



POOTRIVIA



Gustavo Oliveira Pinto nº 2022234185

Pedro Miguel Queiroga Trindade nº 2022230848

Programação Orientada a Objetos

Introdução

Este projeto tem como objetivo aprofundar toda a matéria lecionada na cadeira de Programação Orientada a Objetos. Tendo mais em vista as questões sobre polimorfismo, herança, GUI, ficheiro de texto e de objetos e UML.

Realizaremos um “quiz” onde aplicamos todos esses conceitos lecionados.

Este projeto foi realizado por 2 alunos da licenciatura em Engenharia Informática da Universidade de Coimbra, Pedro Trindade e Gustavo Pinto.

Classes e Heranças

Começamos por criar a classe abstrata “Pergunta”, sendo esta a classe “Mãe”. Outras classes vão herdar desta classe, pois possuem atributos iguais, como por exemplo: String com a pergunta, String com as opções e String com a opção correta.

Logo as classes, “Desporto”, “Artes” e “Ciências” herdam da classe “Pergunta”.

Estas 3 classes diferem entre si principalmente pela pontuação a atribuir em cada pergunta e pelo tipo de opções.

Existem 3 subclasses que herdam da classe “Desporto”, as subclasses: “Natação”, “Futebol” e “Ski”, pois estas perguntas possuem a mesma pontuação que as de desporto, mais uma pontuação adicional.

Estas 3 subclasses diferem entre si pela pontuação a atribuir em cada pergunta e pela forma como é feita a pergunta, e as opções da pergunta.

Criamos a classe “TriviaGUI”, que vai ter como funcionalidade a comunicação com o utilizador. Ou seja, vai abrir janelas de interação com a pergunta e as opções dessa mesma pergunta, vai após responder, dizer se acertou ou não e no final do jogo vai apresentar a pontuação final e pedir ao utilizador o nome completo para poder mandar a informação do jogo para um ficheiro de objetos.

A classe “Ficheiro” contém todos os métodos para realizar operações entre ficheiros e programa, ou seja, inicialmente tem a função “leFicheiro” que recebe como parâmetro o nome de um ficheiro e retorna um array de perguntas caso as perguntas sejam válidas. O método “escreveFicheiroObjeto” escreve toda a informação de um jogo (nome do jogador, pontuação e data) num ficheiro de objetos. O método “leFicheiroObjeto” transforma a informação de um ficheiro de objetos e dá print no terminal de toda a informação do jogo.

Por último, teremos a classe “POOTrivia” que é classe onde temos o main, executando o código inicialmente aqui.

Atributos e métodos

Atributos:

Em relação aos atributos, temos atributos que são mais específicos à sua classe, consoante o que o enunciado pede para cada classe. Mas existem certos atributos que são comuns a maior parte das classes e não fazia sentido repetir todos esses mesmo atributos em todas as classes,

uma vez que as classes que estendem de uma outra ficam com a sua herança, ou seja, as propriedades dessa classe para a classe “filha”.

Como por exemplo, as variáveis “perguntas”, “respostas” e “opções” são comuns às classes que envolvem as subclasses.

Métodos:

A classe abstrata “Pergunta” contém alguns métodos abstratos. Ou seja, não tem uma implementação direta na sua classe, mas sim nas classes que estabelecem com a mesma, uma relação de herança, recorrendo a Polimorfismo para as usarmos. Como por exemplo, os métodos “selecionaRespostaPergunta” e “selecionaOpcoesPergunta”. Estes métodos foram colocados como abstratos porque cada classe tem a sua própria execução da função/método.

É também nesta classe que se vai definir as perguntas que vão ser apresentas ao utilizador. Através do método estático “perguntasUtilizador” são escolhidas 5 perguntas aleatórias do array de perguntas previamente carregado a partir do método “leFicheiro”, que vão ser guardadas num ArrayList<Pergunta>. As perguntas guardadas neste ArrayList<Pergunta> com o nome perguntasUtilizador vão ser usadas no jogo.

Como é pedido no enunciado, tem que haver uma soma dos pontos caso o utilizador acerta a pergunta. Para isso nós usamos um método abstrato para realizar essa mesma ação. Ao usar o método “aumentaPontos”, a pontuação de cada pergunta é definida após responder à própria pergunta, retornando o valor total da pontuação nessa pergunta para adicionar a um somador que no fim do programa irá abrir uma janela a mostrar a pontuação final. Este método é também um exemplo de polimorfismo. Cada área tem a sua pontuação definida e como todas as subclasses estendem da classe “Pergunta”, este método é chamado em cada classe e a pontuação é atribuída consoante as características de cada subclasse.

Na classe “Artes”, consoante o nível da pergunta, são apresentadas 3 opções ou todas as opções. Para colocar essa funcionalidade em prática, nós usamos um método de seleção de perguntas. Ou seja, chamamos o método “selecionaOpcoesArtes” que retorna as opções dependendo do nível.

Na classe “Ciências” é pedido que se apresente questões difíceis e questões difíceis. Para tal, chamamos os métodos “selecionaOpcoesPergunta” e “SelecionaRespostaPergunta” que dão return às opções e à resposta, respetivamente, consoante a dificuldade.

Conclusão

Após a execução deste projeto, consolidamos conhecimentos na cadeira de Programação Orientada a Objetos.

Manual de Utilizador

Este jogo consiste em responder a 5 perguntas.

Para cada pergunta tem de seleccionar uma das opções disponíveis.

Após responder às 5 perguntas é apresentado a pontuação final desse jogo e é pedido ao jogador para inserir o seu nome completo para guardar todas as informações sobre esse jogo.