



## Trabalho Prático 2º bimestre Ultimate Fighting Championship

Valor: 20 pontos

Data de entrega: 05/06/2019

Apresentação - Seminário de POO: 06/06/2019

O objetivo deste trabalho prático é familiarizar o aluno com os principais conceitos da Orientação a Objetos e com a programação utilizando a linguagem Java. Para realização do trabalho o aluno deverá implementar um jogo descrito a seguir. Os princípios da Orientação a Objetos devem ser utilizados.

A presente proposta de trabalho é continuação do trabalho do 1º bimestre. Neste trabalho devem ser observados atentamente os requisitos técnicos necessários da Seção 5.

O trabalho consiste no processamento de informações sobre as ligas do UFC organizadas por anos, extrair algumas informações sobre essa base de dados e apresentar os resultados alcançados.

### 1. Leitura do arquivo de entrada

Arquivo de entrada contendo os dados a serem computados.

Devem ser processados 10 arquivos de entrada, sendo que esses arquivos são organizados da mesma forma como seguem a descrição abaixo.

Cada linha do arquivo corresponde a uma ação dos dois lutadores. Essas ações devem ser computadas no trabalho. Os arquivos são liga2010, liga2011, liga2012, liga2013, liga2014, liga2015, liga2016, liga2017, liga2018 e liga2019.

Para facilitar a divisão de cada campo do arquivo foi criado um arquivo de exemplo com cada divisão exemplificada: **ExemploComIdentificadoresDeCadaPosiçãoDoArquivo**. Esse não deve ser computado, apenas exemplo.

Os campos de cada linha são separados por ponto e vírgula (;). As linhas são organizadas da seguinte forma:

- **Id:** é um identificador de cada linha no arquivo que figura de **1** até **n**.
- **Contador da Luta:** corresponde a um identificador de cada luta. Cada luta terá 19 linhas no arquivo sendo que cada linha corresponde a uma ação do lutador e do seu oponente.
- **Dados dos lutadores:** As informações dos lutadores são descritas duas vezes por linha do arquivo. Sendo que cada linha representa o lutador e seu oponente.
  - **Nome do lutador:** Corresponde ao nome do lutador.
  - **Categoria do lutador:** Corresponde a categoria do lutador.
  - **País do lutador:** Corresponde ao país do lutador.

- **Sexo do lutador:** Corresponde ao sexo do lutador.
- **Ação dos lutadores:** As ações são descritas duas vezes por linha do arquivo. Sendo que em cada linha são descritas as ações do lutador e seu oponente.
  - **Defesa, Ataque, Soco, Chute e Poder**
- **Modalidade:** Corresponde modalidade de luta do lutador que utilizou no momento da ação (cada linha do arquivo de entrada). As modalidades seguem a seguinte nomenclatura:
  - 0 Boxe
  - 1 Jiu-jitsu
  - 2 Muay thai
  - 3 Wrestling
  - 4 Karatê

A modalidade de maior expressão tem maior número. Por exemplo Karatê que é 4 ganha do boxe que é 0. Isso deve ser usado na regra de combate descrita abaixo.

Segue a mesma regra do trabalho do 1º bimestre, porém agora a informação da ação deverá ser lida do arquivo de entrada. A regra de combate está descrita abaixo. Caso seja um ataque, o ataque poderá ser soco, chute ou poder. O 1 representa a ação. Na primeira linha por exemplo, foi um ataque sendo este ataque um soco.

	Ataque	Soco	Chute	Poder	Defesa
1	1	0	0	0	0
0	0	0	0	0	1
1	0	0	1	0	0

1. Soco e defesa: nenhum jogador perde ponto
2. Chute e defesa: nenhum jogador perde ponto
3. Defesa e defesa: nenhum jogador perde ponto
4. Soco e soco: ganha o jogador com maior habilidade (modalidades de lutas) Exemplo: Karatê ganha do Boxe
5. Chute e chute: ganha o jogador com maior habilidade (modalidades de lutas) Exemplo: Karatê ganha do Boxe
6. Poder contra qualquer outro golpe que não seja poder: ganha o jogador que deu o poder
7. Poder contra poder: ganha o jogador com maior habilidade (modalidades de lutas) Exemplo: Karatê ganha do Boxe

## 2. Modelagem de uma base de dados

As informações adquiridas com a leitura dos arquivos devem ser armazenadas em banco de dados. Para isso projeto um banco de dados que possa armazenar as informações lidas do arquivo: dados dos lutadores, dados sobre cada luta e sobre cada liga.

### 3. Relatório de saída:

Por meio de consulta no banco de dados devem ser apresentadas as seguintes informações (tem que consultar do banco, ou seja essas informações devem estar armazenadas no banco de dados):

- Computar os lutadores vencedores de cada luta
- Computar as lutas ocorridas em cada liga (2015, 2016, 2017, 2018 e 2019)
- Computar os pontos ganhos para cada jogador em cada luta
- Número de vitórias por país
- Número de vitórias por jogador
- Maior vencedor por categoria
- Número de lutas computadas no total
- Nomes dos lutadores, categoria, país, sexo e quantidade de vitórias (ordenado em ordem crescente de vitórias dentro de cada categoria)

### 4. Elementos da orientação a objetos que devem ser utilizados

1. Classes objetos
2. Encapsulamento
3. Herança e composição
4. Interfaces
5. Polimorfismo

### 5. Requisitos técnicos

Obs: alguns desses itens ainda serão vistos na disciplina de POO II

1. Junit
2. Exceções do java
3. JavaDoc
4. Interface gráfica (Swing) em algum ponto do trabalho
5. Uso do git e github
6. Uso do Sonar
7. Uso do Log4j
8. Banco de dados (com JDBC puro ou com Hibernate)
9. Uso de threads em algum ponto do trabalho

### 6. Outras observações:

1. Você poderá inferir alguma especificação que não esteja detalhada neste documento, mas para isso você deverá informar na documentação do jogo.
2. Jogos que não sigam a especificação acima não serão avaliados, então, faça seu jogo, não se preocupe em procurar um igual.

### 7. 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. Entrega do JavaDoc.
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.zip', contendo os seguintes arquivos:

- Todo o projeto utilizado no Eclipse (apenas Eclipse);
- A documentação do trabalho.

### **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 individualmente.
- Trabalhos copiados serão penalizados conforme anunciado.
- Penalização por atraso:  $(2^d - 1)$  pontos, onde  $d$  é o número de dias de atraso.