

Trabalho – Campeonato de Futebol

Programação Orientada a Objetos (SCC204)

Prof. Moacir Pereira Ponti Junior

26 de abril de 2011

Trabalho Prático - Parte 2

Tarefas

O Trabalho 2 envolve o uso de interface para implementação de recursos de mais de uma classe existente, interface gráfica, tratamento de exceção e operações com arquivos. Além disso, permitirá uma simulação de jogos.

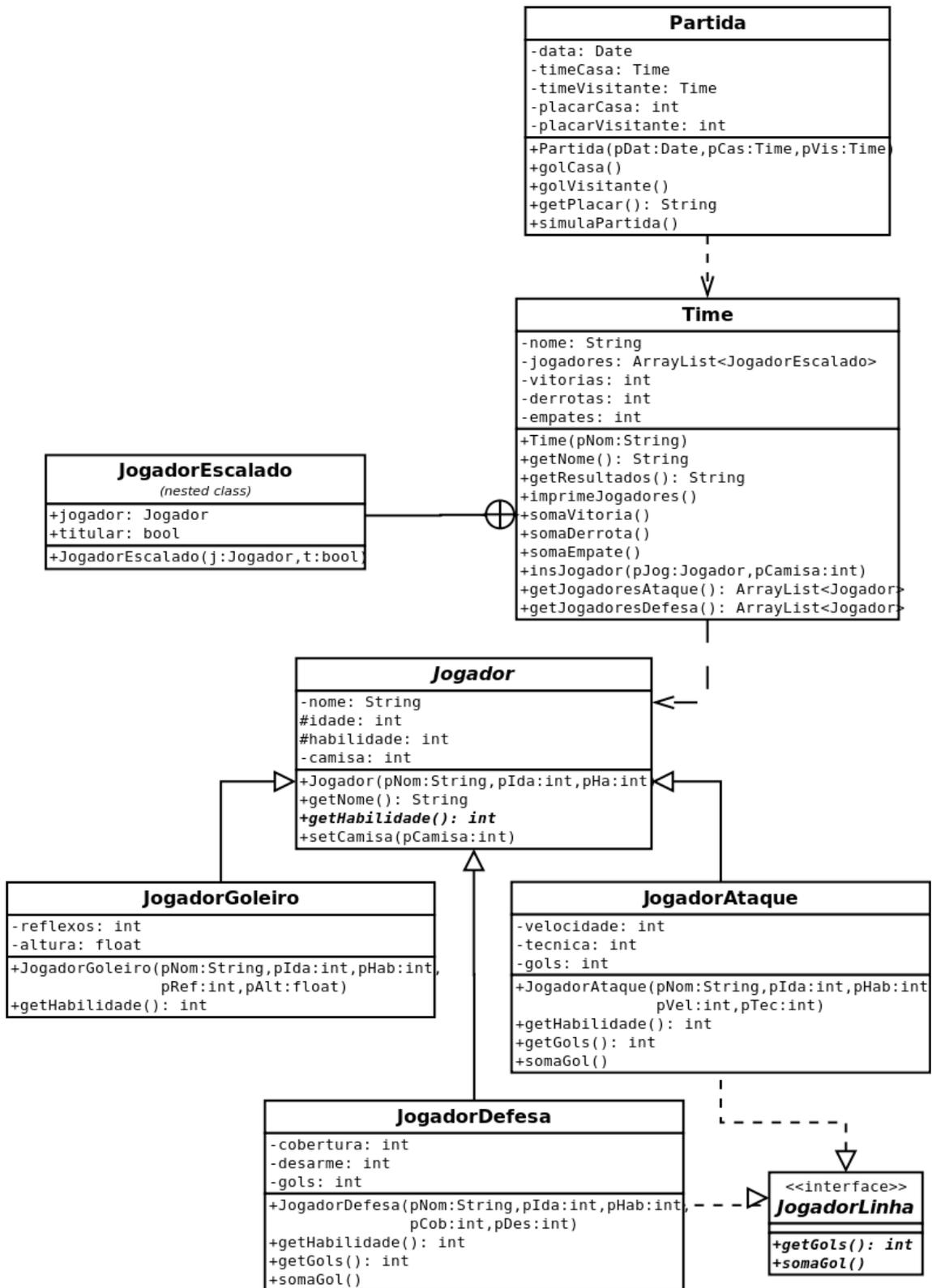


A implementação será **apenas** em **Java**, considerando as modificações realizadas no diagrama UML a seguir:

- A classe Jogador será uma classe abstrata
- A classe Partida possui um novo método chamado simulaPartida()
- A classe Time possui dois métodos novos: getJogadoresAtaque() e getJogadoresDefesa() que retornam um ArrayList (coleção de objetos) de Jogadores

Atacantes e de Jogadores de Defesas respectivamente. Esses métodos serão usados como auxiliares na simulação da partida. Além disso, a classe Time deverá conter uma classe interna Jogador Escalado (*inner class or nested class*) com os atributos jogador do tipo Jogador e titular do tipo boolean para verificar se o jogador é titular. Deve-se construir também um construtor para a classe Jogador Escalado passando o Jogador e o valor da variável titular.

- A classe Jogador não possui mais o atributo gols e nem os métodos getGols(), setNome(pNome: String) e somaGol(). Um novo método deve ser implementado setCamisa(pCamisa: int).
- As classes JogadorAtacante e JogadorDefesa devem conter o atributo inteiro gols e os métodos getGols() que retorna um número inteiro e o método somaGol().
- As classes JogadorAtaque e JogadorDefesa implementam a interface JogadorLinha que possui os métodos getGols() e somaGol() abstratos.



Tarefas para o Trabalho (parte 2):

1) Desenvolver uma interface gráfica conforme a figura a seguir. O programa deve inicialmente ler um arquivo com as informações dos jogadores e dos times e com essas informações deverá instanciar os objetos Jogadores, Times e Partida, para após preencher a interface gráfica. Crie uma classe “Campeonato” para conter a partida (e todos os objetos que serão instanciados)

2) Cada time poderá escalar 5 jogadores, estando presente na escalação pelo menos: 1 goleiro, 1 atacante e 1 defenso. Após cada time marcar os jogadores e pressionar "Escalar", o botão "Simular" irá se tornar disponível. Ao clicar em simular, o programa deverá executar a simulação e mostrar o placar final.

3) A simulação se procederá da seguinte maneira. Cada time terá 10 chances de ataque cada. Em cada chance poderá ou não sair um gol. Para isso será sorteado um número e feito um cálculo com relação às habilidades dos jogadores.

- O time que está atacando fará um gol se $A > B$, onde:

$A = (\text{rand()} * \text{media}(\text{habil_atacantes}) * \text{quant_atacantes}) \% 100$,

$B = (\text{rand()} * \text{media}(\text{habil_defensores} + \text{habil_goleiro}) * \text{quant_defensores}) \% 150$

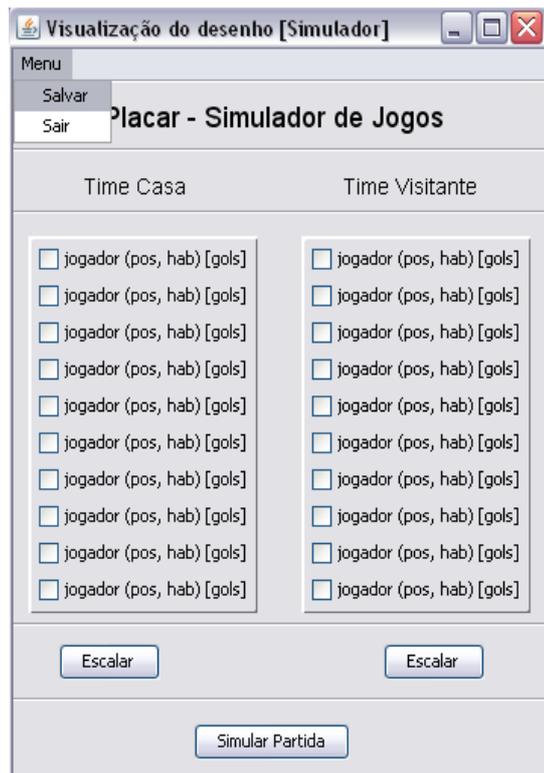
- Se for gol, o programa deverá escolher aleatoriamente qual dos jogadores fez o gol, e esse gol deverá ser registrado no objeto instanciado para o jogador. Você pode, se preferir, dar mais chances para um jogador de ataque ter feito o gol.

4) Após o término da simulação, registrar para cada time o resultado: vitória, empate ou derrota, e exibir o placar na interface gráfica.

5) Ao clicar em "Salvar e Fechar", o programa deverá salvar no arquivo os dados atualizados dos times (e seus resultados), e jogadores (e gols marcados)



Tela 1 – Simulador de Jogos



Tela 2 – Menu Salvar e Sair

Na interface gráfica o texto “jogador (pos, hab) [gols]” deverá ser preenchido através de um arquivo texto padrão no seguinte formato, o sinais de < e > é explicativo e não poderá constar no arquivo texto:

<Nome do time>
 <vitorias>
 <empates>
 <derrotas>
 <Goleiro>
 <Nome do goleiro>
 <idade>
 <habilidade geral>
 <numero da camisa>
 <reflexos>
 <altura>
 <Jogador de Linha (Ataque ou Defesa)>
 <Nome do jogador de linha>
 <idade>
 <habilidade geral>
 <numero da camisa>
 <habilidade1>
 <habilidade2>
 <numero de gols>

Lembrando que habilidade1 e 2 são genéricas e mudam dependendo do jogador de linha. Se for atacante é velocidade e técnica se for defensor é cobertura e desarme. O arquivo texto, deverá conter 2 times e 10 jogadores para cada time em apenas um único arquivo com o nome **dados.txt**.

Instruções

Dúvidas conceituais deverão ser levadas preferencialmente ao Monitor Especial (Eduardo) ou Estagiário PAE (Ricardo), cujos contatos estão na página da disciplina. Envie e-mail com o assunto [poo-trab1] Dúvida.

ATENÇÃO: A detecção de cópia de parte ou de todo código-fonte, de qualquer origem, implicará reprovação direta no trabalho. Partes do código cujas idéias foram desenvolvidas em colaboração com outro(s) aluno(s) devem ser devidamente documentadas em comentários no referido trecho. O que **NÃO** autoriza a cópia de trechos de código. Portanto, compartilhem idéias, soluções, modos de resolver o problema, mas não o código. Qualquer dúvida entre em contato com o professor.