

Lista de Exercícios 09

Programação Orientada a Objetos • 2023.1

Tratamento de Exceções

01. Crie uma classe **Banco** que represente uma conta bancária simples. A classe deve ter um método **sacar** que aceita um valor como parâmetro e tenta realizar o saque da conta. No entanto, trate exceções caso o valor de saque seja maior que o saldo disponível na conta ou caso ocorra algum erro durante a transação. Exiba mensagens adequadas em caso de exceções. Teste o método **sacar** em um programa principal.

Dica: lance a exceção usando **throw**.

02. Crie uma classe **Usuario** que contenha um método **cadastrar()** que aceita o nome e a data de nascimento de um usuário como parâmetros. No entanto, trate exceções caso o nome fornecido seja nulo ou vazio, ou caso a data de nascimento se refira a uma idade menor que 18 anos ou maior que 120 anos. Lance exceções personalizadas. Teste o método **cadastrar()** em um programa principal, capturando e tratando as exceções lançadas.

03. Crie um programa que solicite ao usuário dois números inteiros e realize a divisão do primeiro número pelo segundo número. No entanto, trate possíveis exceções que possam ocorrer durante a divisão, como divisão por zero ou entrada inválida.

Dica: use o bloco **try/catch**.

04. Crie um programa que simule um sistema de autenticação de usuário. O programa deve solicitar ao usuário um nome de usuário e uma senha. Se o usuário inserir um nome de usuário válido ("admin") e uma senha válida ("12345"), o programa exibirá uma mensagem de boas-vindas. Caso contrário, o programa lançará uma exceção personalizada chamada **AcessoNegadoException** indicando que o acesso foi negado. O programa principal deve capturar e tratar a exceção lançada.

Instruções:

- Crie uma classe **AcessoNegadoException** que estenda a classe **Exception** para representar a exceção personalizada.
- Dentro da classe **AcessoNegadoException**, crie um construtor que chame o construtor da classe **Exception** sem argumentos.
- Crie um método chamado **verificarAcesso()** que aceite os parâmetros **usuario** e **senha** do tipo **String**.
- Dentro do método **verificarAcesso()**, verifique se o **usuario** é igual a "admin" e se a senha é igual a "12345".
- Se as condições acima forem verdadeiras, exiba a mensagem "Acesso permitido. Bem-vindo, " seguida do nome de usuário.
- Caso contrário, lance a exceção **AcessoNegadoException**.
- No método principal (main), solicite ao usuário um nome de usuário e uma senha utilizando a classe **Scanner**.

- h. Chame o método `verificarAcesso()` passando o nome de usuário e a senha fornecidos.
- i. Utilize um bloco try-catch para capturar a exceção `AcessoNegadoException` e exiba a mensagem de erro correspondente.
- j. Execute o programa e teste diferentes cenários, fornecendo combinações corretas e incorretas de nome de usuário e senha, verificando se a exceção é lançada e tratada corretamente.

* * *

Orientações para a entrega

- *A entrega deverá ser feita por meio do SIGAA até 28/06/2023 às 23:59.*
- *Apenas os arquivos .java devem ser enviados.*
- *O conjunto dos arquivos .java deve ser compactado em formato .zip*

Obs.: *Em caso de dificuldade para resolver a lista, solicitem ajuda ao monitor da disciplina.*