

Exercícios Trilha Teórica de Back-End:

Exercício Básico I:

Você deve implementar uma classe chamada Calculadora que contenha métodos para realizar operações aritméticas básicas: adição, subtração, multiplicação e divisão. A classe deve permitir a execução dessas operações sobre dois números fornecidos pelo usuário.

Requisitos:

Variáveis:

Crie variáveis para armazenar os dois números sobre os quais as operações serão realizadas.

Crie uma variável para armazenar o resultado de cada operação.

Métodos:

Implemente métodos para cada uma das operações aritméticas:

double adicionar(double a, double b)

double subtrair(double a, double b)

double multiplicar(double a, double b)

double dividir(double a, double b)

Método Main:

No método main, crie uma instância da classe Calculadora.

Solicite ao usuário que insira dois números.

Utilize os métodos da classe Calculadora para realizar as quatro operações básicas com os números fornecidos.

Imprima os resultados de cada operação.

Exercício Básico II:

Desenvolva uma função que recebe uma lista como argumento, ao receber a lista, a função deve verificar se a lista está vazia, caso esteja deve retornar 0. Caso não esteja vazia deve executar um calculo da média de todos os itens da lista e retornar o valor calculado.

Requisitos:

Crie uma lista de INT ou DOUBLE para armazenar a quantidade de números desejados

Crie uma variável que represente a quantidade de itens na lista (para cálculo da média)

Crie um método chamado listMean() que recebe a lista como argumento.

No método main, atribua a função a uma variável e realize a impressão do número no console.

Exercício Intermediário I:

Você deve implementar um sistema de gerenciamento de contas bancárias. O sistema deve permitir a criação de contas, depósito de dinheiro, saque de dinheiro, e verificação de saldo. Além disso, deve incluir uma funcionalidade para transferir dinheiro entre contas.

Requisitos:

Classe ContaBancaria:

Atributos:

String titular
int numeroConta
double saldo

Métodos:

Construtor: public ContaBancaria(String titular, int numeroConta)
void depositar(double valor)
void sacar(double valor)
double verificarSaldo()
void transferir(ContaBancaria contaDestino, double valor)
Classe Banco:

Atributos:

ArrayList<ContaBancaria> contas

Métodos:

void adicionarConta(ContaBancaria conta)
ContaBancaria buscarConta(int numeroConta)

void listarContas()

Método Main:

No método main, crie uma instância da classe Banco.

Adicione algumas contas ao banco.

Realize operações de depósito, saque e transferência entre contas.

Imprima o saldo de algumas contas para verificar se as operações foram realizadas corretamente.

Exercício Intermediário II:

Desenvolva um formulário sem interface (no console) utilizando a classe Scanner, o formulário deve conter um modelo de dados com 4 campos diferentes (ex: Nome, Sobrenome, CPF, email). Para cada entrada feita no formulário, é necessária que a entrada seja verificada utilizando um padrão regex (ex: verificar se o cpf contém "." e "-" na sequência numérica, verificar se o email contém "@", etc). Ao fim da verificação, coloque todos os dados em uma lista e itere sobre todos os itens, imprimindo-os dessa forma: 'Dados Cadastrais: \n' 'Campo1 : Valor2' 'Campo2 : Valor2' 'Campo3: Valor3'

Requisitos:

Um método de verificação para cada campo do formulário

Variáveis para segurar o valor de cada campo

No método main, uma iteração sobre todos os itens da lista e a impressão deles no console, da forma que foi especificado no enunciado.

Exercício Avançado I:

Você deve implementar um sistema de gerenciamento de biblioteca. O sistema deve permitir o cadastro de livros, periódicos e mídias digitais, bem como o gerenciamento de empréstimos e devoluções. Utilize conceitos de orientação a objetos, tratamento de exceções e polimorfismo conforme descrito abaixo.

Requisitos:

Crie uma classe abstrata ItemBiblioteca com os seguintes atributos e métodos:

String titulo

String autor

boolean disponivel

Métodos abstratos void emprestar() e void devolver()

Implemente três classes concretas que herdem de ItemBiblioteca: Livro, Periodico e MidiaDigital. Cada classe deve implementar os métodos abstratos de maneira apropriada.

Crie uma interface Gerenciavel com os métodos:

'void adicionarItem(ItemBiblioteca item)'

'void removerItem(ItemBiblioteca item)'

'ItemBiblioteca buscarItem(String titulo)'

Implemente a interface Gerenciavel na classe Biblioteca, que deve gerenciar uma coleção de ItemBiblioteca utilizando um ArrayList.

Crie uma exceção personalizada ItemNaoDisponivelException que deve ser lançada quando tentar emprestar um item que não está disponível.

Crie uma exceção personalizada ItemNaoEncontradoException que deve ser lançada quando um item não for encontrado na biblioteca ao tentar remover ou buscar.

Método Main:

Implemente um método main na classe Biblioteca que demonstre:

A criação de uma biblioteca.

A adição de diferentes tipos de itens.

O empréstimo e a devolução de itens.

O tratamento das exceções definidas.

Exercício Avançado II:

Sistema de Cadastro e Busca de Usuários

Descrição

Você deve criar um sistema de cadastro e busca de usuários. O sistema deve permitir o cadastro de usuários com validação de dados, busca de usuários por meio de expressões regulares, e manipulação de uma lista de usuários utilizando coleções e generics. Além disso, deve-se fazer uso de classes locais, anônimas, e a classe Optional para o tratamento de possíveis valores ausentes.

Requisitos

Classe Usuario:

Atributos: id, nome, email, idade.

Métodos toString, equals e hashCode.

Classe UsuarioService:

Método para adicionar um usuário, lançando exceção em caso de dados inválidos.

Método para buscar usuários por nome usando regex.

Uso de Optional para retornar usuários buscados por email.

Classe Main:

Utilização de Scanner para interação com o usuário.

Exemplo de uso de classes locais e anônimas.