## GestorDeJogo capturas: Map<Integer, Integer> jogadasValidas: Map<Integer, Integer> Simulador jogadasInvalidas: Map<Integer, Integer> tabuleiro: Tabuleiro numPretas: int gestor: GestorDeJogo numBrancas: int turno: int numTurnosSemCapturas: int iniciaJogo(ficheiroInicial: File): boolean resultado: int getTamanhoTabuleiro(): int é gerido por processaJogada(xO: int. vO: int. xD: int. vD: int): boolean GestorDeJogo(numPretas: int, numBrancas: int) getPecasMalucas(): List<CrazyPiece> getCapturas(idEquipa: int): int iogoTerminado(): boolean getJogadasValidas(idEquipa: int): int getAutores(): List<String> getJogadasInvalidas(idEquipa: int): int getResultados(): List<String> getResultado(): int getIDPeca(x: int, y: int): int quemEstaAJogar(): int getIDEquipaAJogar(): int naoHouveCaptura(): void setGestor(gestor: Gestor): void adicionaCaptura(equipaQueJoga: int): void setTabuleiro(tabuleiro: Tabuleiro): void validaJogada(equipaQueJoga: int): void invalidaJogada(equipaOueJoga: int): void possoTerminarJogo(): boolean apresenta CrazyPiece idPeca: int idTipo: int Tabuleiro idEquipa: int tamanho: int alcunha: String pecas: List<CrazyPiece> x: int v: int Tabuleiro(tamanho: int) contem getPecas(): List<CrazyPiece> CrazyPiece(idPeca: int, idTipo: int, idEquipa: int, alcunha: String) inserirPeca(peca: CrazyPiece): void setCoordenadas(x: int, y: int): void getTamanho(): int aetId(): int getIdPeca(x: int, y: int): CrazyPiece getImagePNG(): String existemCoordenadas(x: int, v: int): boolean toString(): String removerPeca(x: int, y: int): CrazyPiece move(xD: int, vD): boolean getPecaById(idPeca: int) getIdEquipa(): int getX(): int aetY(): int