

Curso de Extensão

Programação Orientada a Objetos com Java

Última atualização: 25/05/2010

Objetivos do Curso

Este curso fornece a você conhecimento e habilidades para:

- Programar e executar aplicações avançadas em Java™,
- Compreender o paradigma de programação orientada a objetos,
- Auxiliar na preparação para a certificação “Sun Certified Programmer for the Java™ 6.0 Platform”.

Visão Geral

Este curso compreende as seguintes áreas:

- A sintaxe da linguagem de programação Java,
- Conceitos de Orientação a Objetos relativos à Java,

Visão Geral Módulo a Módulo

Módulo 01 – Introdução à Tecnologia Java, ao IDE NetBeans e às provas de certificação Java

Módulo 02 – Introdução à Programação Orientada a Objetos

Módulo 03 – Identificadores, Palavras Chaves e Tipos

Módulo 04 – Controle de Fluxo

Módulo 05 – Arrays

Módulo 06 – Desenho de Classes

Módulo 07 – Características Avançadas de Classes e Princípios de Design

Tópicos Não Cobertos

- Conceitos gerais de programação. Este não é um curso para pessoas que nunca programaram antes.
- Servlets, JSP – (Curso de Programação para a Web)
- Enterprise Java Beans – EJB. (Curso de Computação Distribuída em Java)
- JDBC, JPA, AWT, JFC/Swing - (Conceitos vistos em outros cursos de extensão)

Módulo 01

Introdução à Tecnologia Java, ao IDE NetBeans e às Provas de Certificação Java

Sopa de Letrinhas

JSP	EJB	POJO	J2SE	WebSphere	JUnit	JavaFX	WSDP
Servlets	Spring MVC	JAXP	Java SE	Tomcat	Cruise Control	Android	JDK
JSTL	Hibernate	JWS	Java ME	Beanutils	Ant	Custom Tags	WSDK
JSF	IReport	JProfiler	J2EE	Generics	Maven	GlassFish	JAR
JPA	JFreeChart	Tiles	Java EE	RMI	Subversion	JAX-WS	AWT
JFC/Swing	JDBC	EL	GWT	Jxta	Tortoise	JAX-RS	UDDI
JBoss Seam	Swing Framework	Java Annotations	JBoss	JINI	MVC	JVM	JNLP
Struts	Beans	Managed Beans	WebLogic	JNI	J2EE Patterns	JRE	DAO

? ? ? ? ? ? ? ? ?

O que é a tecnologia Java?

- A tecnologia Java é:
 - Uma Linguagem de programação
 - Um ambiente de desenvolvimento
 - Um ambiente de aplicação
 - Um ambiente de execução.
- Similar em sintaxe a C++; similar em semântica em SmallTalk
- É utilizada no desenvolvimento de aplicações e applets.

Principais Metas da Tecnologia Java

- Fornecer uma linguagem de fácil utilização:
 - a) Evita armadilhas de outras linguagens
 - b) Por ser orientada a objetos
 - c) Permite a criação de código mais legível
- Fornecer um ambiente interpretado para:
 - a) Tornar ágil a velocidade de desenvolvimento
 - b) Portabilidade de código.

Principais Metas da Tecnologia Java

- Permitir que usuários executem mais de uma *thread* de atividade,
- Suportar mudanças de forma dinâmica durante o tempo de execução de uma aplicação, carregando classes de fontes distintas;
- Prover melhor segurança.

Principais Metas da Tecnologia Java

As seguintes características realizam estas metas:

- A Máquina Virtual Java (*Java Virtual Machine - JVM*)
- “Coleta de Lixo” em memória (*Garbage collection*)
- Segurança de Código

A Máquina Virtual Java

É um ambiente que:

- Fornece especificações da plataforma de hardware
- Interpreta **bytecodes** que são independentes de plataforma
- É implementada em **software** ou **hardware**
- É implementada em uma ferramenta de desenvolvimento da tecnologia Java ou em um **navegador web**.

A Máquina Virtual Java

- A maior parte de checagem de tipos é feita em tempo de compilação.
- Implementações da **JVM** aprovadas pela **Oracle/Sun Microsystems** são capazes de executar qualquer arquivo classe da linguagem.

Garbage Collection – Coleta de Lixo

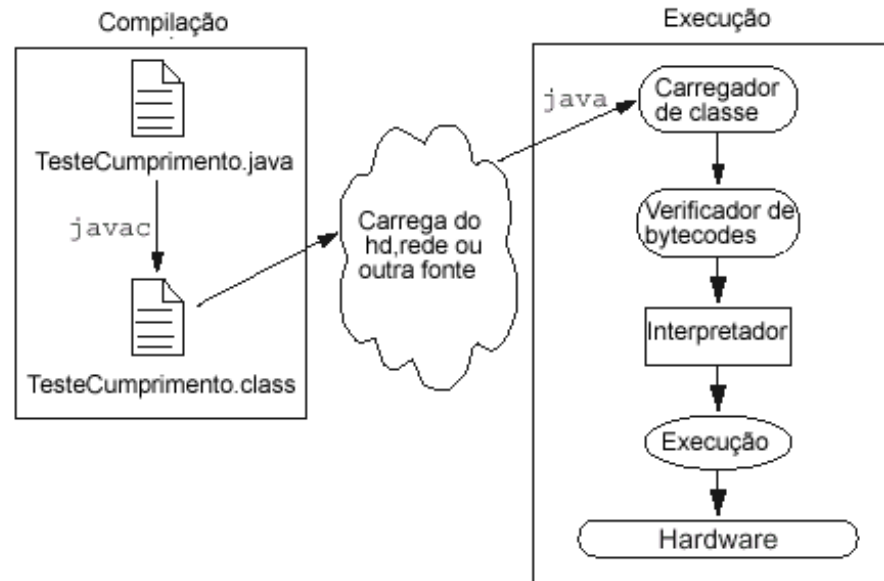
- Memória alocada não mais necessária deve ser desalocada.
- Em outras linguagens desalocar área de memória é responsabilidade do programador
- A linguagem de programação Java fornece uma *thread* em nível de sistema para guardar alocação da memória.

Garbage Collection

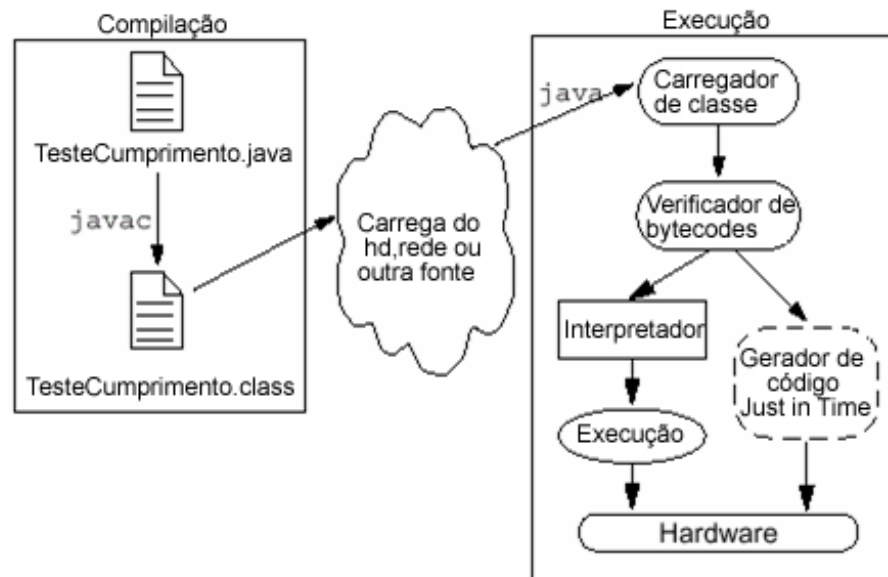
- **Garbage collection:**
 - Verifica e libera memória que não está sendo utilizada.
 - É feita automaticamente
 - Pode variar dramaticamente entre implementações distintas da **JVM**.

Segurança de Código

O ambiente de programação Java executa da seguinte maneira:



Gerador de Código Just-In-Time (JIT)



O Ambiente de Execução Java

Executa três tarefas principais:

- Carrega,
- Verifica,
- Executa o código.

O Carregador de Classe (Class Loader)

- Carrega todas as classes necessárias à execução de um programa,
- Mantém classes do sistema e de arquivos locais em “**namespaces**” distintos.
- Previne *spoofing*.

O Verificador de Bytecodes

Assegura que:

- O código está de acordo com a especificação da **JVM**,
- O código não viola a integridade do sistema,
- O código não causa *underflow* ou *overflow* em pilha de operandos
- Os tipos dos parâmetros para todo o código estão corretos
- Nenhuma conversão ilegal de dados ocorreu.

Demonstração: Exemplo de Uma Aplicação em Java

TesteCumprimento.java

```
1 //
2 // Dizendo oi para todos. Numeração de linhas apenas para tornar mais fácil a referência
3 //
4 public class TesteCumprimento{
5     public static void main (String[] args) {
6         OiPessoal oi = new OiPessoal();
7         oi.cumprimentar();
8     }
9 }
```

OiPessoal.java

```
1 // Declaração da classe OiPessoal.
2 public class OiPessoal{
3     public void cumprimentar() {
4         System.out.println("oi pessoal !!");
5     }
6 }
```

Compilando e Executando o Programa TesteCumprimento

- Compilando TesteCumprimento . java
 - **javac TesteCumprimento.java**
 - OiPessoal.java é compilada automaticamente.
- Executando a aplicação
 - **java TesteCumprimento**

Regras Para Criação de Arquivos Fonte Java

- Arquivos fonte Java terminam com a extensão **.java**
- Embora possível, não é recomendada a criação de mais de uma classe em um arquivo fonte Java
 - Caso opte por ter mais de uma classe em um arquivo fonte, apenas uma poderá ter o **modificador de acesso public**.
- O nome da classe com modificador de acesso **public** deverá ser o nome do **arquivo.java**
 - Exemplo:
 - Definição da classe: **public class Teste { ... }**
 - Nome do Arquivo: **Teste.java**
- Java é **Case Sensitive !**

Localizando Erros Comuns de Compilação e Execução

Erros em Tempo de Compilação:

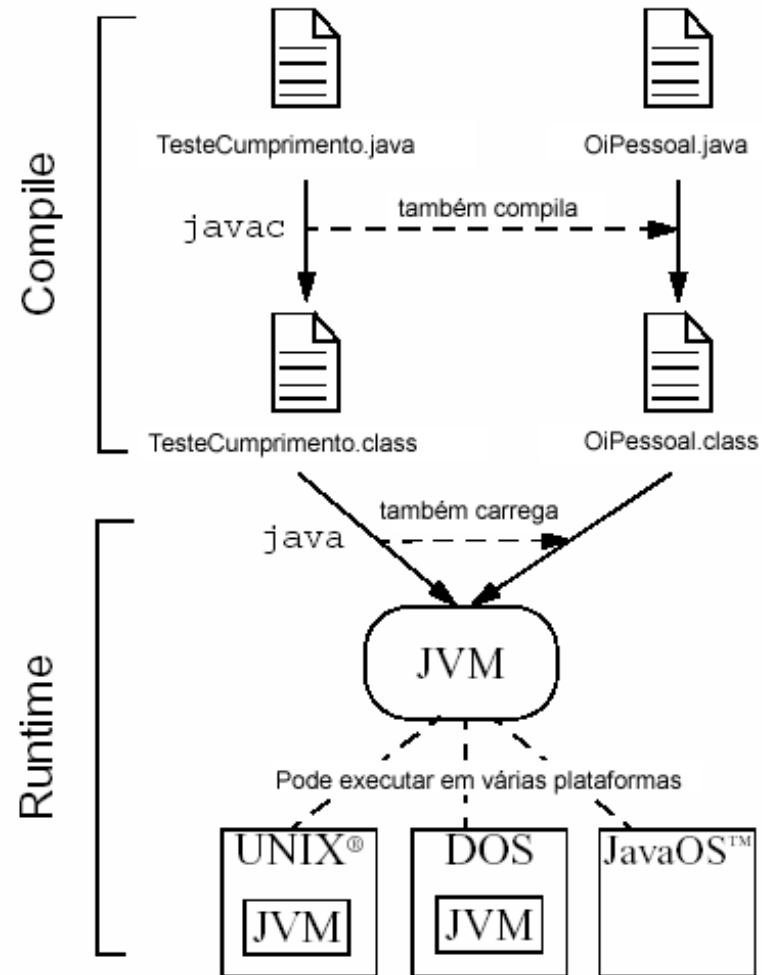
- javac: Command not found
- OiPessoal.java: 4 : cannot resolve symbol. symbol : method printl(java.lang.String) location: class java.io.PrintStream
System.out.printl("Oi Pessoal !!");
- TestaCumprimento.java: 4: Public class TesteCumprimento must Be defined in a file called "TesteCumprimento.java"

Localizando Erros Comuns de Compilação e Execução

Erros em Tempo de Execução:

- Can't find class TestaCumprimento
- Exception in thread "main" java.lang.NoSuchMethodError:main.

Ambiente de Execução Java



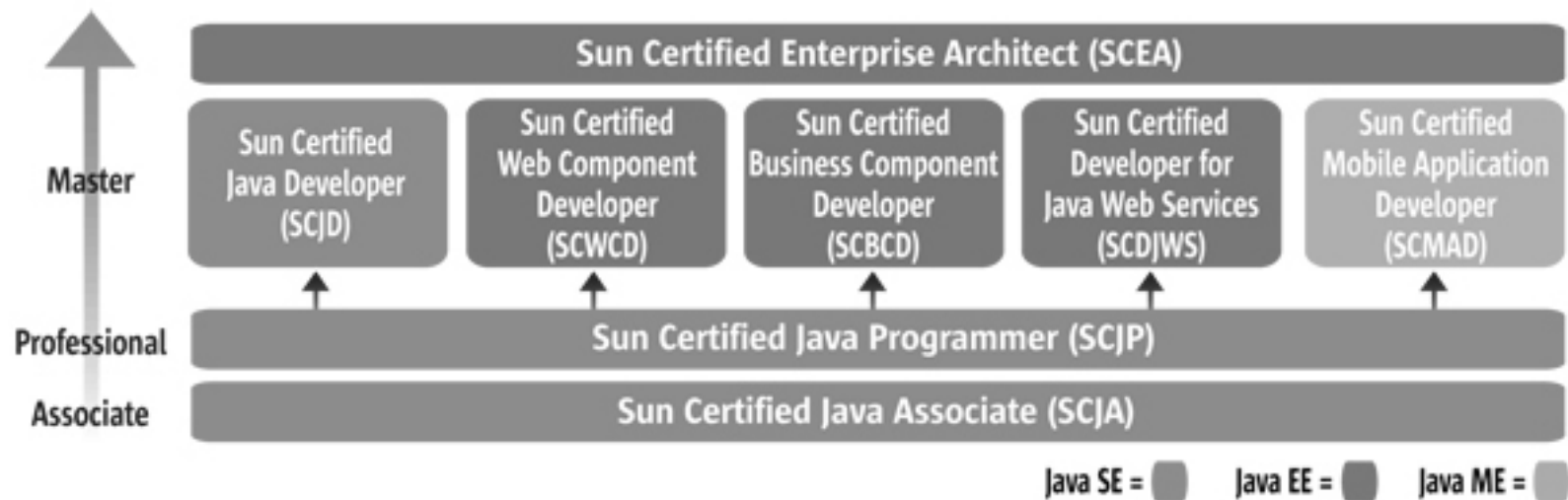
Prova de Certificação para Programador Java 5 - SCJP

- Sempre que possível, os slides deste curso estarão organizados em capítulos de forma a ficarem lado a lado com os objetivos impostos pela **Oracle/Sun**
- Todavia, em alguns casos os objetivos estarão misturados ou parcialmente repetidos em uma ordem mais adequada ao aprendizado.

Dicas de Estudo deste curso e para a prova de certificação

- Estude todos os slides, módulo por módulo.
- Execute todos os exemplos presentes no site entendendo como foram confeccionados e alterando-os para análise das conseqüências.
- Faça todos os exercícios solicitados em cada módulo
- Escreva pequenos códigos sobre um assunto em particular que você estiver estudando

Certificações Java da Oracle/Sun



Fonte: http://education.oracle.com/pls/web_prod-plq-dad/db_pages.getpage?page_id=365
Acessado em, 24/05/2010.

Certificações Java da Oracle/Sun

- **Associate Exam (cx-310-019) – SCJA**
 - Não possui pré-requisito
 - Destinado aos candidatos começando no mundo **Java**
- **Exame de Programador Java 6 (cx-310-065) – SCJP**
 - Não possui pré-requisito
 - Desenvolvido para testar seu conhecimento da linguagem de programação **Java**.
 - Requer conhecimento detalhado da sintaxe da linguagem, conceitos principais e diversas **APIs**

Certificações Java da Oracle/Sun

- **Exame de Desenvolvedor (cx-310-252A, CX-310-027) – SCJD**
 - Continuação do Exame de Programador
 - Pré-requisito: **SCJP**, qualquer versão.
 - Constituído de duas partes: um projeto a ser desenvolvido em casa e uma prova

- **Exame de Desenvolvedor de Componentes Web (cx-310-083) – SCWCD**
 - Pré-requisito: **SCJP**
 - Para desenvolvedores que utilizam as tecnologias **Servlets** e **JSP** para o desenvolvimento de aplicações **web**

Certificações Java da Oracle/Sun

- **Exame de Desenvolvedor de Componentes de Negócio (cx-310-091) – SCBCD**
 - Pré-requisito: **SCJP**
 - Para desenvolvedores que utilizam a **tecnologia EJB** para a construção de aplicações na camada de negócios.
- **Exame de Desenvolvedor de Web Services (cx-310-230) – SCDJWS**
 - Pré-requisito: **SCJP**
 - Para desenvolvedores que utilizam tecnologias de **web services**.

Certificações Java da Oracle/Sun

- **Exame de Arquiteto (cx-310-052, cx-310-301A, CX-310-062) – SCEA**
 - Não possui pré-requisito
 - Constituído de três partes: prova de múltipla-escolha, um projeto arquitetural, e uma prova final
- **Exame para Dispositivos Móveis (cx-310-110) – SCMAD**
 - Pré-requisito: **SCJP**
 - Para desenvolvedores que utilizam a tecnologia **Java ME** para desenvolvimento de aplicativos para dispositivos móveis.

Realizando o Exame SCJP

- Os candidatos são avaliados em um centro **Prometric** através de uma avaliação baseada em computador.
- A prova **SCJP 5.0** possui 72 questões com duração de 175 minutos
 - É necessária nota mínima de 43 (59%) para aprovação.
- No final da prova a nota é exibida com sua aprovação ou reprovação

Formato de Questões - SCJP

- **Questões de Múltipla Escolha**

Quais são declarações válidas?

- A. `short x [];`
- B. `short [] y;`
- C. `short[5] x2;`
- D. `short z2 [5];`
- E. `short [] z [] [];`
- F. `short [] y2 = [5];`

Resposta: alternativas a, b, e

Formato de Questões - SCJP

• Questões de Arrastar e Soltar

Usando os fragmentos abaixo, complete o seguinte código de forma a compilar. Repare que você pode não ter que preencher todas as lacunas

```
class AgedP {  
    _____  
    public AgedP(int x) {  
        _____  
    }  
}  
  
public class Kinder extends AgedP {  
    _____  
    public Kinder (int x) {  
        _____();  
    }  
}
```

Fragmentos : Use os fragmentos a seguir zero ou mais vezes

AgedP	super	this	
()	{	}
;			

Instalação e Testes do Java SE e NetBeans

- Instalando o Java SE 6 no GNU/Linux e Windows
 - Testando a instalação
 - Primeiro aplicativo na “mão”.
- Instalando o IDE NetBeans 6.8 no GNU/Linux e Windows
 - Visão geral do Netbeans
 - Primeiro aplicativo

A demonstração das instalações está disponível através de vídeo-aulas criadas para os alunos do curso.

Bibliografia Recomendada

HORSTMANN, C. **Core Java 2 vol 1: Fundamentos**. Makron Books.



Bibliografia Recomendada

SINTES, A . **Aprenda Programação Orientada A Objetos em 21 dias** . Campus



Bibliografia Recomendada

DEITEL, H. M.; Deitel, P. J. **Java: Como Programar**. Bookman.



Bibliografia Recomendada

SANTOS, R. **Introdução à Programação Orientada à Objetos** . Campus.



Bibliografia Recomendada

HORSTMANN, C. Horstmann. **Big Java**. Bookman .



Bibliografia Recomendada

SIERRA, K. BATES B. “Certificação Sun para Programador Java 5”



Para Saber Mais

Revistas

- Revista Mundo Java. <http://www.mundojava.com.br>
- Revista JavaMagazine. <http://www.devmedia.com.br>

Artigos

- DOEDERLEIN, O. P. “Eclipse x NetBeans. Os Grandes IDEs Open Source Comparados.” Revista Java Magazine, ed. 53

Perguntas?