

Certificação Sun Java Associado
SCJA

Exame CX-310-019

*Guia de viagem para
passar no exame*

Marcelo Castellani

Empresa filiada à Câmara
Brasileira do Livro



Certificação Sun Java Associado

SCJA

Exame CX-310-019

*Guia de viagem para
passar no exame*

Marcelo Castellani



Rio de Janeiro . 2008

Certificação Sun Java Associado SCJA - Guia de viagem para passar no exame

Copyright © 2008 da Editora Alta Books Ltda.

Todos os direitos reservados e protegidos pela Lei 5988 de 14/12/73. Nenhuma parte deste livro, sem autorização prévia por escrito da editora, poderá ser reproduzida ou transmitida sejam quais forem os meios empregados: eletrônico, mecânico, fotográfico, gravação ou quaisquer outros. Todo o esforço foi feito para fornecer a mais completa e adequada informação, contudo a editora e o(s) autor(es) não assumem responsabilidade pelos resultados e usos da informação fornecida. Recomendamos aos leitores testar a informação, bem como tomar todos os cuidados necessários (como o backup), antes da efetiva utilização. Este livro não contém CD-ROM, disquete ou qualquer outra mídia.

Erratas e atualizações: Sempre nos esforçamos para entregar a você, leitor, um livro livre de erros técnicos ou de conteúdo; porém, nem sempre isso é conseguido, seja por motivo de alteração de software, interpretação ou mesmo quando alguns deslizes constam na versão original de alguns livros que traduzimos. Sendo assim, criamos em nosso site, www.altabooks.com.br, a seção Erratas, onde relataremos, com a devida correção, qualquer erro encontrado em nossos livros.

Avisos e Renúncia de Direitos: Este livro é vendido como está, sem garantia de qualquer tipo, seja expressa ou implícita.

Marcas Registradas: Todos os termos mencionados e reconhecidos como Marca Registrada e/ou comercial são de responsabilidade de seus proprietários. A Editora informa não estar associada a nenhum produto e/ou fornecedor apresentado no livro. No decorrer da obra, imagens, nomes de produtos e fabricantes podem ter sido utilizados, e desde já a Editora informa que o uso é apenas ilustrativo e/ou educativo, não visando ao lucro, favorecimento ou desmerecimento do produto/fabricante.

Produção Editorial: Editora Alta Books

Coordenação Editorial: Fernanda Silveira

Revisão: Fernando Macena

Diagramação e Capa: Fernanda Silveira

Impresso no Brasil

O código de propriedade intelectual de 1º de Julho de 1992 proíbe expressamente o uso coletivo sem autorização dos detentores do direito autoral da obra, bem como a cópia ilegal do original. Esta prática generalizada nos estabelecimentos de ensino, provoca uma brutal baixa nas vendas dos livros a ponto de impossibilitar os autores de criarem novas obras.



Rua Viúva Cláudio, 291 – Jacaré
Rio de Janeiro – RJ. CEP: 20970-031
Tel: 21 3278-8069/ Fax: 3277-1253
site: www.altabooks.com.br
e-mail: altabooks@altabooks.com.br

Dedicatória e agradecimentos

Este livro é dedicado a todos que trabalham com desenvolvimento de software, seja qual for seu nível de envolvimento. Sei o trabalho que dá essa profissão, as dores de cabeça, as horas de sono mal dormidas e tudo o que nos aguarda, mas posso garantir que vale a pena.

Dedico principalmente aos meus pais, Nelson e Adélia, os quais estão ao meu lado em todas as horas: fáceis, difíceis, seja qual for, estão presentes. Foi com vocês que aprendi a ler, escrever, amar, sorrir e viver, e sem vocês isto não seria real ou não valeria a pena.

Dedico especialmente à Giuliana Cirelli.

Dedico aos meus irmãos, Renata e Ricardo, por todos os momentos nesses mais de vinte anos juntos. Os lobos observam os cordeiros.

Dedico também à toda minha família: Alana, Breno, Isabella e Adilson; Jaqueline; Olinda; Jailde; Ana Luiza, Adilson, Juliana, Mariana e Alexandre; Sandra, Siqueira e Sheila; Filomena, Rodrigo e Priscilla; Vilma, Romeu e Larissa; Francisco e Valéria; Cido, Flávio e Henrique; Cláudio e Maria; Débora e Ricardo; Luis Cláudio; Aldo, Regina e Mariana; Gabriel e esposa; Juliana e esposo.

E à minha nova família, Ana Beatriz, Francisco, Mithi e Renata.

Dedicado à memória de meus avôs, Aldo e Anacleto, meu tio Deginal e minhas tias Altair e Verônica.

Aos amigos, todos, principalmente ao meu irmão Moacir (in memoriam), que não teve tempo de realizar o que queria, mas o suficiente para deixar saudades.

A todos na editora Alta Books, pela confiança, respeito e dedicação na execução deste trabalho. Que seja o primeiro de vários.

E, como não poderia deixar de ser, a David Bowie, que, com sua música, torna meus dias mais felizes.

Sobre o Autor

Marcelo Fontes Castellani é Analista de Sistemas Sênior da Itaútec S. A., uma das maiores empresas de tecnologia do Brasil presente nos cinco continentes do planeta. Seu envolvimento com Java começou na versão 1.1 da linguagem, há muito tempo atrás (tanto tempo que nem vale a pena contar).

Trabalha com desenvolvimento de sistemas desde 1995, passando pelas mais diversas linguagens e plataformas. Do Pascal ao Visual Basic e Visual C++ (linguagens em que conquistou o selo da Microsoft Certified Professional) e depois pelo Java (e agora pelo Ruby e o Groovy, mas isso é assunto para outro material).

A idéia de escrever o livro surgiu quando se preparava para prestar a prova de SCJA, após passar horas e horas procurando material na Internet e não achando quase nada em português. Tudo para facilitar a vida do leitor.

Se quiser entrar em contato com o autor vá em frente. Ele usa desde sempre o mfcastellani@gmail.com. Só não envie spam, correntes, piadinhas e outros tipos de mensagens sem graça, que elas vão direto e sem dó para o lixo.

Sumário

Introdução -	XVII
O que é a certificação SCJA -	XVII
As certificações -	XVIII
Sobre a prova -	XIX
Ferramentas necessárias -	XX
Exam Objectives -	XX
 Capítulo I - Conceitos Fundamentais de Orientação a Objeto -	 I
Primitivos -	1
Tipos Inteiros -	1
Tipos de ponto flutuante -	3
O tipo boolean -	3
O tipo char -	3
Enumerações -	5
Objetos -	7
O que é uma classe -	8
Como nomear uma classe -	9
Programação orientada por interfaces -	13

Capítulo 2 - Representação em UML dos Conceitos de Orientação a Objetos - 17

O que é UML, afinal de contas? -	17
Classes -	17
Interfaces e classes abstratas -	18
Classes que implementam interfaces -	18
Objetos -	20
Modificadores de visibilidade -	20
Herança -	20
Composição -	21
Associações -	23

Capítulo 3 - Implementações em Java dos Conceitos de Orientação a Objetos - 25

O heap -	26
Usando a palavra reservada new para criar o objeto -	27
Declarando classes concretas -	28
Declarando classes abstratas e as estendendo -	30
Declarando e implementando interfaces -	31
Sobrescrevendo métodos -	32
Associação simples entre classes -	34
Arrays -	36
Polimorfismo -	40
Programação por interfaces -	40

Capítulo 4 - Implementação e Desenho de Algoritmos - 41

O que são variáveis? -	41
Nomeando variáveis -	41
Escopo de variáveis -	43
Condicionais -	46
Interação e controle de fluxo -	50
Operadores de atribuição -	58
Operadores aritméticos -	59
Operadores relacionais -	62
Operadores lógicos -	64
Comparando objetos e primitivos -	66
E o resultado continuaria como: -	68
Operador de concatenação -	69
Métodos de um objeto do tipo String -	70

Capítulo 5 - Fundamentos do Desenvolvimento em Java - 75

O que são pacotes? -	75
Como nomear um pacote? -	76
Exemplificando com um código fonte Java real -	76
Java -	80
Javac -	85
O pacote java.awt e o pacote javax.swing -	88
O pacote java.io -	89
O pacote java.net -	89
O pacote java.util -	90
Recursos Adicionais -	92

**Capítulo 6 - Plataforma Java
e suas Tecnologias - 93**

Um pouco de história -	93
J2SE -	94
J2EE -	95
J2ME -	95
JDBC -	100
SQL -	100
RDBMS -	102
JNDI -	102
Messaging e JMS -	103

Capítulo 7 - Tecnologias do Cliente - 111

Capítulo 8 - Tecnologias do Servidor - 119

JSP -	119
Adicionando ao servidor -	126
Adicionando código -	128
Servlets -	130
Servlets x CGI -	130
O básico da arquitetura servlet -	130
HTTP -	131
Criando um Servlet em nosso projeto JSP -	131
JMS -	133
JNDI -	134
SMTP e JavaMail -	134
JAX-RPC -	134
Web Services -	135

Capítulo 9 - Questões e Respostas -	137
--	------------

Capítulo 10 - Pra Onde ir daqui? -	149
---	------------

Outras referências -	149
----------------------------	-----

Simulados -	149
-------------------	-----

Estou pronto, e agora? -	152
--------------------------------	-----

Prefácio

Sobre este livro

O objetivo deste livro é ser um guia para aqueles que desejam obter a certificação SCJA, detalhada abaixo. Não visa mostrar detalhes da linguagem de programação Java, até por que existe uma infinidade de livros e cursos sobre o assunto, e em muitos casos considerarei que você já conhece a mesma, ainda que superficialmente.

Pelo contrário, o apresentado aqui é apenas o suficiente para que você tenha uma noção do que é necessário para fazer a prova SCJA. Desta forma, o livro foi organizado em oito capítulos e diversos sub-capítulos, cobrindo todos os requisitos da prova apresentados no site da Sun, na ordem em que são apresentados.

As questões apresentadas no final do livro foram retiradas de diversos “brain dumps” disponíveis na Internet, que com certeza é a melhor fonte de informação para todos os assuntos relacionados à Java. Uma visita ao Google também irá lhe ajudar bastante.

Pessoalmente, recomendo que estude, leia, faça simulados e encare a prova sem medo. Certificação é estudo, mérito por um trabalho bem realizado. Não é sorte e nem chute.

Bom trabalho.

Introdução

O que é a certificação SCJA

“Para quem não conhece nada, tudo parece possível.”
Franz Kafka – em “O Castelo”

A certificação SCJA é o ponto de partida para uma carreira de certificações na plataforma Java. Apesar de ser a primeira certificação a se obter, o SCJA é a mais recente do pacote de certificações na plataforma Java, criada apenas em 2005.

Seu conteúdo abrange, de maneira superficial, a tecnologia Java em si, baseando-se principalmente em conceitos de orientação a objeto, às características do Java e linguagem UML. Dessa forma, este livro irá passar muitas vezes de maneira rápida por um determinado assunto, pois o objetivo aqui é cobrir o que a prova pede.

A figura abaixo mostra o caminho do aprendizado para chegar ao nível mais alto de certificação em Java, o SCEA, ou Sun Certified Enterprise Architect:

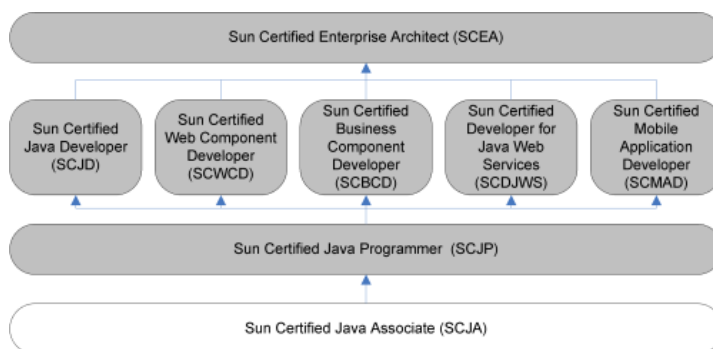


Figura 1 – o caminho do aprendizado

As certificações

Para se tornar um profissional completo na plataforma Java, primeiro você deverá selecionar qual caminho deseja seguir. Todos eles passam pelo SCJA e SCJP, que são as bases das outras certificações.

Para seguir em frente com o J2SE, você deverá, após o SCJP, obter a certificação SCJD:



Figura 2 – certificando-se em J2SE

Para seguir em frente com o J2EE, o trabalho é um pouco maior. Além da SCJA e SCJP, você deverá passar pela SCWCD, pela SCBCD, pela SCDJWS e depois pela SCEA:

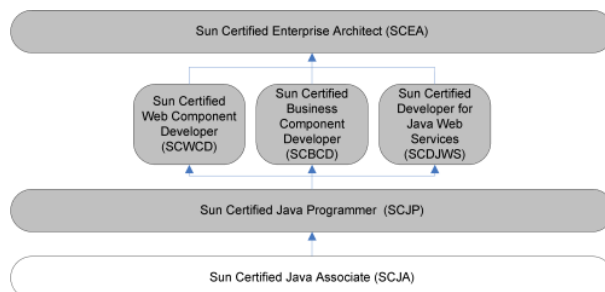


Figura 3 – certificando-se em J2EE

Para certificar-se em J2ME, a plataforma portátil da Sun, você deverá certificar-se em SCJA e SCJP, e depois em SCMD:



Figura 4 – certificando-se em J2ME

Cada uma dessas certificações abrange um nível de conhecimento, como pode ser visto abaixo:

- ❑ SCJA: é o ponto de entrada para uma carreira no desenvolvimento de aplicações ou gerenciamento de projetos de software utilizando a tecnologia Java;
- ❑ SCJP: é a certificação para programadores Java;
- ❑ SCJD: é uma certificação para programadores Java avançados;
- ❑ SCWCD: é a certificação para programadores especializados no desenvolvimento de aplicações com tecnologias JSP e Servlets, usadas para exibir conteúdo dinâmico na web;
- ❑ SCBCD: é a certificação para programadores responsáveis por especificar, desenvolver, testar e distribuir aplicações baseadas na tecnologia Enterprise JavaBeans (EJB);
- ❑ SCDJWS: é a certificação para programadores responsáveis por criar web services baseados em tecnologia Java;
- ❑ SCMD: é a certificação para programadores responsáveis por criar aplicações para dispositivos móveis e celulares usando tecnologia Java;
- ❑ SCEA: é a certificação para arquitetos de software responsáveis por definir e especificar aplicações J2EE.

Sobre a prova

A prova possui 51 questões que devem ser respondidas em 115 minutos. Sua identificação é CX-310-019 e seu nome é “Sun Certified Associate for the Java Platform, Standard Edition”. Dessas 51 questões você deve acertar 35, ou seja, cerca de 68%.

A prova custa US\$ 100,00, mas no Brasil o seu preço é um pouco mais “salgado”. Eu paguei R\$ 330,00 pela mesma e demorou um certo tempo para o boleto e depois para o voucher chegarem em casa, e depois mais um tempo para conseguir marcar a prova. Então o primeiro passo que eu lhe recomendo dar é comprar o seu voucher para fazer a prova, pois a espera é grande e pode ser feita enquanto você estuda.

Isto deve ser feito entrando em contato com o Sun Learning Services, no telefone 0800-55-78-63.

Os tópicos abordados na prova, de maneira superficial, são:

- ❑ Conceitos Fundamentais de Orientação a Objetos,
- ❑ Representação UML de Conceitos Orientação a Objetos,
- ❑ Implementação Java de Conceitos Orientação a Objetos,
- ❑ Implementação e Design de Algoritmos usando a Linguagem Java,
- ❑ Conceitos Fundamentais de Desenvolvimento Java,
- ❑ Plataforma Java e Tecnologias de Integração,
- ❑ Tecnologias Cliente,
- ❑ Tecnologias Servidor.

Esses tópicos serão cobertos a seguir, em oito capítulos dedicados a cada objetivo do exame, além de

XX Certificação SCJA - Guia de Viagem

um capítulo apenas com perguntas e respostas comentadas.

Alguns dos objetivos da prova são cobertos em capítulos relacionados, então em alguns casos você verá textos como “ver capítulo xx, objetivo xx”. Isso ocorre porque os objetivos muitas vezes são complementares e, para não deixar as coisas sem sentido e dividir um conteúdo em duas partes desconexas, os textos foram escritos de maneira a esgotar o conteúdo.

Ferramentas necessárias

Para os exemplos deste livro é recomendado possuir instalado o Eclipse e o Tomcat. Você pode obter os mesmos gratuitamente na Web e os instalar seguindo o tutorial no link abaixo:

<http://www.ibm.com/developerworks/opensource/library/os-ectom/>

Exam Objectives

1. Section 1: Fundamental Object-Oriented Concepts

- 1.1. Describe, compare, and contrast primitives (integer, floating point, boolean, and character), enumeration types, and objects.
- 1.2. Describe, compare, and contrast concrete classes, abstract classes, and interfaces, and how inheritance applies to them.
- 1.3. Describe, compare, and contrast class compositions, and associations (including multiplicity: (one-to-one, one-to-many, and many-to-many), and association navigation.
- 1.4. Describe information hiding (using private attributes and methods), encapsulation, and exposing object functionality using public methods; and describe the JavaBeans conventions for setter and getter methods.
- 1.5. Describe polymorphism as it applies to classes and interfaces, and describe and apply the “program to an interface” principle.

2. Section 2: UML Representation of Object-Oriented Concepts

- 2.1. Recognize the UML representation of classes, (including attributes and operations, abstract classes, and interfaces), the UML representation of inheritance (both implementation and interface), and the UML representation of class member visibility modifiers (-/private and +/public).
- 2.2. Recognize the UML representation of class associations, compositions, association multiplicity indicators, and association navigation indicators.

3. Section 3: Java Implementation of Object-Oriented Concepts

- 3.1. Notes: code examples may use the ‘new’ operator.
- 3.2. Develop code that uses primitives, enumeration types, and object references, and recognize literals of these types.
- 3.3. Develop code that declares concrete classes, abstract classes, and interfaces, code that supports implementation and interface inheritance, code that declares instance attributes and methods, and code that uses the Java access modifiers: private and public.
- 3.4. Develop code that implements simple class associations, code that implements multiplicity using arrays, and recognize code that implements compositions as opposed to simple associations, and code that correctly implements association navigation.
- 3.5. Develop code that uses polymorphism for both classes and interfaces, and recognize code

that uses the “program to an interface” principle.

4. Section 4: Algorithm Design and Implementation

4.1. Describe, compare, and contrast these three fundamental types of statements: assignment, conditional, and iteration, and given a description of an algorithm, select the appropriate type of statement to design the algorithm.

4.2. Given an algorithm as pseudo-code, determine the correct scope for a variable used in the algorithm, and develop code to declare variables in any of the following scopes: instance variable, method parameter, and local variable.

4.3. Given an algorithm as pseudo-code, develop method code that implements the algorithm using conditional statements (if and switch), iteration statements (for, foreach, while, and do-while), assignment statements, and break and continue statements to control the flow within switch and iteration statements.

4.4. Given an algorithm with multiple inputs and an output, develop method code that implements the algorithm using method parameters, a return type, and the return statement, and recognize the effects when object references and primitives are passed into methods that modify them.

4.5. Given an algorithm as pseudo-code, develop code that correctly applies the appropriate operators including assignment operators (limited to: =, +=, -=), arithmetic operators (limited to: +, -, *, /, %, ++, --), relational operators (limited to: <, <=, >, >=, ==, !=), logical operators (limited to: !, &&, ||) to produce a desired result. Also, write code that determines the equality of two objects or two primitives.

4.6. Develop code that uses the concatenation operator (+), and the following methods from class String: charAt, indexOf, trim, substring, replace, length, startsWith, and endsWith.

5. Section 5: Java Development Fundamentals

5.1. Describe the purpose of packages in the Java language, and recognize the proper use of import and package statements.

5.2. Demonstrate the proper use of the “javac” command (including the command-line options: -d and -classpath), and demonstrate the proper use of the “java” command (including the command-line options: -classpath, -D and -version).

5.3. Describe the purpose and types of classes for the following Java packages: java.awt, javax.swing, java.io, java.net, java.util.

6. Section 6: Java Platforms and Integration Technologies

6.1. Distinguish the basic characteristics of the three Java platforms: J2SE, J2ME, and J2EE, and given a high-level architectural goal, select the appropriate Java platform or platforms.

6.2. Describe at a high level the benefits and basic characteristics of RMI.

6.3. Describe at a high level the benefits and basic characteristics of JDBC, SQL, and RDBMS technologies.

6.4. Describe at a high level the benefits and basic characteristics of JNDI, messaging, and JMS technologies.

7. Section 7: Client Technologies

7.1. Describe at a high level the basic characteristics, benefits and drawbacks of creating thin-clients using HTML and JavaScript and the related deployment issues and solutions.

7.2. Describe at a high level the basic characteristics, benefits, drawbacks, and deployment issues

XXII Certificação SCJA - Guia de Viagem

related to creating clients using J2ME midlets.

7.3. Describe at a high level the basic characteristics, benefits, drawbacks, and deployment issues related to creating fat-clients using Applets.

7.4. Describe at a high level the basic characteristics, benefits, drawbacks, and deployment issues related to creating fat-clients using Swing.

8. Section 8: Server Technologies

8.1. Describe at a high level the basic characteristics of: EJB, servlets, JSP, JMS, JNDI, SMTP, JAX-RPC, Web Services (including SOAP, UDDI, WSDL, and XML), and JavaMail.

8.2. Describe at a high level the basic characteristics of servlet and JSP support for

8.3. HTML thin-clients.

8.4. Describe at a high level the use and basic characteristics of EJB session, entity and message-driven beans.

8.5. Describe at a high level the fundamental benefits and drawbacks of using J2EE server-side technologies, and describe and compare the basic characteristics of the web-tier, business-tier, and EIS tier.