

CURSO
BACKEND 1

Programación Orientada a Objetos

Ejercicios

CLASES DE UTILIDADES

Ejercicios de aprendizaje

Los métodos disponibles para las clases de utilidad Integer, Arrays y Date están en esta guía. Recordar que la clase String y Math están explicadas en la guía anterior de Intro Java.



1. Realizar una clase llamada **Cadena**, en el paquete de entidades, que tenga como atributos una frase (String) y su longitud. Agregar constructor vacío y con atributo frase solamente. Agregar getters y setters. El constructor con frase como atributo debe setear la longitud de la frase de manera automática. Crear una clase **CadenaServicio** en el paquete servicios que implemente los siguientes métodos:

```
package guia9.ejer1;

import guia9.ejer1.Servicio.ServicioCadena;
import guia9.ejer1.entidades.Cadena;
import java.util.Scanner;

public class Guia9Ejer1 {

    public static void main(String[] args) {
        Scanner leer = new Scanner(System.in);
        ServicioCadena ss = new ServicioCadena();
        Cadena f1 = ss.pedirFrase();
        System.out.println(f1.toString());
        ss.mostrarVocales(f1);
        ss.invertirFrase(f1);
        System.out.println("Ingrese la letra que desea contabilizar");
        String letra = leer.nextLine();
        ss.vecesRepetido(f1, letra);
        ss.compararLongitud(f1);
        //Cadena f2 = ss.pedirFrase();
        //ss.compararLongitud(f1,f2);
        ss.unirFrase(f1);
        ss.reemplazar(f1);
        ss.contiene(f1);
    }
}
```

```
package guia9.ejer1.Servicio;

import guia9.ejer1.entidades.Cadena;
import java.util.Scanner;

public class ServicioCadena {

    public Cadena pedirFrase(){
        Scanner leer = new Scanner(System.in);
        System.out.println("Ingrese la frase");
        String frase = leer.nextLine();
        return new Cadena(frase);
    }

    public void mostrarVocales(Cadena f){
        int cantVocales = 0;
        for (int i = 0; i < f.getLongitud(); i++) {
            if (f.getFrase().substring(i,i+1).equalsIgnoreCase("A")) {
                cantVocales++;
            } else if (f.getFrase().substring(i,i+1).equalsIgnoreCase("E")) {
                cantVocales++;
            } else if (f.getFrase().substring(i,i+1).equalsIgnoreCase("I")) {
                cantVocales++;
            } else if (f.getFrase().substring(i,i+1).equalsIgnoreCase("O")) {
                cantVocales++;
            } else if (f.getFrase().substring(i,i+1).equalsIgnoreCase("U")) {
                cantVocales++;
            }
        }
        System.out.println("Cantidad de vocales: " + cantVocales);
    }

    public void invertirFrase(Cadena f){
        System.out.println("Frase invertida: ");
        for (int i = f.getLongitud()-1; i > -1; i--) {
            System.out.print(f.getFrase().substring(i,i+1));
        }
        System.out.println("");
    }

    public void vecesRepetido (Cadena f, String letra) {
        int aux = 0;
        for (int i = 0; i < f.getLongitud(); i++) {
            if (f.getFrase().substring(i,i+1).equalsIgnoreCase(letra)) {
                aux++;
            }
        }
        System.out.println("El caracter " + letra + " se repite " + aux + " veces");
    }

    public void compararLongitud(Cadena f){
        System.out.println("Ingrese la frase 2");
        Scanner leer = new Scanner(System.in);
        String frase2 = leer.nextLine();
        if (f.getLongitud() == frase2.length()) {
            System.out.println("La longitud de las frases son iguales");
        } else if (f.getLongitud() > frase2.length()) {
            System.out.println("La frase 1 es más extensa que la frase 2");
        } else {
            System.out.println("La frase 2 es más extensa que la frase 1");
        }
    }

    public void unirFrase(Cadena f){
        System.out.println("Ingrese la frase 2");
        Scanner leer = new Scanner(System.in);
        String frase2 = leer.nextLine();
        System.out.println(f.getFrase().concat(frase2));
    }

    public void reemplazar (Cadena f) {
        Scanner leer = new Scanner(System.in);
        System.out.println("Ingrese el caracter ");
        String caracter = leer.nextLine();
        String nvaFrase;
        nvaFrase = f.getFrase().replaceAll("a", caracter);
        System.out.println(nvaFrase);
    }

    public void contiene (Cadena f) {
        Scanner leer = new Scanner(System.in);
        System.out.println("Ingrese la letra ");
        String letra = leer.nextLine();
        System.out.println(f.getFrase().contains(letra));
    }
}
```

```
package guia9.ejer1.entidades;

public class Cadena {
    //que tenga como atributos una frase (String) y su longitud
    private String frase;
    private int longitud;
    //Agregar constructor vacío y con atributo frase solamente
    public Cadena() {
    }
    //El constructor con frase como atributo debe setear la longitud de la frase de manera automática.
    public Cadena(String frase) {
        this.frase = frase;
        this.longitud = frase.length();
    }
    //Agregar getters y setters.

    public String getFrase() {
        return frase;
    }

    public void setFrase(String frase) {
        this.frase = frase;
    }

    public int getLongitud() {
        return longitud;
    }

    public void setLongitud(int longitud) {
        this.longitud = longitud;
    }

    @Override
    public String toString() {
        return "Cadena{" + "frase=" + frase + ", longitud=" + longitud + "}";
    }
}
```

a) Método mostrarVocales(), deberá contabilizar la cantidad de vocales que tiene la frase ingresada.

b) Método invertirFrase(), deberá invertir la frase ingresada y mostrarla por pantalla. Por ejemplo: Entrada: "casa blanca", Salida: "acnalb asac".

c) Método vecesRepetido(String letra), recibirá un carácter ingresado por el usuario y contabilizar cuántas veces se repite el carácter en la frase, por ejemplo:

d) Entrada: frase = "casa blanca". Salida: El carácter 'a' se repite 4 veces.

e) Método compararLongitud(String frase), deberá comparar la longitud de la frase que compone la clase con otra nueva frase ingresada por el usuario.

f) Método unirFrase(String frase), deberá unir la frase contenida en la clase Cadena con una nueva frase ingresada por el usuario y mostrar la frase resultante.

g) Método reemplazar(String letra), deberá reemplazar todas las letras "a" que se encuentren en la frase, por algún otro carácter seleccionado por el usuario y mostrar la frase resultante.

h) Método contiene(String letra), deberá comprobar si la frase contiene una letra que ingresa el usuario y devuelve verdadero si la contiene y falso si no.

Método Static y Clase Math

- ✓ **2.** Realizar una clase llamada ParDeNumeros que tenga como atributos dos números reales con los cuales se realizarán diferentes operaciones matemáticas. La clase debe tener un constructor vacío, getters y setters. En el constructor vacío se usará el Math.random para generar los dos números. Crear una clase ParDeNumerosService, en el paquete Servicios, que deberá además implementar los siguientes métodos:

- a) Método mostrarValores que muestra cuáles son los dos números guardados.
- b) Método devolverMayor para retornar cuál de los dos atributos tiene el mayor valor
- c) Método calcularPotencia para calcular la potencia del mayor valor de la clase elevado al menor número. Previamente se deben redondear ambos valores.
- d) Método calculaRaiz, para calcular la raíz cuadrada del menor de los dos valores. Antes de calcular la raíz cuadrada se debe obtener el valor absoluto del número.

Clase Arrays

- ✓ **3.** Crear en el Main dos arreglos. El arreglo A de 50 números reales y el arreglo B de 20 números reales. Crear la clase ArregloService, en el paquete servicio, con los siguientes métodos:

- 1) Método inicializarA recibe un arreglo por parámetro y lo inicializa con números aleatorios.
- 2) Método mostrar recibe un arreglo por parámetro y lo muestra por pantalla.
- 3) Método ordenar recibe un arreglo por parámetro y lo ordena de mayor a menor.
- 4) Método inicializarB copia los primeros 10 números del arreglo A en el arreglo B. Luego llenar las últimas 10 posiciones del arreglo B con 0.5.

En el Main nuevamente: inicializar A, mostrar A, ordenar A, inicializar B, mostrar A y B.

Clase Date



4. Vamos a usar la clase Date que ya existe en Java. Crearemos la clase FechaService, en paquete Servicios, que tenga los siguientes métodos:

- a) Método fechaNacimiento que pregunte al usuario día, mes y año de su nacimiento. Luego los pase por parámetro a un nuevo objeto Date. El método debe retornar el objeto Date. Ejemplo fecha: `Date fecha = new Date(anio, mes, dia);`
- b) Método fechaActual que cree un objeto fecha con el día actual. Para esto usaremos el constructor vacío de la clase Date. Ejemplo: `Date fechaActual = new Date();`

El método debe retornar el objeto Date.
- c) Método diferencia que reciba las dos fechas por parámetro y retorna la diferencia de años entre una y otra (edad del usuario).

Si necesitan acá tienen más información en clase Date: [Documentacion Oracle](#)



5. Implemente la clase Persona en el paquete entidades. Una persona tiene un nombre y una fecha de nacimiento (Tipo Date), constructor vacío, constructor parametrizado, get y set. Crear una clase PersonaService, en el paquete servicio, con los siguientes métodos:

- a) Método crearPersona que pida al usuario Nombre y fecha de nacimiento de la persona a crear. Retornar el objeto Persona creado.
- b) Método calcularEdad que calcule la edad del usuario utilizando el atributo de fecha de nacimiento y la fecha actual.
- c) Método menorQue recibe como parámetro una Persona y una edad. Retorna true si la persona es menor que la edad consultada o false en caso contrario.
- d) Método mostrarPersona que muestra la información de la persona deseada.

Uso de vectores como atributos de clase



6. Un profesor particular está dando cursos para grupos de cinco alumnos y necesita un programa para organizar la información de cada curso. Para ello, crearemos una clase entidad llamada Curso, cuyos atributos serán: nombreCurso, cantidadHorasPorDia, cantidadDiasPorSemana, turno (mañana o tarde), precioPorHora y alumnos. Donde alumnos es un arreglo de tipo String de dimensión 5 (cinco), donde se alojarán los nombres de cada alumno. A continuación, se implementarán los siguientes métodos:

- a) Un constructor por defecto.
- b) Un constructor con todos los atributos como parámetro.
- c) Métodos getters y setters de cada atributo.
- d) Método cargarAlumnos(): este método le permitirá al usuario ingresar los alumnos que asisten a las clases. Nosotros nos encargaremos de almacenar esta información en un arreglo e iterar con un bucle, solicitando en cada repetición que se ingrese el nombre de cada alumno.
- e) Método crearCurso(): el método crear curso, le pide los valores de los atributos al usuario y después se le asignan a sus respectivos atributos para llenar el objeto Curso. En este método invocamos al método cargarAlumnos() para asignarle valor al atributo alumnos
- f) Método calcularGananciaSemanal(): este método se encarga de calcular la ganancia en una semana por curso. Para ello, se deben multiplicar la cantidad de horas por día, el precio por hora, la cantidad de alumnos y la cantidad de días a la semana que se repite el encuentro.