

**EJERCICIO 1. Dadas las clases metodos1.java y metodos2.java**

Crea el proyecto tratamientostring y dentro de éste las clases metodos1 y metodos2 usando el adjunto y realiza los pasos siguientes:

- Reorganiza el código de la clase teniendo en cuenta las reglas de refactorización vistas al principio de la unidad y rediseñando la clase y los métodos para poder testear los mismos en test JUnit de caja negra si fuese necesario.
- Crea un test que pruebe los dos métodos de la clase.
- Crea un test parametrizado para el método cuentaVocales con 5 frases distintas.
- Crea un Test Suite que ejecute los dos test anteriores.

Exporta en .zip el proyecto y súbelo en el enlace de entrega.

```
package tratamientostring;
import java.util.Scanner;

/**
 *
 * @author Entornos de Desarrollo - Ruiz Gijón
 */
public class metodos1 {

    public static int repiteCaracter(){
        Scanner sc = new Scanner(System.in);
        System.out.println("Ingrese la cadena:");
        String cadena = sc.nextLine(); //leemos la cadena
        System.out.println("Ingrese el carácter a buscar:");
        String caracter = sc.next(); //ya que no contamos con un
        método específico para
        //los caracteres entonces leemos como cadena, luego obtenemos el
        primer
        //carácter para almacenarlo en un char, que sería el char que
        debemos buscar en
        //la primera cadena ingresada
        char car = caracter.charAt(0);
        int cantidadRepeticiones = 0; //iteramos hasta que el índice
        sea menor al tamaño de la cadena
        for (int i = 0 ; i < cadena.length() ; i++) {
            char aux = cadena.charAt(i); //obtenemos el carácter en la
            posicion i
            if (aux == car) {
                cantidadRepeticiones++;
            }
        }
        //retornamos la cantidad de repeticiones
        return cantidadRepeticiones;
    }
}
```

---



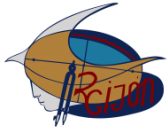
```
package tratamientosttring;
import java.util.Scanner;
/**
 *
 * @author Entornos de Desarrollo - Ruiz Gijón
 */
public class metodos2 {

    public static int cuentaVocales(){
        Scanner sc = new Scanner(System.in);
        String cad = sc.nextLine(); //leemos la cadena
        int cantidadVocales = 0;

        //iteramos mientras el índice sea menor al tamaño total de la cadena
        for (int i = 0; i < cad.length(); i++) {
            char car = cad.charAt(i); //obtenemos el carácter en la
            posición i
            //si el carácter es igual a "a", "e", "i", "o", ó "u" entonces es
            vocal
            if (car == 'a' || car == 'e' || car == 'i' || car == 'o'
            || car == 'u') {
                cantidadVocales++; //contamos cantidad vocales +1
            }
        }
        //devolvemos la cantidad de vocales
        return cantidadVocales;
    }

    public static String invierteCadena(){
        Scanner sc = new Scanner(System.in);
        String cadena = sc.nextLine(); //leemos la cadena
        String cadenaInvertida = "";
        //iteramos de forma invertida, partimos de la última posición valida
        //hasta la primera posición valida que el cero
        //vamos disminuyendo el valor de i en una unidad -1
        for (int i = cadena.length() - 1 ; i >= 0; i--) {
            char car = cadena.charAt(i); //obtenemos el carácter en la
            posición i
            cadenaInvertida = cadenaInvertida + car;
        }
        //devolvemos la cadena invertida
        return cadenaInvertida;
    }

}
```



**EJERCICIO 2. Diseña una clase Test que realice los siguientes test sobre el aula virtual <https://ciclos.iesruizgijon.es>:**

**1) Test loginNombre()**

Este test debe cargar la página en Mozilla Firefox la página y realizar las acciones necesarias de autenticación de tu usuario. Una vez has accedido con tu usuario, comprobará que el nombre que aparece en la parte superior derecha es tu nombre completo.

Para la localización de las cajas de texto vamos a usar un mecanismo de espera explícito de 2 segundos hasta que esté presente.

Para la localización de botones y enlaces vamos a usar un mecanismo de espera explícito de 3 segundos hasta que el elemento sea clickeable.

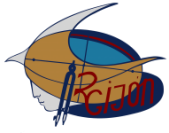
**2) Test datosModuloEntorno()**

Para este test vamos a cargar la página en Chrome y vamos a acceder con nuestro usuario. Una vez accedemos pulsamos sobre el enlace al curso de Entornos de Desarrollo y comprobaremos las siguientes condiciones:

- a. Título de la página es “Curso: Entornos de Desarrollo”.
- b. Encabezado del curso es “Entorno de Desarrollo”.

Para la localización de todos los elementos vamos a usar un mecanismo de espera implícito de 3 segundos.

Exporta en .zip el proyecto y súbelo en el enlace de entrega.



***IES RUIZ GIJÓN***

**CFGS DAW – Entorno de Desarrollo  
Examen Práctico Optimización y Pruebas del Software  
22 de Marzo 2021**