

Ejercicios adicionales tema 5

1.- Definir un array de 10 posiciones de tipo entero y realizar las siguientes operaciones:

- Inicializar todas las posiciones del array a -2 a la hora de definirlo.
- Utilizando un FOR:
 - o Inicializar todas las posiciones a valor -1. Visualizar el resultado.
 - o Inicializar cada posición con un valor distinto. Visualizar el resultado.

2.- Realizar un programa que lea por teclado la temperatura de 10 ciudades y muestre la temperatura media de todas ellas.

3.- Programa que indique la posición del array donde se encuentra un número dado por teclado. En caso de no encontrarlo, visualizar el mensaje "Número no encontrado". Por ejemplo, crear un array con 10 elementos del 1 al 10, preguntar por el número y si el número introducido está comprendido entre 1 y 10 lo encuentra e indica la posición.

4.- Crear un array con 10 elementos e invertirlo en otro array. Muestra el array invertido.

5.- Crear un array con 10 números e invertirlo en el mismo array. Muestra el array obtenido.

6.- Diseñar una clase llamada Bonoloto que tiene como atributo privado un array numérico entero de 6 elementos y los siguientes métodos:

- Un constructor que inicializa el array con 6 números aleatorios distintos comprendidos entre 1 y 49.
- Un método llamado mostrar que devuelva una cadena de texto que muestre el contenido del array.

Nota: No se pueden repetir los números.

7.- Realizar una clase que se componga de:

- Un array como atributo de cinco elementos de tipo entero
- Constructor por defecto que solicite los valores correspondientes a cada elemento.

Y al menos los siguientes métodos:

- Método que multiplique cada elemento por dos.
- Método que muestre el contenido del array por pantalla
- Método que devuelva la suma de todos los elementos del array que se encuentran en posiciones pares.

En el programa principal, prueba todos los métodos desarrollados.

8.- Realizar una clase que se componga:

- Un array de cinco elementos de tipo entero.
- Constructor que solicite al usuario los valores correspondientes a cada elemento.
- Método que devuelva el valor menor del array.
- Método que devuelva el valor mayor del array.
- Método que devuelva la media de todos los valores del array.

Haz un programa principal que pruebe todos los métodos probados.

9.- Se pretende diseñar una clase llamada Invierno que tiene como atributo privado un array de temperaturas (double) en grados Celsius y como métodos públicos:

- Un constructor que recibe como argumento un número n, dimensiona el array con n elementos y pide las n temperaturas por teclado
- Un método llamado temp_media que nos devuelve la temperatura media.
- Un método llamado fahrenheit que devuelve un nuevo array con todas las temperaturas en grados Fahrenheit. Para pasar temperaturas de grados Celsius a Fahrenheit utilizar la fórmula: $F = C * 1.8 + 32$;
- Un método llamado mostrar que devuelve una cadena que contiene los valores de un array. Como parámetro de entrada al método es el array.

Desde una función principal declarar objetos de la clase Invierno y probar los métodos de la clase. Por ejemplo, si declaramos un objeto Invierno y le pasamos como argumento 5, el constructor dimensiona el array con 5 elementos y nos pide las 5 temperaturas.

10.- Diseñar una clase llamada Alumno con los atributos privados: nombre del alumno y un array con las 5 notas enteras de ese alumno y los siguientes métodos públicos:

- Un constructor que dimensiona el array e inicialice los atributos con los valores introducidos por teclado y el nombre se pasa como argumento de entrada.
- Método llamado mayor que devuelve la mejor nota.
- Método llamado menor que devuelve la menor de las notas .
- Método llamado mostrar que devuelve una cadena donde se muestre el nombre del alumno, en mayúsculas, seguido de su nota media.

Desde la clase principal declarar un objeto de tipo alumno y probar los métodos de la clase.

11.- Crear en la clase Principal el array multidimensional:

int

```
matriz1[][]={{101,102,103,104,105},{201,202,203,204,205},{301,302,303,304,305},{401,402,403,404,405},{501,502,503,504,505}};
```

Realizar un menú repetitivo con las siguientes seis acciones:

- 1.-Elegir fila. Pide el número de fila por teclado y muestra elementos de esa fila. Comprueba que el número de fila sea válido.
- 2.- Elegir columna. Pide el número de columna por teclado y muestra los elementos de esa columna. Comprueba que el número de columna sea válido.
- 3.- Mostrar los elementos de la diagonal.
- 4.- Mostrar los elementos de la diagonal inversa.
- 5.- Mostrar los elementos de la diagonal hacia arriba.
- 6.- Salir.

12.- Diseñar una clase Tabla que tenga como elementos privados: un array bidimensional de 4 por 5 y los siguientes métodos:

- Un constructor que inicializa el array con valores aleatorios del 1 al 100.
- Un método llamado mostrar que muestra los valores del array.
- Un método llamado mayor_fila que recibe como argumento una fila y devuelve el mayor de los elementos de esa fila.
- Un método llamado mayor_columna que recibe como argumento una columna y devuelve el mayor elemento de esa columna.

- Un método llamado **media** que devuelve la media de todos los valores de la matriz.

Probar los métodos de la clase en la principal.

13.- Diseñar una clase llamada **Figura** que tiene como atributos:

- un array bidimensional de tipo char llamado **matriz** .
- un carácter llamado **letra**.

Y los siguientes métodos:

- Un **constructor** que recibe como argumento un carácter y un número entero **n** inicializa y dimensiona el array con los valores **n X n**.
- Un método llamado **cuadrado** que devuelve una cadena de texto que dibuja un cuadrado relleno de **n X n** con el carácter especificado en **letra** y rellena **matriz**.
- Un método llamado **cuadrado_vacio** que dibuja un cuadrado vacío (solo los laterales)
- Un método llamado **aspa** que dibuja un aspa (X)

14.- Diseñar una clase llamada **Persona** con los atributos privados **nombre** y **teléfono**, un **constructor** que permite inicializar los atributos con los datos pasados como argumentos y métodos que devuelven el **nombre** y el **teléfono**. Diseñar también una clase llamada **Agenda** que tiene como miembros privados un array de personas y un contador de personas que vamos metiendo en la agenda, un constructor que pone el contador a 0 y dimensiona el array al tamaño que le pasamos como argumento, un método **añadir** que permite añadir una persona a la agenda (controlar que no meta más personas que las que caben en la agenda), un método **mostrar** que lista la información de la agenda, un método **buscar** que pasado un nombre como argumento visualiza los datos de esa persona de la agenda y un método **modificar** que pasado un nombre como argumento si lo encuentra muestra el teléfono actual y pide que introduzca un nuevo teléfono modificando el dato en la agenda.

- Crea en la clase principal un menú repetitivo con las siguientes opciones 1. Nuevo contacto, 2. Mostrar la agenda, 3. Buscar contacto, pide el nombre de la persona a buscar y si la encuentra muestra su información, 4. Ver el número de contactos de la agenda, 5. Modificar teléfono de una agenda, pide el nombre de la persona a cambiar el teléfono y la encuentra se cambia el teléfono, 6. Salir.
- PD. Como sólo se busca por nombre es posible que tengamos más de una persona a mostrar o a modificar.

15.- Calcular la suma y la media aritmética de los valores contenidos en un ArrayList. Programa que lea una serie de valores numéricos enteros desde el teclado (las entradas de datos se realizan utilizando una clase Teclado con métodos estático para validar dichas entradas), y los guarde en un ArrayList de tipo Integer. La lectura de números termina cuando se introduzca el valor -99. Este valor no se guarda en el ArrayList.

A continuación, el programa mostrará por pantalla el número de valores que se han leído, su suma y su media. Por último, se mostrarán todos los valores leídos, indicando cuántos de ellos son mayores que la media.

Vamos a utilizar 3 métodos además del método main para resolverlo:

- Método `lee_valores`: pide por teclado los números y los almacena en el `ArrayList`. La lectura acaba cuando se introduce el valor -99. El método devuelve mediante `return` el `ArrayList` con los valores introducidos.
- Método `calcular_suma`: Recibe como parámetro el `ArrayList` con los valores numéricos y calcula y devuelve su suma. En este método se utiliza un `Iterator` para recorrer el `ArrayList`.
- Método `mostrar_resultados`: Recibe como parámetro el `ArrayList`, la suma y la media aritmética. Muestra por pantalla todos los valores, su suma y su media y calcula y muestra cuantos números son superiores a la media. En este método se utiliza un `for each` para recorrer el `ArrayList`.

16.- Se desea almacenar la información de Partidos de futbol que se celebran en una jornada. La información es:

- `equipoLocal`
- `equipoVisitante`
- `golesLocal`
- `golesVisitante`

El programa deberá realizar las siguientes acciones:

- Almacenar los datos de los partidos de fútbol.
- Indicar el número de partidos almacenados en el `Arraylist`.
- Mostar por pantalla todos los partidos almacenados.
- Elimina los partidos de futbol cuyo resultado no sea empate.
- Imprime de nuevo los partidos que te han quedado.

17.- Realizar un proyecto que permita manejar una lista (`LinkedList`) de los cursos que imparte una academia.

Crear una clase `Curso` (código, descripción, `n_horas`) con dos constructores uno que pida todos los datos por teclado y otro que los recibe como argumentos y los métodos `get` y `set` que estimes necesarios.

Crear una clase Academia que tenga como elemento privado una lista de tipo Curso y los métodos necesarios para llevar a cabo las tareas que presentaremos en el menú que ejecutaremos desde la clase Principal:

- Añadir un nuevo curso al final de la lista.
- Insertar un nuevo curso en cualquier posición.
- Borrar un curso dado su código.
- Modificar el número de Horas de un curso dado su código.
- Mostrar todos los cursos.
- Salir.

18.- Crear un programa que contenga un HashMap con 5 animales, cuyas claves serán números del 1 al 5. Se deberá pedir por teclado al usuario el nombre de un animal y el programa comprobará si están o no incluidos en la colección y nos lo mostrará por pantalla.