

DA1MCTA018-13SA - Programação Orientada a Objetos - Paulo Henrique Pisani - 2021.2

[Painel](#) / [Meus cursos](#) / [POO - DA1MCTA018-13SA - 2021.2](#) / [Tratamento de exceções](#) / [\[EP\] Autenticação usuário](#)

Descrição

[Visualizar envios](#)

[EP] Autenticação usuário

Data de entrega: quarta, 21 Jul 2021, 23:59

Arquivos requeridos: AutenticacaoUsuario.java ([Baixar](#))

Tipo de trabalho: Trabalho individual

Neste exercício, será usada a classe Usuario. Essa classe guarda o login e senha de um Usuario.

Classe **Usuario** (esta classe já está implementada no **pacote acesso** no sistema de correção automática):

- **Usuario(login: String, senha: String)** - construtor de Usuario que armazena o login e a senha.
- **getLogin(): String** - retorna o login da instância atual da classe Usuario.
- **autenticar(senha: String)** - verifica se a senha passada é correta e, caso não seja, lança a exceção SenhaInvalida ou UsuarioBloqueado.

Escreva um programa para autenticar usuários com login e senha que use a classe Usuario, conforme descrito a seguir (o programa deverá ser implementado na classe **AutenticacaoUsuario**):

1. Lê um inteiro n , que é a quantidade de usuários a serem cadastrados.
2. Depois o programa receberá uma sequência de n pares "login senha". O programa deve instanciar a classe Usuario uma vez para cada usuário. O construtor de Usuario recebe o login e senha como argumentos, por exemplo: new Usuario("login", "senha");
3. Após instanciar todos os usuários, o programa recebe mais um inteiro (k), que representa a quantidade de tentativas de autenticação a serem realizadas.
4. Depois o programa receberá uma sequência de k pares "login senha". Desta vez, o programa deverá chamar o método autenticar do usuário correspondente ao login e, para esse usuário, passar a senha lida como argumento (assuma que sempre será passado um login de um usuário que existe). Após chamar o método autenticar, o programa deverá imprimir o resultado usando a classe Impressao de acordo com o caso (os três métodos são static na classe Impressao):
 1. **Impressao.imprimirUsuarioAutenticado(login: String)**: quando o método autenticar não lança exceção
 2. **Impressao.imprimirSenhaInvalida(login: String)**: quando o método autenticar lançar a exceção SenhaInvalida (ocorre se a senha passada é incorreta)
 3. **Impressao.imprimirUsuarioBloqueado(login: String)**: quando o método autenticar lançar a exceção UsuarioBloqueado (essa exceção é lançada se a senha é incorreta 3 ou mais vezes seguidas)

Observação: as classes das exceções SenhaInvalida e UsuarioBloqueado possuem um método chamado getLogin(): String, que pode ser usado para obter o login do usuário de onde a exceção foi lançada.

As classes Usuario, SenhaInvalida, UsuarioBloqueado e Impressao estão no **pacote acesso** e já existem no sistema de correção automática (portanto, não devem ser submetidas). Na aba Editar, há uma implementação inicial do programa com os imports necessários já incluídos.

Importante: Submeta apenas o programa principal (o programa deve estar no **pacote sem nome**, ou seja, sem a linha package). A impressão de dados é permitida apenas com a classe Impressao, usando os três métodos static especificados.

Casos de teste

Formato dos casos de teste (que aparecem ao avaliar as classes no sistema de correção automática):

Entrada:

- n (quantidade de usuários)
- sequência de "login senha" (para cadastro dos usuários)
- k (quantidade de tentativas)
- sequência de "login senha" (para chamar o método autenticar)

Saída:

- dados impressos pelas classes Usuario, SenhaInvalida e UsuarioBloqueado ao longo do programa
- saída dos métodos da classe Impressao

Arquivos requeridos

AutenticacaoUsuario.java

```
1 import java.util.Scanner;  
2 import acesso.Usuario;  
3 import acesso.Impressao;  
4 import acesso.UsuarioBloqueado;  
5 import acesso.SenhaInvalida;  
6  
7 public class AutenticacaoUsuario {  
8     public static void main(String[] args) {  
9  
10    }  
11 }
```

[VPL](#)

◀ [EP] Lâmpada Inteligente com exceção

Seguir para...

[EP] Playlist com exceções ▶



Este é o Ambiente Virtual de Aprendizagem da UFABC para apoio ao ensino presencial e semipresencial. Esta plataforma permite que os usuários (educadores/alunos) possam criar cursos, gerenciá-los e participar de maneira colaborativa.

Informação

Conheça a UFABC

Conheça o NTI

Conheça o Netel

Contato

Av. dos Estados, 5001. Bairro Bangu - Santo André /SP – Brasil. CEP 09210-580.

Siga-nos



[Universidade Federal do ABC](#) - [Moodle](#) (2020)

[Obter o aplicativo para dispositivos móveis](#)