PROGRAMA PEDAGÓGICO

INSTITUIÇÃO DE ENSINO	
RAZÃO SOCIAL:	CURSOS VIRTUAIS LTDA
NOME FANTASIA:	CURSOSVIRTUAIS.NET
CNPJ:	08.179.401/0001-62
REGISTRO ABED:	7734 - CATEGORIA INSTITUCIONAL

CURSO	
NOME:	PROGRAMAÇÃO JAVA - FUNDAMENTOS
MODALIDADE:	CAPACITAÇÃO LIVRE OFERTA - EAD

Metodologia: O conteúdo do curso é disponibilizado ao aluno para estudo online em uma interface diagramada de fácil navegação AVA (Ambiente Virtual de Estudos). O acesso ao material é bastante intuitivo e proporciona uma experiência de interatividade no processo de aprendizagem a distância. O curso conta com a realização de atividade avaliativa ao término de cada aula/módulo e também realização da prova final.

Formato: O curso é ofertado de forma assíncrona e conta com atividades complementares síncronas, permitindo que o aluno organize seus estudos conforme sua disponibilidade. Os módulos de aprendizado são liberados de maneira assíncrona e progressiva, sendo necessário concluir cada etapa para avançar à seguinte. Complementarmente, o curso conta com atividade síncrona por meio do suporte em tempo real com o professor, disponível às terças e quintas-feiras, das 15h às 16h, na ferramenta de tira-dúvidas.

Tutoria e formas de interação: Os alunos recebem suporte de uma tutoria especificamente designada. A interação é realizada por meio do da Área do Aluno, no Ambiente Virtual de Estudos (AVA). A tutoria consiste na assistência didática, compartilhamento de informações, troca de experiências visando o melhor aproveitamento dos conteúdos estudados.

Prova final/Certificação: A prova final é quantitativa. A geração do certificado é condicionada à verificação de aproveitamento mínimo de 70% (setenta por cento) na prova final. O curso conta com ferramenta de avaliação de conteúdo (aprendizagem) correspondente à carga horária certificada.

Organização curricular: O curso apresenta organização curricular elaborada a partir de projetos pedagógicos específicos por uma equipe pedagógica multidisciplinar, que acompanha toda a concepção dos conteúdos.

Tecnologia de EAD/e-learning: Após a elaboração dos conteúdos é realizada a migração para o ambiente de estudos na área do aluno, que é um AVA otimizado para nossa plataforma de ensino.

Materiais didáticos: O conteúdo programático é desenvolvido com base em apostilas e outros materiais didáticos atualizados. Além do material de estudo, o aluno conta com diferentes ferramentas de aprendizagem, como grupo de estudos com tutor/professor, tira-dúvidas por inteligência artificial, atividades avaliativas ao longo das aulas e prova final para avaliação do aprendizado.

Interação e suporte administrativo: O curso conta – além do suporte de tutoria - com uma infraestrutura de apoio que prevê a interação entre alunos e professores/tutores; e alunos e equipe de apoio administrativo. Essa interação é garantida por meios eletrônicos e/ou por meio telefônico, conforme o caso. O Ambiente Virtual de Estudos (AVA) utilizado pela CURSOS VIRTUAIS LTDA é uma

plataforma proprietária, desenvolvida e atualizada permanentemente.

Sobre a instituição de ensino: A CURSOS VIRTUAIS LTDA é uma escola de educação à distância. Iniciamos nossas atividades em 2006 e contamos com mais de 500 mil alunos matriculados em diversos cursos. Além disso, somos associados da ABED - Associação Brasileira de Educação a Distância. Legalmente constituída inscrita no CNPJ 08.179.401/0001-62, atua com a idoneidade e credibilidade auxiliando diversos órgãos públicos e empresas privadas, além de milhares de profissionais, servidores públicos, estudantes e professores de todo o país.

ESTRUTURA DO CURSO - COMPONENTES CURRICULARES

NOME DA CAPACITAÇÃO: Programação Java - Fundamentos

BASE LEGAL: CF/88, arts. 206, II, e 209; na LDB 9.394/96, arts. 39, § 2º, I, e 40; e no Decreto 5.154/2004, art. 3, em consonância com a Resolução CNE/CP 1/2021 (MEC), arts. 14 e 37, quanto à formação e aperfeiçoamento profissional.

OBJETIVO DE APRENDIZAGEM: Ao concluir o curso de Programação Java - Fundamentos, o aluno deverá ser capaz de compreender os princípios da linguagem Java e desenvolver programas utilizando tipos de dados, operadores, estruturas de controle, vetores, matrizes, classes, métodos e pacotes. Deverá aplicar conceitos de orientação a objetos, como encapsulamento, herança, composição, interfaces, sobrecarga e polimorfismo, além de identificar e tratar erros e exceções. Também deverá desenvolver conhecimentos sobre entrada e saída, coleções, interfaces gráficas, concorrência, comunicação em rede, acesso a bancos de dados com JDBC e fundamentos de aplicações web com Servlets, JSP e arquitetura MVC.

ATIVIDADES/AULAS:
1) Java básico
2) Programando com Java
3) Exceções, entrada e saída e programação Swing

CONTEÚDO PROGRAMÁTICO DETALHADO:
História rápida da linguagem
Vantagens do Java
A plataforma Java
Erros comuns de conceito sobre Java
Instalando o ambiente
Meu primeiro programa Java
OOP - Orientação a objetos
O que são objetos?
Sintaxe
Classes, Atributos e Métodos
Construção de Programas em Java
Métodos Construtores e Overloading
Utilização das API's
Conceito de Pacote
Tipos de Métodos: Públicos, Privados e Protegidos
Composição e Herança
Polimorfismo
Métodos Estáticos
Programação Fundamental

Comentários
Tipos de Dados
Conversões Entre Tipos
Operadores
Incremento e Decremento
Operadores Relacionais e Booleanos
Strings
Vetores e Matrizes
Controle de Fluxo do Programa
Outras instruções
Objetivos do Desenho da Linguagem Java
Dicas para a Construção de Classes
Erros Mais Comuns e Suas Soluções
Problemas de Compilador
Erros de Sintaxe
Erros Semânticos
Erros de Execução
Introdução
Programação Orientada a Objetos
Ocultando de Informação
Especialização e Herança
Sobrescrita, Sobrecarga e Polimorfismo
Diagrama de Classes
Introdução à Linguagem Java
Palavras reservadas
Literais
Separadores
Tipos de dados simples
Tipos de dados compostos
Conversão de tipos
Operadores
Expressões e Precedência entre Operadores
Comentários
Blocos e escopo
Estruturas de controle
Seleção
Repetição
Break e continue
Argumentos da linha de comando
Assert (Assertivas)
Sintaxe e semântica
Habilitando e Desabilitando Assertivas
Classes, Packages e Interfaces
Classes
Construtores
Valor de Retorno
Objetos
Modificadores de acesso
Outros Modificadores
Referências compartilhadas
Copiando objetos

O objeto this
Usando Packages
Criando Packages
O Mecanismo de Extensão
Derivando classes
A classe Object
Sobrescrita e Polimorfismo
Classes e métodos abstratos
Interfaces
Classes internas
Classes Internas Anônimas
Conversão
Exceções
A hierarquia de Exceções
Capturando mais de uma exceção
Lançando exceções
Comportamento do Sistema diante das Exceções
Criando suas próprias exceções
A cláusula finally
Documentando o código
Rótulos
HTML embutida
Entrada e Saída (java.io)
Acesso sequencial
Acesso direto
O java.util
Lidando com coleções
As Interfaces Iterator e Enumeration
Vector
Stack
Hashtable
Miscelânea de classes do pacore java.util
Arrays
Date
Observable
StringTokenizer
Serialização e Persistência
AWT (Abstract Window Toolkit)
Tratamento de Eventos com classes Internas
Gerenciando o Layout
Exemplo com BorderLayout
Exemplo com FlowLayout
Exemplo com CardLayout
Exemplo com GridLayout
Exemplo com GridBagLayout
Utilizando listas
Trabalhando com menus e diálogos
Capturando eventos do teclado
Principais classes
Color
Component

Button
Label
List
TextField
TextArea
Containers
Panel
Frame
Applets
Descrição do código HTML
Métodos da classe Applet
Exibindo uma imagem
Áudio
Obtendo parâmetros
Executando um applet como aplicação
Preparando applets para produção e arquivos JARS
Criando os próprios arquivos Manifest
JavaBean
O que é um javabean
Javabeans e ferramentas RAD
Propriedades
Simples
Indexada
Ligada (Bound)
Restringidas(Constrained)
Eventos
TimerEventListener
TimerEvent
TimerBean
Instalando o Beans Development Kit (BDK)
Concorrência
Criando Threads em java
Criando threads por meio da interface Runnable
A classe Thread
Hierarquia
Construtores
Métodos
Variáveis públicas
Ciclo de vida dos Threads
Sleep(), yield(), join(), destroy(), stop(), suspend() e resume()
Daemon Threads
Influência do sistema operacional no comportamento dos Threads
Forma de escalonamento de threads
Relacionamento entre os níveis de prioridades definidas na linguagem Java
Níveis de prioridades definidas nos Sistemas Operacionais
Compartilhamento de memória e sincronização
Atomicidade de Instruções e Sincronização do Acesso à Sessões Críticas
Comunicação entre Threads: wait() e notify()
Animação
Programação em rede
Conceito sobre protocolos usados na Internet

TCP
UDP
Identificando Hosts (IP)
Identificação de Processos (Portas)
Programação em rede com Java
Comunicação Básica Entre Aplicações
Comunicação Sem Conexão (UDP)
Comunicação por meio de URL
Computação Distribuída (RMI)
Implementar interface do objeto remoto
Acesso a Banco de Dados
Tipos de drivers JDBC
Obtendo os Drivers JDBC
Configurando o ODBC
Carregando o Driver
Estabelecendo a conexão
Criando e Executando Comandos
Recuperando valores
Transações e nível de isolamentos
Prepared Statements
Stored Procedures
Servlets e JSP
Applets X Servlets
CGI X Servlets
A API Servlet
Exemplo de Servlet
Compilando o Servlet
Instalando o Tomcat
Criando uma aplicação no Tomcat
Executando o Servlet
Invocando diretamente pelo Navegador
Invocando em uma página HTML
Diferenças entre as requisições GET e POST
Obtendo informações sobre a requisição
Lidando com formulários
Lidando com Cookies
Lidando com sessões
JSP
PHP X JSP
ASP X JSP
Primeiro exemplo em JSP
Executando o arquivo JSP
Objetos implícitos
Tags JSP
Comentários
Diretivas
Extraindo Valores de Formulários
Criando e Modificando Cookies
O Uso de JavaBeans
Reecaminhando ou redirecionando requisições
Uma arquitetura para comércio eletrônico

Tipos de aplicações na Web
Arquitetura MVC para a Web
Agenda Web: Um Exemplo de uma aplicação Web usando a arquitetura MVC
Perguntas Frequentes
Exceções
Seu primeiro encontro com exceções Java
O Bloco try
O Bloco catch
Entrada e saída
Introdução ao Swing
Seu primeiro Swing

DOCUMENTO ASSINADO DIGITALMENTE

CURSOS VIRTUAIS LTDA - CNPJ 08.179.401/0001-62
Assinatura eletrônica qualificada com certificado digital ICP-Brasil

Código: **58**