

## **Trabalho Prático 1 – Olimpíadas Modulares**

**Valor: 10 pontos**

**Data de entrega: 19/09/2016**

O objetivo deste trabalho prático é familiarizar o aluno com os principais conceitos de Orientação a Objetos e com a programação utilizando a linguagem Java. Deverá ser implementada uma estrutura de classes que permita armazenar e manipular os dados de uma olimpíada esportiva.

### **1. Instruções**

Considerando os arquivos de entrada apresentados nas próximas seções, o aluno deve criar os atributos e métodos necessários para acompanhar um campeonato (conjunto de partidas) relativo a 5 modalidades: Corrida, Natação, Levantamento de peso, Salto em altura e Ginástica artística.

Devem ser implementados os métodos para determinar o vencedor em uma determinada modalidade. Para cada categoria, o vencedor é determinado da seguinte forma:

- Corrida: Menor tempo considerando três diferentes corridas
- Natação: Menor tempo considerando três diferentes baterias
- Levantamento de peso: O maior valor considerando a soma de cinco levantamentos
- Salto em altura: Maior altura considerando cinco diferentes saltos
- Ginástica artística: Maior média considerando a nota dada por quatro juízes

Todas as informações deverão ser cadastradas no sistema a partir de um grupo de arquivos de entrada que serão disponibilizados na mesma pasta de execução do programa. Após a inserção de um conjunto de partidas relativas a cada uma das modalidades, o sistema deverá imprimir as estatísticas de acordo com um arquivo de entrada que será informado. As estatísticas são divididas em duas categorias:

- Modalidade: Deverá ser informado a classificação final para um esporte específico
- Geral: Deverá ser informado a classificação geral dos países em todos os esportes

O seu sistema deve tratar possíveis casos de exceção, por exemplo, a requisição de estatísticas de um atleta não cadastrado anteriormente.

O programa deverá ser feito baseado no JAVA SDK 8. Para guardar as informações relativas às coleções, você pode utilizar uma das classes que implementa a interface `Collection<E>`. Você pode obter mais informações em: <http://docs.oracle.com/javase/8/docs/api/java/util/Collection.html>.

#### **1.1. Funcionalidades**

Resumo das funcionalidades esperadas do sistema:

1. Criar objetos representando cada um dos Esportes mencionados;
2. Adicionar um conjunto de Atletas relativo a cada um dos Esportes;
3. Imprimir estatísticas específicas de um determinado Esporte
4. Imprimir estatísticas gerais da classificação dos países

## 2. Arquivos de Entrada e Saída

### 2.1. Exemplo dos arquivos de entrada:

- **Esportes: id;nome**

esportes.txt

```
1;Corrida
2;Natação
3;Levantamento de peso
4;Salto em altura
5;Ginástica artística
```

- **Países: id;nome**

paises.txt

```
1;Japão
2;USA
3;Chile
4;Rússia
5;Brasil
```

- **Atletas e resultados: id;idPaís;idEsporte;nome;<resultados>**

Os resultados seguirão diferentes padrões de acordo com a modalidade, por exemplo:

- Corrida: 10,78|9,85|9,91
- Natação: 20,15|19,66|22,43
- Levantamento de peso: 210|212|214|216|218
- Salto em altura: 1,95|1,97|2,01|2,05|2,10
- Ginástica artística: 8,9|9,3|9,2|9,2

atletas.txt

```
1;1;1;Kawasaki;10,78|9,85|9,91
2;2;1;Smith;10,08|10,22|10,27
3;3;1;Carlos;9,89|10,85|10,91
4;4;1;Sergey;10,55|12,15|10,88
5;5;1;João;11,78|9,15|9,91
6;1;2;Yamaha;20,15|19,66|19,43
7;2;2;John;21,15|19,16|21,43
8;5;2;José;19,55|19,96|20,01
9;2;3;Albert;210|212|214|215|217
10;2;3;Blake;207|209|212|215|218
11;4;3;Alexei;211|213|214|216|218
12;4;3;Maxim;211|214|215|216|216
13;2;4;Christopher;1,95|1,97|2,01|2,05|2,07
14;2;4;Jeffrey;1,93|1,96|2,00|2,04|2,05
15;3;4;Antonio;1,93|1,95|1,98|2,02|2,04
16;4;4;Nikolay;1,94|1,96|1,99|2,03|2,06
17;5;4;Marcos;1,92|1,95|1,97|2,00|2,04
18;1;5;Honda;8,9|9,3|9,0|9,1
19;2;5;Jeremy;9,3|9,3|9,2|9,2
20;4;5;Stanislav;9,6|9,1|9,1|9,4
```

É importante observar que vários atletas podem competir por um mesmo país, o que pode acabar resultando em mais de uma medalha por modalidade. O desempate, caso necessário, é feito pela ordem alfabética dos nomes.

- **Estatísticas: idTipo;<idEsporte>**

Tipos:

- 1: Estatísticas específicas de um Esporte
- 2: Estatísticas gerais da olimpíada

estatisticas.txt

```
1;1
1;5
2
```

## 2.2. Exemplo dos arquivos de saída:

Os arquivos de saída serão relativos apenas às estatísticas. O nome dos arquivos deverá ser composto pelo id do tipo da estatística e o respectivo id do Esporte. Considerando o arquivo ‘estatisticas.txt’ do exemplo acima, os seguintes arquivos devem ser gerados.

estatistica-1-1.txt

```
Corrida

João      9,15
Kawasaki  9,85
Carlos    9,89
Smith     10,08
Sergey    10,55
```

estatistica-1-5.txt

```
Ginástica artística

Stanislav  9,3
Jeremy     9,25
Honda      9,08
```

estatistica-2.txt

```
Quadro de medalhas

País      Ouro      Prata      Bronze
Rússia    2          2          0
USA        2          1          2
Brasil    1          0          1
Japão     0          2          1
Chile     0          0          1
```

### 3. Documentação

Entre outras coisas, a documentação deve conter:

1. Introdução: descrição do problema a ser resolvido e visão geral sobre o funcionamento do programa.
2. Implementação: descrição sobre a implementação do programa. Devem ser detalhadas as estruturas de dados utilizadas (de preferência com diagramas ilustrativos), o funcionamento das principais funções e procedimentos utilizados, bem como decisões tomadas relativas aos casos e detalhes que porventura estejam omissos no enunciado.
4. Testes: descrição dos testes realizados e listagem da saída (não edite os resultados). Você pode propor outros testes além dos fornecidos com o enunciado.
5. Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.
6. Bibliografia: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sites da Internet se for o caso.

#### O que deve ser entregue:

Envie um arquivo ZIP com o nome no formato 'tp1-primeironome1-primeironome2.zip', contendo os seguintes arquivos:

- Arquivo README.txt com os nomes completos dos alunos da dupla.
- O código fonte do programa em Java bem endentado e comentado. Deve ser fornecido junto com o fonte um arquivo Makefile com as opções 'make' e 'make run'.
- A documentação do trabalho bem escrita e detalhada.

#### Comentários Gerais:

- Comece a fazer este trabalho logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está tão longe quanto jamais poderá estar.
- Clareza, endentação e comentários no programa também serão avaliados.
- O trabalho deverá ser feito em dupla.
- Trabalhos copiados serão penalizados conforme anunciado.
- Penalização por atraso:  $(2^d - 1)$  pontos, onde d é o número de dias de atraso.

#### Critérios de avaliação:

- Funcionamento correto (3 pts).
- Uso correto dos conceitos de OO (5 pts).
- Documentação (2 pts).