

**UNIVERSIDADE FEDERAL DE MINAS GERAIS**  
**PROGRAMAÇÃO MODULAR – TRABALHO PRÁTICO 2**

Blackjack  
2017/1

Marina Monteiro Moreira - 2015004348

Ronald Davi Rodrigues Pereira - 2015004437

### **Introdução:**

Blackjack é atualmente o jogo de casino mais popular mundialmente, por ser de fácil entendimento e jogabilidade, além de rápido e dinâmico. Blackjack é um jogo de comparação, em que um jogador joga apenas contra a banca (dealer). O objetivo é conseguir a soma mais próxima de 21, sem ultrapassar 21. O objetivo deste trabalho é implementar um jogo de blackjack, jogado com 6 baralhos, por até 10 jogadores, em uma interface simples de terminal, usando conceitos de orientação a objetos na linguagem Java.

### **Regras do jogo:**

As regras de Blackjack são altamente variáveis dependendo do casino. Nesta implementação escolhemos algumas básicas:

- A mão inicial é de 2 cartas;
- Ao verificar sua mão, o jogador tem 3 opções: stand, hit ou double.
  1. Stand: não recebe mais cartas. A rodada do jogador termina.
  2. Hit: recebe uma carta e pode tomar a decisão novamente.
  3. Double: dobra a aposta e não pode mais receber cartas, fazendo com que sua rodada termine.
- Quando a rodada de um jogador termina, passamos ao próximo, até que todos tenham jogado. Então, é a vez do dealer, cuja mão consiste em uma carta revelada aos jogadores e uma escondida. O dealer revela a carta escondida e compra cartas até que sua soma seja maior ou igual a 17.
- Caso a soma de um jogador passe de 21, ele perde automaticamente.
- Caso a soma do dealer ultrapasse 21, o jogador vence.
- Caso a soma do jogador seja maior que a do dealer, dado que nenhuma seja maior que 21, o jogador vence e recebe o dobro de sua aposta.

### **Implementação:**

Para implementar o jogo, dividimos o problema em 8 classes:

- Baralho: essa classe é, na verdade, mais representativa de uma única carta. Um vetor do tipo Baralho é de fato o baralho. A maior responsabilidade da classe é atribuir a cada carta o valor que ela representa na soma de um jogador (por exemplo, reis, damas e valetes todos valem 10). Representa uma carta com um nome, naipe, seu valor, e uma flag que indica se ela está em jogo ou não.
- Jogador: representa os jogadores. Sua principal responsabilidade é escolher a jogada (com ajuda da classe JogaJogo) e fazer a aposta. Cada jogador é representado basicamente por um nome, seu saldo, e sua mão (que salva apenas a posição das cartas no vetor que é o baralho, por isso as flags são necessárias).

- JogaJogo: essa classe executa o passo a passo da jogada em si, implementando os eventos possíveis. Sua dependência com a classe Jogador está na implementação das decisões que podem ser tomadas, todas contidas em JogaJogo, porque a escolha pertence ao jogador, mas ela surge em um momento específico da rodada, que é conduzida em JogaJogo. Eventos como o início da rodada, as vezes de todos os jogadores, o fim da rodada com a vez do dealer e o pagamento ou recolhimento das apostas são implementados nesta classe.
- CriaBaralho: cria um array de elementos do tipo Baralho (cartas), inserindo uma carta de cada naipe 6 vezes (já que usamos 6 baralhos nesta implementação) e embaralha.
- Dealer: representa o dealer, que é um jogador limitado. Suas únicas funções são distribuir e recolher cartas.
- MenuInterativo: faz a interface com o(s) jogador(es) antes do jogo, pegando os dados básicos como o número de jogadores, seus nomes e o saldo inicial de cada um deles.
- InteligenciaArtificial: age como um jogador, tomando suas decisões de acordo com regras pré definidas. A presença da inteligência artificial no jogo é opcional e pode ser definida no menu interativo.
- Blackjack: a função main, inicializa o jogo.

## Extras:

Para melhorar a experiência do jogo, decidimos por implementar uma inteligência artificial, já descrita acima, e o sistema de apostas, que impede o jogador de continuar na mesa se tiver menos de 10 reais. As apostas deixam o jogo mais real, aproximando-o dos cenários de casinos, enquanto que a inteligência artificial é uma boa comparação para as escolhas dos jogadores, uma vez que segue um padrão de decisões recomendado para cada situação, como mostrado na figura abaixo:

Your Card Total	2	3	4	5	6	7	8	9	10	A
17 - 21	S	S	S	S	S	S	S	S	S	S
16	S	S	S	S	S	H	H	H	H	H
15	S	S	S	S	S	H	H	H	H	H
14	S	S	S	S	S	H	H	H	H	H
13	S	S	S	S	S	H	H	H	H	H
12	H	H	S	S	S	H	H	H	H	H
11	D	D	D	D	D	D	D	D	D	D
10	D	D	D	D	D	D	D	D	H	H
9	H	D	D	D	D	H	H	H	H	H
1 - 8	H	H	H	H	H	H	H	H	H	H
S= Stand		H= Hit		D= Double Down						

Para realizar a criação de uma inteligência artificial, basta digitar, quando for perguntando pelo programa, no lugar do nome do jogador, a palavra-chave “IA” (sem as aspas). Com isso, aquele

jogador será tratado como uma inteligência artificial pelo programa e tomará as decisões automaticamente, incluindo a saída de suas escolhas no console.

### **Conclusão:**

Por fim, o problema proposto foi adequadamente abstraído, modelado, e implementado, de acordo com as regras escolhidas. O maior desafio foi, novamente, a modularização do problema, decidir as responsabilidades de cada classe e suas interrelações.

### **Bibliografia:**

- <http://www.onlinecasinobasics.com/blackjack-strategy/top-5-blackjack-tips/>
- <https://en.wikipedia.org/wiki/Blackjack>
- <https://stackoverflow.com/>
- <https://wizardofodds.com/>
- <https://www.reddit.com/r/blackjack/>