import java.util.Arrays;

import java.util.Scanner;

public class HalcionOnlineBanking {

// Define the correct username and password

private static final String CORRECT\_USERNAME = "user";

private static final char[] CORRECT\_PASSWORD = {'p', 'a', 's', 's', 'w', 'o', 'r', 'd'};

private static final int MAX\_TRIES = 3;

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("--- Welcome to Halcion Online Banking ---");

System.out.println("Please enter your credentials to log in.");

boolean loggedIn = false;

for (int attempt = 1; attempt <= MAX\_TRIES; attempt++) {

System.out.println("\nAttempt " + attempt + " of " + MAX\_TRIES);

// Get username

System.out.print("Enter Username: ");

String username = scanner.nextLine();

// Get password with masking

char[] password = readPasswordWithMasking("Enter Password: ");

// Validate credentials

if (username.equals(CORRECT\_USERNAME)) {

if (Arrays.equals(password, CORRECT\_PASSWORD)) {

System.out.println("\nLogin Successful! Welcome, " + username + ".");

loggedIn = true;

// Clear password array for security

Arrays.fill(password, ' ');

break;

} else {

System.out.println("Error: Incorrect Password.");

}

} else {

System.out.println("Error: Incorrect Username.");

}

// Clear password array after use

if (password != null) {

Arrays.fill(password, ' ');

}

if (!loggedIn && attempt < MAX\_TRIES) {

System.out.println("You have " + (MAX\_TRIES - attempt) + " attempt(s) remaining.");

}

}

if (!loggedIn) {

System.out.println("\nLogin failed after " + MAX\_TRIES + " attempts. Your account has been locked. Please contact support.");

}

scanner.close();

}

/\*\*

\* Reads password input from the user, displaying '\*' for each character.

\* Works in most terminals.

\*/

private static char[] readPasswordWithMasking(String prompt) {

System.out.print(prompt);

// Use a StringBuilder to collect characters

StringBuilder passwordBuilder = new StringBuilder();

try {

while (true) {

int ch = System.in.read();

// Enter key (newline or carriage return)

if (ch == '\n' || ch == '\r') {

break;

}

// Handle backspace (ASCII 8 or 127)

if (ch == 8 || ch == 127) {

if (passwordBuilder.length() > 0) {

passwordBuilder.deleteCharAt(passwordBuilder.length() - 1);

// Move cursor back, overwrite with space, move back again

System.out.print("\b \b");

}

} else if (ch >= 32 && ch <= 126) { // Printable characters

passwordBuilder.append((char) ch);

System.out.print("\*");

}

// Ignore other control characters

}

} catch (Exception e) {

e.printStackTrace();

}

System.out.println(); // Move to next line after input

return passwordBuilder.toString().toCharArray();

}

}