

Se trata de realizar una aplicación que se encargue de gestionar equipos y jugadores de fútbol. La aplicación tendrá la siguiente estructura de clases:

1. Clase Leer. Clase de apoyo a la lectura y validación de datos. Se aporta (no hay que tocarla).
2. Clase Empresa. Sus objetos representan empresas. Sus miembros son:
 - Atributos:
 - CIF.
 - Razón Social.
 - Ciudad.
 - País (dato fijo para todas las empresas, su valor será “España”).
 - Métodos:
 - Constructor. Incorporará los datos de la empresa a través de lecturas desde el teclado.
 - Métodos “*get-set*” para todos los atributos.
3. Clase Equipo. Hereda de la clase Empresa. Sus objetos representan equipos de fútbol. Sus miembros propios son:
 - Atributos:
 - Temporadas en la categoría (numérico).
 - Posición en la última temporada (numérico).
 - Ganador del campeonato en alguna ocasión (booleano).
 - Lista de jugadores (ver clase Jugador).
 - Métodos:
 - Constructor. Incorporará los datos del equipo a través de lecturas desde el teclado.
 - Métodos “*get-set*” para todos los atributos.
 -
4. Clase Jugador. Sus objetos representan jugadores de fútbol. Sus miembros son:
 - Atributos:
 - Nombre.
 - Dorsal.
 - Fecha de nacimiento.
 - Métodos:
 - Constructor. Incorporará los datos del jugador a través de lecturas desde el teclado.
 - Métodos “*get-set*” para todos los atributos.
 -
5. Clase Principal. Se trata de la clase aplicación.

La aplicación mantendrá una estructura de datos que pueda almacenar como máximo a 20 equipos, cada uno de ellos con un máximo de 20 jugadores.

Al ejecutarse aparecerá un menú con las siguientes opciones:

1. Alta de equipo. Pedirá los datos necesarios.
2. Modificación de datos de equipo. El usuario podrá elegir qué dato modificar.
3. Alta de jugador. Pedirá los datos necesarios, incluyendo el equipo al que pertenece.
4. Listado de equipos. Deberá aparecer su nombre, su CIF, su número de jugadores registrados y su clasificación anterior. El listado deberá mostrarse de forma tabular.
5. Listado de jugadores de un equipo. Ordenados por orden de dorsal. El listado deberá mostrarse de forma tabular.
0. Salir.

NOTAS:

- Todas las clases tendrán opciones de visibilidad por defecto.
- Todos los atributos serán privados.
- Además de los métodos exigidos, podrán añadirse cuantos se estimen convenientes.