class Tabuleiro {

constructor() {

this.nome = "Tabuleiro";

this.tabuleiro = [];

// Inicializando o tabuleiro com 8x8 casas

for (let linha = 0; linha < 8; linha++) {

this.tabuleiro[linha] = [];

for (let coluna = 0; coluna < 8; coluna++) {

this.tabuleiro[linha][coluna] = new Casa(linha, coluna);

}

}

console.log(this.tabuleiro);

}

colocarPeca(peca, linha, coluna) {

const casa = this.tabuleiro[linha][coluna];

casa.peca = peca;

casa.elementoHtml.innerHTML = peca.simbolo; // Colocando o símbolo da peça na casa

}

}

class Casa {

constructor(linha, coluna) {

this.linha = linha;

this.coluna = coluna;

this.peca = null;

this.elementoHtml = document.createElement('div');

this.elementoHtml.classList.add('casa');

if ((linha + coluna) % 2 == 0) {

this.elementoHtml.classList.add('clara');

} else {

this.elementoHtml.classList.add('escura');

}

document.getElementById('tabuleiro').appendChild(this.elementoHtml);

}

}

class Peca {

constructor(cor, linha, coluna) {

this.cor = cor;

this.linha = linha;

this.coluna = coluna;

this.simbolo = "";

}

movimentosPossiveis() {

return [];

}

moverPara(novaLinha, novaColuna) {

this.linha = novaLinha;

this.coluna = novaColuna;

}

}

// Definindo classes para as diferentes peças de xadrez

class Rei extends Peca {

constructor(cor, linha, coluna) {

super(cor, linha, coluna);

this.simbolo = cor === 'branca' ? '&#9812;' : '&#9818;'; // Rei branco e preto

}

}

class Rainha extends Peca {

constructor(cor, linha, coluna) {

super(cor, linha, coluna);

this.simbolo = cor === 'branca' ? '&#9813;' : '&#9819;'; // Rainha branca e preta

}

}

class Bispo extends Peca {

constructor(cor, linha, coluna) {

super(cor, linha, coluna);

this.simbolo = cor === 'branca' ? '&#9815;' : '&#9821;'; // Bispo branco e preto

}

}

class Cavalo extends Peca {

constructor(cor, linha, coluna) {

super(cor, linha, coluna);

this.simbolo = cor === 'branca' ? '&#9816;' : '&#9822;'; // Cavalo branco e preto

}

}

class Torre extends Peca {

constructor(cor, linha, coluna) {

super(cor, linha, coluna);

this.simbolo = cor === 'branca' ? '&#9814;' : '&#9820;'; // Torre branca e preta

}

}

class Peao extends Peca {

constructor(cor, linha, coluna) {

super(cor, linha, coluna);

this.simbolo = cor === 'branca' ? '&#9817;' : '&#9823;'; // Peão branco e preto

}

}

// Criando o tabuleiro

const tabuleiro = new Tabuleiro();

// Colocando as peças no tabuleiro

// 8 Peões brancos na linha 1 (segunda linha)

for (let coluna = 0; coluna < 8; coluna++) {

const peaoBranco = new Peao('branca', 1, coluna);

tabuleiro.colocarPeca(peaoBranco, 1, coluna);

}

// 8 Peões pretos na linha 6 (sétima linha)

for (let coluna = 0; coluna < 8; coluna++) {

const peaoPreto = new Peao('preta', 6, coluna);

tabuleiro.colocarPeca(peaoPreto, 6, coluna);

}

// Colocando as peças maiores

tabuleiro.colocarPeca(new Rei('branca', 0, 4), 0, 4); // Rei branco

tabuleiro.colocarPeca(new Rei('preta', 7, 4), 7, 4); // Rei preto

tabuleiro.colocarPeca(new Rainha('branca', 0, 3), 0, 3); // Rainha branca

tabuleiro.colocarPeca(new Rainha('preta', 7, 3), 7, 3); // Rainha preta

tabuleiro.colocarPeca(new Bispo('branca', 0, 2), 0, 2); // Bispo branco (esquerda)

tabuleiro.colocarPeca(new Bispo('branca', 0, 5), 0, 5); // Bispo branco (direita)

tabuleiro.colocarPeca(new Bispo('preta', 7, 2), 7, 2); // Bispo preto (esquerda)

tabuleiro.colocarPeca(new Bispo('preta', 7, 5), 7, 5); // Bispo preto (direita)

tabuleiro.colocarPeca(new Cavalo('branca', 0, 1), 0, 1); // Cavalo branco (esquerda)

tabuleiro.colocarPeca(new Cavalo('branca', 0, 6), 0, 6); // Cavalo branco (direita)

tabuleiro.colocarPeca(new Cavalo('preta', 7, 1), 7, 1); // Cavalo preto (esquerda)

tabuleiro.colocarPeca(new Cavalo('preta', 7, 6), 7, 6); // Cavalo preto (direita)

tabuleiro.colocarPeca(new Torre('branca', 0, 0), 0, 0); // Torre branca (esquerda)

tabuleiro.colocarPeca(new Torre('branca', 0, 7), 0, 7); // Torre branca (direita)

tabuleiro.colocarPeca(new Torre('preta', 7, 0), 7, 0); // Torre preta (esquerda)

tabuleiro.colocarPeca(new Torre('preta', 7, 7), 7, 7); // Torre preta (direita)