

# Relatório POO Xadrez – Fase 1

Autores:

- |                                 |        |
|---------------------------------|--------|
| - Adrian Todt Monteiro da Silva | 759458 |
| - Diego Tupinambá               | 760922 |
| - Victor Mendes                 | 756206 |

Turma: POO 2019/1

Data de envio: 25 de abril de 2019

## Introdução do projeto

Descrição do projeto: Uma implementação orientada a objetos de um jogo de xadrez. A primeira fase é focada em diversos testes de cada método.

Descrição do programa principal: Um programa responsável por executar testes e imprimir os resultados na tela.

Forma de utilização do programa: O programa não pede nenhuma entrada, ele somente executa testes e tem como saída o resultado deles.

Condições de erro no projeto:

Alocação dinâmica: não há alocação dinâmica no código, portanto, não é possível resultar em um erro.

Entradas inválidas: não resultam em um erro porque há uma checagem de parâmetros no começo dos métodos.

Índices inválidos em vetor: o vetor dinamicamente alocado não é utilizado na fase 1.

Dificuldades encontradas no projeto: Não poder usar a biblioteca STL, problemas com o pré-processador de texto (arquivos ".h" incluídos várias vezes), e entender a estrutura de arquivos pedida no projeto.

Possíveis extensões e melhorias do projeto: Usar a biblioteca "ncurses" para uma interface gráfica.

## Classes do projeto

Bibliotecas:

- Vetor
- Mathutils

Classes:

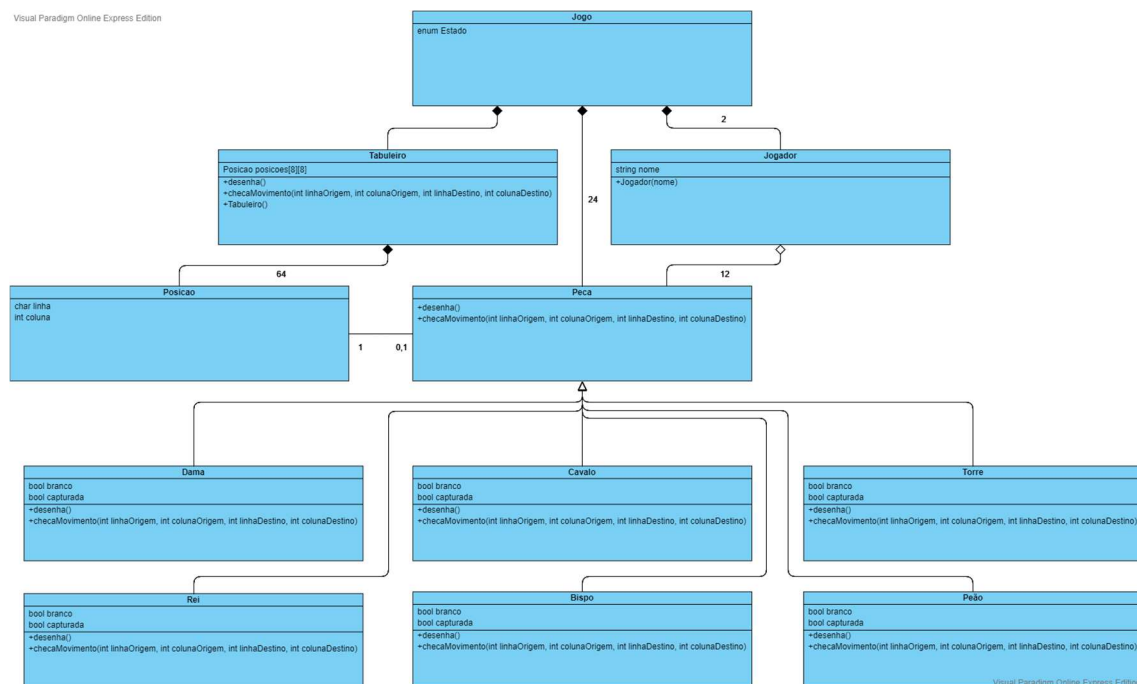
- Jogador
- Jogo
- Peça
- Posição

- Tabuleiro

Classes de peças:

- Bispo
- Cavalo
- Dama
- Peão
- Rei
- Torre

## Diagrama de classes



## Descrição das classes

Vetor:

A biblioteca "Vetor" tem um construtor e funções de: inserir, remover e obter elementos do vetor construído. Ele só será utilizado na fase 2.

Mathutils:

A biblioteca "Mathutils" retorna o sinal de um número. É usada em todas as classes de tipo de peça para as suas respectivas checagens de movimento.

Bispo:

A classe "Bispo" constrói, desenha no tabuleiro, e checa a movimentação de uma peça do tipo bispo.

Cavalo:

A classe "Cavalo" constrói, desenha no tabuleiro, e checa a movimentação de uma peça do tipo cavalo.

Dama:

A classe "Dama" constrói, desenha no tabuleiro, e checa a movimentação de uma peça do tipo dama.

Peão:

A classe "Peão" constrói, desenha no tabuleiro, e checa a movimentação de uma peça do tipo peão.

Rei:

A classe "Rei" constrói, desenha no tabuleiro, e checa a movimentação de uma peça do tipo rei.

Torre:

A classe "Torre" constrói, desenha no tabuleiro, e checa a movimentação de uma peça do tipo torre.

Jogador:

A classe "Jogador" constrói um jogador com o nome em branco e, logo em seguida, um jogador com um nome preenchido. Será usada apenas na fase 2.

Jogo:

A classe "Jogo" tem representações dos estados de jogo e de uma partida de xadrez, mas não será usada até a fase 2.

Peça:

A classe "Peça" constrói uma peça de determinada cor, desenha-a no tabuleiro, e checa a sua movimentação.

Posição:

A classe "Posição" guarda uma posição (linha e coluna), e se ela contém uma peça ou não. Usada na classe "Tabuleiro".

Tabuleiro:

A classe "Tabuleiro" gerencia o tabuleiro de uma partida com um construtor, uma impressão, e uma checagem de movimento por rodada.

## Testes

Teste 1:

Objetivo do Teste: testar a impressão do bispo na tela

Objeto: Bispo

Método testado: desenha()

Valores dos parâmetros: N/D

Valor de retorno: N/D

Tela: B (se branca); b (se preta)

Teste 2:

Objetivo do Teste: testar o movimento do bispo

Objeto: Bispo

Método testado: `checaMovimento()`

Valores dos parâmetros: 1 1 5 5

Valor de retorno: verdadeiro

Tela: N/D

Teste 3:

Objetivo do Teste: testar o movimento do bispo

Objeto: Bispo

Método testado: `checaMovimento()`

Valores dos parâmetros: 1 1 1 2

Valor de retorno: falso

Tela: N/D

Teste 4:

Objetivo do Teste: testar a impressão do cavalo na tela

Objeto: Cavalo

Método testado: `desenha()`

Valores dos parâmetros: N/D

Valor de retorno: N/D

Tela: C (se branca); c (se preta)

Teste 5:

Objetivo do Teste: testar o movimento do cavalo

Objeto: Cavalo

Método testado: `checaMovimento()`

Valores dos parâmetros: 1 1 3 2

Valor de retorno: verdadeiro

Tela: N/D

Teste 6:

Objetivo do Teste: testar o movimento do cavalo

Objeto: Cavalo

Método testado: `checaMovimento()`

Valores dos parâmetros: 1 1 3 3

Valor de retorno: falso

Tela: N/D

Teste 7:

Objetivo do Teste: testar a impressão da dama na tela

Objeto: Dama

Método testado: `desenha()`

Valores dos parâmetros: N/D

Valor de retorno: N/D

Tela: D (se branca); d (se preta)

Teste 8:

Objetivo do Teste: testar o movimento da dama

Objeto: Dama

Método testado: `checaMovimento()`

Valores dos parâmetros: 1 1 5 5

Valor de retorno: verdadeiro

Tela: N/D

Teste 9:

Objetivo do Teste: testar o movimento da dama

Objeto: Dama

Método testado: `checaMovimento()`

Valores dos parâmetros: 1 1 3 2

Valor de retorno: falso

Tela: N/D

Teste 10:

Objetivo do Teste: testar a impressão do peão na tela

Objeto: Peao

Método testado: `desenha()`

Valores dos parâmetros: N/D

Valor de retorno: N/D

Tela: P (se branca); p (se preta)

Teste 11:

Objetivo do Teste: testar o movimento do peão branco

Objeto: Peao

Método testado: `checaMovimento()`

Valores dos parâmetros: 1 1 2 1

Valor de retorno: verdadeiro

Tela: N/D

Teste 12:

Objetivo do Teste: testar o movimento do peão branco

Objeto: Peao

Método testado: `checaMovimento()`

Valores dos parâmetros: 2 1 5 1

Valor de retorno: falso

Tela: N/D

Teste 13:

Objetivo do Teste: testar o movimento do peão preto

Objeto: Peao

Método testado: `checaMovimento()`

Valores dos parâmetros: 6 6 5 6

Valor de retorno: verdadeiro

Tela: N/D

Teste 14:

Objetivo do Teste: testar o movimento do peão preto

Objeto: Peao

Método testado: `checaMovimento()`

Valores dos parâmetros: 5 6 3 6

Valor de retorno: falso

Tela: N/D

Teste 15:

Objetivo do Teste: testar a impressão do rei na tela

Objeto: Rei

Método testado: `desenha()`

Valores dos parâmetros: N/D

Valor de retorno: N/D

Tela: R (se branca); r (se preta)

Teste 16:

Objetivo do Teste: testar o movimento do rei

Objeto: Rei

Método testado: `checaMovimento()`

Valores dos parâmetros: 1 1 5 5

Valor de retorno: verdadeiro

Tela: N/D

Teste 17:

Objetivo do Teste: testar o movimento do rei

Objeto: Rei

Método testado: `checaMovimento()`

Valores dos parâmetros: 1 1 1 2

Valor de retorno: falso

Tela: N/D

Teste 18:

Objetivo do Teste: testar a impressão da torre na tela

Objeto: Torre

Método testado: `desenha()`

Valores dos parâmetros: N/D

Valor de retorno: N/D

Tela: T (se branca); t (se preta)

Teste 19:

Objetivo do Teste: testar o movimento da torre

Objeto: Torre

Método testado: `checaMovimento()`

Valores dos parâmetros: 1 1 1 3

Valor de retorno: verdadeiro

Tela: N/D

Teste 20:

Objetivo do Teste: testar o movimento da torre

Objeto: Torre

Método testado: `checaMovimento()`

Valores dos parâmetros: 1 1 2 2

Valor de retorno: falso

Tela: N/D

Teste 21:

Objetivo do Teste: testar a impressão do tabuleiro na tela

Objeto: Tabuleiro

Método testado: `desenha()`

Valores dos parâmetros: N/D

Valor de retorno: N/D

Tela:

```
| 1 2 3 4 5 6 7 8
--+-----
A | • + • + • + • +
B | + • + • + • + •
C | • + • + • + • +
D | + • + • + • + •
E | • + • + • + • +
F | + • + • + • + •
G | • + • + • + • +
H | + • + • + • + •
```