

# Programação Modular

## Trabalho Prático 1 -

Icaro Harry

Igor Gonçalves Lima

**Setembro 2016**

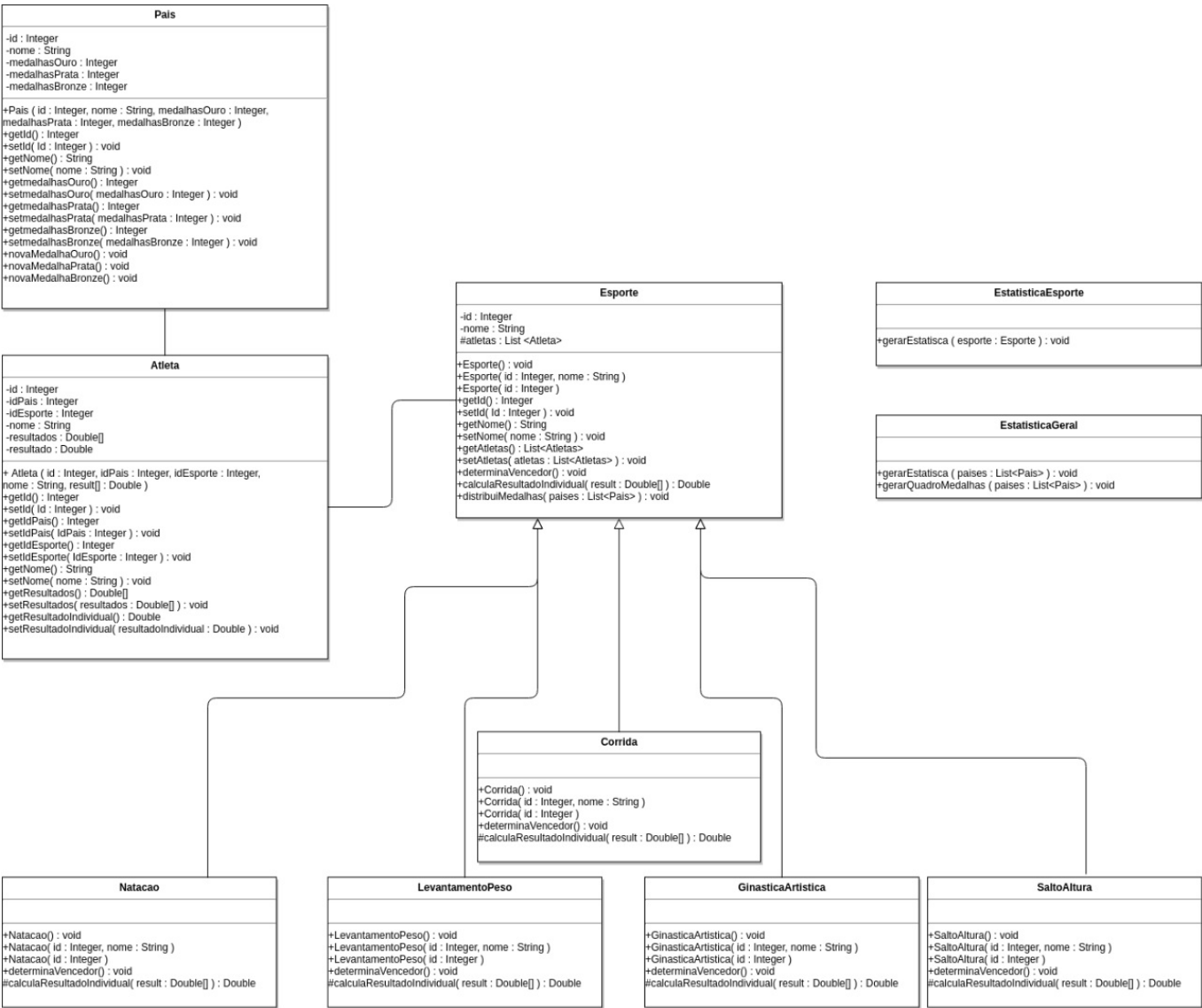
## INTRODUÇÃO

Utilizamos a programação para facilitar e automatizar nossa vida perante situações do mundo real. Porém, programar é algo complicado, e um recurso que seria para facilitar pode apresentar mais dificuldades do que esperávamos.

O problema proposto no trabalho é desenvolver um sistema para que determine o vencedor de cada modalidade de um campeonato, e que determine também o quadro geral de medalhas com os países participantes.

Tanto a entrada (Esportes, Países e Atletas) quanto a saída de dados (estatísticas de cada esporte e quadro geral de medalhas) será feita através de arquivos de texto .txt, e os detalhes da implementação será dada nos tópicos seguintes.

A aplicação foi modelada da seguinte forma:



# IMPLEMENTAÇÃO

O programa foi dividido em diferentes classes. Isso melhora a legibilidade do código e facilita o trabalho de uma equipe no mesmo projeto, possibilitando o desenvolvimento de partes diferentes do código ao mesmo tempo.

As classes principais definidas no projeto foram:

- Atleta
- Esporte (classe abstrata)
  - Corrida
  - Natacao
  - LevantamentoPeso
  - SaltoAltura
  - GinasticaArtistica
- EstatisticaEsporte
- EstatisticaGeral
- País

## **País**

É uma classe simples que representa um país. Seus dados de entrada para construção de um objeto são o id, o nome, o número de medalhas de ouro, de prata e de bronze. Conta com métodos get e set para todos os seus atributos e no final conta com métodos novaMedalhaOuro, novaMedalhaPrata e novaMedalhaBronze, que adicionam uma medalha a mais no contador de medalha de cada tipo.

## **Atleta**

Essa classe é mais elaborada, pois conta com mais dados para a construção de um objeto. Os atributos básicos são o id, idPaís, idEsporte, nome e a lista com os resultados do atleta. Há métodos get para cada atributo e, no final, há um setResultadoIndividual para, futuramente, inserir o resultado final do atleta, uma vez que este resultado é calculado para ser usado na classificação final.

## Esporte(classe abstrata)

É uma classe abstrata pois ela só é usada como superclasse de cada esporte. Apresenta atributos comuns como id e nome, além de uma lista de atleta. Apresenta como métodos novoAtleta, que adiciona novo atleta à lista e distribuiMedalhas, que um objeto do tipo País é verificado para adicionar uma medalha à ele, chamando algum dos métodos novaMedalha deste País. Em cada subclasse, um método determinaVencedor ordena a coleção de atletas daquela modalidade de acordo com os resultados individuais deles e um outro método novoAtleta adiciona um atleta novo no esporte. Em caso de empate de resultados entra atletas, é avaliada a ordem alfabética como critério de desempate.

```
public abstract void determinaVencedor()
```

Método abstrato para determinar o vencedor do esporte em questão. Deve ser sobrescrito pelas classes filhas.

Os vencedores são calculados da seguinte forma: Após todos os atletas e seus resultados individuais terem sido cadastrados no esporte, o vetor de atletas é ordenado de forma crescente ou decrescente, dependendo do esporte. Caso haja empate no resultado, o desempate é feito pela ordem alfabética do nome dos atletas.

```
protected abstract double
```

```
calculaResultadoIndividual(double[] result)
```

Método que calcula o resultado individual de um atleta no esporte. Para cada esporte, é utilizado um método diferente refletido pela

forma como o esporte funciona.

Por exemplo: na corrida o resultado individual do atleta é obtido pelo menor tempo de 3 corridas.

- **Parameters:** `double[]` — result - vetor contendo a série de resultados ou notas do atleta
- **Returns:** double - resultado individual

```
public void novoAtleta(Atleta a)
```

Método que adiciona um novo atleta na lista de atletas do esporte

- **Parameters:** `Atleta` — a - novo atleta a ser adicionado

```
public void distribuiMedalhas(List<País> países)
```

Método que deve ser chamado após a determinação dos vencedores para distribuir as medalhas para os países na lista fornecida

- **Parameters:** `List<país>` — países - lista de países concorrendo medalhas no presente esporte

## Corrida

Subclasse de Esporte Seta o nome para “Corrida” e usa um método para calcular o menor tempo entre três possíveis. A ordenação é feita do menor tempo para o maior.

## Natacao

Subclasse de Esporte Seta o nome para “Natação” e usa um método para calcular o menor tempo entre três possíveis. A ordenação é feita do menor tempo para o maior.

## **LevantamentoPeso**

Subclasse de Esporte Seta o nome para “Levantamento de peso” e usa um método para calcular a soma dos cinco pesos que o atleta levanta. A ordenação é feita da maior soma para a menor.

## **SaltoAltura**

Subclasse de Esporte Seta o nome para “Salto em altura” e usa um método para calcular a maior altura entre cinco dadas. A ordenação é feita da maior altura para a menor.

## **GinasticaArtistica**

Subclasse de Esporte Seta o nome para “Ginástica artística” e usa um método para calcular a média das quatro notas. A ordenação é feita da maior média de notas para a menor.

## **EstatisticaEsporte**

Imprime em um arquivo .txt a classificação final dos atletas em dado esporte, que é dado como entrada no gerarEstatistica. Como os atletas já estão ordenados dentro da classe do esporte dele, basta pegar essa informação e imprimir no arquivo.

# Estatística Geral

Gera a estatística recebendo uma coleção de países. Todos os países criados na execução são avaliados pelo número de medalhas e ordenados por elas. Primeiro critério de ordenação é número de medalhas de ouro, no desempate dessas olhamos o número de medalhas de prata, no desempate desse critério também é pelo número de medalhas de bronze e, persistindo o empate, ordem alfabética. A impressão é feita em arquivo e imprime países e medalhas.

A classe principal primeiramente fazia a leitura do arquivo e dividia o texto por linhas, para depois fazer a divisão por split. O que era número, era convertido em número (int ou double), já que o texto do arquivo era dado em String. Uma a uma as informações eram jogadas nas classes para a criação dos objetos.

```
public static void gerarEstatistica(List<País> países)
```

Método utilizado para gerar estatísticas gerais das olimpíadas.

- **Parameters:** `List<país>` — países - lista países participantes

```
public static void  
gerarQuadroDeMedalhas(List<País> países)
```

Método utilizado para ordenar a lista de países, colocando-os na ordem do número de medalhas de ouro ganhas.

Se o número de medalhas de ouro é o mesmo, utiliza-se as medalhas de prata. Se o número de medalhas de prata também é o mesmo, utiliza-se as medalhas de bronze. Se o número de medalhas de bronze também é o mesmo, utiliza-se a ordem alfabética

Esse método é implementado na classe de estatísticas visto que não tem a ver com um país individualmente para que seja implementado na classe País.

## TESTES

### Entrada 1 - Atletas

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

## Entrada 1 - Países

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

## Entrada 1 - Esportes

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

## Entrada 1 - Estatísticas

1;1
1;5
2

# Saída 1 - estatísticas-1-1.txt

Corrida	
João	9.15
Kawasaki	9.85
Carlos	9.89
Smith	10.08
Sergey	10.55

# Saída 1 - estatísticas-1-5.txt

Ginástica artística	
Stanislav	9.30
Jeremy	9.25
Honda	9.08

# Saída 1 - estatísticas-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

## Entrada 2 - Atletas

1;1;1;Kawasaki;10,78 9,85 9,91
2;1;1;Paul;10,78 9,85 9,91
3;1;1;Mario;10,78 9,85 9,91
4;1;1;Aaron;10,78 9,85 9,91
5;1;1;Jon;10,78 9,85 9,91
6;1;1;Bastian;10,78 9,85 9,91
7;1;1;Yamaha;10,78 9,85 9,91
8;1;1;Ocean;10,78 9,85 9,91
9;1;1;Zanark;10,78 9,85 9,91
10;1;1;Martin;10,78 9,85 9,91
11;1;1;Leo;10,78 9,85 9,91

## Entrada 2 - Países

1;Jap <del>o</del>
2;USA
3;Chile
4;R <del>o</del> ssia
5;Brasil

## Entrada 2 - Esportes

1;Corrida
2;Nata <del>o</del>
3;Levantamento de peso
4;Salto em altura

## Entrada 2 - Estatísticas

1;1

1;5

2

## Saída 2

```
Exception in thread "main" java.lang.IndexOutOfBoundsException: Index: 0, Size: 0
```

```
    at java.util.ArrayList.rangeCheck(ArrayList.java:635)
```

```
    at java.util.ArrayList.get(ArrayList.java:411)
```

```
    at esporte.Esporte.distribuiMedalhas(Esporte.java:91)
```

```
    at app.main(app.java:132)
```

```
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
```

```
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
```

```
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
```

```
    at java.lang.reflect.Method.invoke(Method.java:606)
```

```
    at com.intellij.rt.execution.application.AppMain.main(AppMain.java:147)
```

```
Process finished with exit code 1
```

## Entrada 3 - Atletas

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  
21;6;1;Petro;20,32|8,85|9,67  
22;2;1;Filipe;10,45|10,11|10,23  
23;7;1;Camargo;10,89|7,85|11,12  
24;9;1;Wesley;10,26|12,03|10,90  
25;10;1;Tuco;22,78|8,15|15,91  
26;1;2;Bam;20,67|19,23|21,43

27;5;2;Cazalbe;22,00 20,10 23,43
28;12;2;Marcelo;18,55 15,96 22,01
29;13;3;Mime;198 240 230 220 213
30;12;3;Kaiba;200 200 222 250 218
31;6;3;Joey;211 210 211 234 218
32;8;3;Aldebaran;211 220 203 216 214
33;8;4;Madruguinha;1,99 1,92 2,01 2,05 2,09
34;9;4;Tapia;1,93 1,96 2,00 2,04 2,04
35;7;4;Bendelack;1,93 1,95 1,98 2,02 2,10
36;12;4;Nick;1,94 1,96 1,99 2,03 2,02
37;11;4;Ryu;2,15 1,95 1,97 2,00 2,04
38;10;5;Subzero;8,7 9,1 9,0 9,0
39;2;5;Saruman;9,3 9,3 9,5 9,9
40;4;5;Toro;8,6 8,1 8,1 9,0

## Entrada 3 - Países

1;Japão
2;USA
3;Chile
4;Rússia
5;Brasil
6;Escócia
7;Nicarágua
8;Panamá
9;Espanha
10;China
11;Suriname

12;Marrocos
-------------

13;Egito
----------

## Entrada 3 - Esportes

1;Corrida
-----------

2;Natação
-----------

3;Levantamento de peso
------------------------

4;Salto em altura
-------------------

5;Ginástica artística
-----------------------

## Entrada 3 - Estatísticas

1;1
-----

1;5
-----

1;4
-----

1;3
-----

1;2
-----

2
---

## Saída 3 - estatísticas-1-1.txt

Corrida
---------

Camargo	7.85
---------	------

Tuco	8.15
------	------

Potro	8.85
-------	------

João	9.15
------	------

Kawasaki	9.85
Carlos	9.89
Smith	10.08
Filipe	10.11
Wesley	10.26
Sergey	10.55

## Saída 3 - estatísticas-1-2.txt

Natação	
Marcelo	15.96
John	19.16
Bam	19.23
Yamaha	19.43
Josil <sup>1</sup> / <sub>2</sub>	19.55
Cazalbe	20.10

## Saída 3 - estatísticas-1-3.txt

Levantamento de peso	
Mime	1101
Kaiba	1090
Joey	1084
Alexei	1072
Maxim	1072
Albert	1068



Aldebaran	1064
Blake	1061

## Saída 3 - estatísticas-1-4.txt

Salto em altura	
Ryu	2.15
Bendelack	2.10
Madruguinha	2.09
Christopher	2.07
Nikolay	2.06
Jeffrey	2.05
Antonio	2.04
Marcos	2.04
Tapia	2.04
Nick	2.03

## Saída 3 - estatísticas-1-5.txt

Ginástica artística	
Saruman	9.50
Stanislav	9.30
Jeremy	9.25
Honda	9.08
Subzero	8.95
Toro	8.45

# Saída 3 - estatísticas-2.txt

Quadro de medalhas			
País	Ouro	Prata	Bronze
USA	1	1	1
Marrocos	1	1	0
Nicaríí½gua	1	1	0
Egito	1	0	0
Suriname	1	0	0
China	0	1	0
Ríí½ssia	0	1	0
Escíí½cia	0	0	2
Japíí½o	0	0	1
Panamíí½	0	0	1
Brasil	0	0	0
Chile	0	0	0
Espanha	0	0	0

## CONCLUSÃO

Neste trabalho, foi proposto utilizar conceitos da programação modular para um programa que auxiliasse no controle da realização de um campeonato de esportes. Com isso, tivemos oportunidade de trabalhar com a Orientação a Objetos e a modularização do programa. Sendo um trabalho em dupla, foi possível que dividíssemos as classes para serem feitas individualmente para que futuramente fossem

juntadas para o propósito final do programa.

As classes Atleta, País e Esportes ficaram bem limpas e claras, sendo fácil entender a proposta de cada uma. As classes estatísticas também conseguem demonstrar com clareza o que propõe, sendo classes utilizadas praticamente para impressão. As estatísticas individuais basicamente só imprimem, uma vez que o vetor está ordenado. Já o quadro geral de medalhas necessita de uma ordenação com critérios de desempates, mas nada complicado.

A maior dificuldade encontrada no trabalho foi a classe principal, que ficou um pouco extensa, em compensação à simplicidade das outras classes. Porém, o programa funciona sem problemas, sendo que tivemos cuidados em aspectos como utilizar os modificadores de acesso para garantir o encapsulamento e o tratamento de exceções.

## **BIBLIOGRAFIA**

<http://www.devmedia.com.br/> Java - Como Programar - 8ª Ed. 2010

Deitel Prentice Hall

[RatioCaeli.com | Blog - acesso em 11:30 PM](#)

[java - Makefile with Jar and Package Dependencies - Stack Overflow](#)

[What is the IntelliJ shortcut key to create a javadoc comment? - Stack Overflow](#)

[java - Ordenar uma lista de objetos por mais de um atributo - Stack Overflow em Português](#)

[A printf format reference page \(cheat sheet\) | alvinalexander.com](#)

[sorting - How to sort an ArrayList in Java - Stack Overflow](#)

[java - Comparator and equals\(\) - Stack Overflow](#)

[Como criar e gravar dados em TXT com Java - DevMedia](#)

[How to iterate through Java List? Five \(5\) ways to Iterate Through Loop in Java • Crunchify](#)

[Java: Error: variable might not have been initialized - Stack Overflow](#)

[netbeans - reading text file with utf-8 encoding using java - Stack Overflow](#)

[Resolvido - Caracteres bagunçados \(ex.: ç vira Ã§ e á vira ã\) quando salvo um arquivo do ANTLR - Programação / Java - GUJ](#)

[UTF-8 - Wikipédia, a enciclopédia livre](#)

[file - Java FileReader encoding issue - Stack Overflow](#)

[String \(Java Platform SE 7 \)](#)

[Arrays in Java. Declare Initialize and Use Arrays in Java](#)

[An elegant way of initialising an empty String array in Java - Stack Overflow](#)

[How can I initialize a String array with length 0 in Java? - Stack Overflow](#)

[Parsing string to double - java - Stack Overflow](#)

[Java - Package ou Pacotes - Criar, Definir e Importar](#)