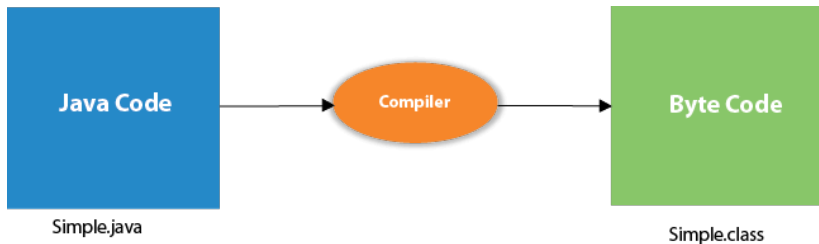
- Compilar: `javac Simple.java`
- Executar: `java Simple`
- Saída:



Hello Java

Fluxo de Compilação

- Quando nós compilamos um programa java usando a ferramenta javac, o compilador java converte o código fonte em bytecode

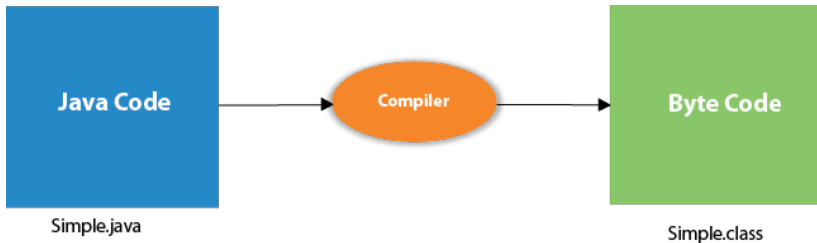


Parâmetros usados no Primeiro Programa Java

- **class:** Palavra reservada que é usada para declarar um classe em java.
- **public:** Palavra reservada que é um modificador de acesso que representa visibilidade. Significa que é visível para todos.
- **static:** É uma palavra reservada. Se nós declararmos algum método como static, é conhecido como método estático. A principal vantagem do método estático é que não é necessário criar um objeto para invocar o método estático. O método principal é executado pela JVM, mas não requer a criação de um objeto para invocar o método principal. Por isso, economiza memória.
- **void:** É um tipo de retorno de método. Significa que não retorna nenhum valor.
- **main:** Representa o ponto de início do programa.
- **String args[]:** É usado para argumento de linha de comando.
- **out.println():** É usada imprimir declarações. Aqui, System é uma class, out é o objeto da classe PrintStream, print() é o método da classe PrintStream.

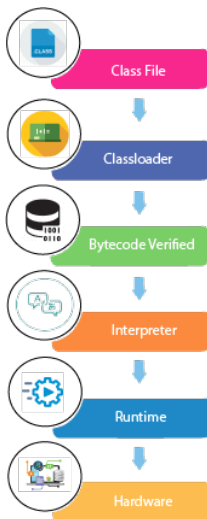
Detalhes Internos

- Em tempo de compilação, o arquivo java é compilado pelo compilador java (ele não interage com o sistema operacional) e converte o código java em bytecode.

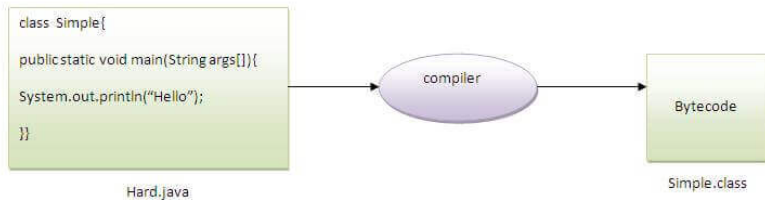


Detalhes Internos

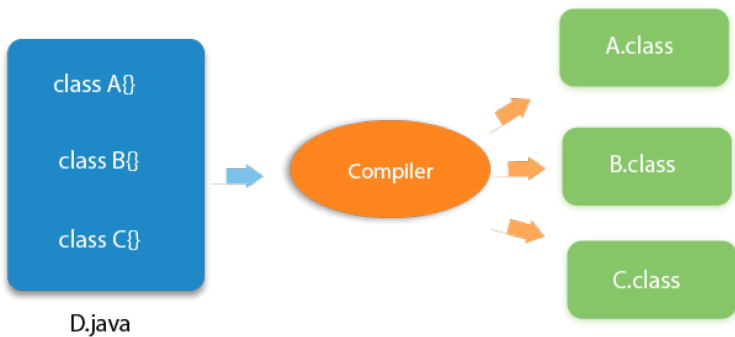
- Em tempo de execução, as seguintes etapas são executadas:



Detalhes Internos



Detalhes Internos



OBRIGADO!