Classe: Inputs

* LerOpcao(): Este método cria uma instância de Scanner para ler a entrada do usuário a partir do sistema de entrada padrão (console). Aguarda e lê um inteiro digitado pelo usuário.

Classe: Jogo

* IniciarJogo(): Método responsável por iniciar o jogo, distribuir peças, registrar jogadas e controlar o fluxo do jogo.
* gerarPecas(): Método que gera e retorna uma lista de todas as peças possíveis para o jogo.
* DistribuirPecas(): Método que distribui aleatoriamente as peças para o jogador humano e o computador.
* BuscarMaiorPeca(Lista PecasHumano, Lista PecasComputador): Retorna uma string indicando se a maior peça pertence ao humano ou ao computador, junto com seu índice na lista.
* JogadaDoComputador(Mesa mesa): Realiza a jogada do computador, escolhendo uma peça e registrando-a na mesa.

Classe: Lista

* getInicio(): Retorna o primeiro nó da lista.
* setInicio(No inicio): Define o primeiro nó da lista.
* getUltimo(): Retorna o último nó da lista.
* setUltimo(No ultimo): Define o último nó da lista.
* getTamanho(): Retorna o tamanho da lista.
* setTamanho(int tamanho): Define o tamanho da lista.
* estaVazia(): Verifica se a lista está vazia.
* inserir(Peca novo): Insere uma nova peça no final da lista.
* inserirNoComeco(Peca peca): Insere uma nova peça no início da lista.
* removerPeca(Peca pecaRemover): Remove uma peça específica da lista.
* Imprimir(): Imprime as peças presentes na lista.
* BuscarIndice(Peca peca): Busca o índice de uma peça na lista.
* IdentificarPecaPorIndice(int indice): Retorna a peça correspondente a um determinado índice na lista.

Classe: Menu

* ApresentarMenu(): Exibe o menu para o usuário e retorna a opção escolhida.
* SolicitarPecaHumano(): Solicita uma peça ao jogador humano e retorna a opção escolhida.

Classe: Mesa

* Construtor Mesa(Lista PecasHumanos, Lista PecasComputador): Inicializa a mesa com as listas de peças dos jogadores humano e computador.
* RegistrarJogadaInicial(Peca peca): Registra a jogada inicial inserindo uma peça na mesa.
* RegistrarJogadaNum1(Peca peca): Registra uma jogada na extremidade inicial da mesa.
* RegistrarJogadaNum2(Peca peca): Registra uma jogada na extremidade final da mesa.
* ValidarSeTemJogadasPossiveis(): Verifica se existem jogadas possíveis para o jogador atual com base nas peças disponíveis na mesa.
* VerificarGanhador(): Verifica se há um vencedor comparando o número de peças entre os jogadores e exibe uma mensagem indicando o resultado.

Classe: No

* public No anterior: Referência para o nó anterior na lista.
* public No proximo: Referência para o próximo nó na lista.
* public Peca peca: Peça armazenada no nó.

Classe: Outputs

* imprimirLista(No inicio): Imprime os elementos de uma lista a partir do nó inicial.
* MensagensIniciais(): Imprime mensagens iniciais para iniciar o jogo.
* HumanoMaiorPeca(Peca peca): Imprime mensagem indicando que o jogador humano possui a maior peça e exibe a peça.
* ComputadorMaiorPeca(Peca peca): Imprime mensagem indicando que o computador possui a maior peça e exibe a peça.
* ImprimirMesa(Lista PecasJogadas): Imprime as peças presentes na mesa.
* ImprimirPeca(Peca peca): Imprime os detalhes de uma peça.
* ImprimirPecaAoContrario(Peca peca): Imprime os detalhes de uma peça de forma invertida.
* exibirMenu(): Exibe o menu para o usuário.
* VezHumano(): Indica que é a vez do jogador humano jogar.
* VezComputador(): Indica que é a vez do computador jogar.
* SolicitarPecaHumano(): Solicita ao jogador humano que digite o índice da peça a ser jogada.
* computadorVenceu(): Imprime mensagem indicando que o computador venceu a partida.
* HumanoVenceu(): Imprime mensagem indicando que o jogador humano venceu a partida.
* Empate(): Imprime mensagem indicando que a partida terminou em empate.

Classe: Peca

* Construtor Peca(int numero1, int numero2): Inicializa uma peça com os números especificados.
* ehDupla(): Verifica se a peça é uma dupla (os dois números são iguais).
* contemNumero(int numero): Verifica se um número está presente na peça.
* InverterPeca(): Inverte os números da peça.
* toString(): Retorna uma representação legível da peça.
* getNumero1(): Retorna o primeiro número da peça.
* setNumero1(int numero1): Define o primeiro número da peça.
* getNumero2(): Retorna o segundo número da peça.
* setNumero2(int numero2): Define o segundo número da peça.

Classe: main

* main(String[] args): Método principal que serve como ponto de entrada da aplicação. Cria uma instância da classe Jogo e chama o método IniciarJogo() para começar o jogo.