

## Práctica 4. Arreglos.

### ① Catálogo de Juegos de Mesa.

El programa debe de:

- Registrar juegos de mesa (nombre, género, calif.)
- Ordenar los datos
- Ver mejor y peor calificadas.

Creación de Clases: JuegoDeMesa, ListaDeJuegos, CatálogoDeJuegos.

#### JuegoDeMesa.

Nombre (String)

Género (String)

Calificación (double)

Constructores: JuegoDeMesa: nombre, género, calif.

Observadores: obtener Nombre  
obtener Género  
obtener Calificación

Modificadores: asignar Nombre  
asignar Género  
asignar Calificación

Especiales: toString

#### ListaDeJuegos

Lista JuegoDeMesa

Contador (int)

Constructores: ListaDeJuegos

Observadores: manejar Juegos

Modificadores: agregar Juegos

Calculadores: ordenar -x nombre  
└ x género  
└ x calificación

Mejor y Peor

└ Observador: obtener mejor  
obtener peor

Especiales: toString.

#### CatálogoDeJuegos

↳ Programa → recibe int. Scanner

public class CatálogoDeJuego

public static void main (String [] args)

new Scanner

Agregar new ListaDeJuegos.

↳ Flujo → Bienvenida → opciones → registrar

└ mostrar

↳ Registro → nombre, género, calificación

↳ Mostrar → Todos, ordenados x nombre, ord. x género, ord. x cal

Mejor

Peor

↳ Terminar.

### ② Memerma

El programa debe de:

- Tener tablero 4x4 8 posos de simbolos. (ultimo en color)
- Flujo del Juego

Creación de Clases: Tablero, Memerma.

#### Memerma

↳ Programa. - recibe int. Scanner

public class Memerma

public static void main (String [] args)

new Scanner

new Tablero (4x4)

numero (int)

el juegoturno (boolean)

Flujo → Bienvenida, indicaciones

↳ Tablero abierto, seleccion 1

↳ Imprimir tablero

seleccion 2

↳ Imprimir tablero (ambas selecciones)

↳ Coordinadora regresa seleccion 1

↳ no coincide → cierra tablero → seleccion 1

↳ al final de encender todos los posgos → Memerma

↳ FIN

#### Tablero

Tablero (char/String)

Centros desubstratos o no (boolean)

Filas (int)

Columnas (int)

Constructores: tablero (fil y col)

Observadores: mostrar tablero

Modificadores: cambiar / desubstar

Calculadores: simbolos okulos

Especiales: toString

\* Tablero inicial

## ⑥ Batalla Naval.

Programa para jugar Batalla Naval

↳ Tablero 10 x 10 y para usuario otro para IA

↳ 1 barco (long 4) 2 barcos (long 3) 1 barco (long 2)

↳ IA - barcos aleatorios

Usuario - selección manual

↳ 1er turno IA

↳ Flyo del Juego.

Crecerán de clases:

### Barco

length (int)

Fila (int)

columna (int)

Posición H/V (barco)

impuesto (int)

Constructores: Barco

(file, col, horizontal, impuesto)

Observadores: distors file, col ...

Estado (Hundido / no)

modificadores (agregar impuesto)

toString(): to String.

### Tablero

tableros (char)

barcos (lista)

files (int)

cañones (int)

Constructores: Tablero new

Barcos new

Observadores: marcar tableros

eliminar barcos

modificadores: posicionar barcos

(aleatorio para IA

mano para usuario)

reabrir impuesto

Destrucción: eliminar impuestos

especial: toString().

## Batalla Naval

↳ Programa

Scanner y scanner

public class BatallaNaval {

public static void main (String [] args) {

Inicio → Bienvenida → Indicaciones:

↳ Tableros → aleatorio para IA,

↳ ingreso de barcos por usuario.

↳ se muestra tablero del jugador y tablero oculto/IA

↳ Turno aleatorio de inicio del juego.

↳ Turno Usuario

↳ marcar tableros

↳ sacar coordenadas

Turno IA

sacar coordenadas

↳ indicar si se acerto o no

↳ Alquilar todos los barcos

↳ Fin del Juego

↳ Ganador o Perdedor.

Juego del  
Juego