

AlgaWorks

STARTER

INICIAÇÃO EM PROGRAMAÇÃO

Lógica de Programação com Java para Iniciantes

1. Introdução

- 1.1. Introdução ao curso de Lógica de Programação
 - 1.2. Introdução a computação
 - 1.3. O que é um algoritmo
 - 1.4. Linguagens de programação
 - 1.5. Como começar e terminar o curso
- ### 2. Começando

2.1. Como o Java funciona?

- 2.2. Instalando o Eclipse
- 2.3. Criando o primeiro projeto no Eclipse
- 2.4. Executando um programa simples no Eclipse
- 2.5. Exercício 01: Alterando a mensagem de Olá
- 2.6. Elaborando um algoritmo no Eclipse
- 2.7. Exercício 02: Cálculo do índice de massa corporal

- 2.8. Como importar um projeto no Eclipse
- 2.9. Palavras reservadas do Java

3. Variáveis e constantes

- 3.1. Introdução a variáveis
- 3.2. Tipos de dados existentes
- 3.3. Tipos numéricos (Byte, Short, Integer, Long)
- 3.4. Exercício 01: Calculando o quadrado de um número
- 3.5. Tipo texto (String)
- 3.6. Exercício 02: Faça a impressão do nome e do sobrenome da pessoa
- 3.7. Tipo lógico (Boolean)
- 3.8. Exercício 03: Crie um programa que informa se o aluno passou de ano
- 3.9. Tipos primitivos
- 3.10. Alterando o valor de uma variável
- 3.11. Exercício 04: Encontre o valor total baseado na quantidade e no desconto
- 3.12. Constantes
- 3.13. Exercício 05: Constantes
- 3.14. Escolhendo bons nomes para variáveis e constantes

4. Operadores

- 4.1. Introdução a operadores
- 4.2. Operadores aritméticos
- 4.3. Exercício 01: Calculadora simples

ALGAWORKS STARTER

- 4.4. Operadores relacionais
- 4.5. Exercício 02: Desconto de frete em compras acima de R\$100,00
- 4.6. Operadores de atribuição
- 4.7. Exercício 03: Cálculo de gastos familiar
- 4.8. Operadores lógicos
- 4.9. Exercício 04: Verificação da possibilidade de aposentadoria
- 4.10. Concatenação de texto
- 4.11. Operadores de incremento e decremento

5. Estruturas de decisão

- 5.1. Introdução a estruturas de decisão
- 5.2. Estrutura "if"
- 5.3. Exercício 01: Verificar se a pessoa passou no concurso público
- 5.4. Utilizando o "if" encadeado
- 5.5. Exercício 02: Calcule o bônus do funcionário(a)
- 5.6. Estrutura "switch"
- 5.7. Exercício 03: Imprima o nome do dia da semana (domingo, segunda-feira, etc.)

6. Iteração

- 6.1. Introdução a iteração
- 6.2. Iterando com o "for"
- 6.3. Exercício 01: Encontre os números divisíveis por 3
- 6.4. Iterando com o "while"
- 6.5. Exercício 02: Repita o exercício anterior com o "while"

7. Vetores

- 7.1. Introdução a vetores
- 7.2. Declarando vetores de 1 dimensão
- 7.3. Exercício 01: Cadastrando as tarefas do dia
- 7.4. Declarando vetores de 2 dimensões
- 7.5. Exercício 02: Encontre a semana de maior faturamento
- 7.6. Declarando vetores com mais de 2 dimensões

8. Métodos

- 8.1. Introdução a métodos
- 8.2. Criando o primeiro método
- 8.3. Exercício 01: Crie um segundo método no algoritmo da aula passada
- 8.4. Recebendo parâmetros
- 8.5. Exercício 02: Identifique um novo padrão no algoritmo da aula passada
- 8.6. Retornando valores
- 8.7. Exercício 03: Crie uma pequena calculadora de subtração e adição
- 8.8. Recursividade
- 8.9. Exercício 04: Exibir a tabuada de um número qualquer
- 8.10. Escolhendo bons nomes para métodos

9. Programação orientada a objetos

- 9.1. Introdução à programação orientada a objetos

ALGAWORKS STARTER

- 9.2. Criando a primeira classe
- 9.3. Exercício 01: Método que informa necessidade de repor estoque
- 9.4. Métodos de instância
- 9.5. Exercício 02: Método de instância que informa necessidade de repor estoque
- 9.6. Encapsulamento
- 9.7. Exercício 03: Exibindo os dados de um pedido
- 9.8. Diferença entre classe e instância

10. Leitura e escrita de dados em arquivos

- 10.1. Introdução à leitura e escrita
- 10.2. Escrevendo informações em arquivos de texto
- 10.3. Exercício 01: Crie uma lista de tarefas e salve a lista em um arquivo
- 10.4. Lendo informações de um arquivo de texto
- 10.5. Exercício 02: Mostre a lista de tarefas para o usuário

11. Utilizando código de outros programadores

- 11.1. Introdução ao uso de códigos de terceiros
- 11.2. Bibliotecas x Frameworks
- 11.3. Utilizando uma biblioteca de terceiro para envio de e-mails
- 11.4. Exercício 01: Crie uma lista de tarefas e envie a lista por email
- 11.5. Criando o sua própria biblioteca

- 11.6. Exercício 02: Use nossa biblioteca

12. Algoritmos avançados

- 12.1. Introdução aos algoritmos avançados
- 12.2. Criando uma lista dinâmica
- 12.3. Removendo elementos da lista
- 12.4. Ordenando a lista
- 12.5. Exercício 01: Ordene a lista de alunos do arquivo

13. Pré-requisitos para nossa primeira aplicação web

- 13.1. Por onde começar a criar uma aplicação web
- 13.2. Evoluindo para o Spring Tool Suite (STS)
- 13.3. O que é HTML?
- 13.4. O que é CSS?
- 13.5. Criando um projeto com o STS
- 13.6. Incluindo as páginas HTML no projeto
- 13.7. Fluxo de uma requisição web

14. Criando sua primeira aplicação

- 14.1. Importando o projeto de base
- 14.2. Criando a classe Contato
- 14.3. Configurando o armazenamento de contatos
- 14.4. Listando os contatos na página

- 14.5. O padrão JavaBean
- 14.6. Fazendo a página de cadastro funcionar
- 14.7. Verbos HTTP
- 14.8. Cadastrando um contato
- 14.9. Preparando para edição
- 14.10. Atualizando um contato
- 14.11. Removendo contatos
- 14.12. Conclusão e próximos passos

Banco de Dados e SQL para Iniciantes

1. Iniciando com banco de dados

- 1.1. O que é banco de dados?
- 1.2. Instalando o MySQL
- 1.3. Tipos de dados
- 1.4. A famosa chave primária
- 1.5. Schemas, inserindo e consultando dados
- 1.6. Desafio: Iniciando um novo projeto

2. Relacionamentos

- 2.1. Por que mais de uma tabela?

- 2.2. A chave estrangeira
- 2.3. Relacionamento muitos-para-um
- 2.4. Relacionamento um-para-muitos
- 2.5. Desafio: Relacionando tabelas
- 2.6. Relacionamento muitos-para-muitos
- 2.7. Desafio: Relacionamento muitos-para-muitos

3. Consultas

- 3.1. Cláusula where
- 3.2. Relacionando várias tabelas em uma consultas
- 3.3. Trabalhando com datas
- 3.4. Desafio: Relacionando tabelas

4. Alterações

- 4.1. Alterando sua tabela
- 4.2. Trabalhando com índices
- 4.3. Deletando dados
- 4.4. Atualizando dados
- 4.5. Desafio: Alterando suas tabelas

5. Consultas avançadas

- 5.1. Função sum
- 5.2. Agrupando dados com group by
- 5.3. Função avg
- 5.4. Desafio: Funções e group by

- 5.5. Ordenando com order by
- 5.6. Filtrando com like
- 5.7. Resultados únicos com distinct
- 5.8. Filtrando com in
- 5.9. Consultas dentro de consultas - subselect
- 5.10. Desafio: subselect

Ignição Java

- 1. Mergulhando em Java e OO
 - 1.1. Introdução e preparação de ambiente
 - 1.2. Fundamentos da linguagem Java
 - 1.3. Orientação a objetos, classes e objetos
 - 1.4. Métodos, construtores e sobrecarga
 - 1.5. Encapsulamento e JavaBeans
 - 1.6. Pacotes
- 2. Orientação a objetos avançada
 - 2.1. Herança e sobrescrita de métodos
 - 2.2. Upcasting e polimorfismo
 - 2.3. Classes abstratas
 - 2.4. Interfaces
 - 2.5. Exceções
 - 2.6. Boxing, enumerações, datas e valores monetários

- 3. Coleções, Lambdas e Streams API
 - 3.1. Collections Framework
 - 3.2. Interfaces funcionais e expressões lambda
 - 3.3. Streams API
 - 3.4. Optional
 - 3.5. Conclusão

Ignição Spring REST

- 1. Fundamentos de REST e Spring
 - 1.1. Boas vindas e as oportunidades do mercado
 - 1.2. Quem é você? Quem sou eu?
 - 1.3. Alguns combinados antes de continuar
 - 1.4. O que é uma API?
 - 1.5. O que é REST?
 - 1.6. Conhecendo o protocolo HTTP
 - 1.7. Entendendo os Recursos REST
 - 1.8. Identificando recursos REST
 - 1.9. Por que Spring?
 - 1.10. Conhecendo o ecossistema de projetos Spring
 - 1.11. Estudos de caso

2. Construindo uma REST API

- 2.1. Preparando o ambiente de desenvolvimento
- 2.2. Conhecendo o modelo de domínio do projeto
- 2.3. Alternativas para criar projetos Spring Boot
- 2.4. Criando o projeto com Spring Initializr
- 2.5. Entendendo a estrutura do projeto Maven
- 2.6. Gerando o FatJAR e iniciando a aplicação
- 2.7. Implementando e testando a requisição de um recurso
- 2.8. Implementando uma Collection Resource
- 2.9. Configurando e usando o Lombok
- 2.10. Métodos e códigos de status HTTP
- 2.11. Content Negotiation
- 2.12. Turbinando a produtividade com DevTools

3. Persistindo os dados

- 3.1. Configurando a conexão com o banco de dados no projeto
- 3.2. Conhecendo e adicionando o Flyway no projeto
- 3.3. Criando a primeira migration com Flyway
- 3.4. Conhecendo o Jakarta Persistence (JPA)
- 3.5. Mapeando entidades com Jakarta Persistence
- 3.6. Implementando uma consulta com JPQL
- 3.7. Conhecendo o Spring Data JPA (SDJ) e criando um repositório
- 3.8. Injetando e usando o repositório do SDJ
- 3.9. Implementando Query Methods no repositório
- 3.10. Implementando endpoint de busca de recurso
- 3.11. Implementando endpoint de inclusão de recurso

- 3.12. Implementando endpoint de atualização de recurso
- 3.13. Implementando endpoint de exclusão de recurso

4. Evoluindo a API

- 4.1. Conhecendo e adicionando Jakarta Bean Validation no projeto
- 4.2. Validando entrada de dados com Jakarta Bean Validation
- 4.3. Implementando Domain Services
- 4.4. Implementando regra de negócio para restringir e-mails duplicados
- 4.5. Capturando exceções do controlador com `@ExceptionHandler`
- 4.6. Adicionando migration para criação da tabela de veículos
- 4.7. Criando e mapeando a entidade de veículo
- 4.8. Implementando os endpoints de consulta de veículos
- 4.9. Implementando o endpoint de inclusão de veículos
- 4.10. Implementando regras de negócio no cadastro de veículos
- 4.11. Protegendo propriedades somente-leitura
- 4.12. Validando em cascata
- 4.13. Validando com Validation Groups

5. Aplicando as boas práticas

- 5.1. Capturando exceções globais com `@RestControllerAdvice`
- 5.2. Usando a RFC 7807 (Problem Details for HTTP APIs)
- 5.3. Customizando as informações do ProblemDetail
- 5.4. Adicionando campos customizados no ProblemDetail
- 5.5. Customizando as mensagens de validação
- 5.6. Tratando exceções customizadas de forma global

- 5.7. Boas práticas para trabalhar com data/hora
- 5.8. Isolando o Domain Model do Representation Model
- 5.9. Criando o Representation Model do recurso de veículo
- 5.10. Transformando objetos com ModelMapper
- 5.11. Implementando assembler de Representation Model
- 5.12. Compondo objetos no Representation Model
- 5.13. Criando um Representation Model para entrada de dados

6. Modelando sub-recursos e ações não-CRUD

- 6.1. Criando e mapeando a entidade de autuação
 - 6.2. Implementando serviço de autuação
 - 6.3. Modelando sub-recursos
 - 6.4. Implementando o endpoint de cadastro do recurso de autuação
 - 6.5. Especializando a exceção de entidade não encontrada
 - 6.6. Implementando recurso de coleção de autuações
 - 6.7. Implementando regras de negócio de apreensão de veículo
 - 6.8. Modelando ações não-CRUD
 - 6.9. Implementando endpoints de ações não-CRUD
- ### 7. Alcançando o próximo nível
- 7.1. Traçando um plano