Java Programmer

Carga Horária: 100

Pré-requisitos

Para acompanhar o curso com facilidade é indicado que tenha feito o nosso curso básico de Lógica de Programação e SQL ou que tenha uma base de conhecimento semelhante.

Sobre o curso

A linguagem java é uma das mais utilizadas no mundo, e suas aplicações vão desde páginas de sites, jogos online minecraft é um deles, até ferramentas como OpenOffice, semelhante ao Microsoft Office, só que feito para rodar no Linux.
Se você já tem noções básicas ou programa em outra linguagem, esse curso é ideal para aprimorar suas habilidades.
Durante as aulas você aprenderá os fundamentos do Java que te qualificarão para desenvolver diversos tipos de aplicações inclusive para mobile, já que o próprio google usa a linguagem como base do sistema operacional Android!

Conteúdo Programático

Introdução à linguagem Java

Histórico.
Características.
Edições disponíveis.
Java Development Kit (JDK): Java Virtual Machine (JVM).
Ambientes de desenvolvimento (IDEs).
Estrutura básica de um programa Java.
Características do código: Case sensitive; Nomes de arquivo; Nomenclatura; Estrutura; Comentários; Palavras reservadas.
Compilando e executando um programa.
JShell (Java Interativo): Utilização básica.

Tipos de dados, literais e variáveis

Tipos de dados primitivos.
Literais: Literais inteiros; Literais de ponto flutuante; Literais booleanos; Literais de caracteres (Caracteres de escape).
Variáveis: Definindo uma variável; Declarando uma variável (Usando o qualificador final); Escopo de variáveis.
Casting.
Tipos de referência comuns: String; Enum; Classes Wrapper.

Operadores

Operador de atribuição.
Operadores aritméticos: Operadores aritméticos de atribuição reduzida.
Operadores incrementais e decrementais.
Operadores relacionais.
Operadores lógicos.
Operador ternário.
Precedência dos operadores.

Estruturas de controle

Estruturas de desvios condicionais: if / else; switch.
Estruturas de repetição: While; Do / while; For.
Outros comandos: Break (Instruções rotuladas); Continue.

Introdução à orientação a objetos

Apresentação.
Classes.

Objeto: Instanciação.
Atributos.
Tipos construídos: Atribuição entre objetos de tipos construídos; Variáveis não inicializadas; O uso do this.
Encapsulamento.
Pacotes: Criando um pacote; Acessando uma classe em outro pacote.
UML - Diagramas de casos de uso, classes e pacotes: Diagrama de casos de uso; Diagrama de classes; Diagrama de pacotes.

Métodos

Estrutura de um método.
Comando return.
Chamando um método (mensagens).
Passagem de parâmetros.
Varargs.
Métodos assessores: Método getter; Método setter.
Modificadores de métodos.
Modificador static: Atributos estáticos; Métodos estáticos; Exemplos práticos de membros estáticos.
Método main().
Sobrecarga de métodos.
UML - Diagrama de sequência.

Construtores

Construtor padrão.
Considerações sobre os construtores.

Arrays

Tipos de array: Array unidimensional; Array bidimensional; Array multidimensional.
Acessando elementos de um array: Acesso aos elementos em um for tradicional; Acesso aos elementos usando enhanced for (foreach).
Modos de inicializar e construir um array: Por meio de uma única instrução; Por meio de um array anônimo.
Passando um array como parâmetro: Variáveis de referência para arrays unidimensionais; Variáveis de referência para arrays multidimensionais.
Array de argumentos.

Herança, classes abstratas e polimorfismo

Herança e generalização.
Estabelecendo herança entre classes: Acesso aos membros da superclasse; O operador super; Chamada ao construtor da

superclasse.
Herança e classes: Classes finais; Classe Object.
Classes abstratas: Métodos abstratos.
Polimorfismo: Ligação tardia (late binding); Polimorfismo em métodos declarados na superclasse; Operador instanceof.
UML - Associações entre classes: Tipos de associação (Associação Simples, Agregação, Composição, Herança); Herança x Composição.

Interfaces

O conceito de interface: Variáveis de referência; Constantes.
Métodos em interfaces: Métodos estáticos; Métodos default; Métodos privados.

Tratamento de exceções

Bloco try/catch: Manipulando mais de um tipo de exceção.
throws.
finally.
Exceções e a pilha de métodos em Java.
Hierarquia de exceções: Exceções verificadas; Exceções não verificadas.
Principais exceções: Throwable (Exceções encadeadas); Error; Exception; NullPointerException; NumberFormatException; ArrayIndexOutOfBoundsException; ArithmeticException; ClassCastException; IOException; Classe SQLException.
Exceções personalizadas.

As bibliotecas Java e o Javadoc

Conceito de API.
Javadoc e a documentação oficial Java.
Criação de uma documentação Javadoc: Geração da página de documentação.

Testes unitários com Java

Conceito de teste unitário.
Como implantar o teste unitário.
Utilizando o JUnit: Criando um teste unitário (Ciclo de vida de um teste, Assertions).
Conclusão.

Programação funcional

Introdução à programação funcional: Vantagens da programação

funcional; Um primeiro exemplo.
Interface funcional: A anotação @FunctionalInterface; Exemplos de interface funcional.
Expressões lambda: Forma geral; Expressões com parâmetros; Expressões sem parâmetros; Expressões com um único parâmetro; Corpo da expressão lambda; Expressões com valor de retorno.
Referenciando métodos.
O pacote java.util.function.

Coleções e mapas

O que são coleções.
Principais operações de coleções.
Principais interfaces das coleções: Características das classes de implementação.
Generics: Tipos genéricos.
Coleção List.
Coleção Set: Classe Iterator; Equivalência de objetos (equals) (As regras de equals()); Hashing (As regras de hashCode()); Método forEach(); Método removeIf(); Interface Comparable; Interface Comparator.
Manipulando coleções com Streams: Método sorted(); Método filter(); Método limit(); Método skip(); Método map(); Método distinct(); Método count(); Métodos min() e max().
Interface Map: Principais métodos.
Collections Framework.

Arquivos - I/O e NIO

I/O: Classe OutputStream (Métodos); Classe InputStream (Métodos); Leitura de arquivos binários; I/O - Arquivos e diretórios (classe File).
try-with-resources: Exceções suprimidas.
Leitura de arquivos de texto: Classe FileReader; Classe BufferedReader.
NIO - Arquivos e diretórios: Visão Geral de NIO; Path, Paths e Files.

Threads

Programação multithreaded: Multitarefa baseada em processo; Multitarefa baseada em threads.
Implementando multithreading: java.lang.Thread; java.lang.Runnable.
Construtores.
Estados da thread.
Scheduler.
Prioridades das threads: Método yield(); Método join(); Método isAlive(); Método sleep().
Sincronização: Palavra-chave synchronized (Race condition); Bloco sincronizado.

Bloqueios.
Deadlock.
Interação entre threads.

Geração de Pacotes - Instalação de Aplicações Java (JAR)

Conceito de aplicações e bibliotecas.
Geração de bibliotecas e executáveis: Geração de um pacote executável; Utilização de uma biblioteca em projetos.

Banco de dados com Java - JDBC

Pacote java.sql.
Conexões com banco de dados: Estabelecendo uma conexão; Interface Connection; Classe DriverManager; Estabelecendo a conexão com o banco de dados; Método Close.
Operações na base de dados.
Operações parametrizadas.
Transações.
Consultas.

IMPACTA

Avenida Paulista, 1009
11 3254 2200

atendimento@impacta.com.br