

1.1.- Escribe el código en Java de una clase para representar un empleado de una empresa. Del empleado vamos a querer tener los datos de su nombre, apellido, edad y salario.

Realiza dos métodos constructores, uno con los valores de los cuatro atributos y otro sin atributos.

Crea también los métodos que devuelvan los valores de los atributos y otros que permitan modificarlos, así como un método que si el empleado tiene más de 40 años se le aumente el sueldo una cantidad que se pasa por parámetro. A este último método le llamaremos comprobarPlus y devuelve true si se aumenta el sueldo y falso si no se aumenta.

Se debe crear también un método toString() para imprimir un objeto Empleado.

Utiliza la clase JOptionPane para recibir los valores del usuario para ir creando los empleados. Para ello, crearás un método denominado **leerEmpleado()** en la misma clase donde tengamos el método main(), este método pide los datos de un empleado y devuelve un objeto de la clase Empleado.

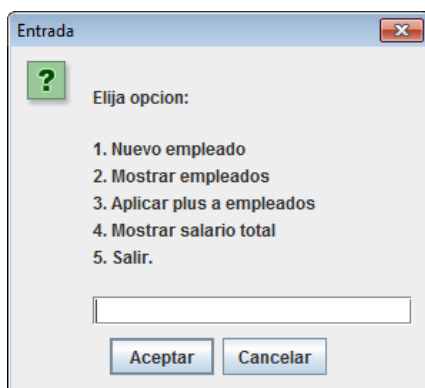
Crea tres empleados para probar los métodos descritos.

1.2.- Haz el ejercicio creando un array de empleados, donde se puedan acumular hasta 100 empleados (usa menos para probar el programa). Crea los empleados guardándolos en el array, para posteriormente comprobar si hay que aumentar el plus a los empleados. Muestra los datos de cada empleado del array.

Muestra la suma del salario de todos los empleados antes y después de realizar el aumento que corresponda.

Haz el ejercicio con JOptionPane.

1.3.- Haz el ejercicio ahora con un menú como el siguiente:



Para ir creando los empleados usarás la **1ª opción** y un **atributo estático** que va contando los empleados según se van creando en el/los constructor/es.

En la **2ª opción** mostrarás todos los empleados creados hasta el momento en que se selecciona esa opción

En la **3ª opción**, se recorre el array y se sube la cantidad indicada como plus a los mayores de 40 años. Fíjate en que si se elige varias veces esta opción, habrá empleados a los que se les aplique la subida más de una vez. Busca una solución a este problema.

En la **4ª opción**, se muestra la suma de todos los empleados creados hasta el momento en que se selecciona la opción.

En esta versión del programa, deberás crear métodos estáticos en el fichero fuente del main, que reciban el array de empleados y lleven a cabo lo que se pide en cada opción. Es decir, **en el menú aparecerán solo las llamadas a los métodos**.

Si el usuario selecciona alguna opción y no hay empleados, el programa deberá notificarlo.

2.- Implementa una clase en Java que permita realizar promedios. La clase debe tener dos métodos, uno para ingresar un nuevo número, llamado *agregarNumero(int numero)* y otro para obtener el promedio hasta el momento, llamado *obtenerPromedio()*. Determina qué atributos son necesarios para implementarla.

Implementa luego una clase de prueba que permita introducir algunos valores y que muestre el promedio.

Realiza el ejercicio con un menú que muestre las opciones:

- 1 Agregar número
- 2 Obtener promedio hasta el momento
- 3 Ver los datos introducidos
- 4 Salir

3.- Implementa una clase en Java que represente una fracción de números enteros. Implementa asimismo los siguientes métodos en la clase Fracción:

- Suma de dos fracciones, que será una nueva fracción. Resta de dos fracciones, similar a la suma.
- División de dos fracciones.
- Multiplicación de dos fracciones.
- Cálculo de la inversa de una fracción (cambiar numerador por denominador y viceversa).

Todos los métodos se realizan sobre un objeto Fracción pasándole por parámetro la segunda fracción cuando sea necesario.

Intenta simplificar los resultados usando un nuevo método.

4.- Realiza un programa en Java que, mediante un menú con varias opciones haga las siguientes tareas (hazlo con Scanner y con la clase JOptionPane):

Opción 1: Pide los datos de un alumno (nombre, apellido, curso, nota) y da de alta al alumno en nuestro programa.

Opción 2: Muestra los datos de los alumnos cuya nota media es mayor o igual a 5 y el número de ellos que hay que cumplan esa media.

Opción 3: Muestra los datos de los alumnos cuya nota media es menor a 5 y el número de ellos que hay que cumplan esa media.

Utiliza funciones (métodos estáticos) para llevar a cabo las tareas pedidas.

5.- Crea una clase en Java llamada Cubo con dos atributos:

```
int capacidad; // capacidad máxima en litros
int contenido; // contenido actual en litros
```

La clase tendrá un constructor que recibe la capacidad inicial del cubo. Una vez establecida la capacidad, ya no será posible modificarla.

Habrà un método llamado **llena()**, que llenará el cubo hasta su capacidad máxima y otro **vacia()** que pone el contenido a 0 litros.

Haz un método llamado **vuelcaEn()** que vuelca el contenido de un cubo sobre otro. Antes de echar el agua se comprueba cuánto le cabe al cubo destino (será el que se manda por parámetro (void vuelcaEn(Cubo destino)).

Se pide también un método llamado **pinta()** que pinta un cubo en la pantalla. Se muestran los bordes del cubo con el carácter # y el agua que contiene con el carácter ~. Cada litro se representa con una línea.

Para probar el ejercicio, ve haciendo las sentencias correspondientes a las siguientes salidas:

Cubo pequeño:

```
#   #
#   #
#####
```

Cubo grande:

```
#   #
#   #
#   #
#   #
#   #
#   #
#   #
#####
```

Lleno el Cubo pequeño:

```
#~::~~#
#~::~~#
#####
```

El Cubo grande sigue vacio:

```
#      #
#      #
#      #
#      #
#      #
#      #
#      #
#####
```

Ahora vuelco lo que tiene el Cubo pequeño en el Cubo grande.

Cubo pequeño:

```
#      #
#      #
#####
```

Cubo grande:

```
#      #
#      #
#      #
#      #
#      #
#~::~~#
#~::~~#
#####
```

Ahora vuelco lo que tiene el Cubo grande en el Cubo pequeño.

Cubo pequeño:

```
#~::~~#
#~::~~#
#####
```

Cubo grande:

```
#      #
#      #
#      #
#      #
#      #
#      #
#      #
#####
```