



UNIVERSIDADE FEDERAL DE PELOTAS
CENTRO DE DESENVOLVIMENTO TECNOLÓGICO
CURSOS DE CIÊNCIA E ENGENHARIA DE COMPUTAÇÃO
Disciplina de Programação Orientada a Objetos - 2021/1
Prof.: Felipe de Souza Marques (felipem@inf.ufpel.edu.br)

Trabalho Final Super Trunfo da Reciclagem

O objetivo deste trabalho é implementar um jogo de cartas virtual simples, chamado Super Trunfo da Reciclagem. O jogo foi desenvolvido pela Equipe de Robótica Sentinelos em conjunto com a Robótica do SESI-SP. As regras do jogo e definições de cartas estão disponíveis no seguinte link: <https://atividadesinfantis.com.br/super-trunfo-da-reciclagem/>.

Você deve escrever uma classe principal chamada ***SuperTrunfoDaReciclagem***, onde tudo irá acontecer. O diagrama de classes do projeto está no Anexo I da descrição do trabalho. O conjunto de requisitos básicos para a construção da hierarquia de classes e algumas funcionalidades são descritos a seguir. Lembre-se de explorar os conceitos que são os pilares da POO: abstração, encapsulamento, herança e polimorfismo.

Conjunto de Requisitos básicos - (7,0 pontos)

1. Todas as classes que aparecem no diagrama de classes devem fazer parte do projeto. Algumas destas classes serão fornecidas pelo professor.
2. As informações das cartas devem ser lidas a partir de um arquivo texto disponibilizado pelo professor e atribuídas a um objeto modelado pela classe *Carta*. Os objetos das cartas devem ser mantidos na classe *Baralho*.
3. O jogo deve permitir a participação de no mínimo 2 jogadores (caso deseje possibilitar a participação de mais jogadores, o número máximo deve ser 4).
4. A classe *SuperTrunfoDaReciclagem* deve implementar as regras do jogo, conforme especificado em <https://atividadesinfantis.com.br/super-trunfo-da-reciclagem/>, utilizando o suporte das demais classes.

Simulação do Jogo - (1,5 ponto)

1. Distribuir todas as cartas do baralho, de forma aleatória, entre os jogadores.
2. Iniciar o processo de simulação do jogo, definindo aleatoriamente qual jogador iniciará o jogo.
3. No turno de cada jogador, o critério que será comparado também será definido de forma aleatória.
4. A simulação deve continuar até que um dos usuários tenha todas as cartas do baralho. Mantenha um contador de turnos para saber quantos foram necessários até a conclusão do jogo.

Interface com o Usuário (gráfica ou textual) - (1,5 ponto)

1. Implemente uma interface com o usuário que permita a definição do número (caso a implementação permita mais de 2 jogadores) e dos nomes dos jogadores.
2. Os dois primeiros itens da seção “Simulação do Jogo” também devem ser realizados quando a interface for utilizada para jogar.
3. A interface deve criar um mecanismo para que os jogadores, no seu respectivo turno, possam escolher qual critério será utilizado para comparação das cartas.
4. Lembre-se de utilizar o tratamento de exceções para tratar situações adversas.

Dicas:

1. Utilizem a classe Random para gerar os números aleatórios (<https://www.devmedia.com.br/numeros-aleatorios-em-java-a-classe-java-util-random/26355>).
2. Utilizar o método toString() para facilitar as impressões de dados.
3. O uso de coleções facilita a implementação do projeto. Vocês terão uma vídeo aula para partir de uma base de conhecimentos sobre coleções.
4. Crie métodos para facilitar a comparação entre os atributos das cartas. Por exemplo:

```
int comparaDecomposicao(Carta carta) {  
    // Retorna zero se os valores do atributo decomposição forem iguais  
    // Retorna 1 se o atributo decomposição de this (instância do objeto que invoca o  
    // método) for maior que o parâmetro carta e -1 caso contrário  
}
```
5. O uso de arquivos será demonstrado por uma vídeo aula. Isso não impede que o desenvolvimento seja iniciado, pois é possível testar todos os mecanismos injetando um “baralho reduzido” diretamente no código.

Anexo I - Diagrama de Classes do Super Trunfo da Reciclagem



