

Informe

Nombre: Robinson Josué Romero Nolivos

NRC: 1322

Estructura General

El programa está compuesto por varias clases que trabajan en conjunto para gestionar equipos, jugadores y partidos en un torneo. Las clases principales son:

Main: Clase principal que contiene el método “main”, donde se ejecuta el programa.

Jugador: Clase que representa a un jugador de fútbol.

Equipo: Clase que representa a un equipo de fútbol.

Partido: Clase que representa un partido entre dos equipos.

SistemaTorneo: Clase que gestiona el sistema del torneo, incluyendo el registro de equipos y partidos.

Descripción de las Clases

1. Clase Main

La clase Main es el punto de entrada del programa. En su método “main” se llevan a cabo las siguientes acciones:

Registro de Equipos: Se solicitan los nombres de los equipos, los entrenadores y los jugadores. Cada equipo puede tener hasta 11 jugadores, que se agregan mediante un ciclo for.

```

import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        SistemaTorneo sistema = new SistemaTorneo();

        // Registrar equipos
        System.out.print("Introduce el nombre del equipo local: ");
        String equipol = scanner.nextLine();
        System.out.print("Introduce el nombre del entrenador: ");
        String entrenadorl = scanner.nextLine();
        Equipo equipol = new Equipo(equipol, entrenadorl);
        for (int i = 1; i < 12; i++) {
            System.out.print("Introduce el nombre del jugador "+i+" : ");
            String jugador = scanner.nextLine();
            equipol.agregarJugador(new Jugador(jugador));
        }
        sistema.registrarEquipo(equipol);

        System.out.print("Introduce el nombre del equipo visitante: ");
        String equipov = scanner.nextLine();
        System.out.print("Introduce el nombre del entrenador: ");
        String entrenadorv = scanner.nextLine();
        Equipo equipo2 = new Equipo(equipov, entrenadorv);
        for (int i = 1; i < 12; i++) {
            System.out.print("Introduce el nombre del jugador "+i+" : ");
            String jugador = scanner.nextLine();
            equipo2.agregarJugador(new Jugador(jugador));
        }
        sistema.registrarEquipo(equipo2);
    }
}

```

Creación de un Partido: Se pide al usuario la fecha del partido y se crea una instancia de Partido con los equipos registrados.

```

// Crear partido
System.out.print("Introduce la fecha del partido: ");
String fecha = scanner.nextLine();
Partido partido = new Partido(equipol, equipo2, fecha);
sistema.crearPartido(partido);

```

Listar Equipos: Se llama a un método de SistemaTorneo para mostrar todos los equipos y sus respectivos jugadores.

```

// Listar equipos
sistema.listarEquipos();

```

Mostrar Historial de Partidos: Se despliega el historial de partidos registrados.

```
// Mostrar historial de partidos
sistema.mostrarHistorialPartidos();
```

Actualizar Resultados: Se solicita el resultado del partido y se actualiza en el sistema.

```
// Actualizar resultados
System.out.print("Introduce el resultado del partido: ");
String resultado = scanner.nextLine();
sistema.actualizarResultado(0, resultado);
```

Mostrar Historial de Partidos de Nuevo: Se vuelve a mostrar el historial de partidos después de la actualización.

```
// Mostrar historial de partidos después de actualizar
sistema.mostrarHistorialPartidos();

scanner.close();
```

2. Clase Jugador

Esta clase representa a un jugador de fútbol. Tiene un atributo nombre_jugador y un método constructor para inicializarlo. También tiene un método getNombre para obtener el nombre del jugador.

```
class Jugador {
    private String nombre_jugador;

    public Jugador(String nombre) {
        this.nombre_jugador = nombre;
    }

    public String getNombre() {
        return nombre_jugador;
    }
}
```

3. Clase Equipo

La clase Equipo tiene atributos para el nombre del equipo, el entrenador y una lista de jugadores. Incluye:

```
import java.util.ArrayList;
import java.util.List;

class Equipo {
    private String nombre_equipo;
    private String entrenador;
    private List<Jugador> jugadores;
```

Un constructor para inicializar el nombre del equipo y el entrenador.

```
public Equipo(String nombre, String entrenador) {
    this.nombre_equipo = nombre;
    this.entrenador = entrenador;
    this.jugadores = new ArrayList<>();
}
```

Un método agregarJugador para añadir jugadores a la lista.

```
public void agregarJugador(Jugador jugador) {
    jugadores.add(jugador);
}
```

Métodos getNombre, getEntrenador y getJugadores para acceder a la información del equipo y sus jugadores.

```
public String getNombre() {
    return nombre_equipo;
}

public String getEntrenador() {
    return entrenador;
}

public List<Jugador> getJugadores() {
    return jugadores;
}
```

4. Clase Partido

El Partido representa un encuentro entre dos equipos. Sus atributos incluyen:

equipoLocal y equipoVisitante, que son instancias de Equipo; fecha, resultado, y un booleano finalizado que indica si el partido ha terminado.

```
class Partido {  
    private Equipo equipoLocal;  
    private Equipo equipoVisitante;  
    private String fecha;  
    private String resultado;  
    private boolean finalizado;  
  
    public Partido(Equipo equipoLocal, Equipo equipoVisitante, String fecha) {  
        this.equipoLocal = equipoLocal;  
        this.equipoVisitante = equipoVisitante;  
        this.fecha = fecha;  
        this.resultado = "Pendiente";  
        this.finalizado = false;  
    }  
}
```

Un método finalizarPartido que actualiza el resultado y marca el partido como finalizado.

```
public void finalizarPartido(String resultado) {  
    this.resultado = resultado;  
    this.finalizado = true;  
}
```

Un método isFinalizado que devuelve el estado del partido.

```
public boolean isFinalizado() {  
    return finalizado;  
}
```

Un método toString que devuelve una representación en forma de cadena del partido.

```
@Override  
public String toString() {  
    return "Partido: " + equipoLocal.getNombre() + " vs " + equipoVisitante.getNombre() +  
        " | Fecha: " + fecha + " | Resultado: " + resultado;  
}
```

5. Clase SistemaTorneo

La clase SistemaTorneo se encarga de gestionar los equipos y partidos. Incluye:
Listas para almacenar equipos y partidos.

```
import java.util.ArrayList;
import java.util.List;

class SistemaTorneo {
    private List<Equipo> equipos;
    private List<Partido> partidos;

    public SistemaTorneo() {
        equipos = new ArrayList<>();
        partidos = new ArrayList<>();
    }
}
```

Métodos para registrar equipos (registrarEquipo),

```
public void registrarEquipo(Equipo equipo) {
    equipos.add(equipo);
}
```

crear partidos (crearPartido),

```
public void crearPartido(Partido partido) {
    partidos.add(partido);
}
```

listar equipos (listarEquipos),

```
public void listarEquipos() {
    for (Equipo equipo : equipos) {
        System.out.println("Equipo: " + equipo.getNombre() + ", Entrenador: " + equipo.getEntrenador());
        System.out.print("Jugadores: ");
        for (Jugador jugador : equipo.getJugadores()) {
            System.out.print(jugador.getNombre() + " \n");
        }
    }
}
```

mostrar historial de partidos (mostrarHistorialPartidos),

```

public void mostrarHistorialPartidos() {
    for (Partido partido : partidos) {
        System.out.println(partido);
    }
}

```

y actualizar resultados (actualizarResultado).

```

public void actualizarResultado(int indicePartido, String resultado) {
    if (indicePartido >= 0 && indicePartido < partidos.size()) {
        partidos.get(indicePartido).finalizarPartido(resultado);
    } else {
        System.out.println("Índice de partido no válido.");
    }
}
}

```

Flujo del Programa

1. Se inicia el programa y se crea una instancia del sistema de torneo.
2. Se registran dos equipos con sus respectivos entrenadores y jugadores.
3. Se crea un partido entre los equipos registrados y se asigna una fecha.
4. Se listan los equipos y se muestra el historial de partidos.
5. Se solicita un resultado para el partido y se actualiza en el sistema.
6. Finalmente, se muestra el historial de partidos de nuevo.