

## Ejercitación 1a

### Interfaces:

#### Ejercicio 1:

Crear una interfaz cuyo nombre sea `Imprimible`. Definir en esta interfaz el método `imprimir`, con la siguiente firma: **`public void imprimir()`**

Luego, crear las siguientes clases y hacer que cada una implemente la interfaz `Imprimible`. Además, definir el comportamiento del método `imprimir` en cada caso:

- Contrato: imprime por pantalla "Soy un contrato muy legal"
- Foto: imprime por pantalla "Soy una selfie"
- Documento: "Soy un documento de word".

Crear una clase `Impresora` que contenga una lista de imprimibles. Luego crear dos métodos en la clase impresora:

- **`public void imprimirTodo()`**. Este método será el encargado de recorrer la lista de imprimibles e imprimirlos.
- **`public void agregarImprimible(Imprimible unImprimible)`**. Este método será el encargado de agregar un imprimible a la lista de la impresora.

Crear en la clase `Main`:

- un contrato
- una foto
- documento
- impresora.

Agregar el contrato, la foto y el documento a la impresora. Pedirle a la impresora que imprima todo. Observar el comportamiento

## Ejercicio 2:

Crear una interfaz cuyo nombre sea Voleable. Definir en esta interfaz el método volar, con la siguiente firma:

**public void volar()**

Luego, crear las siguientes clases y hacer que cada una implemente la interfaz Voleable. Además, definir el comportamiento del método volar en cada caso:

- Pato: un pato tiene una energía. Cada vez que el pato vuela pierde 5 unidades de energía e imprime por pantalla "Estoy volando como un pato"
- Avión: un avión tiene horas de vuelo. Cada vez que el avión vuela incrementa 13 hs de vuelo e imprime por pantalla "Estoy volando como un avión."
- Superman: superman tiene experiencia. Cada vez que superman vuela aumenta 3 unidades su experiencia e imprime por pantalla "Estoy volando como un campeón"

Crear una clase TorreDeControl que contenga una lista de volables. Luego crear dos métodos en la clase TorreDeControl:

- **public void vuelenTodos()**. Este método será el encargado de recorrer la lista de volables y hacerlos volar.
- **public void agregarVolador(Voleable unVoleable)**. Este método será el encargado de agregar un volable a la lista de voladores.

Crear en la clase Main:

- un pato
- un avion
- un superman
- una torre de control.

Agregar el pato, el avion y superman a la torre de control. Pedirle a la torre de control que haga volar a todos los volables. Observar el comportamiento

Continúa en la próxima página ---->

## Colecciones:

### Ejercicio 3:

Definir e inicializar una nueva lista vacía. Agregar a la lista vacía los siguientes elementos: 1,5,5,6,7,8,8,8.  
Imprimir el resultado por pantalla.

### Ejercicio 4:

Definir e inicializar un nuevo conjunto vacío. Agregar al conjunto vacío los siguientes elementos: 1,5,5,6,7,8,8,8.  
Imprimir el resultado por pantalla. ¿Qué diferencia existe con el ejercicio anterior?

### Ejercicio 5:

Definir e inicializar una nueva lista vacía. Agregar a la lista vacía los primeros cien números enteros positivos.  
Imprimir el resultado por pantalla.

### Ejercicio 6:

Definir e inicializar una nuevo conjunto vacío. Agregar al conjunto vacío cien veces el número 1.  
Imprimir el resultado por pantalla.

### Ejercicio 7:

Crear una nueva clase llamada Prueba. Definir e implementar el siguiente método:

```
public void imprimirPorPantalla(List<String> nombres)
```

El método debe recorrer la lista de nombres e imprimir por pantalla cada nombre agregando delante del nombre "El nombre del alumno es: ".

### Ejercicio 8:

Crear una nueva clase llamada Prueba. Definir e implementar el siguiente método:

```
public void sumaTotal(Set<Integer> conjuntoDeEnteros)
```

El método debe recorrer el conjunto de enteros, ir acumulando el total de la suma de los valores y al finalizar imprimir por pantalla el valor de la suma total.