

PSP PREPARACION EXAMEN FEBRERO EJERCICIO 5

Realiza un programa Java que lea una cadena desde la entrada estándar o a través de una tubería y visualice en pantalla si la cadena es o no palíndromo o si la cadena está vacía (la longitud es 0).

Realiza un segundo programa Java que mande COMPILAR y, después, EJECUTAR el anterior, mostrando por pantalla la salida que genere el primero. Solucion de Rafa...

Clase Palindromo

```
package ej5_mandarcompileyexecutarclase;

import java.io.*;
import java.util.Scanner;

public class Palindromo {

    public static void main(String[] args) {

        // Leemos una cadena desde el teclado o
        // desde las tuberías

        InputStreamReader streamTuberia = null;
        BufferedReader tuberia = null;
        InputStream streamTeclado = null;
        Scanner teclado = null;
        String palindromo = "";
        try {
            // Prepara el buffer para leer de tubería
            streamTuberia = new InputStreamReader(System.in);
            tuberia = new BufferedReader(streamTuberia);
            // Prepara el buffer para leer de teclado
            streamTeclado = System.in;
            teclado = new Scanner(streamTeclado);

            boolean preparado = tuberia.ready();
            if (preparado) {
                // Hay datos en la tubería y se leen
                palindromo = tuberia.readLine();
            } else {
                // No hay datos en la tubería y se solicitan por teclado
                palindromo = teclado.nextLine();
            }
            if (palindromo == null || palindromo.isEmpty()) {
                System.out.println("\nLa cadena que se ha recibido esta nula o vacia\n.");
            } else if (esPalindromo(palindromo)) {
                System.out.println("\nLa palabra " + palindromo + " si es un palindromo.\n");
            } else {
                System.out.println("\nLa palabra " + palindromo + " no es un palindromo.\n");
            }
        } catch (IOException ex) {
            System.err.println("Error debido a la E/S general.");
        } catch (Exception e) {
            System.err.println("Ha ocurrido un error: " + e.getMessage());
        } finally {
            // Se cierran las conexiones
        }
    }

    private static boolean esPalindromo(String s) {
        if (s == null || s.isEmpty()) return true;
        String reversed = new StringBuilder(s).reverse().toString();
        return s.equals(reversed);
    }
}
```

```

    try {

        if (streamTuberia != null) {
            streamTuberia.close();
        }
        if (tuberia != null) {
            tuberia.close();
        }
        if (streamTeclado != null) {
            streamTeclado.close();
        }
        if (teclado != null) {
            teclado.close();
        }
    } catch (IOException ex) {
        System.err.println("Error debido a la E/S en el cierre de las conexiones.");
    }
}

}

private static boolean esPalindromo(String palindromo) {
    // La cadena la convertimos a minúsculas y le quitamos
    // los espacios, puntos y comas. También quitamos las
    // tildes.
    palindromo = palindromo.toLowerCase().replace("á", "a").replace("é", "e").replace("í",
    "i").replace("ó", "o")
        .replace("ú", "u").replace(" ", "").replace(".", "").replace(",", "");

    // Invertimos la cadena y si es igual que la original entonces
    // es un palíndromo
    String invertida = new StringBuilder(palindromo).reverse().toString();

    return invertida.equals(palindromo);
}
}

```

Main

```
package ej5_mandarcompilaryejecutarclase;

import java.io.*;

public class Main {

    public static void main(String[] args) {

        compilarClase_ProcessBuilder(); // Compilamos el programa Palindromo

        ejecutarClase_ProcessBuilder(); // Ejecutamos el programa Palindromo
    }

    public static void compilarClase_ProcessBuilder() {
        try {
            Process proceso = new ProcessBuilder("CMD", "/C",
                "JAVAC src\\ej5_mandarcompilaryejecutarclase\\Palindromo.java").start();
            int estado = proceso.waitFor();
            if (estado == 0) {
                // OK
                System.out.println("Clase compilada.");
            } else {
                // KO
                System.out.println("Error al compilar la clase");
            }
        } catch (SecurityException eSec) {
            System.out.println("Error de seguridad. Sin permiso para ejecutar el proceso");
        } catch (Exception e) {
            System.err.println("Ha ocurrido un error: " + e.getMessage());
        }
    }

    private static void ejecutarClase_ProcessBuilder() {
        InputStream is = null;
        try {
            // WAAKKAAW es un palindromo
            Process proceso = new ProcessBuilder("CMD", "/C",
                "ECHO WAAKKAAW | java src\\ej5_mandarcompilaryejecutarclase\\Palindromo.java").start();

            int estado = proceso.waitFor();
            if (estado == 0) {
                // OK
                System.out.println("\nClase ejecutada.");
                is = proceso.getInputStream();
                int c;
                while ((c = is.read()) != -1) {
                    System.out.print((char) c);
                }
            } else {
                // KO
                System.out.println("Error al ejecutar la clase\n");
            }
        } catch (SecurityException eSec) {
            System.out.println("Error de seguridad. Sin permiso para ejecutar el proceso");
        } catch (Exception e) {
            System.err.println("Ha ocurrido un error: " + e.getMessage());
        } finally {
            // Se cierran las conexiones
        }
    }
}
```

```
    try {
      if (is != null) {
        is.close();
      }
    } catch (IOException ex) {
      System.err.println(
        "Error debido a la E/S en los cierre de los flujos.");
    }
  }
}
```