



UNIVERSIDADE D
COIMBRA

Projeto Final

POOTrivia

Programação Orientada aos Objetos

Trabalho Realizado por:

Tiago Nuno Matos Cabo Lopes de Almeida nº 2021221615

João Tomás Grangeia Querido nº 2021240660

O projeto POOTrivia tem como objetivo o desenvolvimento de um jogo de perguntas e respostas, implementado em Java e utilizando uma interface gráfica Java (JFrame). O jogo permite que um jogador responda a 5 perguntas de diferentes áreas, como Artes, Desporto e Ciências, cada uma com características específicas.

Características do Jogo

1. Áreas de Perguntas

- Artes: Perguntas com uma lista de opções de resposta. Se apresentadas antes da 3ª pergunta, apenas três opções são exibidas;
- Desporto: Perguntas com solução "Verdadeiro" ou "Falso". A subárea de futebol apresenta duas listas de opções consoante o número da pergunta do jogo;
- Ciências: Perguntas com uma lista fácil e outra difícil de opções de resposta.

2. Interatividade:

- O jogo é composto por 5 perguntas escolhidas aleatoriamente (sem repetição);
- Após responder a cada pergunta, o jogador recebe feedback sobre a resposta correta e a pontuação obtida;
- No final das 5 perguntas, o jogador fornece o seu nome para que seja guardado um ficheiro objeto na base de dados;
- É apresentada uma lista com 3 melhores pontuações (TOP 3) do jogo, juntamente com a pontuação do jogo atual.

3. Dados:

- No início do programa as perguntas são carregadas a partir de um arquivo de texto, perguntas.txt;
- Após cada jogo, é guardado um ficheiro objeto com o nome do jogador e com a data e hora da conclusão do jogo e com os detalhes das perguntas que o jogador acertou ou falhou;
- O programa permite a impressão de uma lista dos jogadores com as 3 melhores pontuações. Esta lista inclui também os detalhes dos jogos destes 3 jogadores.

Classe *POOTrivia*

A classe POOTrivia em Java é a espinha dorsal de um jogo interativo de trivia, projetado para proporcionar aos usuários uma experiência envolvente e desafiadora. Através de uma interface gráfica amigável, o jogo oferece uma variedade de perguntas, desde verdadeiro/falso até perguntas de submissão de resposta, proporcionando uma experiência diversificada aos jogadores.

O carregamento dinâmico de perguntas a partir do arquivo "pootrivia.txt" permite a expansão fácil do conjunto de perguntas, garantindo uma jogabilidade renovada a cada sessão. A classe gerencia habilmente o fluxo do jogo, apresentando perguntas, opções de resposta e calculando pontuações em resposta às ações do jogador.

O jogo não apenas desafia a capacidade de conhecimento do jogador, mas também oferece uma abordagem visualmente atraente, graças ao uso eficiente de elementos gráficos Swing. A modularidade do código e os princípios de programação orientada a objetos são evidentes, facilitando a manutenção e futuras expansões do jogo.

Ao finalizar uma partida, a classe não apenas exibe a pontuação final do jogador, mas também fornece uma visão dos melhores jogadores em uma janela de liderança. Além disso, a capacidade de salvar as informações do jogador em um arquivo de objeto permite uma experiência personalizada e a possibilidade de acompanhar o progresso ao longo do tempo.

Em resumo, a classe POOTrivia encapsula a essência de um jogo de trivia, combinando elementos visuais atraentes, variedade de perguntas e uma estrutura de código robusta. É um exemplo de como a programação orientada a objetos pode ser aplicada para criar jogos interativos e educacionais de forma eficaz.

Classe *Player*

A classe *Player* desempenha um papel fundamental no jogo de trivia POO Trivia em Java, representando e gerenciando as informações de um jogador. Com atributos como nome, pontuação e listas de perguntas corretas e incorretas, a classe encapsula dados relevantes para a experiência do jogador.

Ao criar um objeto *Player*, o construtor inicializa as estruturas de dados necessárias, oferecendo uma base sólida para o acompanhamento do desempenho do jogador ao longo do jogo. Métodos simples, como *getPlayerName* e *getScore*, proporcionam acesso direto às informações essenciais do jogador.

A capacidade de aumentar a pontuação (*increaseScore*) e registrar perguntas corretas e incorretas (*addCorrectQuestion*, *addIncorrectQuestion*) destaca a interatividade da classe, permitindo uma representação detalhada da performance do jogador.

A implementação da interface *Serializable* possibilita a criação de arquivos de objeto (*createObjectFile*), facilitando a persistência das informações do jogador entre sessões de jogo. A obtenção das iniciais do jogador (*getInitials*) adiciona um toque personalizado à nomenclatura dos arquivos de objeto.

Além disso, o método estático *getTopPlayers* oferece uma funcionalidade crucial, recuperando e exibindo informações dos melhores jogadores com base em suas pontuações. Essa abordagem proporciona uma perspectiva competitiva e incentiva a participação continuada no jogo.

Em resumo, a classe *Player* é uma peça-chave na estrutura do POO Trivia, incorporando funcionalidades que vão desde o gerenciamento de dados do jogador até a persistência e a avaliação do desempenho, contribuindo assim para uma experiência envolvente e personalizada no universo do jogo.

Classe *Questions*

A classe abstrata *Questions* desempenha um papel crucial no projeto do jogo de trivia POO Trivia em Java, atuando como a classe base para diferentes tipos de perguntas. Seu propósito principal é fornecer uma estrutura comum para lidar com elementos compartilhados entre todas as perguntas, como texto da pergunta, opções, resposta correta e pontuação associada.

No âmbito da implementação, a classe é projetada para ser estendida por subclasses que representam tipos específicos de perguntas, como perguntas de Arte, Ciência, Futebol, entre outras. O construtor é responsável por inicializar os atributos comuns com base nos parâmetros fornecidos.

A presença do método *questionType* permite que subclasses forneçam informações específicas sobre o tipo de pergunta que representam. Métodos de acesso simples, como *getQuestion*, *getOptions* e *getCorrectAnswer*, facilitam a obtenção de informações essenciais sobre a pergunta.

A abstração é evidenciada pelos métodos abstratos como ``getAnswers``, ``isCorrectAnswer``, ``returnPoints`` e ``getSublistOptions``, os quais devem ser implementados por cada subclasse para definir comportamentos específicos.

Além disso, destaca-se o método ``isTrueFalseCorrect``, que verifica se uma resposta selecionada é correta para perguntas do tipo Verdadeiro/Falso.

Em resumo, a classe ``Questions`` oferece uma base sólida e flexível para a construção de diferentes tipos de perguntas no jogo POO Trivia, promovendo a reutilização de código e a extensibilidade do sistema.

Classe *ArtesQ*

A classe ``ArtsQ`` representa uma pergunta relacionada às artes no jogo POO Trivia. Essa classe estende a classe abstrata ``Questions`` e fornece implementações específicas para a recuperação de detalhes da pergunta, obtenção de respostas, verificação de correção, cálculo de pontos e criação de uma sublista de opções.

O construtor da classe ``ArtsQ`` recebe o texto da pergunta, uma lista de opções e a resposta correta, inicializando os atributos correspondentes por meio do construtor da classe base.

A implementação do método ``questionType`` retorna o tipo da pergunta, que é "Arts". Os métodos de acesso, como ``getQuestion``, ``getAnswers`` e ``getCorrectAnswer``, fornecem informações essenciais sobre a pergunta e suas opções.

O método ``isCorrectAnswer`` verifica se uma resposta fornecida está correta para a pergunta de artes, comparando-a com a resposta correta.

O método ``returnPoints`` calcula os pontos associados à pergunta de artes, assumindo que cada pergunta dessa categoria vale 50 pontos.

Por fim, o método ``getSublistOptions`` cria uma sublista de opções para a pergunta de artes, embaralhando as opções e selecionando as três primeiras.

Em resumo, a classe ``ArtsQ`` encapsula o comportamento específico das perguntas de artes no contexto do jogo POO Trivia, seguindo a estrutura definida pela classe abstrata ``Questions``.

Classe *ScienceQ*

A classe *ScienceQ* representa uma pergunta relacionada à ciência no jogo POO Trivia. Esta classe estende a classe abstrata *Questions* e fornece implementações específicas para perguntas de ciência.

No construtor, é criado um objeto *ScienceQ* com o texto especificado da pergunta de ciência, uma lista de opções e a resposta correta.

O método *questionType* sobrescreve o método correspondente na classe base para retornar o tipo da pergunta, que é "Ciência" para objetos da classe *ScienceQ*.

Métodos como *getQuestion*, *getAnswers*, *isCorrectAnswer* e *getCorrectAnswer* oferecem acesso ao texto da pergunta de ciência, à lista de respostas possíveis e à resposta correta.

O método *returnPoints* calcula os pontos atribuídos por responder a uma pergunta de ciência, assumindo uma distribuição de pontos de 5 pontos por selecionar a resposta correta e mais 5 pontos adicionais.

O método *getSublistOptions* cria uma sublista embaralhada de opções para a pergunta de ciência, incluindo três opções, e depois as embaralha.

Em resumo, a classe *ScienceQ* encapsula o comportamento específico de perguntas de ciência no contexto do jogo POO Trivia, ao mesmo tempo que segue a estrutura definida pela classe abstrata *Questions*.

Classe *SportsQ*

A classe *SportsQ* é uma classe abstrata que serve como base para representar perguntas relacionadas a desportos no jogo POO Trivia. Esta classe estende a classe abstrata *Questions* e fornece uma estrutura comum para perguntas sobre desportos.

No construtor, é criado um objeto *SportsQ* com o texto especificado da pergunta sobre desportos, uma lista de opções e a resposta correta.

Os métodos *getAnswers* e *isCorrectAnswer* fornecem acesso à lista de respostas possíveis e verificam a correção de uma resposta fornecida para perguntas sobre desportos.

O método *returnPoints* é declarado como um método abstrato, indicando que deve ser implementado por subclasses. Este método define quantos pontos são atribuídos por responder a uma pergunta sobre desportos, permitindo flexibilidade nos cálculos de pontos com base nos requisitos específicos de cada pergunta desportiva.

O método *getSublistOptions* é implementado para criar uma sublista embaralhada de opções para perguntas sobre desportos, incluindo três opções (incluindo a resposta correta), proporcionando variedade e desafio aos jogadores.

Em resumo, a classe *SportsQ* encapsula comportamentos e estrutura comuns para perguntas sobre desportos, permitindo implementações específicas em subclasses, ao mesmo tempo que mantém consistência por meio da classe abstrata *Questions*.

Classe *SwimmingQ*

A classe `SwimmingQ` estende a classe abstrata `SportsQ` e está projetada para representar perguntas relacionadas com natação no jogo POO Trivia. Vamos analisar as suas características principais:

1. **Construtor:**

- A classe possui um construtor que recebe o texto da pergunta sobre natação, uma lista de opções e a resposta correta como parâmetros.
- Chama o construtor da superclasse `SportsQ` para inicializar atributos comuns.

2. **Tipo e Respostas:**

- O método `questionType` especifica que o tipo de pergunta é "Natação."
- O método `getAnswers` retorna a lista de respostas possíveis para a pergunta sobre natação.

3. **Verificação da Resposta Correta:**

- O método `isCorrectAnswer` verifica se uma resposta fornecida está correta para a pergunta sobre natação.

4. **Cálculo de Pontos:**

- O método `returnPoints` calcula e retorna os pontos atribuídos por responder à pergunta sobre natação. A fórmula utilizada envolve uma combinação de pontos fixos (5 e 3) e mais 10 pontos adicionais.

5. **Sublista de Opções:**

- O método `getSublistOptions` cria uma sublista de opções para a pergunta sobre natação. Embaralha as opções e retorna as três primeiras, incluindo a resposta correta.

6. **Verificação Específica de Verdadeiro/Falso:**

- O método `isTrueFalseCorrect` verifica se uma resposta Verdadeiro/Falso fornecida está correta para a pergunta sobre natação. Determina a resposta correta com base nas opções embaralhadas.

Em resumo, a classe `SwimmingQ` fornece funcionalidades específicas para lidar com perguntas relacionadas com natação, incluindo identificação do tipo, validação da resposta correta, cálculo de pontos e gestão de opções. Aproveita a estrutura definida na classe abstrata `SportsQ` ao adaptar o seu comportamento ao domínio das curiosidades sobre natação.

Classe *SkiQ*

A classe `SkiQ` estende a classe abstrata `SportsQ` e foi concebida para representar perguntas relacionadas com esqui no jogo POO Trivia. Vamos examinar as suas principais características:

1. **Construtor:**

- A classe possui um construtor que recebe o texto da pergunta sobre esqui, uma lista de opções e a resposta correta como parâmetros.
- Chama o construtor da superclasse `SportsQ` para inicializar atributos comuns.

2. **Tipo e Respostas:**

- O método `questionType` especifica que o tipo de pergunta é "Esqui."
- O método `getAnswers` retorna a lista de respostas possíveis para a pergunta sobre esqui.

3. ****Verificação da Resposta Correta:****

- O método ``isCorrectAnswer`` verifica se uma resposta fornecida está correta para a pergunta sobre esqui.

4. ****Cálculo de Pontos:****

- O método ``returnPoints`` calcula e retorna os pontos atribuídos por responder à pergunta sobre esqui. A fórmula utilizada envolve uma combinação de pontos fixos (5 e 3) multiplicados por 2.

5. ****Sublista de Opções:****

- O método ``getSublistOptions`` cria uma sublista de opções para a pergunta sobre esqui. Embaralha as opções e retorna as três primeiras, incluindo a resposta correta.

6. ****Verificação Específica de Verdadeiro/Falso:****

- O método ``isTrueFalseCorrect`` verifica se uma resposta Verdadeiro/Falso fornecida está correta para a pergunta sobre esqui. Determina a resposta correta com base nas opções embaralhadas.

Em resumo, a classe ``SkiQ`` fornece funcionalidades específicas para lidar com perguntas relacionadas com esqui, incluindo identificação do tipo, validação da resposta correta, cálculo de pontos e gestão de opções. Aproveita a estrutura definida na classe abstrata ``SportsQ`` ao adaptar o seu comportamento ao domínio das curiosidades sobre esqui.

Classe *SoccerQ*

A classe ``SoccerQ`` estende a classe abstrata ``SportsQ`` e foi desenvolvida para representar perguntas relacionadas com futebol no jogo POO Trivia. Vamos analisar as suas principais características:

1. ****Construtor:****

- A classe possui um construtor que recebe o texto da pergunta sobre futebol, uma lista de opções e a resposta correta como parâmetros.
- Chama o construtor da superclasse ``SportsQ`` para inicializar atributos comuns.

2. ****Tipo e Respostas:****

- O método ``questionType`` especifica que o tipo de pergunta é "Futebol."
- O método ``getAnswers`` retorna a lista de respostas possíveis para a pergunta sobre futebol.

3. ****Verificação da Resposta Correta:****

- O método ``isCorrectAnswer`` verifica se uma resposta fornecida está correta para a pergunta sobre futebol.

4. ****Cálculo de Pontos:****

- O método ``returnPoints`` calcula e retorna os pontos atribuídos por responder à pergunta sobre futebol. A fórmula utilizada envolve uma combinação de pontos fixos (5 e 3) mais um ponto adicional.

5. ****Sublista de Opções:****

- O método ``getSublistOptions`` cria uma sublista de opções para a pergunta sobre futebol. Embaralha as opções e retorna as três primeiras, incluindo a resposta correta.

Em resumo, a classe ``SoccerQ`` fornece funcionalidades específicas para lidar com perguntas relacionadas com futebol, incluindo identificação do tipo, validação da resposta correta, cálculo de pontos e gestão de opções. Aproveita a estrutura definida na classe abstrata ``SportsQ`` ao adaptar o seu comportamento ao domínio das curiosidades sobre futebol.

Conclusão

Estas classes colaboram como a espinha dorsal do jogo POO Trivia, ilustrando um design robusto que implementa diversas facetas. A classe POOTrivia orquestra o jogo, integrando perguntas, dados dos jogadores e interação do utilizador através de uma interface gráfica amigável. O carregamento dinâmico de perguntas a partir de um ficheiro externo promove a escalabilidade, garantindo uma experiência de jogo sempre renovada. Esta versatilidade, aliada a um código bem organizado e a princípios de programação orientada a objetos, facilita a manutenção e futuras expansões.

A classe Player atua como um contenedor de dados, gerindo informações cruciais dos jogadores ao longo do jogo. Não só monitoriza o desempenho, como também assegura a persistência através de serialização, proporcionando uma experiência personalizada e acompanhamento de progresso entre sessões de jogo. A sua simplicidade ao fornecer acesso a dados essenciais dos jogadores e métodos interativos para avaliação de desempenho aprimora a experiência de jogo.

A classe abstrata Questions estabelece um enquadramento para vários tipos de perguntas, permitindo que subclasses como ArtsQ e ScienceQ implementem comportamentos específicos de perguntas. Através da sua estrutura definida, otimiza o tratamento de detalhes das perguntas, incluindo texto, opções, respostas corretas e métodos de pontuação. A abstração possibilita tipos de perguntas versáteis, enquanto impõe uma interface comum, promovendo a reutilização de código e a extensibilidade do sistema.

Estas subclasses concretas de perguntas, como ArtsQ e ScienceQ, adaptam o seu comportamento aos respetivos domínios, mantendo-se fiéis à estrutura comum fornecida pela classe Questions. Elas encapsulam detalhes específicos das perguntas e mecanismos de pontuação, oferecendo uma variedade de perguntas no jogo e garantindo consistência no tratamento de respostas e cálculos de pontos.

Além disso, SportsQ e suas subclasses, como SwimmingQ e SkiQ, destacam a flexibilidade na gestão de perguntas relacionadas com desportos. Elas tiram partido de um enquadramento comum, ajustando as suas funcionalidades para acomodar as características únicas de cada desporto. Esta abordagem promove um comportamento consistente em diferentes perguntas sobre desportos, permitindo implementações específicas de regras e alocações de pontos.

No geral, a interconexão e modularidade destas classes ilustram um design bem arquitetado e adaptável, demonstrando o poder da programação orientada a objetos na criação de uma experiência envolvente e educativa no jogo de trivia.