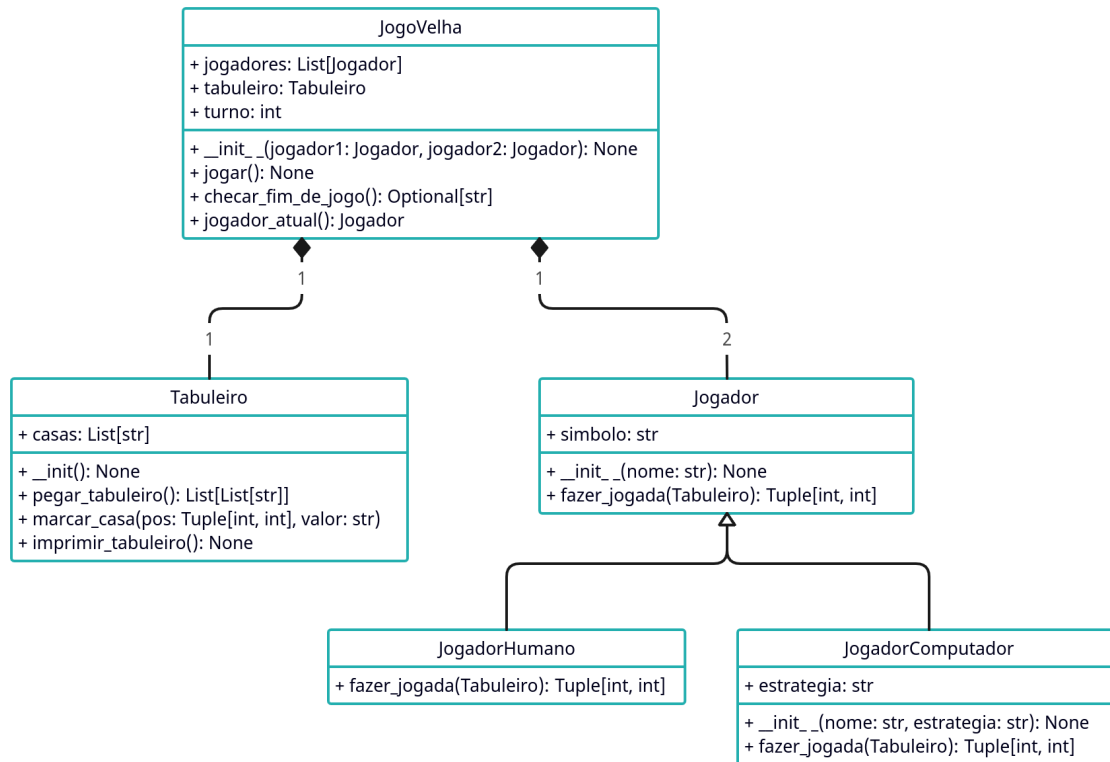


Trabalho 01

Programação Orientada a Objetos

Crie um pacote python que implemente um jogo da velha, seu pacote deve implementar no mínimo as seguintes classes:



- **JogoVelha**: Principal classe, onde a mecânica do jogo será implementada.
 - Atributos:
 - `jogadores`: lista contendo os dois jogadores.
 - `tabuleiro`: um objeto do tipo **Tabuleiro**.
 - `turno`: variável para marcar de qual jogador é a vez.
 - Métodos:
 - `__init__`: Método de inicialização da classe **JogoVelha**, deve receber dois objetos da classe **Jogador** (ou suas subclasses) e inicializar os atributos `tabuleiro`, `jogadores` e `turno`.
 - `jogar`: método que roda a mecânica do jogo esse método deve conter um loop para as rodadas com os seguintes passos:
 - Executa o método `JogoVelha.jogador_atual`;
 - Pedir a jogada ao jogador atual (pelo método `Jogador.fazer_jogada`) (Não peça a jogada ao usuário daqui, se o jogador for humano a classe **JogadorHumano** que deve fazer isso);
 - Registrar a jogada no tabuleiro com o método `Tabuleiro.marcar_casa`;

- Imprimir a jogada e o tabuleiro (Tabuleiro.imprimir_tabuleiro).
 - Conferir as condições de fim de jogo com o método JogoVelha.checar_fim_de_jogo e caso tenha acabado imprimir a condição que determinou o fim de jogo sair do loop.
 - `checar_fim_de_jogo`: Caso uma condição de fim de jogo seja atendida retorna uma string determinando a condição de fim de jogo ou None caso o jogo não tenha terminado.
 - `jogador_atual`: Retorna o objeto Jogador do jogador atual.
- Tabuleiro: Classe para representar o tabuleiro.
 - Atributos:
 - Casas: Uma lista contendo o valor das casas, o valor de cada casa deve estar em branco caso a casa não tenha sido jogada ou o símbolo do jogador que marcou aquela casa.
 - Métodos:
 - `__init__`: Inicializa o tabuleiro vazio.
 - `pegar_tabuleiro`: retorna o tabuleiro no formato lista de lista onde cada sub-lista é uma linha do tabuleiro.
 - `marcar_casa`: Marca uma casa com o símbolo de um jogador. Deve conferir se a casa está vazia.
 - `imprimir_tabuleiro`: Imprime o tabuleiro na tela com separadores.
- Jogador: Classe abstrata para os diferentes tipos de jogadores.
 - Atributos:
 - `simbolo`: O símbolo daquele jogador.
 - Métodos:
 - `__init__`: Inicializa um jogador
 - `fazer_jogada`: Método em branco para ser implementado nas subclasses.
- JogadorHumano: Classe para um jogador humano.
 - Métodos:
 - `fazer_jogada`: Imprime qual é o jogador e pede uma jogada ao usuário.
- JogadorComputador: Classe para um jogador computador.
 - Atributos:
 - `estrategia`: String determinando a estratégia para o jogador computador (implementar no mínimo a estratégia "aleatoria" onde o computador joga em uma casa vazia aleatória).
 - Métodos:
 - `__init__`: Verifica se a estratégia é válida e armazena no atributo `estrategia`.
 - `fazer_jogada`: Determina a jogada de acordo com a estratégia selecionada na inicialização e o tabuleiro atual.

Seu módulo deve se chamar `jogovelha` e as classes `JogoVelha`, `JogadorHumano` e `JogadorComputador` devem estar disponíveis para serem importadas dele:

```
from jogovelha import JogoVelha, JogadorHumano, JogadorComputador
```

Estrutura do pacote

- Um módulo em uma pasta com um arquivo `__init__.py` e mais pelo menos um arquivo `.py` com o código do seu pacote.
- Um arquivo de exemplo de uso do seu pacote.
- O `setup.py` configurado para instalar a sua biblioteca.
- Um `README.md` e `LICENSE`.
- Caso seu código use bibliotecas de terceiros, um arquivo `requirements.txt`.
- Um arquivo de texto `relations.txt` descrevendo o tipo e o funcionamento das relações entre as classes implementadas.

Avaliação

- Organização de código (3 pontos):
 - Typing: 1 ponto
 - Docstrings: 1 ponto
 - Comentários: 1 ponto
- Estrutura (3 pontos):
 - Uso de classes e objetos: 2 pontos
 - Uso de loops e condicionais: 1 ponto
- Organização do pacote: 1 ponto
- Execução do trabalho (3 pontos):
 - Execução e aderência aos requisitos: 2 pontos
 - Descrição das relações de classes: 1 ponto

Prazo

Os trabalhos serão coletados da pasta `trabalho_01` na plataforma do curso **dia 11/10**, não será tolerado nenhum atraso.