

Упражнения – Раздел 8: Футбольная лига

Обзор

Прошлая зима была очень холодной и снежной. Никто из ваших друзей не хотел играть в футбол. Однако с приходом весны может начаться очередной сезон чемпионата. Ваша задача — написать программу, которая моделирует футбольную лигу и отслеживает статистику сезона.



В лиге 4 команды. Пары противников определяются случайным образом. Каждый вторник проходят 2 игры, и это позволяет еженедельно участвовать каждой команде. Количество игр в сезоне не ограничено. Сезон продолжается до наступления зимы.

Активность лиги сильно зависит от температуры. В жаркие дни защитники пассивны. Поэтому в такие дни растет вероятность увеличения количества голов за игру. Если температура ниже нуля, а такую неделю игры не проводятся. Если температура ниже нуля в течение 3 последовательных недель, тогда считается, что пришла зима и сезон завершен.

Задачи

Напишите программу, которая моделирует футбольную лигу и отслеживает статистику сезона. Тщательно рассмотрите, какие данные следует хранить в массиве, а какие данные следует хранить в `ArrayList`. На основе описания лиги разработайте классы с полями и методами. Вам также потребуется тестовый класс, который содержит главный метод. Все поля должны иметь значение `private`. Предоставьте все необходимые методы чтения и устанавливающие методы.

Команды

У каждой команды есть название. Для каждой команды программа должна также отслеживать общее число побед, общее число проигрышей, общее число дополнительных матчей, общее число засчитанных голов и общее число голов. Создайте массив команд, которым будет управлять планировщик.

После завершения сезона распечатайте статистику каждой команды.

Игры

В игре важно отметить название каждой команды, число очков каждой команды и температуру в этот день. Каждой игре следует назначить целочисленный идентификационный номер. Это число возрастает с каждой игрой. Необходимо отследить все игры, сыгранные в этой сезоне. Этот класс хранит `ArrayList` всех игр как поле.

Ваша программа должна определять очки случайным образом. Максимальное число голов, которое может забить любая команда, должно возрастать пропорционально температуре. Однако такие числа должны быть разумными.

После завершения сезона распечатайте статистику каждой игры. Напечатайте самую высокую температуру и среднюю температуру сезона.

Планировщик

Должен принимать ввод пользователя с помощью метода `JOptionPane` или `Scanner`. Во время работы приложения следует запросить у пользователя ввод температуры. Ввод пользователя не должен вызвать сбой программы. Если достаточно тепло, чтобы играть, запланируйте 2 игры. Противники выбираются случайным образом. Убедитесь в том, что по плану командам не предлагается играть против самих себя. Если температура ниже нуля в течение 3 последовательных недель, тогда сезон завершен.

Пример выходных данных:

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

*****РЕЗУЛЬТАТЫ*****

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
```

Подсказка

Во время этого упражнения ваша программа может застрять в бесконечном цикле. Это не редкость при разработке программного обеспечения. В окончательной программе должна быть способ собственного завершения. Но пока вы не реализовали эту функцию, NetBeans предоставляет кнопку **Stop**.

