Programação Orientada aos Objetos

Licenciatura em Engenharia Informática

2023

Relatório | Projeto

POOTrivia

Docente:

Luís Filipe Vieira Cordeiro

Discentes:

Luís Miguel Ferreira Vaz (2022214707)

Sérgio Lopes Marques (2022222096)

Índice

Introdução

O objetivo do projeto realizado prende-se com o desenvolvimento de um jogo de perguntas e respostas na linguagem de programação java. O jogo tem por base proporcionar ao utilizador um questionário com cinco perguntas. Estas perguntas podem ser de diversas áreas (artes, desporto e ciência) e subáreas (futebol, ski e natação). O jogo deve ainda ter em conta o fator dificuldade, que promove determinadas alterações na forma como as opções de resposta são apresentadas ao utilizador.

Manual do Utilizador:

1. Ao executar pela primeira vez o programa, o utilizador será confrontado com a janela inicial do jogo, que lhe apresentará duas opções: “Jogar” e “Sair”. Se pretender começar o jogo deve selecionar a opção “Jogar”. Caso pretenda sair do jogo deve selecionar a opção “Sair”.

2. Uma vez selecionada a opção “Jogar”, começará o jogo “pooTrivia”, que consiste em uma sequência de cinco perguntas de diferentes categorias. Para cada pergunta o utilizador deve ler atentamente o enunciado que aparecerá o topo da janela, e selecionar apenas a opção que considerar correta.

3. No final de responder à pergunta será confrontado com uma mensagem que lhe dirá se acertou a pergunta e consequentemente o número de pontos que amealhou. Ou caso tenha selecionado uma opção errada, o jogo mostrará a mensagem a informá-lo que selecionou uma opção errada. Independentemente da mensagem que lhe aparecer deve selecionar “Ok” de forma a prosseguir com o jogo.

4. No final de responder às cinco perguntas do jogo, será confrontado com uma janela onde lhe será pedido o Nome de Utilizador, este será o nome que ficará associado à sua prestação no jogo que acabou de finalizar.

5. Depois de inserir o Nome de Utilizador, o programa apresentará uma lista com os melhores resultados já obtidos no jogo.

6. Por fim, o programa mostrar-lhe-á uma janela de fim de jogo, onde será perguntado se pretende jogar novamente. Caso pretenda repetir o jogo, deverá selecionar a opção “Sim” que o levará de volta para o ecrã inicial (ponto 1). Caso pretenda sair deve selecionar “Não” e o jogo fechar-se-á.

Classe “Perguntas”

A classe “Perguntas” é a classe principal de todas as perguntas, a chamada superclasse. Todas as outras classes (as suas subclasses), que correspondem às categorias e subcategorias das perguntas (Artes, Ciências, Desporto (que inclui Futebol, Ski e Natação)), herdam os seus atributos e métodos.

O construtor desta classe é composto por apenas dois atributos: o “enunciado” (do tipo String) que corresponde ao enunciado da pergunta lida a partir do ficheiro de texto e um atributo do tipo int “pontosPergunta”, que contém o valor da pergunta em questão.

Quanto aos métodos que constituem esta classe, podemos encontrar os getters e os setters para cada um dos atributos e, também o método toString da nossa classe.

Posteriormente, construímos 3 métodos que, através do uso do polimorfismo, implementando-os nas suas subclasses, serão extremamente importantes para um bom funcionamento do nosso jogo.

Um desses métodos é o “majoracao”, que irá calcular (tendo em conta a majoração de cada categoria), para cada subclasse, os pontos a serem atribuídos ao utilizador caso este escolher a opção correta. De seguida, temos o método “janelaPergunta”, que irá apresentar a janela do GUI com o enunciado e as opções de resposta referentes à pergunta. Por último, o método “verificaResposta”, tal como o seu nome indica, irá verificar se a resposta dada pelo utilizador está, ou não, correta.

Classe “Artes”

Esta classe, que corresponde às perguntas da categoria das Artes, é uma subclasse da classe “Perguntas”, herdando, então, os seus métodos e atributos. Do construtor desta classe, fazem parte, também, os atributos “correctAnswerArte” e “OpcoesRespostaArte” e os seus getters e setters, bem como o método toString desta subclasse.

Nesta classe, o método “majoracao”, caso a resposta à pergunta seja a correta, irá multiplicar o número de pontos por 10, como pedido no enunciado, para mostrar, posteriormente, o número de pontos ganhos ao utilizador. No método “janelaPergunta” irá ser criada uma nova janela com o enunciado da pergunta e as opções de resposta. Opções estas que serão apenas 3, se se tratar de uma pergunta fácil (primeiras 2 perguntas do jogo), ou 5, se a pergunta for difícil. Depois, será feita a verificação da resposta com o método “verificaResposta”, a partir da opção escolhida pleo utilizador e do atributo “correctAnswerArte”, que corresponde à resposta correta.

Classe “Ciências”

A classe “Ciências”, também subclasse da classe “Perguntas”, recebe, no seu construtor, os atributos da superclasse e, para além disso, uma lista de opções fáceis, “OpcoesFacil”, e uma de opções difíceis, “OpcoesDificil”, bem como a resposta correta da questão, “correctAnswerCiencias”. Tal como nas outras classes, estão implementados os métodos getter e setter destes atributos e o método toString correspondente a esta classe.

Nesta classe, o método “majoracao”, chamado se o utilizador responder corretamente à pergunta, acrescentará 5 pontos ao valor da pergunta. Este valor deverá, depois, ser apresentado ao utilizador. Já o método “janelaPergunta” irá abrir uma nova janela, tal como na classe “Artes”, com as informações relativas à pergunta em questão.

Se a pergunta for fácil, será apresentada ao utilizador uma lista de 5 opções, com um grau de dificuldade inferior. Caso se trate de uma pergunta difícil, ocorrerá o contrário. De modo a perceber se a opção escolhida pelo utilizador é a correta, iremos fazer a verificação da resposta, utilizando, novamente, o método "verificaResposta".

Classe "Desporto"

Esta classe, em relação às outras subclasses da classe "Perguntas", tem uma pequena particularidade, que se deve ao facto de estar dividida em 3 subclasses: classes "Futebol", "Ski" e "Natacao". Bem como nas classes "Artes" e "Ciencias", esta classe recebe, por herança, os atributos e métodos da classe "Perguntas".

Uma vez que as perguntas sobre Ski e Natação têm algo em comum (são ambas de resposta Verdadeiro/Falso), esta classe serve de molde para estas categorias e, por isso, definimos no construtor desta classe uma ArrayList do tipo String com as opções "verdadeiro" e "falso".

Nesta classe podemos, também, encontrar os métodos getters e setters dos atributos, bem como o método toString, comuns a todas as outras classes. De resto, é importante realçar que o método "janelaPergunta" desta classe recebe a ArrayList "OpcoesVerdadeiroFalso", com o intuito de construir a janela destinada às perguntas de Ski e Natação. A verificação da resposta correta é realizada da mesma maneira que é feita para as outras classes, utilizando o método verificaResposta.

Classe "Futebol"

Esta classe, apesar de se tratar de uma subclasse da classe "Desporto", não tem muito em comum com a mesma, herdando apenas os atributos "enunciado" e "pontosPergunta", oriundos da superclasse "Perguntas".

Nesta classe, são definidos alguns atributos relativos à lista de opções de nomes de jogador e à lista de opções de números da camisola e, ainda, às opções corretas para cada uma das listas. Consequentemente, estão construídos, na classe, os getters e setters correspondentes a cada um dos atributos em questão.

Quanto ao método "majoracao", caso a resposta à pergunta seja a correta, serão adicionados (3 + 1) pontos à pontuação inicial da pergunta, referentes à majoração da categoria Desporto e da subcategoria Futebol, respetivamente. No método "janelaPergunta", ir-se-á perceber se a pergunta é fácil ou difícil. Caso esta seja do grau de dificuldade "fácil", irá ser usada, como lista de opções de resposta à pergunta, a lista de opções com os nomes dos jogadores. Caso contrário, a lista de opções é a referente ao número da camisola dos jogadores. O mesmo acontece no método "verificaResposta". Se a pergunta for fácil, a verificação será feita através da opção correta para os nomes dos jogadores. Pelo contrário, se a pergunta for difícil, a verificação passa a ser feita com a solução dos números das camisolas dos jogadores.

Classes "Ski" e "Natação"

Estas classes vão ter muito em comum com a classe Desporto e vão herdar os seus métodos e atributos. A única particularidade que vão ter será no método "majoracao",

onde para cada uma, será efetuado um cálculo diferente, baseado naquilo que nos é dito no enunciado do trabalho. Para a classe "Ski", caso o utilizador acerte uma pergunta desta categoria, a pontuação da pergunta será aumentada em 3 pontos e será multiplicada por 2. No caso da "Natacao", a pontuação será aumentada em (3 + 10) pontos.

Classe "GUI"

É na classe GUI que vão estar os métodos responsáveis pela interação gráfica entre o utilizador e o programa. Nela vão estar os métodos que geram o ecrã inicial, para apresentar as perguntas, para o utilizador inserir o seu nome, top 3 de melhores resultados e o ecrã final.

Classe "Jogo"

Nesta classe serão guardadas todas as informações relativas a um respetivo jogo: o nome do utilizador, a data e hora do jogo e as respostas corretas e erradas do utilizador. Estes atributos entram, portanto, no construtor desta classe, bem como o atributo "pontuacaoJogo", que é utilizado, apenas no momento do cálculo do top 3.

De seguida, estão definidos todos os getters e setters destes atributos, assim como o método "compareTo", que nos permite comparar todos os jogos (objetos Jogo), a partir do atributo "pontuacaoJogo".

Classe "pooTrivia"

É na classe pooTrivia onde vai ser executado a função "main" do código e onde vão estar os métodos:

- **"cicloJogo"** - responsável pela execução da sequência dos acontecimentos que vão ter lugar durante o decorrer do programa;
- **"criaStringTop3**" - responsável pela criação das strings que vão ser introduzidas na janela que cria o top3 e que vai ser exibida ao utilizador após o término do jogo;
- **"passarArrayParaArrayList**" - que, mais uma vez, como o nome indica vai receber um array e passar o seu conteúdo para uma arrayList;
- **"perguntasReader**" - que é responsável por fazer a leitura do ficheiro ".txt", realizar o "parsing" , a partir dessa informação cria o objeto respetivo a cada classe e , por fim, armazena a pergunta numa arrayList de perguntas;
- "**calculaNomeFicheiro**" – responsável pela criação da string que vai dar nome ao ficheiro de objetos, vai concatenar a data da criação do ficheiro com a primeira letra de cada nome que o utilizador colocar na janela em que lhe é pedido o nome;
- "**sortJogosByTotalPontos"** – responsável pela ordenação da lista de jogos que recebe;

Método “lerFicheiroObjetos”

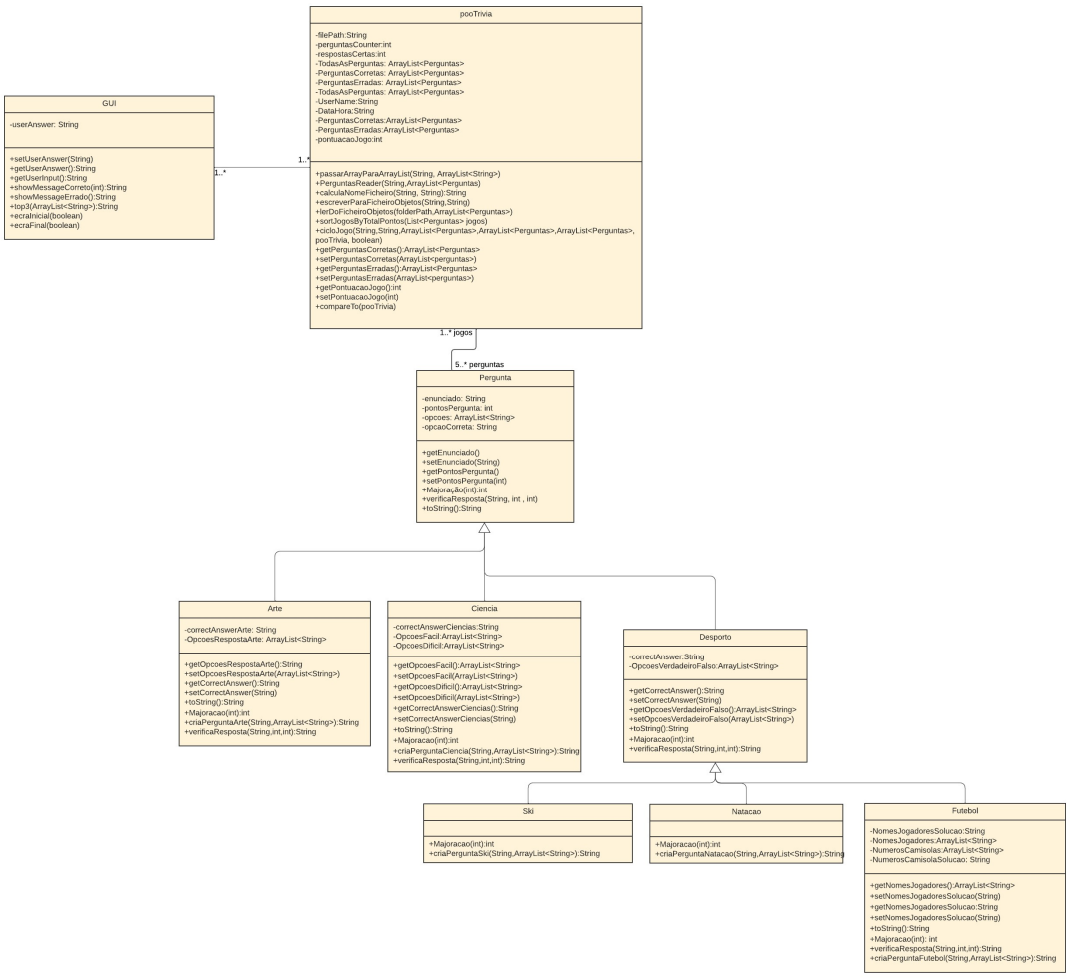
O método lerFicheiroObjectos começa por fazeres as respetivas verificações a nível da existência da pasta e verifica se existem ficheiros para ler dentro da pasta. Feito isto começa a iterar sobre cada ficheiro existente na pasta, para cada ficheiro vai ler o objeto “Jogo” e aceder à arrayList de perguntas certas, a partir daí calcula a pontuação geral do jogo acedendo às pontuações de cada pergunta (tendo em conta a majoração de cada uma). Uma vez calculada a pontuação final, vai dar setPontuacaoJogo de forma a guardar no objeto jogo a pontuação total e vai armazenar o jogo numa ArrayList. É importante denotar que a pontuação nunca é guardada no ficheiro em momento algum, apenas existe um período entre o cálculo da pontuação final e a exibição da mesma no top3 em que essa pontuação está armazenada no objeto de forma a ser possível sua a ordenação com base nessa mesma pontuação. O método vai dar return da última pontuação calculada, que corresponde à pontuação do ficheiro mais recente, de forma a ser possível exibir a pontuação do jogo mais recente ao utilizador.

UML

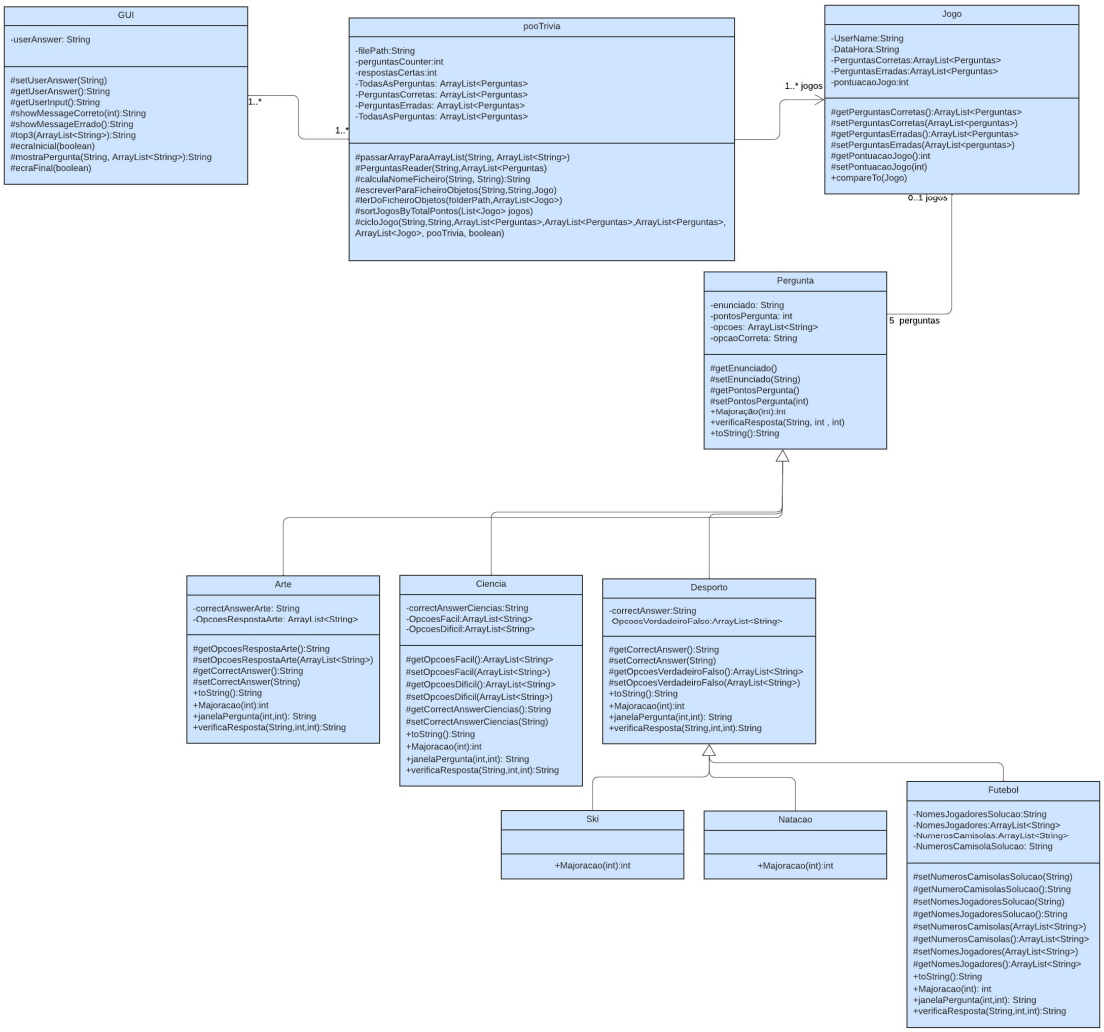
Site utilizado para a criação de ambos os diagramas UML:

https://www.lucidchart.com/pages/pt

UML Inicial:



UML Final:



Discussão

Ao compararmos os diagramas UML inicial e final são notórias algumas diferenças: Inicialmente considerámos a abordagem de criar um método para cada classe e subclasse que iria criar uma janela com as especificações de cada tipo de pergunta. Daí a existência do método criaPergunta(Respetiva Classe) no primeiro UML, que iria receber o enunciado e a lista de opções para cada tipo de pergunta. Mas chegamos à conclusão de que o método que cria as perguntas recebe sempre a mesma base para qualquer classe ou subclasse, assim optamos por criar um método que gera uma pergunta genérica na classe GUI ("mostraPergunta"), que vai ser adaptado, usando o método "janelaPergunta" para receber as respetivas informações e condicionantes de cada pergunta.

Nota ainda para o facto de que no UML inicial, não existe a classe Jogo, esta foi pensada posteriormente, aquando da necessidade de armazenar a informação do jogo e a escrever para o ficheiro de objetos. Inicialmente antes de começar a desenvolver o jogo, acreditávamos que seria possível armazenar e escrever a informação respetiva ao jogo utilizando a classe pooTrivia, mas rapidamente nos apercebemos do papel fundamental que a classe jogo teria quer na organização do nosso projeto.

Conclusão

Para concluir, com o desenvolvimento deste projeto, pusemos em prática conteúdos essenciais no ramo da programação tais como a hierarquia e o polimorfismo que foram lecionados nas aulas. O facto de o projeto ter sido desenvolvido na linguagem de programação java, deu-nos também uma visão da grande utilidade e das inúmeras funcionalidades presentes na linguagem que auxiliam no desenvolvimento de aplicações e jogos como foi o caso deste projeto. Para além de que, por o trabalho ser feito em grupos, aprendemos também a discutir, com outras pessoas, diferentes abordagens para os mesmos problemas e melhorar a nossa forma de trabalhar e cooperar em grupo.

Referências

- https://docs.oracle.com/javase/tutorial/uiswing/index.html
- Java Swing, Second Edition – Loy, Eckstein, Wood, Elliott, Cole – O'Reilly Associates
- Java AWT Reference – Zukowski – O'Reilly Associates
- https://developer.ibm.com/articles/the-class-diagram/
- https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/
- Slides disponibilizados na plataforma UC Student da disciplina Programação Orientada aos Objetos.