

1 Descrição do Trabalho

Construir um programa em Java que implemente um jogo parecido com o jogo conhecido como Mau-Mau ou Uno®.

2 Descrição do Jogo

O jogo deve ser criado considerando o seguinte conjunto características básicas:

- Jogo deve prever dois tipos de cartas: simples e especiais.
- Toda carta deve ter um símbolo e uma cor.
- O símbolo de toda carta simples será um número de um algarismo (0 à 9).
- O símbolo de toda carta especial será uma letra maiúscula.
- Qualquer carta pode ser descartada, desde que atenda a requisitos básicos descritos mais a frente.
- Uma carta simples descartada não tem efeito adicional
- Uma carta especial descartada causa um efeito especial, dependendo da carta.
- O jogo pode ser jogado a partir de dois jogadores.

2.1 Mecânica do Jogo

Inicialmente, o programa deverá criar o baralho. Em termos de programação, criar esse baralho é, basicamente, instanciar todas as cartas, inserindo estas em uma lista, constituindo, assim, nosso deque principal. As quantidades de cartas do baralho serão descritas em breve. O deque deverá ser embaralhado. A classe `Collections` tem um método para isso, chamado `shuffle`.

Após a criação do baralho (ou antes disso), o programa deverá pedir para o usuário informar a quantidade de jogadores e o nome de cada jogador.

Em seguida, cada jogador irá receber 7 (sete) cartas, vindas de um dos lados do deque (início e fim, tanto faz). Sugere-se que as cartas sejam distribuídas alternadamente, distribuindo apenas uma carta a cada jogador, depois outra carta para cada jogador, depois outra, e assim até completar sete cartas para cada jogador.

Após a distribuição das cartas, uma carta adicional deve ser retirada do deque principal e jogada em outro deque, que representará o monte morto. Todas as cartas devem ser descartadas sobre esse monte de cartas. Em termos de programação, sugere-se utilizar uma estrutura de pilha (`Stack`) para armazenar essas cartas. Portanto, esse monte será composto, inicialmente, apenas por essa carta retirada do deque principal logo após a distribuição de cartas.

Em seguida, deverá ser sorteado um dos jogadores para que este comece a jogar, sendo seguido pelo próximo jogador e assim por diante, em uma alternância circular. Em sua vez, cada jogador terá duas alternativas:

- Descartar uma de suas cartas no monte morto;
- ou "passar", e, como penalidade, comprar uma carta adicional no deque principal.

Para a segunda alternativa de ação (comprar carta e passar), não existe pré-requisito. Trata-se apenas de uma decisão que o jogador, em sua vez, pode tomar.

Para a primeira alternativa de ação (descartar uma de suas cartas), a jogada deverá passar por uma validação simples: a carta só pode ser descartada se e somente se, ela for especial ou se for simples e coincidir símbolo ou cor com a carta que está no topo do monte morto. Note que, quando a carta que está no topo do monte morto for uma carta especial, cartas simples só podem ser descartadas caso a cor seja a mesma.

Obviamente, diante da impossibilidade de descartar uma de suas cartas, o jogador será obrigado a tomar a outra ação, de compra de carta. Caso compre uma carta, diferente dos jogos Mau Mau e Uno, o jogador NÃO poderá descartar uma carta no mesmo turno.

O jogo termina quando um dos jogadores descarta todas as suas cartas.

2.2 Cartas especiais

As cartas especiais são cartas que, tais como qualquer outra carta, podem ser descartadas. No entanto, ao serem descartadas, efetuam uma ação especial. As ações especiais podem ser diversas. Neste trabalho, deverão haver 8 cartas especiais. A seguir, são apresentadas as cartas especiais e suas ações especiais:

Carta	Símbolo	Ação
Carta de Pulo	P	faz com que o próximo jogador seja pulado, de modo que o jogador seguinte a ele passa a ser o próximo.
Carta de Reversão	R	faz com que o sentido do fluxo que determina de quem é a vez de jogar se inverta, de modo que o próximo passa a ser o anterior, e depois o anterior, e assim por diante.
Carta de Ataque	A	faz com que o próximo jogador seja obrigado a comprar duas cartas do deque. Isso não faz com que o próximo jogador perca a vez.
Carta de Ataque Cruel	C	faz com que o próximo jogador seja obrigado a comprar quatro cartas do deque e ainda faz com que ele perca a vez (ou seja pulado).
Carta de Ataque Retroativo	T	faz com que o jogador anterior seja obrigado a comprar uma carta do deque. A vez não passa a ser do jogador anterior. Ele só é obrigado a comprar a carta.
Carta de Ataque Geral	G	faz com que TODOS os jogadores sejam obrigados a comprar duas cartas do deque.
Carta de Desintegração	X	sorteia um jogador aleatoriamente para que este seja removido do jogo. Caso após o uso da carta reste apenas um jogador, este terá vencido a partida.
Carta Customizada	Z	A aluno deverá criar uma ação para essa carta. Use a criatividade.

2.3 Construção do baralho

Ao gerar o baralho, as cartas deverão ser instanciadas seguindo o seguinte quantitativo:

- 40 cartas simples, sendo 10 verdes, 10 vermelhas, 10 azuis e 10 amarelas. Em cada grupo de cores, deverão haver 10 cartas, cada uma com um número diferente: 0 a 9;
- 2 cartas especiais do tipo "Carta de Desintegração", sendo uma da cor vermelha e uma da cor amarela;
- 2 cartas especiais do tipo "Ataque Cruel", sendo uma da cor verde e uma da cor azul;
- 4 cartas de todos os demais tipos especiais, sendo uma de cada cor.

2.4 Construção do Programa

O jogo deverá ser construído com o paradigma de orientação a objeto utilizando a linguagem Java. Portanto, será fundamental a definição de classes. Evidentemente, a melhor forma de construir o programa é utilizar alguns recursos interessantes da orientação a objetos para a modelagem do jogo.

Para modelar o jogo, sugere-se que sejam utilizados recursos como herança e polimorfismo. Seguem algumas sugestões:

- Classe ou interface **Carta**.
 - Uma classe abstrata contendo informações mínimas da carta, como símbolo e cor, e ações básicas como descartar. Cada uma das cartas deveria ser uma implementação ou subclasse de **Carta**.
 - Evidentemente, não é necessário criar uma classe para cada carta. Apenas para cada tipo das citadas na tabela anterior. Ou seja, além da interface/classe **Carta**, deveriam ser criadas mais 8 (oito) classes que implementam/estendem **Carta**.
- Motor do jogo.
 - Crie um motor que seja capaz de executar todo o jogo sem que a implementação desse motor precise saber detalhes de implementação de cada uma das cartas. A única referência que o motor teria das cartas seria apenas a classe/interface **Carta**.

3 Avaliação

Parte da avaliação será realizada por meio da pontualidade na entrega do código até a data estipulada pelo professor. A outra parte da avaliação se dará no dia da apresentação.

3.1 Bônus

Será atribuída uma nota adicional aos trabalhos feitos com interface gráfica.