

Técnicas de programación segura: Programación de mecanismos de control de acceso:

Explicación

Este código realiza una simulación de una base de datos para llevar a cabo una autenticación básica consistente en verificar las credenciales de los usuarios ingresados.

Al iniciar el programa a través del IDE a elegir:

1. Se inicializa una base de datos simulada de usuarios y contraseñas utilizando un HashMap. Esta base de datos se inicializa con 2 usuarios y contraseñas, los cuales se muestran en el propio código. Esta práctica NO es segura, pero debido a la naturaleza del trabajo, no era necesario recurrir a mecanismos de cifrado para guardar los datos.
2. Se muestra un mensaje de bienvenida al usuario.
3. Se solicita al usuario su nombre de usuario y contraseña.
4. Si el usuario se equivoca 3 veces, o si escribe las credenciales correctas, la aplicación termina.
5. El mensaje final dependerá de si se ha logueado correctamente o no.

Es un ejemplo de aplicación muy simple para mostrar uno de los muchos mecanismos de control de acceso. Otros ejemplos pueden ser:

1. Programas que manejen los permisos de los usuarios registrados. No es lo mismo un empleado que un administrador.
2. Programas que gestionen accesos a recursos restringidos.
3. Programas de gestión de sesiones seguras.

Y muchos más ejemplos, los cuales irán aumentando tanto en número como en complejidad.

Código fuente.

```
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

public class Autenticacion {
    // Base de datos simulada para usuarios y contraseñas
    private Map<String, String> usuarios = new HashMap<>();

    public Autenticacion() {
        // Inicialización de usuarios y contraseñas
        usuarios.put("usuario1", "contraseña1");
        usuarios.put("usuario2", "contraseña2");
    }

    // Método para autenticar usuarios
    public boolean autentUsuario(String nombre, String contraseña) {
        // Verificar si el usuario existe y la contraseña es correcta
        return usuarios.containsKey(nombre) &&
            usuarios.get(nombre).equals(contraseña);
    }

    public static void main(String[] args) {
        Autenticacion authSystem = new Autenticacion();
        Scanner scanner = new Scanner(System.in);
        // Crear las variables del sistema de autenticación
        boolean sesionIniciada = false;
        int contador = 0;
        // Inicializar el sistema de autenticación
        System.out.println("Bienvenido al sistema de autenticación.");

        // Bucle principal del sistema de autenticación
        // Si el numero de intentos llega a 3, cerrar el sistema
        while (!sesionIniciada && contador < 3) {
            // Solicitar credenciales al usuario

            System.out.print("Ingrese su nombre de usuario: ");
            String nombre = scanner.nextLine();
            System.out.print("Ingrese su contraseña: ");
            String contraseña = scanner.nextLine();

            // Autenticar al usuario
            if (authSystem.aumentUsuario(nombre, contraseña)) {
                System.out.println("¡Inicio de sesión exitoso!");
                sesionIniciada = true;
                // Aquí se modifica el valor de sesión iniciada para
                romper el bucle
            } else {
                System.out.println("Nombre de usuario o contraseña
                incorrectos.");
                contador++;
                // Si el usuario es incorrecto, incrementar el
                contador de intentos y notificar al usuario
            }
        }
        // Notificación de cierre de sistema
        if (contador == 3) {
            System.out.println("Demasiados intentos fallidos. Cerrando
            el sistema");
        }
    }
}
```

```
scanner.close();  
}  
}
```

Bibliotecas y dependencias:

Este programa no utiliza ninguna dependencia que no sea una clase estándar de Java.

A continuación, explicaré sus usos en esta aplicación:

- `java.util.HashMap` y `java.util.Map`: Se utilizan para simular la base de datos de usuarios.
- `java.util.Scanner`: Se utiliza para permitir la funcionalidad de la aplicación, ya que es necesario para introducir los valores de inicio de sesión.

Ubicación del JAR

Desde la raíz del repositorio:

`out/artifacts/MecanismosControlAcceso_jar/MecanismosControlAcceso.jar`

Enlace a GitHub

<https://github.com/Anher97/MecanismosControlAccesoAHS>