The code provided is for changing a user's password in an LDAP Active Directory (AD) environment. Below is a detailed workflow and breakdown of each component.

**Workflow Breakdown:**

1. **Password Generation**:
   * The program generates a random alphanumeric password using a secure random generator.
2. **Input User Details**:
   * The user is prompted to input the username (Common Name) for which the password will be updated.
3. **LDAP Connection Setup**:
   * It prepares the necessary configurations to connect to the LDAP server using the administrator credentials.
4. **Password Encoding**:
   * The password is encoded in UTF-16LE format (required by Active Directory).
5. **Modify LDAP Entry**:
   * The password attribute for the user is updated in the Active Directory by using LDAP commands.
6. **Connection Closure**:
   * After the password is successfully changed, the connection to the LDAP server is closed.

**Code Components Explained:**

1. **generateRandomPassword(int length)**:
   * **Purpose**: Generates a random password of a specified length.
   * **Working**:
     + A string of valid characters (A-Z, a-z, 0-9) is defined.
     + A SecureRandom object is used to randomly select characters.
     + A loop runs for the length specified, building a password from random characters.
2. **encodePasswordForAD(String password)**:
   * **Purpose**: Encodes the password in UTF-16LE format (required by Active Directory).
   * **Working**:
     + The password is first wrapped in double quotes ("password") as required by AD.
     + The password is then converted to a byte array in UTF-16LE format for LDAP communication.
3. **Scanner to take input**:
   * **Purpose**: Captures user input from the console.
   * **Working**:
     + The username of the LDAP user is entered.
     + This username is used to construct the distinguished name (DN) of the user in AD.
4. **LDAP Connection Configuration**:
   * **LDAP URL**: ldaps://app010w001.minjtech.xyz:636 — The secure LDAP server URL, using port 636 for an encrypted connection.
   * **Admin DN**: The distinguished name of the admin account (CN=Administrator,CN=Users,DC=minjtech,DC=xyz).
   * **Admin Password**: The administrator’s password to authenticate.
   * A Hashtable is used to set these LDAP connection parameters.
5. **Establishing LDAP Connection (InitialDirContext)**:
   * **Purpose**: Creates an LDAP connection using the provided admin credentials and URL.
   * **Working**:
     + Uses InitialDirContext to create a connection to the LDAP server using the parameters set in the env hashtable.
6. **Password Modification**:
   * **Purpose**: Changes the user's password in Active Directory.
   * **Working**:
     + ModificationItem[] is an array that holds the modifications (here, it replaces the user's unicodePwd attribute with the new encoded password).
     + ctx.modifyAttributes(userDN, mods) is the LDAP operation to apply the modification.
     + If successful, the password is changed for the user specified by userDN.
7. **Error Handling**:
   * **NamingException Handling**: If any issue occurs while communicating with the LDAP server (e.g., incorrect credentials, invalid user DN), the exception is caught and printed.
8. **Resource Cleanup**:
   * **Scanner Closure**: Closes the Scanner resource after the input is taken to avoid resource leaks.
   * **LDAP Context Closure**: The LDAP context (ctx.close()) is closed after the password is modified to free up resources.

**Complete Flow:**

1. User inputs the username for which they want to change the password.
2. The program generates a random password.
3. It establishes a connection to the LDAP server using admin credentials.
4. The random password is encoded and formatted for AD.
5. The user's password is updated in AD through an LDAP operation.
6. The connection is closed, and any errors are handled.

This code efficiently handles password updates in Active Directory for users via LDAP protocol.

import javax.naming.Context;

import javax.naming.NamingException;

import javax.naming.directory.DirContext;

import javax.naming.directory.InitialDirContext;

import javax.naming.directory.ModificationItem;

import javax.naming.directory.BasicAttribute;

import java.security.SecureRandom;

import java.util.Hashtable;

import java.util.Scanner;

public class ChangePassword {

    // Method to generate a random alphanumeric password

    public static String generateRandomPassword(int length) {

        final String chars = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";

        SecureRandom random = new SecureRandom();

        StringBuilder password = new StringBuilder(length);

        for (int i = 0; i < length; i++) {

            password.append(chars.charAt(random.nextInt(chars.length())));

        }

        return password.toString();

    }

    // Method to format password for AD (UTF-16 with quotes)

    public static byte[] encodePasswordForAD(String password) {

        String quotedPassword = "\"" + password + "\""; // Password must be in quotes

        try {

            return quotedPassword.getBytes("UTF-16LE");

        } catch (Exception e) {

            throw new RuntimeException("Error encoding password to UTF-16LE", e);

        }

    }

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        // Taking input for username

        System.out.print("Enter the username (cn) to update the password: ");

        String username = scanner.nextLine(); // Get username input

        // LDAP server connection details

        String ldapUrl = "ldaps://app010w001.minjtech.xyz:636"; // LDAP URL

        String userDN = "CN=" + username + ",CN=Users,DC=minjtech,DC=xyz"; // Construct User DN from input

        String adminDN = "CN=Administrator,CN=Users,DC=minjtech,DC=xyz"; // Admin DN

        String adminPassword = "Login%12345"; // Admin Password

        // Generate a new random password

        String newPassword = generateRandomPassword(10);

        System.out.println("Generated new password: " + newPassword);

        Hashtable<String, String> env = new Hashtable<>();

        env.put(Context.INITIAL\_CONTEXT\_FACTORY, "com.sun.jndi.ldap.LdapCtxFactory");

        env.put(Context.PROVIDER\_URL, ldapUrl);

        env.put(Context.SECURITY\_AUTHENTICATION, "simple");

        env.put(Context.SECURITY\_PRINCIPAL, adminDN); // Administrator DN

        env.put(Context.SECURITY\_CREDENTIALS, adminPassword); // Admin password

        try {

            DirContext ctx = new InitialDirContext(env);

            // Prepare the modification

            ModificationItem[] mods = new ModificationItem[1];

            mods[0] = new ModificationItem(DirContext.REPLACE\_ATTRIBUTE, new BasicAttribute("unicodePwd", encodePasswordForAD(newPassword)));

            // Change the password

            ctx.modifyAttributes(userDN, mods);

            System.out.println("Password changed successfully for user: " + username);

            ctx.close();

        } catch (NamingException e) {

            e.printStackTrace();

        } finally {

            scanner.close(); // Close the scanner resource

        }

    }

}