



**Caelum**  
Ensino e Inovação



**FJ-19**

Preparatório para a  
Certificação de Programador Java

[www.caelum.com.br](http://www.caelum.com.br)

design by aef.com.br

**EF-5**

Certificado de Habilidades  
Habilidades básicas



Georges  
L'Oréal



A Caelum atua no mercado com consultoria, desenvolvimento e ensino em computação. Sua equipe participou do desenvolvimento de projetos em vários clientes e, após apresentar os cursos de verão de Java na Universidade de São Paulo, passou a oferecer treinamentos para o mercado. Toda a equipe tem uma forte presença na comunidade através de eventos, artigos em diversas revistas, participação em muitos projetos *open source* como o VRaptor e o Stella e atuação nos fóruns e listas de discussão como o GUJ.

Com uma equipe de mais de 60 profissionais altamente qualificados e de destaque do mercado, oferece treinamentos em Java, Ruby on Rails e Scrum em suas três unidades - São Paulo, Rio de Janeiro e Brasília. Mais de 8 mil alunos já buscaram qualificação nos treinamentos da Caelum tanto em suas unidades como nas próprias empresas com os cursos *incompany*.

O compromisso da Caelum é oferecer um treinamento de qualidade, com material constantemente atualizado, uma metodologia de ensino cuidadosamente desenvolvida e instrutores capacitados tecnicamente e didaticamente. E oferecer ainda serviços de consultoria ágil, mentoring e desenvolvimento de projetos sob medida para empresas.

## Comunidade



Nossa equipe escreve constantemente artigos no **Blog da Caelum** que já conta com 150 artigos sobre vários assuntos de Java, Rails e computação em geral. Visite-nos e assine nosso RSS:

→ [blog.caelum.com.br](http://blog.caelum.com.br)



Acompanhe também a equipe Caelum no **Twitter**:

→ [twitter.com/caelumdev/equipe](http://twitter.com/caelumdev/equipe)



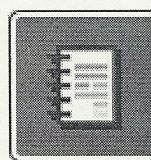
O **GUJ** é maior fórum de Java em língua portuguesa, com 700 mil posts e 70 mil usuários. As pessoas da Caelum participam ativamente, participe também:

→ [www.guj.com.br](http://www.guj.com.br)



Assine também nossa **Newsletter** para receber as novidades e destaques dos eventos, artigos e promoções da Caelum:

→ [www.caelum.com.br/newsletter](http://www.caelum.com.br/newsletter)



No site da Caelum há algumas de nossas **Apostilas** disponíveis gratuitamente para download e alguns dos **artigos** de destaque que escrevemos:

→ [www.caelum.com.br/apostilas](http://www.caelum.com.br/apostilas)

→ [www.caelum.com.br/artigos](http://www.caelum.com.br/artigos)



сторе, как подтверждение для выдачи вами прописки. Требование, что в паспорте прописано место жительства, не является обязательным условием, но оно может быть полезным для проверки достоверности информации о месте проживания.

Важную информацию об апостоле от вашего бывшего гражданства вы можете найти в паспорте ОБСР. В нем есть даже паспортные данные, которые можно использовать для проверки гражданства. Для этого вам потребуется паспорт и ксерокопия его страницы с фотографией и подписью.

Помимо этого, вы можете использовать паспорт ОБСР для проверки гражданства в других странах. Для этого вам потребуется паспорт и ксерокопия его страницы с фотографией и подписью.

## Задачи

Ваша задача – это проверка гражданства ОБСР на основании предоставленных вами документов. Для этого вам потребуется паспорт ОБСР и ксерокопия его страницы с фотографией и подписью.

Задача №1: Проверка гражданства ОБСР на основании предоставленных вами документов.

Задача №2: Проверка гражданства ОБСР на основании предоставленных вами документов.

Задача №3: Проверка гражданства ОБСР на основании предоставленных вами документов.

Задача №4: Проверка гражданства ОБСР на основании предоставленных вами документов.

---

# Índice

<b>1 A certificação de programador Java</b>	<b>1</b>
1.1 O que é a prova de certificação de programador Java . . . . .	1
1.2 Por que fazer a SCJP . . . . .	2
1.3 Como se preparar? . . . . .	3
1.4 Links . . . . .	3
1.5 Tirando dúvidas . . . . .	4
1.6 Bibliografia . . . . .	4
1.7 Sobre o treinamento . . . . .	4
1.8 Para saber mais: Outras certificações . . . . .	5
<b>2 Fundamentos - Parte 1</b>	<b>7</b>
2.1 Código Fonte . . . . .	7
2.2 Variáveis . . . . .	9
2.3 Operador de atribuição . . . . .	13
2.4 Operadores aritméticos . . . . .	13
2.5 Comparadores . . . . .	15
2.6 Operadores lógicos . . . . .	15
2.7 Aritmética com Atribuição . . . . .	16
2.8 Operador ternário - Condicional . . . . .	17
2.9 Operador de referência . . . . .	17
2.10 Concatenação de strings . . . . .	18
2.11 Detalhes: Precedência . . . . .	18
2.12 Pontos importantes . . . . .	18
2.13 Casting de tipos primitivos . . . . .	19

2.14 Array . . . . .	20
2.15 Exercícios . . . . .	24
2.16 Testes . . . . .	25
<b>3 Fundamentos - Parte 2</b>	<b>27</b>
3.1 Compilação e Execução . . . . .	27
3.2 O método main . . . . .	29
3.3 Classpath . . . . .	30
3.4 Regras de Compilação e Execução . . . . .	32
3.5 Pacotes e Imports . . . . .	33
3.6 Controle de Acesso . . . . .	36
3.7 Passagem de parâmetro, Pilha de Execução e Heap . . . . .	38
3.8 Unreachable Code e Missing return . . . . .	40
3.9 Testes . . . . .	41
<b>4 Controle de Fluxo</b>	<b>45</b>
4.1 Controlar o fluxo de execução . . . . .	45
4.2 if / else . . . . .	45
4.3 switch . . . . .	46
4.4 while e do...while . . . . .	48
4.5 for . . . . .	51
4.6 Rótulos em laços (labeled loops) . . . . .	52
4.7 Exceptions . . . . .	53
4.8 Exceptions para a prova . . . . .	60
4.9 Assertion . . . . .	63
4.10 Exercícios . . . . .	65
4.11 Testes . . . . .	65
<b>5 Classes e interfaces</b>	<b>69</b>
5.1 Classes . . . . .	69
5.2 Atributos . . . . .	70
5.3 Métodos . . . . .	73
5.4 Static . . . . .	74
5.5 Abstract . . . . .	75
5.6 Parâmetros . . . . .	76

5.7	Sobrecarga (overload) . . . . .	77
5.8	Herança e reescrita . . . . .	78
5.9	Construtores . . . . .	83
5.10	Blocos de inicialização . . . . .	86
5.11	Interfaces . . . . .	88
5.12	Convenções de nomenclatura . . . . .	89
5.13	Exercícios . . . . .	89
5.14	Testes . . . . .	91
<b>6</b>	<b>Polimorfismo e Casting</b>	<b>97</b>
6.1	Polimorfismo . . . . .	97
6.2	Casting de referências . . . . .	99
6.3	instanceof . . . . .	102
6.4	Exercícios . . . . .	102
6.5	Testes . . . . .	104
<b>7</b>	<b>Enums e classes aninhadas</b>	<b>105</b>
7.1	Enums . . . . .	105
7.2	Classes aninhadas . . . . .	109
7.3	Exercícios . . . . .	113
7.4	Testes . . . . .	114
<b>8</b>	<b>Concorrência</b>	<b>117</b>
8.1	Porque usar threads . . . . .	117
8.2	Como criar thread . . . . .	117
8.3	Escalonador . . . . .	118
8.4	Ciclo de vida de threads . . . . .	119
8.5	Prioridades de threads . . . . .	120
8.6	Yield e sleep . . . . .	120
8.7	Garbage collector e finalize . . . . .	121
8.8	Deamons . . . . .	122
8.9	Problemas ao trabalhar com threads . . . . .	123
8.10	synchronized . . . . .	124
8.11	Dicas para analisar códigos com threads . . . . .	126
8.12	Exercícios . . . . .	127
8.13	Testes . . . . .	128

<b>9 Collections e Generics</b>	<b>131</b>
9.1 Conhecendo as Collections e suas interfaces . . . . .	131
9.2 Principais implementações das coleções . . . . .	132
9.3 Usando as coleções e o básico do generics . . . . .	133
9.4 Iterator e o enhanced for . . . . .	133
9.5 equals . . . . .	133
9.6 hashCode . . . . .	134
9.7 java.util.Collections e java.util.Arrays . . . . .	136
9.8 Comparable e Comparator . . . . .	137
9.9 Criando classes com generics . . . . .	138
9.10 Generics: extends . . . . .	139
9.11 Generics: o coringa - wildcard . . . . .	139
9.12 Generics: mais de um tipo genérico . . . . .	141
9.13 Generics: Métodos genéricos . . . . .	141
9.14 Generics: super . . . . .	142
9.15 Exercícios . . . . .	143
9.16 Testes . . . . .	143
<b>10 APIs do Java e outros conceitos</b>	<b>147</b>
10.1 Visão geral das APIs cobradas na SCJP . . . . .	147
10.2 Object . . . . .	147
10.3 Classes Wrapper . . . . .	148
10.4 Autoboxing . . . . .	150
10.5 String, StringBuilder e StringBuffer . . . . .	152
10.6 Expressões regulares . . . . .	155
10.7 Date e Calendar . . . . .	160
10.8 Locale e formatação de datas e números . . . . .	162
10.9 Formatter e printf . . . . .	165
10.10 java.util.Scanner . . . . .	167
10.11 java.io . . . . .	170
10.12 Console . . . . .	173
10.13 Exercícios . . . . .	174
<b>11 Apêndice - Simulado</b>	<b>177</b>

<b>12 Apêndice - Tópicos da prova</b>	<b>191</b>
12.1 Declarations, Initialization and Scoping . . . . .	191
12.2 Flow Control . . . . .	191
12.3 API Contents . . . . .	192
12.4 Concurrency . . . . .	192
12.5 OO Concepts . . . . .	192
12.6 Collections / Generics . . . . .	193
12.7 Fundamentals . . . . .	193
<b>13 Apêndice - Tópicos antigos</b>	<b>195</b>
13.1 wait, notify e notifyAll . . . . .	195
13.2 APIs de java.io antes cobradas . . . . .	197
13.3 Serialização . . . . .	197
13.4 Argumentos variáveis: varargs . . . . .	199
13.5 JavaBeans . . . . .	200
<b>14 Respostas dos Exercícios</b>	<b>203</b>

Versão: 12.1.1

