/\*Class PasswordVerifyException. It extends the Exception class and from it we can create  
our own exceptions \*/  
public class PasswordVerifyException extends Exception {  
 String username;  
 String password;  
  
 //constructor  
 public PasswordVerifyException(String username, String password, String message) {  
 super(message);  
 this.username = username;  
 this.password = password;  
 }  
}

/\*Class PasswordVerify. The constructor of this class throws an exception if the  
username or password violates any of the rules, and displays a message to the user  
to let them know which password/username rule was broken.  
 \*/  
public class PasswordVerify {  
 String username;  
 String password;  
  
 //Constructor. It throws a PasswordVerifyException  
 public PasswordVerify(String username, String password) throws PasswordVerifyException{  
  
 //if condition to check whether the username has at least 6 characters  
 if (username.length() < 6) {  
 throw new PasswordVerifyException(username, password, "Username must be at least 6 characters long. " +  
 "Press enter to try again. ");  
 }  
  
 //The following code checks whether the username has at least 2 vowels  
 int vowelCount = 0; //variable to store the number of vowels  
  
 //for loop that checks whether each character is a vowel and increments vowelCount accordingly  
 for (int i = 0; i < username.length(); i++ ){  
 if (username.charAt(i) =='a' || username.charAt(i) =='e' || username.charAt(i) =='i'  
 || username.charAt(i) =='o' ||username.charAt(i) =='u'){  
 vowelCount++;  
 }  
 }  
 //checks whether the username has at least 2 vowels  
 if (vowelCount < 2){  
 throw new PasswordVerifyException (username, password, "The username needs to have at least 2 vowels. " +  
 "Press Enter to try again. ");  
 }  
  
 //check if password has exactly 8 characters  
 if (password.length() != 8){  
 throw new PasswordVerifyException(username, password, "The password needs to have exactly 8 characters. " +  
 "Press Enter to try again. ");  
 }  
  
 //checks if the 4th character of the password is an '%'  
 if(password.charAt(3)!= '%'){  
 throw new PasswordVerifyException(username, password, "The 4th character of the password needs to be an '%." +  
 " Press Enter to try again. ");  
 }  
  
 this.username= username;  
 this.password = password;  
 }  
  
 //ToString method to print class attributes  
 @Override  
 public String toString(){  
 return "Username: " + username + ", Password: " + password;  
 }  
  
}

/\*  
Marla Peraza Ravelo  
COP-3330C-17193  
9/22/2024  
The objective of this program is to showcase how to throw and catch exceptions, as well as  
how to create our custom exceptions so that we can have specific types of exceptions that suit our  
problems.  
For the first part, the user is going to input two integer numbers, and the output is going to be  
the result of the first one divided by the second one. If the second number is a 0, the program  
catches an exception and outputs that division by 0 is not possible.  
For the second part, the user is going to input a series of usernames and passwords. The usernames need to  
be at least 6 characters long and need to have at least 2 vowels. The passwords can only be exactly  
8 characters and the 3rd character needs to be '%'. If they do not follow the rules,  
an exception is thrown and a message is going to print for the user to try again. After all  
the accounts are created, the details are printed on the screen.  
  
 \*/  
  
import java.util.Scanner; //imports Scanner  
//Main class. In this class the PasswordVerify objects are going to be created and exceptions are going to  
//be demonstrated.  
public class Main {  
 public static void main(String[] args) {  
  
 //Part 1: Throwing and catching a basic exception  
 //creates a new Scanner object  
 Scanner scanner = new Scanner(System.*in*);  
  
 try {//try block starts  
 // Prompt the user to enter the first integer  
 System.*out*.print("Enter the first integer: ");  
 int num1 = scanner.nextInt();  
  
 //Prompt the user to enter the second integer  
 System.*out*.print("Enter the second integer: ");  
 int num2 = scanner.nextInt();  
  
 // Check for division by zero  
 if (num2 == 0) {  
 throw new ArithmeticException("Division by zero is not allowed.");  
 }//try block ends  
  
 // Performs division and prints the result as a double with two decimal places  
 double result = (double) num1 / num2;  
 System.*out*.printf("%d/%d = %.2f%n", num1, num2, result);  
  
 } catch (ArithmeticException e) { //catch block starts  
 // Catch divide-by-zero exception and display a message  
 System.*out*.println(e.getMessage());  
  
 }//catch block ends  
  
 //Clears the buffer  
 scanner.nextLine();  
  
 //Part 2: Creating custom exceptions  
 //Number of times to ask the user for input  
 int dataCount = 4;  
  
 //temporary variables  
 String pass = null;  
 String user = null;  
  
 //Array to store each object information  
 PasswordVerify[] accounts = new PasswordVerify[dataCount];  
  
 //Display the rules for entering the username and the password  
 System.*out*.println("Creating account Rules:");  
 System.*out*.println("1. Username must be at least 6 characters long and contain at least 2 vowels");  
 System.*out*.println("2. Password must be exactly 8 characters long, and the 4th character must be a '%'");  
 System.*out*.println();  
  
 /\*for loop that instantiates the PasswordVerify objects using input from the user,  
 checks for exceptions, and stores objects in array \*/  
 for (int i = 0; i < dataCount; i++) {  
 try { //try block starts  
  
 //asking from input  
 System.*out*.println("Enter the username for account " + i);  
 user = scanner.nextLine();  
  
 System.*out*.println("Enter the password for the account " + i);  
 pass = scanner.nextLine();  
  
 //add the new account to the array  
 accounts[i] = new PasswordVerify(user, pass);  
  
 System.*out*.println("New Account created successfully");  
  
 } //try block ends  
 catch (PasswordVerifyException pwe) { //catch block starts  
  
 //prints message if an exception is caught  
 System.*out*.println(pwe.getMessage());  
 //clear the buffer  
 scanner.nextLine();  
 //decrements the index so the user can try again  
 i--;  
  
 } //catch block ends  
  
 }  
 //prints all the usernames with their corresponding passwords  
 System.*out*.println("All accounts created:");  
 for (int o=0; o <accounts.length; o++){  
 System.*out*.println(accounts[o]);  
 }  
  
 //closes the scanner  
 scanner.close();  
  
 }  
  
  
}