

Nombre: _____ Grupo: S1DAM

1. Dado un array de números enteros, realiza un programa que crea dos arrays con los números pares e impares.

Ejemplo: Tenemos el array A

0	4	7	3	5	-1	10	10	20
---	---	---	---	---	----	----	----	----

Debe generar los arrays

Pares

0	4	10	10	20
---	---	----	----	----

Impares

7	3	5	-1
---	---	---	----

No sirve con imprimir los valores en pantalla, es necesario generar los dos arrays. Se valorará la eficiencia de la solución.

2. La DGT necesita una aplicación para poder almacenar la ingente cantidad de multas que está imponiendo. Se necesita una clase `AlmacenDeMultas` donde se almacenen todas las sanciones. Esta clase debe tener los métodos necesarios para poder gestionar toda esta cantidad de sanciones.

Las sanciones pueden ser de varios tipos, las hay leves, graves o muy graves. Las sanciones leves tienen una cuantía económica, las graves tienen además de la cuantía económica, una pérdida de puntos, y las muy graves añaden un tercer atributo que son los meses de retirada del carnet.

Todas las sanciones en general deben contener:

- Día y Hora en la que se efectuó la sanción.
- Dni del conductor.
- Matrícula del vehículo.
- Descripción de la sanción.

Se pide:

- **Realizar la clase `AlmacenDeMultas` que almacene todas las sanciones. Esta clase debe contener**
 1. Los métodos necesarios para el buen funcionamiento de la clase (añadir, get, ...).
 2. Un método *obtenerSanciones(matricula)* al que se le pasa una matrícula y nos retorna todas las sanciones que tiene ese vehículo.

3. Un método *puntosRestados()*, que retorne la cantidad de puntos totales que la DGT ha restado a todos los conductores.(Tened en cuenta que no todas las sanciones conllevan resta de puntos)

- Diseñar la jerarquía de clases para las sanciones.

3. Un pasatiempo clásico es la “sopa de letras”. Se trata de un array de dos dimensiones donde hay una letra almacenada en cada hueco.

El juego consiste en descubrir donde está colocada una palabra. Puede estar en una fila o en una columna.

Realiza un método llamado *buscaPalabra()*, que recibe un array de letras y la palabra a buscar, y retorna como resultado en que fila o que columna está la palabra. El resultado será “F3” si ha encontrado la palabra en la fila 3, o bien retornará “C1” si ha encontrado la palabra en la columna 1. Si no encuentra la palabra retorna “NE” (no encontrado).

Ejemplo: Dado el siguiente array llamado *sopa*

A	N	Z	E	T
D	F	R	M	A
R	A	U	C	F
I	S	T	O	E
O	C	A	S	A

Si al método le pedimos buscar una palabra dará el siguiente resultado:

buscaPalabra(sopa, "CASA") retornará “F4”

buscaPalabra(sopa, "RUTA") retornará “C2”

buscaPalabra(sopa, "DIA") retornará “NE”

Nota: Suponemos que las palabras están siempre al derecho, nunca al revés.

Es mejor que conviertas cada fila o cada columna a un String y luego busques dentro la palabra con el método *indexOf()*.

4. Programa la siguiente ventana para calcular cuánto nos va a costar un taxi.

Calcula el presupuesto de tu Taxi

Km de distancia:

☒ Ida y Vuelta Total : 13,80 €

☒ Maletas (suplemento del 15%)

Precio por Km (0.30 €)

La ventana funciona tecleando una cantidad de kilómetros a recorrer, y al pulsar sobre calcular nos da el precio de cuanto nos va a costar. Si pulsamos sobre los check de Ida y Vuelta(el precio se duplica) o sobre Maletas(se añade un 15% al precio).

Indicaciones:

- Se dispone de 2 horas (desde las 11:30 hasta las 13:20). Los errores disminuyen la calificación y los errores graves la determinan.
- Al finalizar el examen se debe entregar una copia de la carpeta del proyecto java con las soluciones a los ejercicios propuestos. Este proyecto java debe ser nombrado de la siguiente manera: "Recup_15Abril_NombreAlumno".
- La carpeta del proyecto la debéis comprimir y subirla al espacio de classroom habilitado.

Puntuación Recuperación:

Ejercicio	1	2
Puntos	5	5

Aprobados:

Ejercicio	3	4
Puntos	5	5