



PRACTICA 1. PROGRAMACIÓN EN JAVA Y SERIALIZACIÓN

OBJETIVO:

En esta práctica tomaremos contacto con

- Aprender/Refrescar los conceptos básicos de Java
- Almacenar información en disco usando la Serialización que proporciona Java

DOCUMENTACIÓN:

Tutoriales de Java en la Web y en la página oficial de Java:

<http://www.oracle.com/technetwork/es/java/javase/documentation/index.html>

<http://zarza.usal.es/~fgarcia/doc/tuto2/Index.htm>

DURACIÓN: 3 sesiones

FECHA TOPE DE ENTREGA Y DEFENSA:

- Eval Continua: 10 de Octubre en clase de prácticas
- Eval. No Continua: 21-Nov. Previa solicitud cita en horario de tutorías

FORMA DE ENTREGA:

- Subir los archivos de código fuente (.java) a la plataforma.

REQUISITOS PARA EL DESARROLLO

- Instalación JRE
- Herramienta de Desarrollo Java (Eclipse, Netbeans, ...)

ENUNCIADO:

Queremos desarrollar una sencilla aplicación para gestionar los datos de los partidos de fútbol de Primera División. Para ello necesitamos guardar la siguiente información:

- Personas: - Jugadores -Arbitros
- Equipos
- Partidos
- Estadios

CLASE Personas
+ int id; + String nombre; - String email - String Tlf
+ Personas ([int id], String nombre, [...]) + setEmail (...) + setTlf (...) + getEmail (...) + getTlf (...) ...



CLASE Jugador EXTENDS Personas

-int salario
-String posicion
-Boolean titular
-int num

+ Jugador([int id], String posición,[...])

...

CLASE Arbitro EXTENDS Personas

-String tipo

+ Arbitro([...])

...

CLASE Estadio

-int idestadio
-String Direccion
-String Ciudad
-int Capacidad

+ Estadio([int idestadio],[...])

...

CLASE Equipo

-int idequipo;
-Estadio esta
-int posicion; //posición que está en la liga
+Collection <Jugador> ljug;

+ Equipo (int ideq, Estadio esta)

+ AltaJugador (Jugador j)

+ BajaJugador (Jugador j o String id)

...

CLASE Partido

- int idpartido
- Estadio e
- Fecha o String x
- Equipo eq1
- Equipo eq2
- Boolean ida
- Collection (<Arbitro>) larbitros
- Collection (<Jugador>) leq1
- Collection (<Jugador>) leq2
- int golesA
- int golesB

+ Partido([...])

...



CLASE AppFutbol

```
-Map<Integer, Equipo> lEquipos ; //el integer será el idequipo
-Map<Integer, Jugador> lJugadores; //Integer será idjugador
-Map<Integer, Arbitro> lArbitros ; // ..igual
-Map<Integer, Estadio> lEstadios ; //..igual
-Collection<Partidos> lpartidos;

+ AppFutbol () //Aquí se pueden cargar los datos o en un nuevo método
+ Equipo altaEquipo(..)
+ boolean bajaEquipo(..)
+ Jugador altaJugador(..Equipo e.) //Se da de alta en un equipo y si no está en el
sistema también
+ boolean bajaJugador() // de un equipo, no del sistema
+ Arbitro altaArbitro()
+ boolean bajaArbitro()
+ Arbitro altaEstadio() //del sistema
+ Partido altaPartido()
+ boolean bajaPartido()

+ void listarEstadios()
+ void listarEquipos()
+ void listarArbitros()
+ int ContarPartidos()
+ void listarPartidos(dada una fecha) //devuelve info del partido
+ void listarPartidos(dado un equipo) //devuelve partidos y fechas
+ void listarJugadores(..dada una posición en el campo..)
+ void listarJugadoresEquipo( ..dado un equipo..)

+ public void Salvar()
+ public void CargarDatos()

+ void CalcularCampeonTemporada() **OPCIONAL**
+ void CalcularPosicionesEquipos(lequipos) **OPCIONAL**
```

CLASE AppFutbolMenu

```
public static String[] posiciones= {"Portero","Defensa","Medio","Ataque"} //Sugerencia
String[] posicionesarbitro = {"Principal","Asistente"} //sugerencia

void main(){
    Repetir menú hasta salir (pulsar 0) - (menú en la línea de comandos)
    Menu: Introduzca Opcion:
        1: Alta Equipo
        2: Baja Equipo
        3: Alta Jugador
        4: Baja Jugador
        5: Alta Arbitro
        6: Baja Arbitro
        7: Alta Estadio
        8: Alta Partido
        9: Baja Partido
        10: Listar Equipos
        11: Listar Estadios
        12: Listar Árbitros
```



```
13: Devolver Total de Partidos
14: Listar Información de Partidos dada una fecha
15: Listar los partidos hechos por un equipo
16: Listar los jugadores que hay de una posicion
17: Listar los jugadores de un equipo y sus posiciones
18: Cargar Sistema
19: Salvar los datos

20: Calcular el campeón OPCIONAL
21: Calcular posiciones OPCIONAL
```

```
}
```

SE PIDE:

Dado el esquema de clases anterior, desarrollar un programa Java que utilizando un menú con las opciones indicadas en la clase AppFutbolMenú. Teniendo en cuenta que:

- Se deben incluir TODAS las clases anteriormente especificadas
- Deben estar todas definidas en un mismo paquete
- Las listas que hemos indicado como Map <>, se deben implementar como HashMap o TreeMap, y las Collections<> como ArrayList o LinkedList.
- Posición en un equipo de futbol utilizaremos las siguientes convenciones (pódeis usar un array y guardáis esta información si os resulta más cómodo, o usar una tabla. También se pueden añadir más posiciones).
`String[] posiciones = {"Portero", "Defensa", "Medio", "Ataque"}`
`String[] posicionesarbitro = {"Principal", "Asistente" }`
- La operación de Carga deberá especificar el archivo/s dónde se encuentran los datos del sistema de gestión de partidos de Fútbol, previamente guardados usando serialización.
- La operación de Salvar deberá guardar en archivo/s todos los datos de los jugadores, árbitros, partidos, estadios y equipos usando serialización.
- El alumno tiene libertad para modificar/cambiar/añadir/prescindir de los métodos indicados en el esquema proporcionado, justificándolo en el momento de la defensa. Las modificaciones NO PUEDEN ser cambios drásticos en el programa planteado.
- **Para conseguir la nota máxima en esta práctica 0.4 ptos. El alumno deberá hacer una práctica correcta que incluya todos los métodos indicados y que funcione correctamente y sin fallos.**
- El alumno también tiene la libertad de hacer cualquier tipo de interfaz gráfica (y no la propuesta en línea de comandos), no siendo necesario para la correcta evaluación de la práctica. Se considera opcional.
- Cualquier actividad realizada por el alumno que se indica como **opcional** en esta práctica, se tendrá en cuenta en la nota final de curso.

NOTAS ACLARATORIAS:

- [] significa que el alumno es libre de incluir los parámetros que estime conveniente
- +/- significa public/private: Esto es orientativo, para una situación ideal, sin embargo el alumno es libre de modificar la visibilidad de los elementos de sus clases libremente para ayudar a la realización de la práctica con más rapidez. Esto no perjudica la nota final de la misma.
- ... significa que el alumno incluya los métodos que considere oportunos