Homework-11-Leo-2

2： 请用你熟悉的编程语言写一个用户密码验证函数，

Boolean checkPW(String 用户ID，String 密码明文，String 密码密文)

返回密码是否正确boolean值，密码加密算法使用你认为合适的加密算法。

```
package com.ll;

import javax.crypto.SecretKeyFactory;
import javax.crypto.spec.PBEKeySpec;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.security.spec.InvalidKeySpecException;
import java.util.Arrays;
import java.util.Base64;
import java.util.Random;

public class PasswordUtils {
    private static final Random RANDOM = new SecureRandom();
    private static final String ALPHABET =
"0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";
    private static final int ITERATIONS = 10000;
    private static final int KEY_LENGTH = 256;

    private static byte[] hash(char[] password, byte[] salt) {
        PBEKeySpec spec = new PBEKeySpec(password, salt, ITERATIONS,
KEY_LENGTH);
        Arrays.fill(password, Character.MIN_VALUE);
        try {
            SecretKeyFactory skf =
SecretKeyFactory.getInstance("PBKDF2WithHmacSHA1");
            return skf.generateSecret(spec).getEncoded();
        } catch (NoSuchAlgorithmException | InvalidKeySpecException e) {
            throw new AssertionError("Error while hashing a password: " +
e.getMessage(), e);
        } finally {
            spec.clearPassword();
        }
    }

    public static String generateSecurePassword(String plainPassword, String
userID) {
        String ret = null;
```

```
        byte[] securePassword = hash(plainPassword.toCharArray(),
userID.getBytes());

        return Base64.getEncoder().encodeToString(securePassword);
    }

    public static boolean checkPW(String userID, String plainPassword, String
encodePassword) {
        boolean ret = false;

        String newSecurePassword = generateSecurePassword(plainPassword,
userID);

        return newSecurePassword.equalsIgnoreCase(encodePassword);
    }
}
```

```
package com.ll;

public class Main {

    public static void main(String[] args) {
  // write your code here
        String plainPassword = "myPassword123456";
        String userId = "user-98527";

        String securePassword =
PasswordUtils.generateSecurePassword(plainPassword, userId);

        boolean passwordMatch = PasswordUtils.checkPW(userId, plainPassword,
securePassword);

        if (passwordMatch) {
            System.out.println("Provided user password " + plainPassword + "
is correct.");
        } else {
            System.out.println("Provided user password is incorrect!");
        }
    }
}
```