

Projeto

Programação de Sistemas II

Prof. Leandro Pupo Natale / Prof. Tomaz Mikio Sasaki
1º semestre de 2018



Introdução

- Este projeto corresponde à nota de laboratório do 2º bimestre (os demais exercícios realizados no 2º bimestre contribuirão para a nota de participação).
- O projeto deve ser desenvolvido em grupos de 2 ou 3 alunos.
- A entrega e demonstração do funcionamento do projeto será realizada na aula de laboratório da semana de 21 a 25 de maio.

Preparação

Criação do projeto NetBeans

- a) Crie um projeto no NetBeans.
- b) Baixe o arquivo **JogadoresLiga.jar** que está no Moodle e copie este arquivo dentro do diretório do seu projeto.
- c) Dentro do projeto no NetBeans, clique com o botão direito do mouse em **Bibliotecas** e selecione o item **Adicionar Jar/pasta**. A seguir, selecione o arquivo **JogadoresLiga.jar**.

Biblioteca

Na biblioteca que você baixou e instalou há uma classe chamada **LeitorDadosJogadores** que permite ler os dados dos jogadores, times, ligas e confederações. Esta classe possui somente o método **ler**:

- **String[] ler()**: retorna uma lista, onde cada string corresponde aos dados de um jogador (confederação, liga, time, nome, data de nascimento e local de nascimento) separados pelo caractere ";".

A seguir serão mostrados alguns exemplos de uso desta classe. Preste atenção na explicação do professor para cada exemplo, e a seguir digite e execute o código.

Exemplo 1

Este programa lê e imprime **todos os registros**.

```
import jogadoresliga.LeitorDadosJogadores;

public class App1 {

    public static void main(String[] args) {
        LeitorDadosJogadores leitor = new LeitorDadosJogadores();
        String[] dados = leitor.ler();
        for (String dado : dados) {
            System.out.println("Dados do jogador: " + dado);
        }
        System.out.println("Quantidade de registros: " + dados.length);
    }
}
```

Formato dos registros

Note que cada registro é uma string com as informações do jogador (confederação, liga, time, nome, data de nascimento e local de nascimento) separados pelo caractere ":"

Exemplo:

```
Ásia;J1 League;Júbilo Iwata;Guilherme Santos;05/02/1988;Jequié
```

Para separar estas informações e obter somente uma informação do jogador (por exemplo, o nome) podemos utilizar o método **split** da classe **String**.

Exemplo 2

Este programa separa e imprime apenas os **nomes dos jogadores** (note que o nome corresponde ao 4º campo do registro, por isso utilizamos o índice 3 para acessá-lo).

```
import jogadoresliga.LeitorDadosJogadores;

public class App2 {
    public static void main(String[] args) {
        LeitorDadosJogadores leitor = new LeitorDadosJogadores();
        String[] dados = leitor.ler();
        for(String dado: dados) {
            String[] info = dado.split(";");
            String nome = info[3];
            System.out.println("Nome do jogador: " + nome);
        }
        System.out.println("Quantidade de registros: " + dados.length);
    }
}
```


Exemplo 3

Este programa separa e imprime apenas os nomes dos **times** (note que o time corresponde ao 3º campo do registro, por isso utilizamos o índice 2 para acessá-lo).

O nome de cada time é impresso várias vezes porque há vários jogadores em cada time.

```
import jogadoresliga.LeitorDadosJogadores;

public class App3 {
    public static void main(String[] args) {
        LeitorDadosJogadores leitor = new LeitorDadosJogadores();
        String[] dados = leitor.ler();
        for(String dado: dados) {
            String[] info = dado.split(";");
            String time = info[2];
            System.out.println("Nome do time: " + time);
        }
        System.out.println("Quantidade de registros: " + dados.length);
    }
}
```

Exemplo 4

Este programa também separa e imprime apenas os nomes dos **times**, mas imprime apenas uma vez o nome de cada time.

```
import java.util.ArrayList;
import java.util.List;
import jogadoresliga.LeitorDadosJogadores;

public class App4 {
    public static void main(String[] args) {
        LeitorDadosJogadores leitor = new LeitorDadosJogadores();
        String[] dados = leitor.ler();
        List<String> times = new ArrayList<>();
        for(String dado: dados) {
            String[] info = dado.split(";");
            String time = info[2];
            if (!times.contains(time)) {
                times.add(time);
            }
        }
        for (String time: times) {
            System.out.println("Nome do time: " + time);
        }
        System.out.println("Quantidade de times: " + times.size());
    }
}
```

Exemplo 5

Este programa separa e imprime apenas os nomes das **ligas**.

```
import java.util.ArrayList;
import java.util.List;
import jogadoresliga.LeitorDadosJogadores;

public class App5 {
    public static void main(String[] args) {
        LeitorDadosJogadores leitor = new LeitorDadosJogadores();
        String[] dados = leitor.ler();
        List<String> ligas = new ArrayList<>();
        for(String dado: dados) {
            String[] info = dado.split(";");
            String liga = info[1];
            if (!ligas.contains(liga)) {
                ligas.add(liga);
            }
        }
        for (String liga: ligas) {
            System.out.println("Nome da liga: " + liga);
        }
        System.out.println("Quantidade de ligas: " + ligas.size());
    }
}
```

Exemplo 6

Este programa separa e imprime apenas os nomes das **confederações**.

```
import java.util.ArrayList;
import java.util.List;
import jogadoresliga.LeitorDadosJogadores;

public class App6 {
    public static void main(String[] args) {
        LeitorDadosJogadores leitor = new LeitorDadosJogadores();
        String[] dados = leitor.ler();
        List<String> confederacoes = new ArrayList<>();
        for(String dado: dados) {
            String[] info = dado.split(";");
            String confederacao = info[0];
            if (!confederacoes.contains(confederacao)) {
                confederacoes.add(confederacao);
            }
        }
        for (String confederacao: confederacoes) {
            System.out.println("Nome da confederação: " + confederacao);
        }
        System.out.println("Quantidade de confederações: " + confederacoes.size());
    }
}
```

Exemplo 7

Este programa separa e imprime o nome, a data de nascimento e o local de nascimento de cada jogador.

```
import jogadoresliga.LeitorDadosJogadores;

public class App7 {

    public static void main(String[] args) {
        LeitorDadosJogadores leitor = new LeitorDadosJogadores();
        String[] dados = leitor.ler();
        for(String dado: dados) {
            String[] info = dado.split(";");
            String nome = info[3];
            String dataNascimento = info[4];
            String localNascimento = info[5];
            System.out.println(nome + " nasceu no dia "
                               + dataNascimento + " em "
                               + localNascimento);
        }
        System.out.println("Quantidade de registros: " + dados.length);
    }
}
```

Desenvolvimento do projeto

Jogador, Time, Liga e Confederação

Sabendo que:

- Cada jogador tem seu nome, data e local de nascimento e pertence a um único time.
- Cada time tem seu nome e pertence a uma única liga.
- Cada liga tem seu nome e pertence a uma única confederação.

Declare em Java as classes **Jogador**, **Time**, **Liga** e **Confederacao** de forma a obedecer estas definições.

Jogos e Ligas

Cada jogo possui dois times.

Nosso sistema deverá organizar os jogos de cada liga de acordo com os seguintes critérios:

- Se a liga tiver um número de times menor ou igual a 10, os jogos serão organizados no formato **todos contra todos com turno e retorno**.
- Se a liga tiver um número de times maior do que 10, os jogos serão organizados no formato **todos contra todos em turno único**.

Declare em Java as classes **Jogo** e **Liga** de forma a obedecer a estas definições. A classe **Liga** deverá ter um método **organizarJogos** que deverá retornar a lista de jogos desta liga.

Programa

Desenvolva um programa que:

- a) Utilize a classe **LeitorDadosJogadores** para ler as informações da base de dados.
- b) Crie as instâncias das classes **Jogador**, **Time**, **Liga** e **Confederacao** correspondentes às informações lidas.
- c) Apresente um menu principal com as seguintes opções:

MENU PRINCIPAL:

(1) Listar confederações

(0) SAIR

Escolha uma opção: __

Programa (continuação)

d) Se o usuário escolher a opção Listar confederações, o programa deverá listar os nomes das confederações e solicitar que o usuário escolha uma delas (o exemplo abaixo supõe que as confederações cadastradas são Europa e Ásia):

LISTA DE CONFEDERAÇÕES:

(1) Europa

(2) Ásia

(0) VOLTAR

Escolha uma confederação (ou VOLTAR): ____

Programa (continuação)

e) Após o usuário escolher um item da LISTA DE CONFEDERAÇÕES, o programa deverá apresentar os nomes das ligas da confederação escolhida e solicitar que o usuário escolha uma delas (o exemplo abaixo supõe que o usuário selecionou a confederação Ásia e que há apenas 4 ligas nesta confederação):

LISTA DE LIGAS DA CONFEDERAÇÃO ÁSIA:

- (1) A-League
- (2) Hong Kong PL
- (3) League One
- (4) Super League

(0) VOLTAR

Escolha uma liga (ou VOLTAR): __

Programa (continuação)

f) Após o usuário escolher um item da LISTA DE LIGAS DA CONFEDERAÇÃO, o programa deverá apresentar um menu com as seguintes opções (o exemplo abaixo supõe que o usuário escolheu a opção **Super League** no menu anterior):

LIGA SUPER LEAGUE (CONFEDERAÇÃO ÁSIA):

(1) Listar times

(2) Listar jogos

(0) VOLTAR

Escolha uma opção: __

Programa (continuação)

g) Se o usuário escolher a opção LISTAR TIMES no menu do item (f), o programa deverá apresentar os nomes dos times da liga escolhida e solicitar que o usuário escolha um deles (o exemplo abaixo supõe que o usuário selecionou a liga **Super League** e que há apenas 5 times nesta liga):

```
TIMES DA LIGA SUPER LEAGUE (CONFEDERAÇÃO ÁSIA):
```

```
(1) BJ Renhe  
(2) Changchun  
(3) HB CFRC  
(4) Henan  
(5) Pahang FA
```

```
(0) VOLTAR
```

```
Escolha um time (ou VOLTAR): __
```

Programa (continuação)

h) Após o usuário escolher um time no menu anterior, o programa deverá apresentar os nomes dos jogadores do time escolhido e solicitar que o usuário escolha um deles (o exemplo abaixo supõe que o usuário selecionou o time **Henan** e que há 11 jogadores no time):

JOGADORES DO TIME HENAN (LIGA SUPER LEAGUE DA CONFEDERAÇÃO ÁSIA):

- (1) Guoyuan Yang
- (2) Christian Bassogog
- (3) Changjie Du
- (4) Juan Cala
- (5) Qi Jin
- (6) Hao Chen
- (7) Cheng Long
- (8) Yao Lu
- (9) Xingyu Ma
- (10) Guoming Wang
- (11) Wenxiang Gu

(0) VOLTAR

Escolha um jogador (ou VOLTAR): __

Programa (continuação)

i) Após o usuário escolher um jogador no menu anterior, o programa deverá apresentar os dados do jogador escolhido escolhido (o exemplo abaixo supõe que o usuário selecionou o jogador **Christian Bassogog**):

CHRISTIAN BASSOGOG (DO TIME HENAN, LIGA SUPER LEAGUE, CONFEDERAÇÃO ÁSIA):

- Data de nascimento: 18/10/1995
- Local de nascimento: Bamenda

Pressione ENTER para voltar...

Programa (continuação)

j) Se o usuário escolher a opção LISTAR JOGOS no menu do item (f), o programa deverá apresentar os jogos da liga escolhida (o exemplo abaixo supõe que o usuário selecionou a liga **Super League** e que há apenas 5 times nesta liga, por isso o torneio da liga possui **turno e retorno**):

JOGOS DA LIGA SUPER LEAGUE (CONFEDERAÇÃO ÁSIA):

- BJ Renhe vs. Changchun
- Changchun vs. BJ Renhe
- BJ Renhe vs. HB CFFC
- HB CFFC vs. BJ Renhe
- BJ Renhe vs. Henan
- Henan vs. BJ Renhe
- BJ Renhe vs. Pahang FA
- Pahang FA vs. BJ Renhe
- Changchun vs. HB CFFC
- HB CFFC vs. Changchun
- Changchun vs. Henan
- Henan vs. Changchun
- Changchun vs. Pahang FA
- Pahang FA vs. Changchun
- HB CFFC vs. Henan
- Henan vs. HB CFFC
- HB CFFC vs. Pahang FA
- Pahang FA vs. HB CFFC
- Henan vs. Pahang FA
- Pahang FA vs. Henan

(0) VOLTAR

Pressione ENTER para voltar...



Bom estudo!



The seal of the University of Mackenzie is a circular emblem. The outer ring contains the text 'Universidade Presbiteriana' at the top and 'Mackenzie' at the bottom. Inside the ring is a shield with a large 'M' in the center. The shield is flanked by the years '1870' on the left and '1952' on the right. Below the shield, a banner reads 'Tradição e Pioneirismo na Educação'.