

Dia 3 – Introdução ao java



O termo Java é utilizado, geralmente, quando nos referimos a:

- -Linguagem de programação orientada a objetos;
- -Ambiente de desenvolvimento composto pelo compilador, interpretador, gerador de documentação etc.;
- -Ambiente de execução que pode ser praticamente qualquer máquina que possua Java Runtime Environment (JRE) instalado;

A linguagem de programação Java é uma linguagem de alto-nível com as seguintes características:

- Simples: O aprendizado da linguagem de programação Java pode ser feito em um curto período;
- Orientada a objetos: Desde o início do seu desenvolvimento esta linguagem foi projetada para ser orientada a objetos;
- Familiar: A linguagem Java é muito familiar para os programadores C/C++;
- Robusta: Ela foi pensada para o desenvolvimento de softwares confiáveis
- Segura: Aplicações Java são executadas em ambiente próprio (JRE) o que inviabiliza a intrusão de código malicioso;
- Portável: Programas desenvolvidos nesta linguagem podem ser executados em praticamente qualquer máquina desde que esta possua o JRE instalado;

Dia 3 – Introdução ao java

JRE e JDK

JRE: O Java Runtime Environment contém tudo aquilo que um usuário comum precisa para executar uma aplicação Java (JVM e bibliotecas), como o próprio nome diz é o “Ambiente de execução Java”;

JDK: O Java Development Kit é composto pelo JRE e um conjunto de ferramentas úteis ao desenvolvedor Java.

JVM: é uma máquina imaginária que emula uma aplicação em uma máquina real. É a JVM que permite a portabilidade do código Java, isto ocorre porque todo código Java é compilado para um formato intermediário, bytecode, este formato é então interpretado pela JVM.



Primeiro contato com o Java

Definições:

- Classe: É a estrutura que, quando construída, produzirá um objeto, dizemos “todo objeto é instância de alguma classe”;
- Objeto: Em tempo de execução, quando a JVM encontra a palavra reservada new é criada uma instância da classe apropriada;
- Estado: É definido pelo conjunto de atributos de uma classe, isto é, cada instância da classe possuirá um estado independente dos demais objetos.
- Comportamento: São os métodos da classe, comportamento é aquilo que uma classe faz (algoritmos), muitas vezes, um determinado comportamento (método) muda o estado do objeto, isto é, após a execução do método um ou mais atributos mudaram de valor;

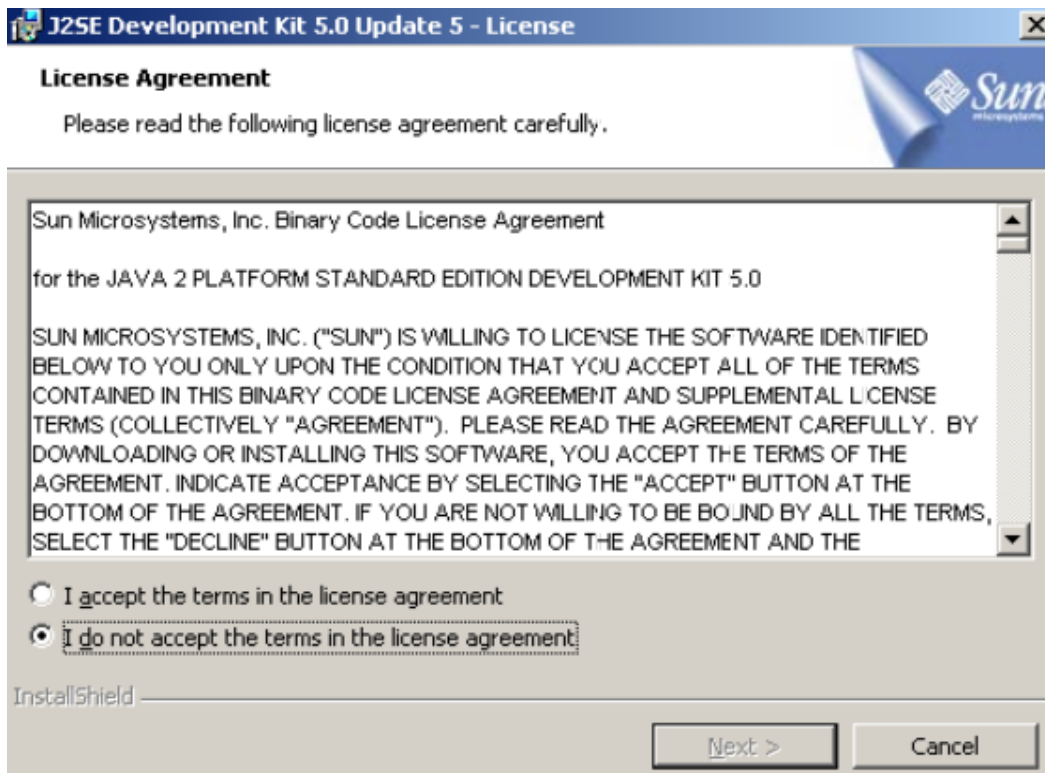
Instalação do JDK

Dia 3 – Introdução ao java

A seguir veremos os passos necessários para a instalação do JDK no Windows, porém, antes de iniciarmos, vamos dar uma olhada nas variáveis de ambiente que devem ser atualizadas e criadas quando instalamos o JDK:

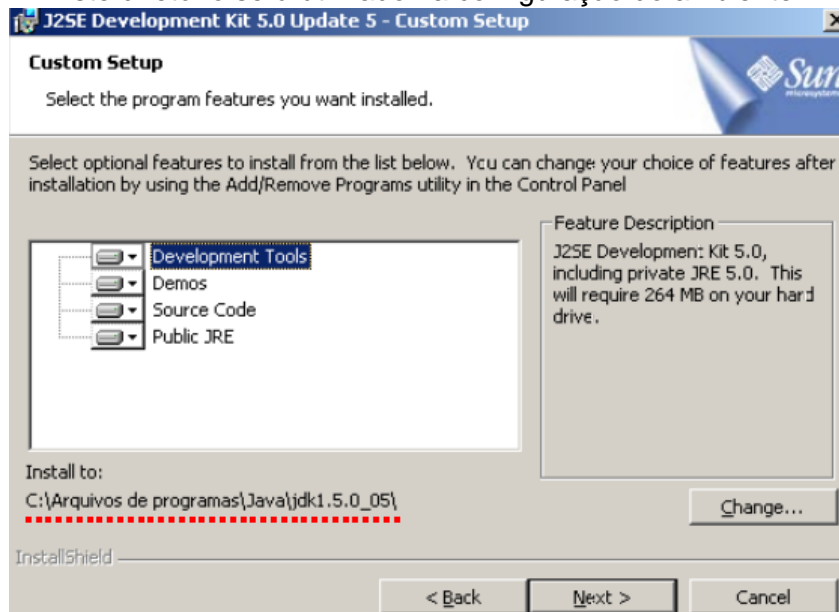
- JAVA_HOME: Indica o diretório onde foi instalado o JDK, muito utilizado por frameworks e por outros programas para localizar o JDK;
-
- PATH: Identifica o local onde encontram-se as ferramentas de desenvolvimento (compilador, interpretador, gerador de documentação etc.), devemos adicionar o diretório JAVA_HOME\bin;
-
- CLASSPATH: Identifica diretório onde o ClassLoader pode encontrar classes que são utilizadas pela sua aplicação.

Faça o download do JDK(<https://www.oracle.com/java/technologies/javase-downloads.html>) no site da SUN e execute o instalador.

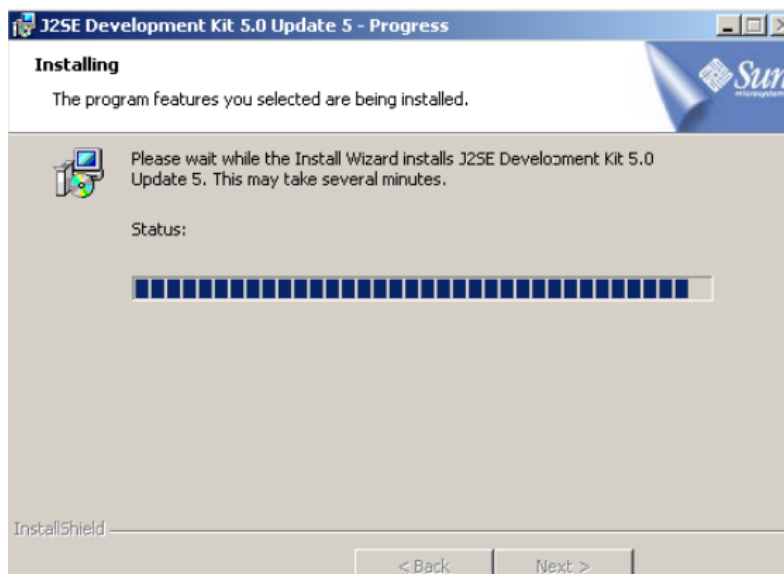


Dia 3 – Introdução ao java

Este diretório será utilizado na configuração do ambiente.

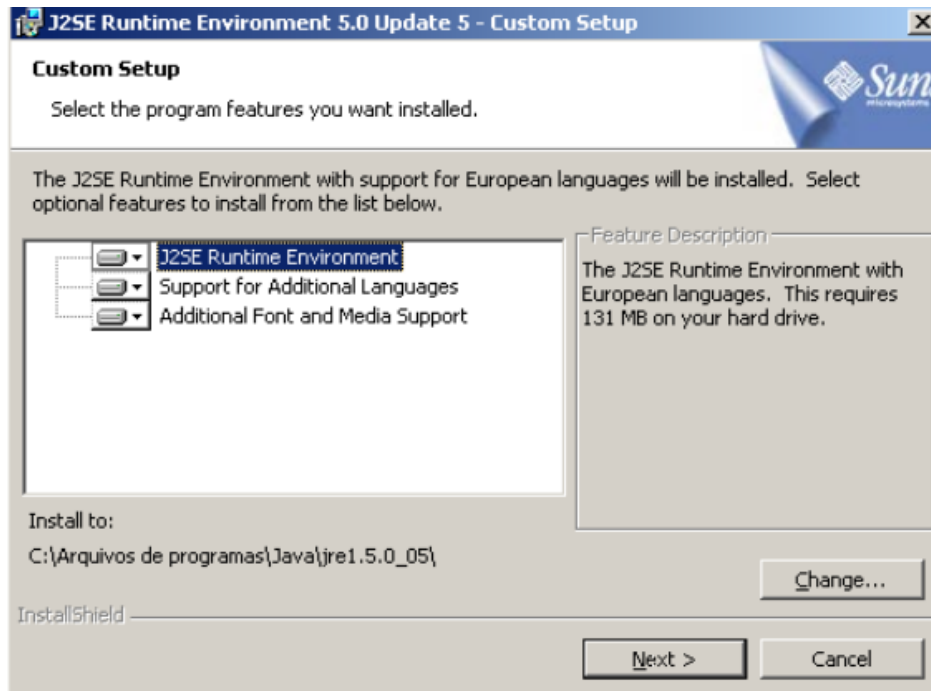


A seguir clique em Next> até que a instalação seja iniciada.

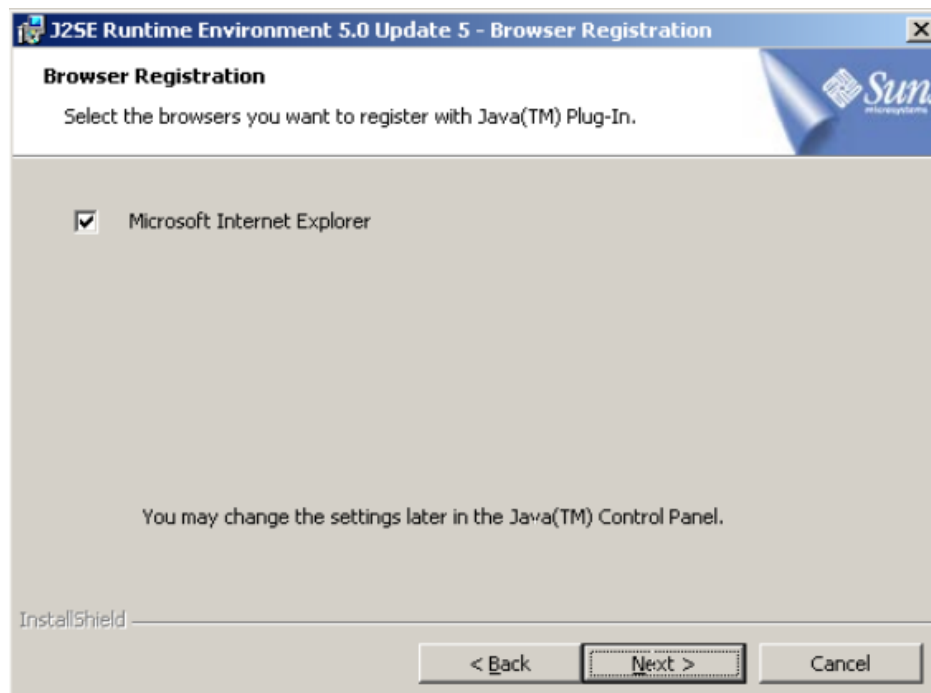


Dia 3 – Introdução ao java

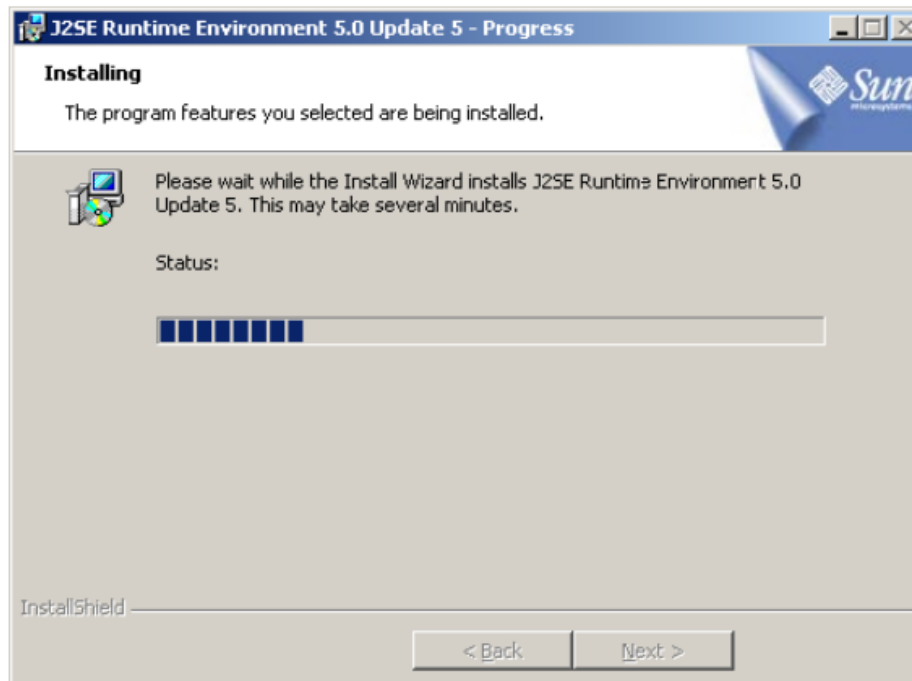
Neste momento inicia-se a instalação do JRE, continue clicando em Next>.



Nesta tela selecione todos os navegadores disponíveis.



Dia 3 – Introdução ao java



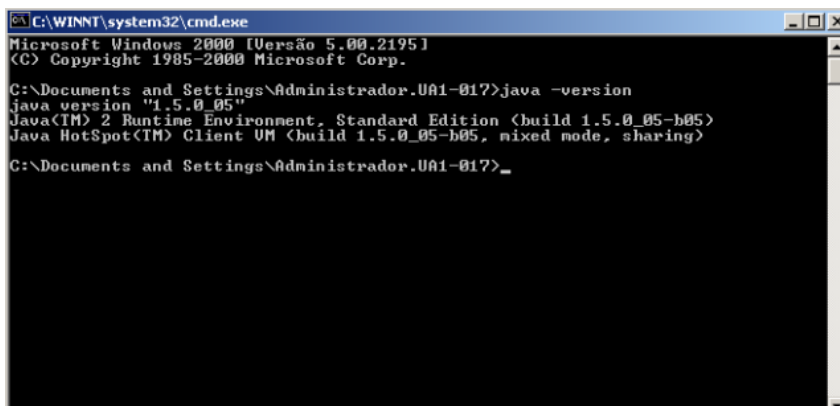
Clique em Finish para terminar o processo.



Dia 3 – Introdução ao java

Após a instalação do Java nós devemos iniciar a configuração das variáveis de ambiente.

1. Clique com o botão direito em cima do ícone “Meu Computador”;
 2. Vá em “Propriedades”;
 3. Selecione a aba “Avançado”;
 4. Clique no botão “Variáveis de ambiente”;
 5. Clique no botão “Nova” em “Variáveis do sistema”;
 - 5.1. Nome da variável: JAVA_HOME
 - 5.2. Valor da variável: Coloque aqui o endereço de instalação neste caso = C:\Arquivos de programas\Java\jdk1.5.0_05
 - 5.3. Clique em OK
 6. Clique novamente no botão “Nova” em “Variáveis do sistema”;
 - 6.1. Nome da variável: CLASSPATH
 - 6.2. Valor da variável;
%JAVA_HOME%\lib; %JAVA_HOME%\lib\tools.jar; %JAVA_HOME %\lib\dt.jar;
%JAVA_HOME%\lib\htmlconverter.jar; %JAVA_HOME%\ jre\lib;
%JAVA_HOME%\jre\lib\rt.jar
 - 6.3. Clique em OK
 7. Selecione a variável PATH em “Variáveis do sistema”;
 - 7.1. Adicione o seguinte endereço ao campo Valor da variável:
 - 7.2. %JAVA_HOME%\bin
 - 7.3. Clique em OK;
 8. Clique em OK. Agora vamos testar a instalação. Clique no botão Iniciar, vá em Executar e digite cmd.
- digite java -version, deverá aparecer algo parecido com isto:



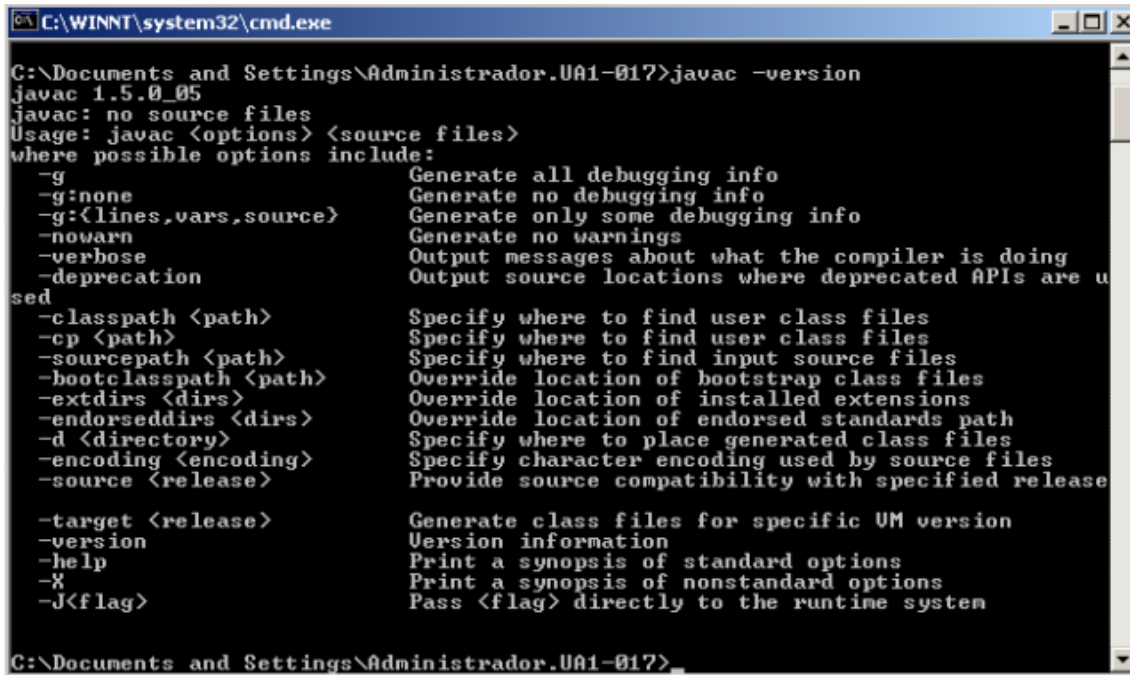
```
C:\WINNT\system32\cmd.exe
Microsoft Windows 2000 [Versão 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\Documents and Settings\Administrador.U01-017>java -version
java version "1.5.0_05"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_05-b05)
Java HotSpot(TM) Client VM (build 1.5.0_05-b05, mixed mode, sharing)

C:\Documents and Settings\Administrador.U01-017>
```

Dia 3 – Introdução ao java

digite javac -version, deverá aparecer algo parecido com isto:



```
C:\WINNT\system32\cmd.exe
C:\Documents and Settings\Administrador.UA1-017>javac -version
javac 1.5.0_05
javac: no source files
Usage: javac <options> <source files>
where possible options include:
  -g               Generate all debugging info
  -g:none          Generate no debugging info
  -g:<lines,vars,source> Generate only some debugging info
  -nowarn          Generate no warnings
  -verbose         Output messages about what the compiler is doing
  -deprecation     Output source locations where deprecated APIs are used
  -classpath <path> Specify where to find user class files
  -cp <path>       Specify where to find user class files
  -sourcepath <path> Specify where to find input source files
  -bootclasspath <path> Override location of bootstrap class files
  -extdirs <dirs>   Override location of installed extensions
  -endorseddirs <dirs> Override location of endorsed standards path
  -d <directory>   Specify where to place generated class files
  -encoding <encoding> Specify character encoding used by source files
  -source <release> Provide source compatibility with specified release
  -target <release> Generate class files for specific VM version
  -version         Version information
  -help           Print a synopsis of standard options
  -X             Print a synopsis of nonstandard options
  -J<flag>        Pass <flag> directly to the runtime system
C:\Documents and Settings\Administrador.UA1-017>
```

Método main

Antes de iniciar o método main, vamos abrir um breve parênteses para introduzir a forma como comentamos o código em Java. Basicamente existem duas formas:

1. // texto: Esta é a forma de comentar apenas uma linha de código
2. /* texto */: Esta declaração é utilizada quando desejamos comentar mais de uma linha de código

Vejamos a classe anterior agora com comentários:

```
//Este é o comentário de uma linha

public class Teste {

/*
Este é o comentário de mais de uma linha
*/

public static void main(String[] parametros) {

System.out.println("30code Java Basico");

}

}
```

Retornando ao main, temos o seguinte:

Dia 3 – Introdução ao java

```
public static void main(String[] parametros)
```

O main é o método que inicia as aplicações Java, quando solicitamos ao interpretador que execute uma determinada classe compilada ele procura o método main, se este método não existir irá ser gerada uma exceção informando que o método não foi localizado.

A JVM só irá reconhecer o método main se ele possuir as seguintes características:

1. Ser público (public);
2. Estático (static);
3. Não retornar nenhum valor (void);
4. O nome deve ser “main”;
5. Receber como parâmetro um array de String