

Trabalho Final de POO

Projeto em Java: **Jogo Super Trunfo**

Versão do projeto: 1.0.0

Professor: MATEUS MACHADO LUNA

Valor: 20% da Nota final

Data de Entrega: 24 de Fevereiro de 2023

Atividade feita em grupos de no máximo 4 integrantes

Para demonstrar os conhecimentos adquiridos nesta disciplina, você deverá implementar um projeto em Java que cubra grande parte dos tópicos que discutimos. Trata-se de uma versão digital do popular jogo de cartas Super Trunfo.

Sobre o Super Trunfo

O jogo original é vendido como um baralho sobre alguma temática (Carros, Dinossauros, Países, Times de Futebol, etc). Cada baralho deste consiste de 32 cartas e todas elas possuem as seguintes características:

- Um nome (por exemplo, Tiranossauro Rex, Brasil, Ferrari);
- Um código (que é único para cada carta, sendo que seu valor é dividido em 8 grupos de 4 cartas - 1A, 1B, 1C, 1D, 2A....8C e 8D);
- Um conjunto de 1 a 3 tipos de curiosidades (informações extras) que não tem peso para o jogo (por exemplo, para Dinossauros, seu período de existência; para países, seu idioma oficial e continente; para carro, seu modelo);
- Um conjunto de exatos 4 atributos relevantes para o jogo, sempre com as seguintes informações:
 - Nome do atributo (por exemplo, peso, velocidade, altura, potência, etc - depende da temática);
 - Valor numérico deste atributo (123, 10.000,00, 0.40, etc);
 - Unidade de medida deste atributo (kg, km/h, m, ou vazio);
- Por fim, uma carta apenas é chamada "SuperTrunfo", sendo uma espécie de coringa, cuja funcionalidade mencionaremos a seguir.

O jogo começa com a distribuição entre os jogadores de todas as cartas do baralho. Cada jogador vê apenas as cartas do seu monte e as retira, uma por vez em cada turno, do topo de seu baralho. Aleatoriamente, escolhe-se o primeiro jogador que inicia um turno. O turno consiste em:

1. O jogador que começa o turno elege um dos atributos relevantes das cartas para ser o critério de competição da rodada.
2. Todos os jogadores apresentam suas cartas, anunciando o valor dos seus atributos.
3. Ganha o jogador cuja carta tiver o maior valor do atributo escolhido. Neste caso, ele recebe todas as cartas em jogo no turno e as coloca no final do seu monte.

O jogador vencedor do turno também terá o direito de iniciar o turno seguinte. O jogo ocorre deste modo até que todas as cartas estejam na mão de um jogador, que será o vencedor.

A carta "SuperTrunfo", que é única no baralho, tem maior poder de competição. Se ela for apresentada na rodada, seu dono automaticamente ganha o turno, exceto se algum dos jogadores tiver uma carta do grupo A (1A, 2A...). Neste caso, compete-se normalmente, podendo a carta ganhar ou perder.

O jogo possui ainda regras para um turno de desempate, para situações onde cartas possam ter valores iguais dos atributos. Para este trabalho, desconsideraremos estas regras e daremos a vitória automaticamente ao último jogador na sequência do turno, dentre os empatados.

Sobre a implementação do projeto

O seu projeto deve ser capaz de gerar um baralho SuperTrunfo, a partir de uma escolha inicial do usuário, com no mínimo 4 temáticas disponíveis. Escolhida a temática, o baralho será gerado, embaralhado e distribuído em dois montes: o do jogador e o do adversário, que deverá ser simulado pelo seu próprio programa, usando escolhas aleatórias.

Classes esperadas:

- **Baralho** - É uma implementação da Java *LinkedList* de *Cartas*, com algumas operações relacionadas à elas como *Embaralhar* e *Distribuir* entre os jogadores. Seu método "Carregar" deve instanciar as *Cartas*. De início, você irá inserir estes dados direto no código, mas mais pra frente em nossa matéria veremos como ler dados de arquivos, para que as informações das cartas sejam importadas de um arquivo com todas as cartas definidas.
- **Carta** - Contém os atributos necessários para representar as características descritas anteriormente. Note que nesta classe será importante definir seus *getters*, que provavelmente serão usados na lógica de competição entre cartas. O método *toString()* deve tentar exibir as informações da carta de uma maneira objetiva.
- **Atributo** - Usada para cada um dos atributos relevantes da carta. Não é apenas o valor, mas o conjunto de informações que compõe um atributo (*valor*, *nome* e *unidade de medida*). O campo *valor* desta classe é um desafio, pois o seu tipo pode ser inteiro (*anos*, *habitantes*, *idade*) ou um número com casas decimais (*peso*, *altura*, *largura*, *velocidade*). Cabe a você decidir uma estratégia para viabilizar essa dualidade. A classe também não tem *setters* públicos, somente *getters*, sendo que estes devem ter a lógica genérica para retornar o valor apropriado de acordo com o tipo. Seu *toString()* deve mostrar uma versão sucinta do atributo (*por exemplo*: "*Peso: 20Kg*").
- **JogadorAbstrato** - Em sua função *main*, serão instanciados dois destes, um de cada tipo: *JogadorReal* e *JogadorRandômico*. Eles contêm um atributo *Monte*, que é um objeto *Baralho* retornado do método *Distribuir* do *Baralho* principal do Jogo. O jogador também tem métodos como o *JogarTurno*, que implica na escolha de um atributo e na conferência dos valores de cada carta, que é retornada pelo método *pegarDoTopo* do *Baralho*. A diferença entre o *JogadorReal* e o

`JogadorRandômico` é que o primeiro faz escolhas baseando-se em dados enviados pelo usuário via um `Scanner`. O segundo faz escolhas completamente randômicas (é o que emula seu adversário no jogo).

- `Jogo` - É uma instância de jogo, que contém um `Baralho` e os dois Jogadores. Esta instância recebe como construtor o tema do jogo, para que invoque o método `Carregar um Baralho` passando o tema apropriado. Ela tem métodos como `"mostrarVencedor"`, `"mostrarStatus"` (que diz quantas cartas cada jogador tem no momento) e `"Jogar"`. No corpo do método `Jogar` deve estar a lógica de controle dos turnos, cuidando de atribuir para cada jogador que vencer o turno a lógica de tentar competir pelas cartas *enquanto* (`while`) não houver um vencedor. Lembre-se nas operações do turno de identificar cada ação que é feita com algum `System.out.println` antes, para que se possa saber o que está acontecendo.

Na sua função `main`, você deve ter a lógica de uma apresentação e um menu, que oferece quatro temas diferentes de jogos para o `SuperTrunfo`, para então escolhido o tema, ser instanciado um objeto `Jogo`.

Serão avaliadas boas práticas de encapsulamento e organização geral do código. Além dos requisitos listados acima, seu trabalho deve implementar, em pelo menos algum dos métodos pedidos, uma lógica de Tratamento de Erros, conceito que estudaremos mais adiante na matéria.

Interface Gráfica

Para aqueles que se aventurarem a implementar uma interface gráfica para o Jogo, será dado um total de 1 ponto extra (10%) na média final. Nesta proposta, espera-se:

- Uma janela principal contendo a apresentação do jogo;
- Uma etapa de seleção do tema do jogo através de um combo de seleção, mostrando as opções disponíveis;
- A exibição, nos turnos, da carta atual do jogador e um dispositivo de seleção dos atributos disponíveis;
- A exibição, a qualquer momento do jogo, em algum local da janela, do status atual d jogo (quantas cartas cada jogador tem);
- A exibição, nos turnos, das duas cartas selecionadas após a escolha do atributo, e uma identificação visual da carta vencedora;
- A exibição, ao encerrar o jogo, do jogador vencedor e uma opção para se jogar novamente ou trocar de tema.

A pontuação extra será concedida baseando-se na apresentação destes recursos.