

Projeto em grupos - 2024 1º semestre - 1,0 ponto - Prof. Ledón

Estrutura de Dados - I

Devemos elaborar um aplicativo Java para processar os dados de um campeonato de atletismo onde só será efetuada a competição na modalidade de 200 m rasos.

Serão utilizadas três classes: **Pessoa** (com os atributos nome e sexo), **Atleta** (um atleta possui os dados nome do atleta: string, país: string, sexo: char ou string, tempo: real; mas, na verdade, a classe Atleta deve ser derivada da classe Pessoa, então herdará suas características, ou seja, os atributos nome e sexo não devem ser declarados novamente na classe Atleta) e uma terceira classe principal para a interface gráfica, chamada **CampeonatoAtletismo2024**.

Este app deverá atender as características seguintes e ter um visual (aproximado) como mostrado na figura.

1. Cadastrar os dados de atletas.

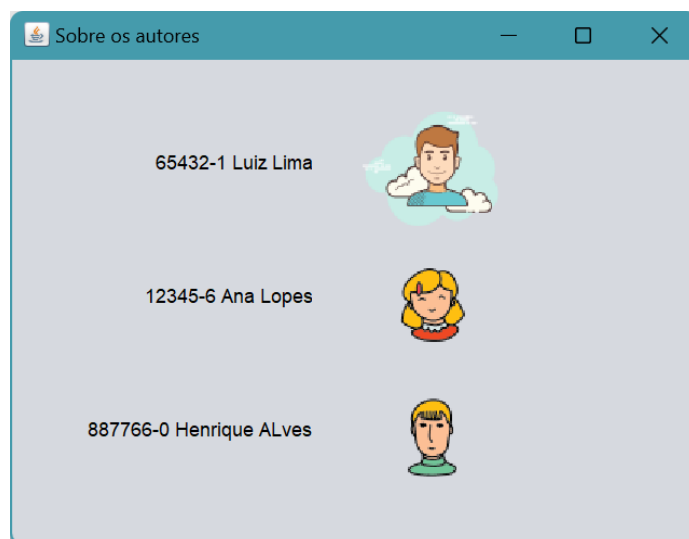
Depois de cadastrar os dados de um atleta, mostre uma mensagem e limpe os dados digitados pelo usuário. Apresente uma mensagem caso aconteça algum problema na gravação; por exemplo, dados vazios ou chave primária repetida.

Mas tem um detalhe importante, ao clicar no botão **Cadastrar**, o **atleta** deverá ser inserido, dependendo do sexo informado, em uma **pilha** se for uma **mulher** ou em uma **fila** se for um **homem**.

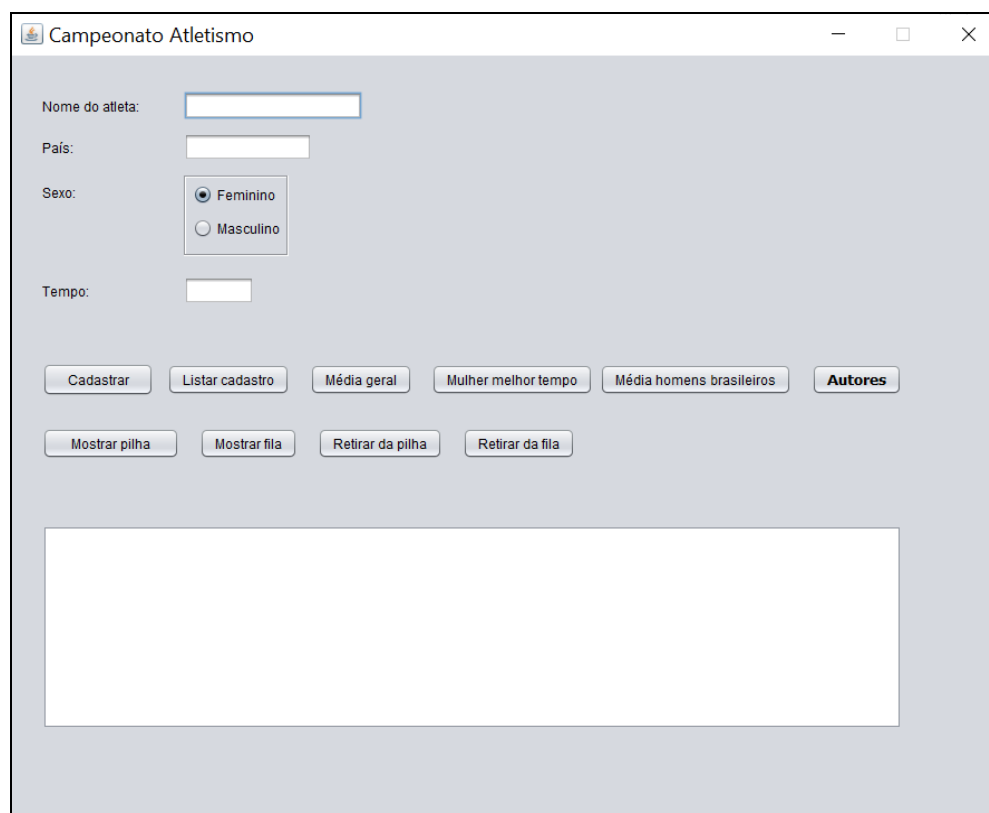
As lógicas dos itens **2. e 3.** a seguir se referem e processam todos os atletas, isto é, devemos processar os dados da pilha de mulheres e da fila de homens, ou seja, considerar os dados de todos os atletas, homens e mulheres. Apresente mensagens adequadas nos casos em que: a pilha ou a fila ou ambas se encontrem vazias (**2. e 3.**) ou não existam homens (**5.**) ou não existam mulheres cadastradas (**4.**).

2. Implementar uma opção que permita listar todos os dados dos atletas (homens e mulheres). Obs: utilize uma área de texto com mecanismo de rolagem, porque poderiam ser muitos. Apresente primeiro os dados das mulheres que se encontram na pilha, obrigatoriamente em ordem alfabética e depois os dados dos homens na fila, também em ordem alfabética. Sugestão: utilize a função `toArray()` para não alterar as estruturas de dados e utilize qualquer método de ordenação estudado (Bubble Sort etc.).
3. Implementar uma opção que permita calcular a média geral de todos os tempos, considerando todos os participantes, mulheres e homens. Sugestão: utilize a função `toArray()` para não alterar as estruturas de dados.
4. Implementar uma opção que determine a atleta (mulher, sexo F ou feminino) com o melhor tempo na competição na modalidade de 200 metros rasos e mostre todos seus dados na tela. Observe que o 'melhor tempo' é o 'menor valor'.

5. Implementar uma opção que calcule e visualize a média dos tempos obtidos pelos atletas brasileiros homens, ou seja, do Brasil e cadastrados com sexo masculino.
6. As lógicas dos botões: **Mostrar pilha**, **Mostrar fila**, **Retirar da pilha** e **Retirar da fila**, devem executar exatamente essas operações relacionadas com a **pilha de mulheres** e a **fila** que guarda os atletas **homens**. Apresente mensagens adequadas caso uma estrutura de dados se encontre vazia!
7. Adicione um botão ou opção de menu **Autores**, que visualize em uma **segunda tela** os RGMs e nomes completos dos alunos do grupo. Coloque também (obrigatório) uma foto ou caricatura de cada integrante do grupo (utilize um Label com a propriedade icon para mostrar a figura). Ou seja, este aplicativo deverá ter duas telas (da classe JFrame). Para abrir a segunda tela utilize, por exemplo, os comandos: `Sobre sob = new Sobre();`
`sob.setVisible();` No exemplo a seguir, utilizo ícones gratuitos do site [icons8.com](https://www.icons8.com). Somente os alunos declarados nesta tela serão considerados com nota neste Projeto.



Visual aproximado da tela principal em CampeonatoAtletismo2024



The screenshot shows a web application window titled "Campeonato Atletismo". It features a form with the following fields and controls:

- Nome do atleta:
- País:
- Sexo: ☒ Feminino ☐ Masculino
- Tempo:

Below the form, there are two rows of buttons:

- Row 1: Cadastrar, Listar cadastro, Média geral, Mulher melhor tempo, Média homens brasileiros, Autores
- Row 2: Mostrar pilha, Mostrar fila, Retirar da pilha, Retirar da fila

At the bottom of the window is a large, empty rectangular box, likely intended for displaying a list of athletes or results.

Observações finais obrigatórias

- Os grupos deverão estar compostos preferivelmente por **4 ou 5 alunos**. **Não é permitida a elaboração individual deste Projeto**, porque visa também a avaliação de habilidades para trabalho em equipe.
- Apenas um aluno da equipe deverá enviar o Projeto (app)** pelo Blackboard, dentro de um arquivo compactado .zip do projeto NetBeans, mas a **correção** do Projeto deverá ser feita **no laboratório**.