

# Programação Orientada aos Objetos

## 2023/2024

### POOTrivia

---

#### Relatório

#### **Alunos:**

André Magalhães de Carvalho 2020237655  
Daniel Coelho Pereira 2021237092

## Introdução:

No âmbito da cadeira de Programação Orientada aos Objetos foi-nos proposto o desenvolvimento de uma aplicação que contém um jogo de perguntas e respostas.

Estas perguntas podem ser de 3 áreas diferentes: artes, desporto e ciências. Dentro da área desporto as perguntas podem ser sobre natação, ski e futebol. Todas as informações sobre as perguntas são transferidas para o programa através de um ficheiro de texto “pootrivia.txt”. Após as 5 perguntas que cada jogo tem, as características do jogo do jogador são guardadas num ficheiro de objetos. Esses ficheiros de objetos são comparados e é apresentado um top 3.

## Classes:

### POOTrivia

Na classe POOTrivia, a função *CriaPergunta()* cria e retorna uma lista de perguntas a partir da leitura de um arquivo chamado “pootrivia.txt”.

Nela também está a função *main* que inicia a interface gráfica e a aplicação.

### Pergunta

Trata-se de uma classe abstrata que implementa a interface *Serializable*.

Serve como uma base para implementar vários tipos de perguntas. As suas subclasses apresentam implementações mais específicas.

### Artes

Classe “filha” da classe Pergunta e representa uma pergunta específica à categoria Artes.

Tem um método *calcularPontuação()* que retorna a pontuação da já pergunta majorada(x10).

### Ciências

Classe “filha” da classe Pergunta e representa uma pergunta específica à categoria Ciências.

Tem um método *calcularPontuação()* que retorna a pontuação da já pergunta majorada(+5).

Também é de realçar que tem o método *getRespostaDifícil()* que retorna escolhas múltiplas onde é mais difícil acertar a correta.

### Desporto

Classe “filha” da classe Pergunta e representa uma pergunta específica à categoria Desporto.

Esta classe vai ter 3 classes “filhas”: Futebol, Ski e Natação.

### Futebol

Classe “filha” da classe Desporto e representa uma pergunta específica à subcategoria Futebol.

Tem um método *calcularPontuação()* que retorna a pontuação da já pergunta majorada(+4).

### Ski

Classe “filha” da classe Desporto e representa uma pergunta específica à subcategoria Ski.

Tem um método *calcularPontuação()* que retorna a pontuação da já pergunta majorada((pontuaçãoBase+4)x2).

### Natação

Classe “filha” da classe Desporto e representa uma pergunta específica à subcategoria Natação.

Tem um método *calcularPontuação()* que retorna a pontuação da já pergunta majorada(+13).

### Jogador

A classe Jogador representa um jogador no contexto de um jogo. É caracterizado por um nome, a data de início do jogo, uma lista com as perguntas que respondeu acertadamente, uma lista com as perguntas que respondeu erradamente.

Tem métodos que permitem retornar as características acima e os abaixo representados:

1. *calcularPontuaçãoTotal(ArrayList<Pergunta> perguntasCertas)*: calcula a pontuação total do jogador com base nas respostas corretas.
2. *formatarDataHora(LocalDateTime dataHora)*: formata a data e a hora para um formato apropriado.
3. *extrairIniciais(String nome)*: extrai as iniciais do nome do jogador.
4. *criarNomeFicheiro(LocalDateTime dataHora, String nomeJogador)*: cria um nome para o ficheiro de objetos a criar.
5. *salvarDadosJogador(Jogador jogador)*: salva os dados do jogador num ficheiro de objetos.
6. *loadObjectsFromDirectory()*: carrega os objetos dos ficheiros de objetos.
7. *top3()*: retorna uma String com o top 3.
8. *toString()*: retorna uma representação em String dos atributos desta classe.

### POOTriviaApp

Trata-se de uma interface gráfica (GUI) que representa o menu inicial que contem os botões: botãoNovoJogo e botãoSair.

Ao clicar no botãoSair a aplicação é fechada.

Ao clicar no botãoNovoJogo fecha o menu e é iniciada a apresentação das perguntas com a classe ApresentarPergunta.

## ApresentarPergunta

A ApresentarPergunta é uma classe Java que estende JFrame, projetada para gerenciar a interação do usuário com um jogo de perguntas e respostas. Ela lida com a apresentação de perguntas, a coleta de respostas e a administração do fluxo do jogo.

1. Construtor: O construtor recebe uma lista de perguntas (ArrayList<Pergunta>) e inicializa a interface gráfica do usuário (GUI), configurando o tamanho, layout e visibilidade do frame.
2. Método pergunta: Este método é responsável por configurar e exibir uma pergunta aleatória e suas respectivas opções de resposta. A escolha da pergunta é feita evitando perguntas já respondidas, sejam elas corretas ou erradas. O método ajusta o layout do texto da pergunta e cria um painel de botões para as respostas, adaptando-se à categoria da pergunta atual.
3. Classe Interna ButtonClickListener: Implementa ActionListener e é responsável por gerenciar ações do botão. Quando um botão é clicado, o método actionPerformed verifica se a resposta selecionada está correta, atualiza as listas de perguntas certas ou erradas e invoca o método caminho.
4. Método caminho: Este método decide se o jogo deve continuar com uma nova pergunta ou terminar. Se todas as 5 perguntas forem feitas, o jogo termina, os dados do jogador são salvos e a GUI do top 3 é exibida.
5. Método top3GUI: Apresenta uma interface gráfica que mostra as três melhores pontuações. Ele é chamado ao final do jogo para exibir o desempenho do jogador em comparação com os recordes anteriores.
6. Método cortaRespostasImprime: Seleciona e exibe um número limitado de respostas baseado na categoria da pergunta. Este método é vital para apresentar opções de resposta de forma aleatória e equilibrada.
7. Método escolherPerguntaRandom: Escolhe uma pergunta aleatoriamente da lista fornecida, garantindo que não seja uma pergunta já respondida, seja correta ou incorretamente.
8. Método obterNomeUsuario: Solicita ao usuário que insira seu nome e garante que um nome válido seja fornecido.

## Formato do Ficheiro de texto:

O ficheiro de texto “pootrivia.txt” tem o seguinte formato de texto:

### **-para perguntas da categoria Artes:**

categoria;texto;respostas;respostaCorreta

### **-para perguntas da categoria Desporto/Futebol:**

categoria;texto;respostas;respostasCorreta;jogadores;jogadorResposta

### **-para perguntas das categorias Desporto/Natação e Desporto/Ski:**

categoria;texto;respostas;respostasCorreta (verdadeiro ou falso)

-para perguntas da categoria Ciências:

categoria;texto;respostasrespostaDificil;respostaCorreta

## Conclusão

A realização deste trabalho prático permitiu aprimorarmos as nossas competências de Programação Orientada aos Objetos. Este projeto foi também vantajoso no que toca a Herança e Polimorfismo, dois conceitos tão importantes nesta cadeira.

Ao refletirmos sobre a concretização deste projeto percebemos que a POO permite realizar programas bastante complexos e eficientes.