// jojo8/juno/Password.java
/*
 * Copyright 2003 Bill Campbell and Ethan Bolker
 */
/**
 * Model a good password.
 *
 * A password is a String satisfying the following conditions (close to those required of Unix passwords, according to the
 * man passwd command in Unix):
 * - A password must have at least PASSLENGTH characters, where PASSLENGTH defaults to 6. Only the first eight characters are significant.
 * - A password must contain at least two alphabetic characters and at least one numeric or special character. In this case, "alphabetic" refers to all upper or lower case letters.
 * - A password must not contain a specified string as a substring. For comparison purposes, an upper case letter and its corresponding lower case letter are equivalent.
 * - A password must not be a substring of a specified string. For comparison purposes, an upper case letter and its corresponding lower case letter are equivalent.
 *
 * A password string may be stored in a Password object only in encrypted form.
 */

public class Password
{

private String password;

/**
 * Construct a new Password.
 *
 * @param password the new password.
 * @param notSubstringOf a String that may not contain the password.
 * @param doesNotContain a String the password may not contain.
 *
 * @exception BadPasswordException when password is unacceptable.
 */

public Password(String password, String notSubstringOf, String doesNotContain)
throws BadPasswordException
{
    this.password = encrypt(password);
}

// Rewrite s in a form that makes it hard to guess s.

private String encrypt( String s )
{
    return Integer.toHexString(s.hashCode());
}

/**
 * See whether a supplied guess matches this password.
 *
 * @param guess the trial password.
 *
 * @exception BadPasswordException when match fails.
 */

public void match(String guess)
throws BadPasswordException
{
}

/**
 * Unit test for Password objects.
 */

public static void main( String[] args )
{
}