

Examen Programación 03/03/2023

A lo largo del examen, sigue los siguientes criterios para nombrar los proyectos y ficheros:

- La entrega final será un único archivo comprimido con el nombre **Examen3323TuNombreTuApellido**.
- Cada ejercicio se realizará en un proyecto independiente cuyo nombre será **Ejercicio 1, Ejercicio 2, Ejercicio 3 y Ejercicio 4**.
- Para cada ejercicio deberás entregar el **proyecto completo** de NetBeans de cada ejercicio en un archivo comprimido y después incluirlos en la entrega final.
- En cada ejercicio se nombrará las clases y los paquetes como se indique, respetando las mayúsculas. En caso de que no se especifique ningún nombre, se nombrará como se considere.
- Se valorará el orden y legibilidad del código.

Ejercicio 1 (2.5 puntos)

Crea un proyecto con las siguientes clases dentro del paquete ejercicio1:

- Clase Personal:
 - Sus atributos privados son dni (String) y salario (int).
 - Incluye el constructor por defecto y con parámetros.
 - Incluye los métodos set y get que consideres necesarios.
- Clase Gestion:
 - Incluye un atributo privado que será una colección de tipo LinkedList donde se almacenarán objetos de la clase Personal para gestionar la información de los empleados.
 - Incluye un constructor por defecto que inicializa la colección.
 - Incluye un método inserta_empleado() que recibe como parámetros su dni y su salario, crea un objeto de la clase Personal con ese dni y ese salario y lo añade al LinkedList. El método debe comprobar si hay un empleado almacenado con ese dni, en cuyo caso no se podrá añadir y se mostrará un mensaje por pantalla. Utiliza un for para recorrer la colección.
 - Incluye un método borra_empleado() que recibe como parámetro el dni del empleado que se quiere borrar, comprueba si está almacenado en el LinkedList y lo elimina. El método debe comprobar si el empleado está en la colección para poder borrarlo y si no está no se podrá borrar y se mostrará un mensaje por pantalla. Utiliza un for-each para recorrer la colección.
 - Incluye un método lista_empleados() que imprime por pantalla el dni y el salario de todos los empleados incluidos en la colección (LinkedList). Utiliza un iterador para recorrer la colección.

- Clase Principal:
 - Incluye un método main donde se prueban los tres métodos de la clase Gestion. Añade 5 objetos de la clase Personal con el método inserta_empleado() a un objeto de la clase Gestion, borra uno de ellos con borra_empleado() y muestra los 4 empleados restantes con lista_empleado().

Ejercicio 2 (2.5 puntos)

- Crea un proyecto que imprima por pantalla los siguientes caracteres utilizando un array bidimensional:

```

+ - + - +
+ - + - +
+ - + - +
+ - + - +

```

Ejercicio 3 (3 puntos)

- Escribe un programa que genere un array de una dimensión que contenga 30 números aleatorios enteros entre 0 y 9 (ambos incluidos). Se debe generar un HashMap de tal forma que sus claves sean los números del 0 al 9 y sus valores la cantidad de veces que cada número aparece en el array. Si un número del 0 al 9 no aparece en el array no debe incluirse en el mapa. El programa debe imprimir por pantalla el array y el HashMap. Una vez se hayan imprimido por pantalla ambos se deberán visualizar después los valores y las claves del mapa por separado. Para ello el programa debe recorrer el mapa con un bucle for-each y utilizar durante el recorrido los métodos values() y keySet() respectivamente.

Ejercicio 4 (2 puntos)

- Escriba un programa Java que cree un ArrayList de objetos de tipo String y que permita al usuario introducir por teclado nuevos elementos en el ArrayList. El programa deberá imprimir por pantalla el ArrayList completo cada vez que se añada un nuevo elemento. Si el usuario introduce la palabra "fin", el programa deberá terminar. Debes considerar que el usuario puede introducir la palabra para terminar en mayúsculas o minúsculas y en ambos casos debe terminar.