Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Disciplina Programação Modular (DCC052) - 2014/2
Professor: Prof. Douglas G. Macharet.

Trabalho Prático 1 - Campeonato PM

Alunos: Elias Soares e Alvaro Souza.
Matrículas: 2011049053 e 2008030851.

1. Introdução:

O trabalho consiste em implementar um campeonato relativo a vários esportes sendo eles: basquete, futebol (salão, areia e campo), vôlei (praia de quadra e de praia), totalizando seis esportes no total. Além dos métodos os esportes serão compostos por seleções e as seleções serão compostas por atletas. Dado um arquivo contendo os resultados dos jogos o programa deverá ser capaz de imprimir estatísticas gerais das seleções e dos esportes.

2. Implementação:

Nosso trabalho está dividido nas seguintes classes:

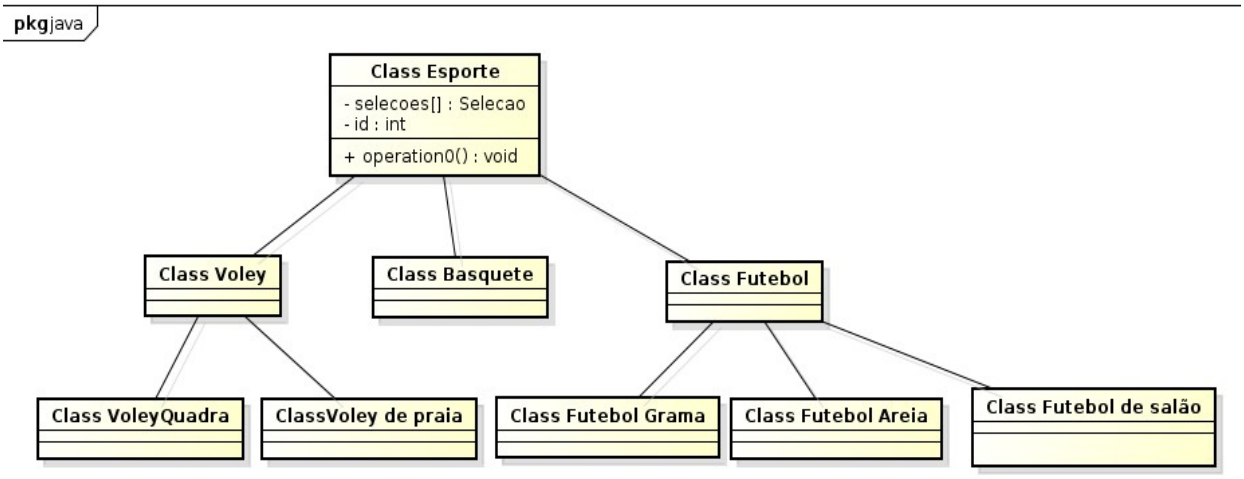
- Atletas;
- Basquete;
- EntradaESaida;
- Esporte;
- Futebol;
- FutebolAreia;
- FutebolGrama;
- FutebolSalao;
- Partida;
- Selecao;
- TP1PM;
- Voley;
- VoleyPraia;
- VoleyQuadra.

Tem-se um grande número de classes devido a exploração de dois dos principais conceitos de Programação Orientada a Objetos, são eles hierarquia e composição. Exploramos a hierarquia criando várias especializações de esportes, especializações de futebol e de vôlei. Utilizando composição, definimos que Atletas compõe a classe Selecoes e Selecoes compõe a classe Esporte, a ligação entre essas classes é forte, não faz sentido ter atletas sem seleção e seleção sem esporte.

Interpretamos as entradas de arquivos como um banco de dados, desta forma fazemos o acesso aos dados percorrendo o mesmo e utilizando os métodos para retornar os valores solicitados.

As classes têm as seguintes relações:

- FutebolGrama, FutebolSalao e FutebolAreia herdam características de Futebol;
- VoleyQuadra e VoleyPraia herdam de Voley;
- Voley, Basquete e Futebol herdam de Esporte;



Além dos conceitos de herança temos implementado os conceitos de polimorfismo, que aparece em suas diversas formas, porém destacamos o polimorfismo Universal de Inclusão onde há definição de métodos em classes inferiores, além das outras formas de polimorfismo, como o Ad-hoc de coerção e universal paramétrico.

Os nome utiizados para os esportes devem ser os mesmos citados na especificação sem alterações mesmo para maiusculas ou minusculas. Vôlei, Vôlei de Praia, Futebol, Futebol de areia, Futebol de salão, Basquete.

A entrada de estatisticas

Problemas:

- Tivemos um problema com a formatação indevida de arquivos, o partida.txt e estatisticas.txt, tinha um espaço em branco após o último elemento da primeira linha, ao usar a função split, que quebra Strings, tentávamos fazer casting para inteiro, gerando um erro durante a compilação, fomos “salvos” pelo **println**.

- Ao fazer os procedimentos para leitura dos arquivos, o aluno que fez os métodos ficou um dia e meio tentando saber porque ia direto para captura da exceção, não encontrando os arquivos, ele pediu pro colega tentar executar no PC dele, e rodou, os arquivos deveriam estar na pasta fora de onde os códigos estavam, diferentemente da Linguagem C que por padrão procura os arquivos de leitura na pasta onde se encontram os códigos.
- Dificuldade em nos acostumar com as facilidades do ArrayList e do Iterator.

Execução:

Os testes devem estar no mesmo diretório que o código .java e executar o makefile com o comando “make run”.

3. Testes:

Fizemos 3 testes sendo que um deles está com uma seleção e um time que não se encontra no bando de dados, dessa forma atestamos uma forma de tratamento para o caso onde não há uma seleção válida. Os outros testes foram bem sucedidos.

4. Conclusão:

Concluimos que a orientação a objeto é banstante, coesa para implementação, dado que a dupla criou cada um um grupo de métodos , classes e atributos, e ainda sim conseguiu-se gerar o encapsulamento de forma que cada uma pode utilizar as implementações do outro, sem o menor problema ou alterações de adaptação, tendo assim fixado os conceitos de orientação a objeto e encapsulamento dentre outros concedidos pela programação modular.

5. Referências:

http://www.caelum.com.br/apostila-java-orientacao-objetos/
http://www.guj.com.br/
http://www.javaprogressivo.net/
http://docs.oracle.com/javase/7/docs/api/java/util/
http://www.devmedia.com.br/java/