



*Programação
orientada a
objetos*

com **JavaTM 2 SE**

- *Este curso tem como objetivo iniciá-lo em Java*
 - *... mas não apenas isto*
- *Visa também a ajudá-lo a desenvolver **boas práticas de desenvolvimento** que tornarão sua vida de programador mais fácil no futuro.*
 - *Uso eficiente da documentação,*
 - *Uso de diagramas de classe,*
 - *Estilo e convenções de codificação,*
 - *Técnicas de depuração, testes e manutenção*
 - *Uso e conhecimento de padrões de projeto*

Saber programar em C

ou

*Saber programar em uma
linguagem orientada a objetos*

Assuntos abordados

- *Este curso explora os seguintes assuntos*
 - *Como escrever programas em Java, como compilá-los e como executá-los*
 - *Conceitos essenciais de programação orientada a objetos*
 - *A sintaxe da linguagem Java*
 - *Recursos fundamentais como estruturas de dados utilitárias, manipulação de strings, leitura e gravação de bytes e caracteres*
 - *Tópicos essenciais de programação concorrente*
 - *Como utilizar a documentação*
 - *Boas práticas, testes e noções de padrões de projeto*
 - *Como montar um ambiente de desenvolvimento Java baseado no Ant e outras ferramentas de código aberto*

Assuntos abordados superficialmente

- *Os seguintes assuntos são abordados de maneira muito superficial e incompleta neste curso*
 - *Como criar aplicações gráficas usando JFC/Swing (javax.swing) ou AWT e Applets*
 - *Como desenvolver aplicações integradas a bancos de dados usando JDBC (java.sql)*
 - *Como desenvolver aplicações de rede (java.net)*
 - *Como desenvolver aplicações de objetos distribuídos (java.rmi e org.omg.CORBA)*
 - *Multithreading*
 - *Expressões regulares, reflection, class loader*
 - *Padrões de projeto (design patterns)*

Assuntos não abordados

- Os seguintes assuntos **não são** abordados neste curso
 - Programação elementar estruturada (estruturas de dados, variáveis, funções, laços de controle, compilação, pilhas, algoritmos, etc.): como foi mencionado antes, saber programar é um **pré-requisito essencial** para este curso.
 - Como desenvolver aplicações para a Web (servlets e JSP)
 - Como criar aplicações e componentes para servidores de aplicação transacionais (Enterprise JavaBeans)
 - Análise, design ou técnicas e práticas para desenvolver aplicações robustas, seguras e eficientes
 - UML (Unified Modelling Language)
 - Uso de ambientes integrados (IDEs) gráficos, debuggers, profilers, sistemas de controle de versão, etc.

- *Parte I - Introdução prática*
 - *1. Conceitos básicos e plataforma Java*
 - *2. Programação orientada a objetos*
 - *3. Como configurar e usar o ambiente*
 - *4. Como usar a documentação da API*
- *Parte II - Programação OO com Java*
 - *5. Tipos, literais, operadores e controle de fluxo*
 - *6. Como criar classes e objetos*
 - *7. Pacotes e encapsulamento*
 - *8. Gerenciamento de projetos com o Ant*

- *Parte III - Mais programação OO com Java*
 - *9. Reuso com herança e composição*
 - *10. Interfaces e polimorfismo*
 - *11. Controle de erros, exceções e asserções*
 - *12. Testes de unidade com o JUnit **
- *Parte IV - Threads, strings e I/O*
 - *13. Fundamentos de programação concorrente*
 - *14. Coleções, propriedades, resources e strings*
 - *15. Entrada e saída, logs e serialização*
 - *16. Classes internas*

* *módulo opcional*

- *Parte V - Swing e persistência de dados*
 - *17. Fundamentos de Swing e aplicações gráficas*
 - *18. Fundamentos de JDBC (java.sql)*
 - *19. Fundamentos de Sockets (java.net)*
 - *20. Fundamentos de Objetos remotos (java.rmi) **

A abordagem dos assuntos nos módulos da parte V é superficial.

Características importantes sobre este curso

- *Este curso dedica 70% do tempo à aprendizagem dos conceitos fundamentais da linguagem*
 - *Orientação a objetos*
 - *Metologias de desenvolvimento, padrões*
 - *Boas práticas, testes, roteiros, organização*
 - *Utilização da documentação*
- *Por outro lado, sobra menos tempo para tratar de APIs (são abordadas superficialmente)*
 - *Swing, sockets, I/O, JDBC, RMI, Reflection*
 - *Muitas exigiriam bem mais tempo*
- *Se seus conceitos são sólidos, aprender novas APIs será muito mais fácil*

- *Ao final deste curso você deve ser capaz de*
 - *Desenvolver aplicações simples em Java (inclusive aplicações gráficas), compilá-las e executá-las*
 - *Analisar programas maiores, identificar seus componentes e compreender seu funcionamento*
 - *Consultar a documentação da API e descobrir como usar novas classes, objetos e métodos.*
 - *Descrever os principais recursos do pacote Java 2 SE*
 - *Construir e utilizar um ambiente de desenvolvimento Java baseado em ferramentas gratuitas*
 - *Explorar assuntos mais complexos em OO e Java*
 - *Entender os assuntos requeridos para a certificação de programador Java da Sun*

Como tirar o melhor proveito deste curso

- *Faça perguntas*
- *Faça os exercícios*
- *Explore os exemplos*
- *Vá além dos exemplos e exercícios: invente exemplos similares, teste trechos de código*
- *Explore e se familiarize com a documentação*
- *Procure desenvolver um projeto que utilize Java, seja no trabalho, seja no seu tempo livre*
 - *Não fique sem programar nos próximos meses ou todo o esforço terá sido em vão!*
- *Leia revistas, artigos e livros sobre Java e mantenha-se atualizado.*

O todo e as partes

- *Este curso introduz uma nova linguagem e muitos novos conceitos*
- *Como qualquer novo conhecimento, vários de seus conceitos mais complexos dependem de outros mais simples*
- *Nem sempre é possível compreender um conceito mais abrangente na primeira vez*
 - *Ele as vezes depende do conhecimento de partes que só poderão ser abordadas mais adiante*
 - *Mas as partes, às vezes dependem dele!*
- *Solução: repetição. Assuntos complexos serão abordados superficialmente e depois revisitados mais de uma vez*
 - *Se tiver dúvidas, pergunte na hora*
 - *Cada dia haverá mais dúvidas novas e menos dúvidas antigas*

Exercícios, testes e projetos

- *Exercícios são propostos ao final de cada módulo*
 - *Incluídos na carga-horária*
 - *Geralmente aplicações triviais (para fixar conceitos)*
 - *Distribuídos separadamente*
- *Projetos e testes (opcionais)*
 - *Não incluídos na carga-horária*
 - *Projetos usando Java e as principais APIs do J2SE são propostos para quem desejar fixar os conceitos aprendidos. Alguns são aplicações que devem ser completadas. Fazer pelo menos um dos projetos (leva + ou - entre 2 e 8 horas) é fortemente recomendado*
 - *Testes (similares aos de certificação) são propostos como uma revisão dos principais conceitos abordados no curso*

Fontes suplementares

- *Este material serve apenas de roteiro de aula*
 - *Use-o como um resumo*
- *Para informações mais detalhadas, exemplos extras, testes e projetos utilize um dos livros abaixo*
 - *"Thinking in Java 2", Bruce Eckel www.bruceeckel.com (PDF - download gratuito) - **livro-texto principal***
 - *"The Java Tutorial", da Sun, por Mary Campione e Kathy Walrath. java.sun.com/tutorial - **livro-texto para exemplos com Swing***
 - *"Java: como programar", Deitel & Deitel (em português)*
 - *"Aprenda OO em 21 dias", A. Sintes (em português)*
 - *"Core Java 2", Cay Hortsman et al. (em português)*

- *Instrutor: Helder da Rocha (helder@acm.org)*
 - *Utiliza Java desde 1995*
 - *XML, J2EE, JSP, servlets, Web*
 - *<http://www.argonavis.com.br>*
- *Alunos?*
 - *Nome?*
 - *O que faz? Onde trabalha?*
 - *Background (sabe C? Java? Web? Que linguagem?)*
 - *Expectativas?*

Curso J100: Java 2 Standard Edition

Revisão 17.0

© 1996-2003, Helder da Rocha
(helder@acm.org)

 argonavis.com.br

Investindo em Java desde 1995