

2º. Trabalho – Jogo de Tabuleiro com Padrões

Valor: 4 pontos

Data de entrega: 12 de fevereiro

Equipe: até 3 pessoas

Estenda e melhore o jogo de tabuleiro feito na primeira etapa para seguir as seguintes especificações:

- Seu jogo deve permitir até 6 participantes ao mesmo tempo. Cada jogador possui uma cor que o identifica, uma pontuação em moedas que ele recebe quando cai em uma casa Normal do tabuleiro. Cada movimento de um competidor é determinado por dois dados: o competidor andará o número de casas igual à soma dos valores dos dois dados. Existem 3 tipos de jogador: jogador com sorte, cuja soma dos valores dos dados é sempre maior ou igual à 7; jogador azarado, cuja soma dos valores dos dados é sempre menor ou igual a 6; jogador normal, que pode obter tanto valores altos como baixos para a soma dos dados.

- O tabuleiro deve ser uma classe que contém um ArrayList de jogadores e outro de casas e os métodos que você achar necessários. As casas devem seguir uma hierarquia, sendo a classe mãe abstrata possuindo o atributo número e declarando o método *aplicarRegra()*. Classes filhas podem ser do tipo

- * **Simples:** acrescenta 1 moeda à pontuação do jogador;

- * **Surpresa:** o jogador deve tirar uma carta aleatória que o fará mudar de tipo de jogador de acordo com a carta.

- * **Prisão:** se o jogador parar em uma dessas casa, ele fica preso e não joga a próxima rodada, estando liberado em duas rodadas ou se na rodada seguinte pagar taxa de 2 moedas.

- * **Sorte:** o jogador anda 3 casas para frente, desde que ele não seja um jogador azarado, e recebe mais 3 moedas.

- * **Azar:** o jogador perde 3 moedas, desde que não seja jogador sortudo;

- * **Reversa:** se o jogador parar em uma delas, ele troca de lugar com o jogador que está mais atrás no jogo. Caso ele seja o último, ele não sai do lugar.

- * **JogaDeNovo:** o jogador irá jogar os dados mais uma vez e ganha a quantidade de moedas igual à soma dos dados da nova jogada.

- * **Troca:** o jogador poderá trocar seus pontos incrementando seu avatar escolhendo os seguintes elementos (um de cada apenas por vez) e recebendo os respectivos benefícios:

- boné: ganha 3 moedas e anda uma casa a mais a cada rodada de dados;

- moleton: 4 moedas e anda duas casas a mais a cada rodada de dados;

- óculos escuro: 5 moedas e anda três casas a mais a cada rodada de dados;

Requisitos:

- Cada competidor começa da casa 0 e com 0 moedas.

- O jogo tem que ter pelo menos dois jogadores de tipos diferentes, mas pode haver dois ou mais jogadores do mesmo tipo.

- Permita que o usuário crie o tabuleiro com a quantidade de casas que quiser e selecione as casas de acordo com as regras indicadas.

- Ganha o competidor que chegar primeiro na última casa (ou passar dela).

- A cada rodada você deve mostrar a posição de cada jogador no tabuleiro e a quantidade de moedas de cada um (ex: Azul na casa 2 com 5 moedas, verde na casa 6 com 2 moedas e boné e moleton...), e de quem é a vez para jogar;
- Após jogar os dados, você deve mostrar o valor da soma dos dados
- Se um jogador cair em uma das casas ou condições acima, mostrar uma mensagem indicando
- Crie um modo *Debug* que permite, ao invés de jogar dados, que o usuário insira o número da casa que o jogador deverá ir.
- Você tem que mostrar o vencedor
- Você deve mostrar a quantidade de jogadas de cada jogador ao final do jogo, a posição e a quantidade de moedas de cada um.
- Você pode usar uma interface gráfica, se quiser.
- Você deve usar o padrão **Strategy** para representar a hierarquia de casas e seleção de casas;
- Use o padrão **Injeção de dependência** para passar às classes que representam as casas o objeto jogador para verificar se possui bônus;
- Você deve usar o padrão **Factory** para criar jogadores e casas no tabuleiro;
- Você deve usar o padrão **Singleton** para garantir que só pode ter 1 tabuleiro;
- Você deve criar uma classe Jogo, que funcionará como um **Facade**, de forma que ele implemente pelo menos os seguintes métodos conforme mostrados na classe Main:

```
public class Main {
    public static void main(String args[]){
        Jogo jogo = new Jogo();
        int numCasas = // ler do teclado o número de casas
        int numJogadores = // ler do teclado o número de casas;
        jogo.configTabuleiro(numCasas);
        jogo.config(numJogadores);
        jogo.printTabuleiro();
        jogo.start();
    }
}
```

Seu Diagrama de classes do projeto deve ficar mais ou menos assim:

