Universidade Federal de Minas Gerais
Departamento de Ciência da Computação
DCC052: Programação Modular
Trabalho Prático 1
Grupo:
Nome: Gabriel de Azevedo Cardoso **/ Matrícula:** 2015086689
Nome: Danilo Fabrino Favato **/ Matrícula:** 2016058069

---

1. Introdução

O objetivo do programa implementado é controlar as informações de uma competição olímpica que envolve atletas de diferentes países e diversas modalidades esportivas.
A entrada do programa é composta por 4 diferentes arquivos de entrada:

- esportes.txt: Esse arquivo armazena o ID e o nome de cada modalidade esportiva presente na competição, são 5 diferentes modalidades possíveis: Corrida, Natação, Levantamento de peso, Salto em altura e Ginástica Artística
- paises.txt: Arquivo com as informações de ID e nome de cada um dos países que participaram da competição.
- atletas.txt: Esse arquivo contém as informações relativas aos atletas como ID, nome, país de origem, esporte praticado e os resultados alcançados.
- estatisticas.txt: Arquivo que armazena as estatísticas devem ser produzidas pelo programa. Podem ser de dois tipo: 1) Estatísticas relativas a uma modalidade específica; ou 2)Quadro de medalhas da competição por país.

2. Funcionamento Básico

Basicamente o funcionamento do programa se dá da seguinte forma:

1. O arquivo de entrada dos esportes é lido e as informações guardadas em uma coleção de esportes.
2. Cada esporte possui uma lista de resultados que armazena o atleta e o valor resumo do rendimento do atleta naquela determinada modalidade.
3. O arquivo de entrada dos países é lida e as informações guardadas em uma coleção de países.
4. Cada país possui variáveis que armazenam quantas medalhas de cada tipo aquele país ganhou.
5. O arquivo de entrada dos atletas é lido
6. Cada atleta possui acesso às informações de seu país de origem.
7. As pontuações individuais de cada atleta são calculadas em um valor único, a fórmula deste cálculo depende do esporte.
8. Os resultados sumarizados de cada atleta são armazenados na lista da modalidade que cada um disputou.

9. Após processar todas as entradas a coleção de Esportes realiza a classificação dos atletas de acordo com as regras de cada Esporte.
10. Após a classificação as medalhas são distribuídas aos 3 primeiros lugares de cada esporte atualizando o quadro de medalhas por país.
11. Por fim o arquivo de estatísticas é lido e as saídas impressas nos respectivos arquivos.

3. Implementação

O programa foi dividido em 3 pacotes para implementação:

1. athletes: Pacote responsável por gerir as informações dos atletas, dos seus países de origem e resultados.
2. main: Pacote responsável pela interação do programa com usuário, o método MAIN se encontra nesse pacote assim como algumas funções utilitárias.
3. sports: Pacote responsável por gerir o comportamento das diferentes modalidades de esportes praticados.

As figuras a seguir resumem a organização dos pacotes, classes e suas relações. A Figura 1 mostra o pacote main enquanto a Figura 2 mostra os pacotes athletes e sports. A seguir serão descritos cada um dos pacotes com suas respectivas classes e comportamentos.

Figura 1 - Pacotes main

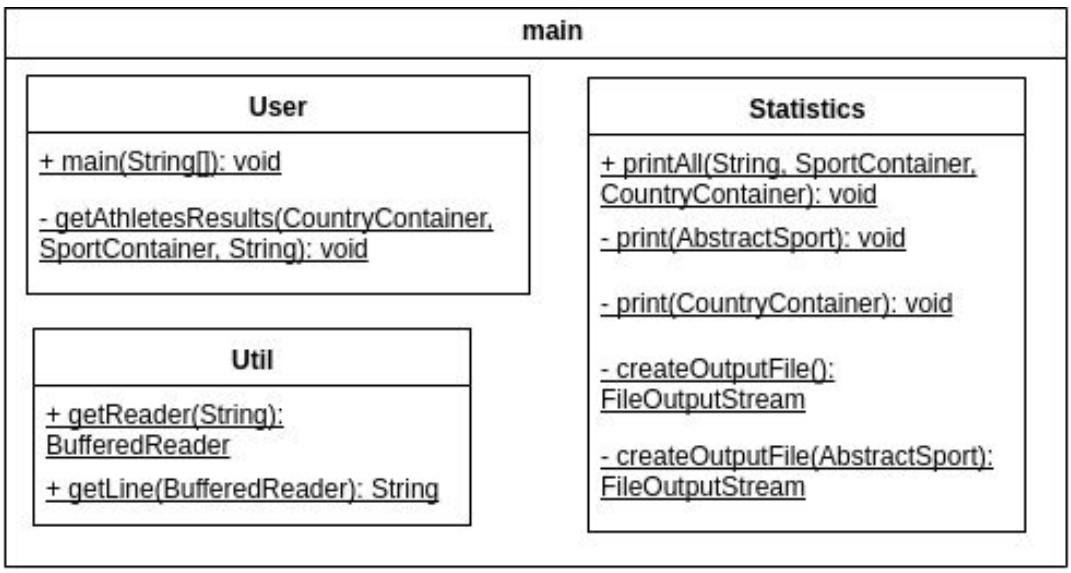
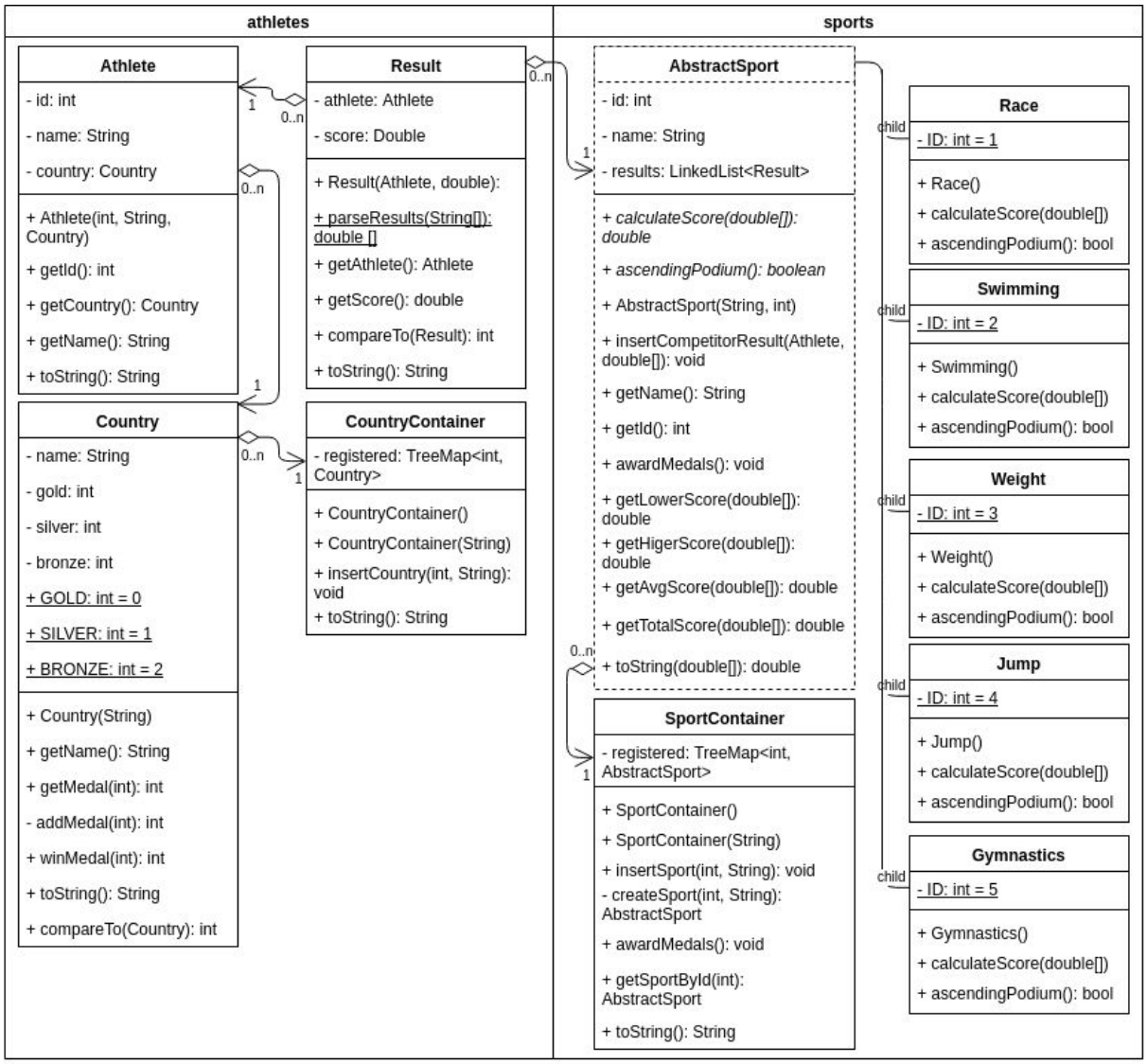


Figura 2 - Pacotes athletes e sports



3.1. Pacote athletes

Como dito anteriormente este pacote reúne as informações dos atletas participantes. Para isso foram implementadas 5 classes cujo comportamento será detalhado a seguir.

3.1.1. Classe Athlete

Essa classes representa o atleta, ela armazena as informações básicas do atleta como ID e nome e possui um apontador para o país que aquele atleta representa. Os métodos dessa classe são:

- public Athlete(int id, String name, Country country)
    - Método construtor, cria a instância e inicializa os atributos com o argumentos informados
- public int getId()
    - Retorna o número identificador do atleta

- public Country getCountry()
    - Retorna o país que aquele atleta representa
- public String getName()
    - Retorna o nome do atleta
- public String toString()
    - Retorna uma representação do atleta como um String que contém o ID, o nome e o nome do país.

3.1.2. Classe Country

Armazena as informações dos países como nome e quantidade de cada tipo de medalha.
Os métodos dessa classe são:

- public Country(String strName)
    - Construtor básico
- public String getName()
    - Retorna o nome do país
- public int getMedal(int type)
    - Retorna a quantidade de medalhas de um certo tipo que esse país possui. GOLD = 0, SILVER =1 e BRONZE = 2. Caso seja solicitado um tipo diferente disso retorna 0.
- private int addMedal(int type)
    - Adiciona uma medalha de um certo tipo às medalhas do país e retorna quantas medalhas deste mesmo tipo aquele país possui.
- public int winMedal(int type)
    - Método público responsável por chamar addMedal e distribuir a medalha de um certo tipo.
- public String toString()
    - Retorna uma representação do país como uma string, já formatado para saída no quadro de medalhas.
- public int compareTo(Country o)
    - Método responsável pela lógica de ordenação dos países no quadro de medalhas. São ordenados primeiro pela número de ouros, em caso de empate o número de pratas e depois o de bronzes é utilizado no desempate. Esse método é necessário pois Country implementa a interface Comparable.

3.1.3. Classe CountryContainer

Essa classe possui um TreeMap que guarda os países indexados pelos seus números de identificação. Os métodos implementados são:

- public CountryContainer()
    - Cria o objeto com o TreeMap vazio.

- public CountryContainer(String fileAddress)
    - Overload do método construtor que permite criar uma instância já com o TreeMap preenchido com os registros de um arquivo que deve ser informado como argumento.
- public GetCoutryById(int id)
    - Retorna uma referência para o país com a id igual ao valor passado por parâmetro, caso ele exista.
- public void insertCountry(int id, String name)
    - Cria um objeto Country e insere no TreeMap com o Id informado.
- public String toString()
    - Retorna um string com uma representação da classe que é exatamente o quadro de medalhas. Antes de gerar a string os países são ordenados conforme o número de medalhas.

3.1.4. Classe Result

Essa classe é a responsável por ligar um resultado a um atleta e um esporte. Cada esporte possui um coleção de classes deste tipo, que armazena todos os resultados das competições para este esporte. As instâncias dessa classe possuem dois atributos: um apontador para o atleta e outro que contém sua pontuação. Os métodos são:

- public Result(Athlete athlete, double score)
    - Construtor básico
- public static double[] parseResults(String[] line)
    - Método utilitário que serve para converter o string de resultados do arquivo de entrada em um arranjo de double para que os resultados sejam comparados corretamente.
- public Athlete getAthlete()
    - Retorna o Atleta responsável pelo resultado
- public double getScore()
    - Retorna o valor do resultado obtido pelo atleta.
- public int compareTo(Result o)
    - Método necessário para comparar os resultados de dois atletas e utilizado pelo Collections.Sort(). O valor dos resultados são comparados, em caso de empate compara-se os nomes dos atletas.
- public String toString()
    - Retorna uma String que representa o resultado, já formatado da maneira que será utilizado no arquivo de saída.

3.2. Pacote sports

As modalidades disputadas compartilham várias características comuns. Todas possuem um nome e um número de identificação, além disso todas

distribuem medalhas aos 3 melhores classificados. Por outro lado o método de classificação dos competidores nem sempre é comum. Na corrida e natação, os atletas são classificados de acordo com seu menor tempo dentre 3 corridas. Para o levantamento de peso, considera-se o peso total levantado após cinco levantamentos. O salto em altura por suas vez considera o salto mais alto entre cinco saltos. Por último, a ginástica artística considera a melhor média das notas de quatro juízes.

Este cenário fez com que fosse escolhida a implementação da classe abstrata AbstractSport, com as funcionalidades comuns de todos os esportes considerados. Nesta classe são implementados métodos gerais que recebem um array de pontos e então retornam uma pontuação final usando diversas métricas diferentes, como maior valor, menor valor e média. A partir dessa classe abstrata as modalidades foram criadas como subclasses, que por sua vez chamam o método correto de sua classe pai para o cálculo da pontuação de seus competidores.

3.2.1. Classe AbstractSport

Todo objeto do tipo AbstractSport possui os seguintes atributos: id, nome, e uma lista encadeada com os resultados dos atletas. Os métodos dessa classe são:

- AbstractSport(String name, int id)
    - Construtor básico, inicializa os atributos a partir dos argumentos e cria uma lista encadeada de Results vazia.
- public abstract double calculateScore(double scores[])
    - Como cada modalidade resume as diferentes “tentativas” de um maneira esse método é abstrato para que possa ser implementado pelas modalidades esportivas. Esse método recebe um arranjo com as “tentativas” e retorna ou o maior valor do arranjo, ou o menor, ou a média, ou a soma, dependendo da modalidade.
- public abstract boolean ascendigPodium()
    - Assim como cada modalidade resume as “tentativas” de uma maneira, algumas modalidades classificam os atletas por resultados ascendente (quanto menor melhor), é o caso da natação e da corrida. Outras modalidades classificam descendentemente (quanto maior melhor) é o caso do salto em altura, levantamento de peso e ginástica artística. Por isso esse método é abstrato. Ele retorna verdadeiro caso a lógica de classificação seja ascendente e falso caso contrário.
- public void InsertCompetitorResult(Athlete athlete, double scores[])
    - Insere o resultado de um atleta na lista de resultados da modalidade. Para tanto o método calculateScore(double scores[]) é chamado para calcular a medida de resumo das tentativas, após isso o Result é instanciado e armazenado na lista encadeada.
- public String getName()
    - Retorna o nome da modalidade esportiva.

- public int getId()
    - Retorna o ID do esporte.
- public void awardMedals()
    - Distribui as medalhas para os países dos 3 melhores classificados. Para isso a lista encadeada de resultados é ordenada dependendo do valor retornado por ascendigPodium. Após a ordenação o método Country.winMedal(int type) é chamado para cada um dos países dos 3 primeiros colocados.
- public double getLowerScore(double[] scores)
    - Retorna o menor dos scores fornecidos no argumento
- public double getHigherScore(double[] scores)
    - Retorna o maior dos scores fornecidos no argumento
- public double getAvgScore(double[] scores)
    - Retorna a médio dos scores fornecidos no argumento
- public double getTotalScore(double[] scores)
    - Retorna a soma dos scores fornecidos no argumento
- public String toString()
    - Retorna uma representação da modalidade esportiva como string já formatada para o arquivo de saída a ser gerado.

3.2.2. Subclasses de AbstractSport

As subclasses são bastante simples pois grande parte da lógica está implementada na classe abstrata. Basicamente cada uma das subclasses possui uma constante, que é o ID da modalidade, implementa o método construtor e os dois métodos abstratos da classe pai.

3.2.3. Classe SportContainer

Essa classe possui atributo TreeMap que armazena todas as modalidades esportivas disputadas de acordo com o ID das mesmas. Os seguintes métodos são implementados:

- public SportContainer()
    - Construtor básico, inicia a instância com o TreeMap vazio.
- public SportContainer(String fileAddres)
    - Overload do construtor que permite inicializar a instância já com o TreeMap preenchido com os registros do arquivo informado no argumento.
- private AbstractSport createSport(int id, String name)
    - Método responsável por escolher qual construtor será chamado, o que depende do valor do ID do esporte. Dado o id o construtor da modalidade esportiva é chamado e a instância criada retornada.
- public void insertSport(int id, String name)

    - Chama o método createSport para que a instância correta seja criada e adiciona a modalidade ao TreeMap
- public void awardMedals()
    - Chama o método AbstractSports.awardMedals para cada um dos AbstractSports armazenados no TreeMap
- public AbstractSport getSportByid(int id)
    - Retorna uma modalidade dado o id da mesma.
- public String toString()
    - Retorna um String que é a representação da instância.

3.3. Pacote main

O pacote main é composto de classes que possuem apenas métodos estáticos que são utilizados para ler os arquivos de entrada e gerar os arquivos de saída.

3.3.1. Classe Statistics

Essa classe é responsável por processar o arquivo de entrada estatisticas.txt. Ela implementa basicamente dois métodos print e createOutputFile, cada um dos dois possui um overload para cada um dos dois tipos de estatísticas que podem ser solicitadas (por esporte ou quadro de medalhas).

- private static FileOutputStream createOutputFile()
    - Cria e retorna o arquivo de saída para o Quadro de medalhas.
- private static FileOutputStream createOutputFile(AbstractSport sport)
    - Cria e retorna o arquivo de saída para as estatísticas do esporte.
- private static void print(AbstractSport sport)
    - Chama o método createOutputFile e descarrega o conteúdo de sport.toString (estatísticas do esporte) no arquivo criado.
- private static void print(CountryContainer countries)
    - Assim como acima, cria o arquivo e descarrega o quadro de medalhas no arquivo.
- public static void printAll(String fileName, SportContainer sports, CountryContainer countries)
    - Processa todo o arquivo fornecido como argumento e chama o método print de acordo com a entrada.

3.3.2. Classe Util

Possui dois métodos estáticos que auxiliam a leitura de arquivos.

3.3.3. Classe User

É a classe responsável pelo método main. Além disso possui o método getAthletesResults que basicamente processa o arquivo de entrada atletas.txt de

forma a criar os atletas informados e registrar seus resultados em cada uma das modalidades disputadas.

4. Testes

Foram realizado um teste cuja competição simulada continha 5 países, 5 esportes e 20 atletas. Foram pedidas estatísticas sobre o esporte “Corrida”, o esporte “Ginástica Artística” e o Quadro de Medalhas. Os resultados foram os exibidos a seguir:

- Estatísticas sobre a corrida (arquivo estatistica-1-1.txt)

```
Corrida

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

- Estatísticas sobre a ginástica artística (arquivo estatistica-1-5.txt)

```
Ginástica artística

Stanislav  9,30
Jeremy     9,25
Honda      9,08
```

- Quadro de medalhas (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
```

5. Conclusão

Uma das principais dificuldades encontradas no desenvolvimento do projeto foi a entrada de informações a partir de arquivos txt, já que para realizar tal tarefa em Java é necessário iniciar três classes diferentes. Outra dificuldade foi organizar diversos atletas, países e esportes de maneira prática. Esta dificuldade logo foi superada com o melhor entendimento dos Containers e com a generalização de todos os esportes diferentes em uma classe AbstractSport.

O restante do trabalho correu sem grandes problemas, considerando que ambos integrantes do grupo tinham algum conhecimento prévio sobre orientação a objetos e algum conhecimento superficial sobre a linguagem Java.

6. Bibliografia

- http://www.oxfordmathcenter.com/drupal7/node/35
- https://docs.oracle.com/javase/7/docs/api/java/util/TreeMap.html
- https://www.dotnetperls.com/format-java
- https://docs.oracle.com/javase/tutorial/essential/io/formatting.html
- http://docs.oracle.com/javase/8/docs/api/java/nio/file/Files.html#lines-java.nio.file.Path-
- http://www.tutorialspoint.com/java/java_files_io.htm