

Relatório de Aplicação

Liga dos Campeões

Introdução

O presente trabalho foi desenvolvido no âmbito da UFCD “816 - Programação de Sistemas distribuídos em Java” do curso “Programador de Informática” do Instituto de Emprego e Formação Profissional.

O objetivo deste trabalho é criar uma aplicação, com a tecnologia Java, que faça uma simulação da completa da competição de futebol “ Liga dos Campeões”, a partir da fase de grupos.

Com este trabalho pretendo adquirir e melhorar os meus conhecimentos em Java e em programação orientada a objetos assim como a minha capacidade de análise e resolução de problemas.

Este relatório visa explicar todas as decisões tomadas por mim na criação da aplicação e do algoritmo assim como explicar a sua estrutura e as funcionalidades de cada um dos seus componentes.

Resumo da aplicação

A aplicação “Liga dos Campeões” tem como objetivo fazer a simulação completa da competição desportiva. Esta é composta por 4 ficheiros (que correspondem aos potes iniciais das equipas qualificadas) e 6 classes(eliminatória, equipa, final_match, grupo, index, Jogo, SortbyPoints).

A aplicação simula os seguintes aspetos, pela ordem cronológicas:

- **Sorteio de grupos** realizado de acordo com as seguintes regras:
 - Cada grupo tem uma equipa de cada pote
 - Cada grupo não pode ter mais do que uma equipa do mesmo país.
 - Sorteio aleatório das equipas de cada grupo.
- **Simulação dos fase de grupos** (Jogos,Classificação):
 - Cada equipa joga duas vezes com todas as outras equipas do mesmo grupo.
 - A vitória dá 3 pontos à equipa vencedora e 0 à equipa derrotada.
 - O empate dá um ponto a ambas as equipas.
 - As duas equipas com mais pontos de cada grupo passam à fase seguinte.
- **Simulação dos oitavos de final** (Sorteio, Jogos, Classificação):
 - O sorteio de cada eliminatória é feito de forma aleatória sendo as equipas de cada eliminatória não podem ser do mesmo país.
 - Cada eliminatória é composta por dois jogos.
 - A equipa vencedora é aquela que tem mais pontos nos dois jogos sendo que, no caso de empate, ganha a equipa com mais golos marcados fora (no estádio do adversário).
- **Simulação dos oitavos de final** (Sorteio, Jogos, Classificação):
 - O sorteio de cada eliminatória é feito de forma aleatória.
 - Cada eliminatória é composta por dois jogos.
 - A equipa vencedora é aquela que tem mais pontos nos dois jogos sendo que, no caso de empate, ganha a equipa com mais golos marcados fora (no estádio do adversário).

Resumo da aplicação

- **Simulação dos quartos de final** (Sorteio, Jogos, Classificação):

- O sorteio de cada eliminatória é feito de forma aleatória.
- Cada eliminatória é composta por dois jogos.
- A equipa vencedora é aquela que tem mais pontos nos dois jogos sendo que, no caso de empate, ganha a equipa com mais golos marcados fora (no estádio do adversário).

- **Simulação das meias finais** (Sorteio, Jogos, Classificação):

- O sorteio de cada eliminatória é feito de forma aleatória.
- Cada eliminatória é composta por dois jogos.
- A equipa vencedora é aquela que tem mais pontos nos dois jogos sendo que, no caso de empate, ganha a equipa com mais golos marcados fora (no estádio do adversário).

- **Simulação da final** (Jogos, Classificação):

- A final é composta por um jogo.
- A equipa vencedora é aquela que marcar mais golos, no caso de de empatarem o resultado é aleatório.

Na aplicação irão aparecer as classificações de cada grupo, as equipas classificadas para as fases seguintes e, a partir dos oitavos de final, irão aparecer os jogos das eliminatórias assim como a equipa vencedora de cada uma.

Ficheiros

A aplicação é composta por 4 ficheiros correspondentes aos potes das equipas classificadas para a competição sendo eles: pote1.txt, pote2.txt, pote3.txt e pote4.txt.

Cada ficheiro é composto por várias linhas sendo que cada linha corresponde a uma equipa e contém a seguinte informação (por esta ordem):

(nome de equipa) (sigla do país) (nome do estádio)

Estes ficheiros são essenciais à aplicação porque contém toda a informação das equipas participantes

Classe - equipa

A classe “equipa” visa representar cada equipa como sendo um objeto. Ela é composta pelos seguintes atributos:

- **name** -> nome da equipa
- **country** -> país da equipa.
- **stadium** -> estádio da equipa.
- **points** -> pontos da equipa.

O construtor desta classe obriga a que sejam inseridos o nome, país e o estádio como argumentos para a sua instanciação.

A classe também é composta pelos seguintes métodos:

- **(overriding) toString()**-> sempre que a classe é chamada vai retornar o seu atributo “name”.
- **get/setters de cada atributo** -> dado que os atributos são privados te-mos de usar funções publicas para os trabalhar.

Classe - Jogo

A classe “Jogo” visa representar cada jogo como sendo um objeto. Ela é composta pelos seguintes atributos:

- **equipa_casa**-> objeto do tipo equipa que representa a equipa da casa.
- **equipa_fora**-> objeto do tipo equipa que representa a equipa visitante.
- **stadium** -> estádio onde joga a equipa.
- **golos_efora**-> golos da equipa visitante.
- **golos_ecasa**-> golos da equipa da casa.
- **resultado** -> resultado do jogo.
- **equipa_vencedora**-> objeto do tipo equipa que representa o vencedor.

O construtor desta classe obriga a que sejam inseridos dois objetos do tipo equipa, o primeiro representa a equipa visitada e o segundo representa a equipa visitante. O construtor automaticamente completa o atributo ‘stadium’ com o estádio da equipa visitada.

A classe também é composta pelos seguintes métodos:

- **gera_resultado**-> este método atribui um numero de golos aleatório aos golos marcados pela equipa visitante e pela equipa visitada. No fim, compara os valores, atribui a equipa vencedora ao atributo “equipa_vencedora” e regista o resultado: 0 se a equipa visitada ganhou, 1 se deu empate e 2 se ganhou a equipa fora. Para além disto, também é atribuída a pontuação às equipas.
- **dados_jogo** -> mostra o numero de golos marcado por cada equipa.
- **get/setters de cada atributo** -> dado que os atributos são privados temos de usar funções publicas para os trabalhar.

Classe - grupo

A classe “grupo” visa representar cada grupo como sendo um objeto. Ela é composta pelos seguintes atributos:

- **nome**-> nome do grupo.
- **teams**-> lista de objetos do tipo equipa.
- **games**-> lista de objetos do tipo Jogo.
- **golos_efora**-> golos da equipa visitante.
- **golos_ecasa**-> golos da equipa da casa.
- **resultado** -> resultado do jogo.
- **equipa_vencedora**-> objeto do tipo equipa que representa o vencedor.

A classe também é composta pelos seguintes métodos:

- **simula_jogos**-> faz com que cada equipa jogue duas vezes com todas as outras equipas do grupo, uma vez no seu estádio e outra na da equipa adversária.
- **classificacao**-> retorna a pontuação de cada equipa do grupo, desempate por numero de golos marcado.
- **get/setters de cada atributo** -> dado que os atributos são privados temos de usar funções publicas para os trabalhar.

Classe - eliminatória

A classe “eliminatória” visa representar cada eliminatória como sendo um objeto. Ela é composta pelos seguintes atributos:

- **nome**-> nome da eliminatória.
- **teams**-> lista de objetos do tipo equipa.
- **games**-> lista de objetos do tipo Jogo.
- **winning_team** -> objeto do tipo equipa que representa o vencedor.

A classe também é composta pelos seguintes métodos:

- **simula_jogos**-> faz com que cada equipa jogue duas vezes com todas as outras equipas do grupo, uma vez no seu estádio e outra na da equipa adversária.
- **classificacao**-> retorna a pontuação de cada equipa do grupo, no caso de empate faz a validação dos golos fora.
- **get/setters de cada atributo** -> dado que os atributos são privados temos de usar funções publicas para os trabalhar.

Classe - final_match

A classe “final_match” visa representar a final da competição como sendo um objeto. Ela herda a classe “Jogo” sendo que a única diferença é o método **gera_resultado**:

Nesta classe, a função gera_resultado faz com que seja apenas realizado um jogo sendo que o resultado desse jogo irá determinar o vencedor da competição. No caso de dar empate, o resultado será gerado aleatoriamente.

Classe - Index

A classe index é onde se vai realizar todo o programa através da utilização das outras classes.

A classe é composta pelas seguintes funções:

main -> função onde se vai desenvolver a aplicação.

criar_grupos -> função que cria grupos.

simular_grupos -> simula a fase de grupos de cada grupo.

oitavos_sorteio -> faz o sorteio das eliminatórias dos oitavos

simular -> simula uma eliminatória

sorteio_elimina -> sorteia uma eliminatória.

fill_pote -> preenche uma lista com as equipas que estão num ficheiro

givenList_shouldReturnARandomElement -> retorna um elemento aleatório de uma lista

verify_team -> verifica se duas equipas são do mesmo país.

fill_groups -> preenche os grupos.