

Java Foundations

Prácticas - Sección 8: Liga de fútbol

Descripción general

Ha sido un invierno muy frío y con mucha nieve. Ninguno de sus amigos ha querido jugar al fútbol. Pero ahora que ha llegado la primavera, puede empezar otra temporada de la liga. El reto es escribir un programa que modele una liga de fútbol y realice un seguimiento de las estadísticas de la temporada.

Hay cuatro equipos en la liga. Los encuentros se determinan de forma aleatoria. Cada martes se juegan dos partidos, lo cual permite que todos los equipos participen semanalmente. No hay definido un número de partidos por temporada. La temporada continúa hasta que llega el invierno.

La liga es muy sensible a la temperatura. Los defensas son muy lentos en días calurosos. Los días más cálidos aumentan la posibilidad de más goles durante un partido. Si la temperatura es helada, no se juegan partidos esa semana. Si hay tres semanas consecutivas de temperaturas heladas, significa que ha llegado el invierno y ha terminado la temporada.



Tareas

Escriba un programa que modele una liga de fútbol y realice un seguimiento de las estadísticas de la temporada. Considere atentamente qué datos se deben almacenar en una matriz y qué datos se deben almacenar en un `ArrayList`. Diseñe clases con campos y métodos basados en la descripción de la liga. También necesitará una clase de prueba que contenga un método `main`. Todos los campos deben ser `private`. Proporcione los métodos `getter` y `setter` necesarios.

Equipos

Cada equipo tiene un nombre. El programa también debe realizar un seguimiento del total de victorias, pérdidas, empates, goles marcados y goles permitidos. Cree una matriz de los equipos que gestionará el programador.

Imprima las estadísticas del equipo cuando termine la temporada.

Partidos

En un partido, es importante tener en cuenta el nombre del equipo, la puntuación del equipo y la temperatura del día. Numere cada partido con un número de ID. Este número aumenta a medida que se juega cada partido. Realice un seguimiento de cada partido jugado esta temporada. Esta clase almacena un `ArrayList` de todos los partidos como un campo.

El programa debe determinar las puntuaciones de forma aleatoria. El número máximo de goles que puede marcar un equipo debe aumentar proporcionalmente con la temperatura. Pero asegúrese de que estos números sean bastante razonables.

Cuando la finalice la temporada, imprima las estadísticas de cada partido. Imprima la temperatura más alta y la temperatura media de la temporada.

Scheduler

Acepte la entrada del usuario mediante `JOptionPane` o `Scanner`. Mientras se ejecuta la aplicación, pida al usuario que introduzca una temperatura. El programa no debería fallar debido a la entrada del usuario. Si la temperatura es suficientemente cálida para jugar, programe dos partidos. Los oponentes se eligen de forma aleatoria. Asegúrese de que los equipos no se programen para jugar contra sí mismos. Si hay tres semanas consecutivas de temperaturas heladas, la temporada ha terminado.

Salida de ejemplo:

```
run:
Too cold to play.
Too cold to play.
Too cold to play.
Season is over

*****RESULTS*****

Team 1
Wins: 1, Losses: 1, Ties:0
Points Scored: 9, Points Allowed: 9

Team 2
Wins: 1, Losses: 1, Ties:0
Points Scored: 8, Points Allowed: 8

Team 3
Wins: 0, Losses: 1, Ties:1
Points Scored: 6, Points Allowed: 9

Team 4
Wins: 1, Losses: 0, Ties:1
Points Scored: 8, Points Allowed: 5

Game #1
Temperature: 90
Away Team: Team 2, 4
Home Team: Team 4, 7

Game #2
Temperature: 90
Away Team: Team 1, 8
Home Team: Team 3, 5

Game #3
Temperature: 35
Away Team: Team 1, 1
Home Team: Team 2, 4

Game #4
Temperature: 35
Away Team: Team 3, 1
Home Team: Team 4, 1

Hottest Temp: 90
Average Temp:62.5
```

Indicación

Es posible que el programa se quede bloqueado en un bucle infinito durante este ejercicio. Es algo habitual durante el desarrollo de software. El programa final debe tener una forma de terminarse por sí mismo. Pero hasta que haya implantado esa función, el IDE proporciona un botón **Stop**.

