

Regiões partilhadas

As entidades para interagirem tem de aceder a regiões que são partilhadas entre si, para este problema foram indicadas as zonas que devíamos usar.

Cada zona tem algumas especificações que devem de ser respeitadas, para evitar condições de corrida ou deadlocks.

1.1. Contestants Bench

1.1.1. Estrutura de dados interna

- **MemFIFO<Integer>[] teams:** Memória para guardar elementos de cada equipa
- **int[] contestantCounter:** Contador de elementos por equipa
- **boolean selectedContestant[][]:** Booleano para indicar que elementos foram selecionados para jogar.

1.1.2. Pontos de sincronização

- **Semaphore acess:** Garante o acesso de regime de exclusão mutua.
- **Semaphore refereeSem:** Ponto de sincronização para o árbitro
- **Semaphore[] coachesSem:** Ponto de sincronização para os treinadores
- **Semaphore[][] contestantsSem:** Ponto de sincronização para cada elemento (Contestant)

1.1.3. Construtor

- **ContestantsBench(Repository logger):** O construtor da classe recebe como argumentos a variável *logger*, para proceder ao registo de todas as interações do programa no ficheiro.

1.2. RefereeSite.

1.2.1. Pontos de sincronização

- **Semaphore[] coachesSem:** Ponto de sincronização para os treinadores

1.2.2. Construtor

- **RefereeSite(Repository logger):** O construtor da classe recebe como argumentos a variável *logger*, para proceder ao registo de todas as interações do programa no ficheiro.

1.3. Playground

1.3.1. Estrutura de dados interna

- **MemFIFO<Integer>[] teams:** Memória para guardar elementos de cada equipa
- **Int countContestants:** Contador de elementos para identificar o último elemento
- **Int countCoaches:** Contador de treinadores para identificar o último treinador
- **int[] contestantCounter:** Contador de elementos por equipa

1.3.2. Pontos de sincronização

- **Semaphore acess:** Garante o acesso de regime de exclusão mútua.
- **Semaphore refereeSem:** Ponto de sincronização para o árbitro
- **Semaphore[] coachesSem:** Ponto de sincronização para os treinadores
- **Semaphore[][] contestantsSem:** Ponto de sincronização para cada elemento (Contestant)

1.3.3. Construtor

- **Playground(Repository logger):** O construtor da classe recebe como argumentos a variável *logger*, para proceder ao registo de todas as interações do programa no ficheiro.

1.4. Repository

O repositório geral é uma região partilhada distinta das outras. Nesta região, todas as interações feitas por cada entidade são registadas num ficheiro, e esta região é a responsável por isso.

O ficheiro produzido contém algumas informações que necessitam de ser legendadas, pois para cada entidade existe um determinado número de estados, e cada estado tem o seu código representativo.

1.4.1. Estados do Referee

- STM: START_OF_THE_MATCH
- STG: START_OF_THE_GAME
- TSR: TEAMS_READY
- WTC: WAIT_FOR_TRIAL_CONCLUSION
- ETM: END_OF_THE_MATCH
- ETG: END_OF_THE_GAME

1.4.2. Estados dos Coaches

- WFRC: WAIT_FOR_REFEREE_COMMAND
- AETM: ASSEMBLE_TEAM
- WHTL: WATCH_TRIAL

1.4.3. Estados dos Contestants

- STB: SEAT_AT_THE_BENCH
- SIP: STAND_IN_POSITION
- DYB: DO_YOUR_BEST