Pontifícia Universidade Católica do Rio Grande do Sul – PUCRS
Faculdade de Informática
Programação Distribuída – Turma 128 – Prof. Roland Teodorowitsch – 2018/1

TRABALHO 2: UNO AUTOMATIZADO EM JAVA WEB SERVICES

Introdução

Uno é o nome da versão comercial (produzida pela empresa Mattel) de um jogo de cartas em que o objetivo dos jogadores é se livrar de suas cartas. Há variações jogadas com um baralho comum (por exemplo, *Mau-mau*) ou produzidas por outros fabricantes (por exemplo, *Can Can* da Grow Jogos e Brinquedos Ltda.). Para a definição deste trabalho serão usadas as regras do Uno.

Uno é um jogo formado por 108 cartas:

- 19 cartas azuis (uma com valor 0 e duas de cada um dos valores de 1 a 9);
- 19 cartas verdes (uma com valor 0 e duas de cada um dos valores de 1 a 9);
- 19 cartas vermelhas (uma com valor 0 e duas de cada um dos valores de 1 a 9);
- 19 cartas amarelas (uma com valor 0 e duas de cada um dos valores de 1 a 9);
- 8 cartas +2 (comprar 2 cartas) – 2 de cada cor (azul, verde, vermelho e amarelo);
- 8 cartas Inverter – 2 de cada cor (azul, verde, vermelho e amarelo);
- 8 cartas Pular – 2 de cada cor (azul, verde, vermelho e amarelo);
- 4 Curingas;
- 4 Curingas +4 (comprar 4 cartas).

A Figura 1 mostra 9 exemplos de cartas de Uno. O verso das cartas corresponde à ilustração que está na coluna mais à esquerda e na linha inferior da Figura 1.

Figura 1 – Algumas cartas de Uno (BARALHO-DO-JOGO-UNO.JPG, [s.d.])

O jogo pode ser disputado de 2 a 10 jogadores, acumulando pontos de várias partidas. No entanto para a finalidade deste trabalho será considerada uma única partida entre 2 jogadores.

Desta forma, as regras apresentadas a seguir foram em grande parte transcritas do Manual de Instruções do jogo (UNO, 2003), com as adaptações e simplificações para 2 jogadores jogando de forma distribuída já aplicadas.

Inicialmente as 108 cartas são embaralhadas, sendo que cada jogador receberá 7 destas cartas. A próxima carta do baralho é colocada com a face voltada para cima e corresponderá à pilha de descarte. As demais cartas corresponderão à pilha de compra e devem ficar com a face voltada para baixo. Iniciará a partida o primeiro jogador a ter se registrado no servidor para jogar Uno. A partir daí os jogadores farão jogadas alternadas tentando se livrar das cartas que têm em suas mãos.

Eventualmente a sequência de jogo poderá ser alterada por uma das Cartas de Ação. Para dois jogadores a única alteração possível na sequência de jogo corresponde a pular o próximo jogador. Em uma partida de dois jogadores, pular um jogador corresponderá a jogar duas vezes.

A maioria das cartas possui duas características básicas: uma cor e um número. Na sua vez de jogar o jogador deve descartar uma carta que tenha ou a mesma cor ou o mesmo valor que a carta que está no topo da pilha de descarte. Por exemplo, se a carta que está no topo da pilha de descarte é a carta com o número 7 em azul. Pode-se jogar qualquer carta com 7 (de qualquer cor), qualquer carta azul (com qualquer número), ou uma das cartas de curinga (que permitem inclusive que o jogador selecione a cor ativa para a próxima jogada).

Desta forma, na sua vez de jogar o jogador deve descartar uma das cartas de sua mão que corresponda ao número, ou à cor ou que seja um curinga. Caso o jogador não possua nenhuma destas cartas, ele deverá comprar uma carta da pilha de compra e ~~. Se esta carta servir, o jogador poderá descartá-la imediatamente. Se não servir, o jogador~~ perderá a vez.

Mesmo que o jogador tenha uma carta que possa ser jogada, pode ser uma estratégia de jogo não descartá-la. Por exemplo, se tiver em sua mão uma carta e não quiser jogá-la, o jogador deverá comprar uma carta da pilha de compra, no entanto, perdendo a vez. ~~Se tiver comprado uma carta da pilha de compra, o jogador poderá também optar em descartar esta carta ou não. Mas não é permitido comprar uma carta da pilha de compra, para na sequência descartar uma carta que o jogador já tinha na sua mão.~~

Algumas cartas não contêm números. Estas são chamadas Cartas de Ação. Elas são as seguintes:

- “+2”: quando esta carta é jogada, o próximo jogador deverá comprar 2 cartas e perderá a vez (ou seja, em um jogo com dois jogadores, o próximo jogador compra 2 cartas e o mesmo jogador continua jogando). Esta carta pode ser jogada sobre uma de mesma cor ou sobre outras cartas “+2”. Se ela for aberta no começo do jogo, as mesmas regras se aplicam (ou seja, o jogador que iniciaria jogando, compra 2 cartas e perde a vez e a carta “+2” permanece no topo do descarte).
- “pular”: esta carta pular o próximo jogador, o que em um jogo entre dois jogadores corresponde a fazer com que o jogador que a jogou continue jogando. Esta carta somente pode ser jogada sobre uma carta da mesma cor ou sobre uma outra carta “pular”. Se uma carta “pular” for aberta no início do jogo, o jogador que estava previsto para iniciar o jogo perde a vez e o outro jogador iniciará o jogo (a carta “pular” permanece no topo do descarte).
- “inverter”: em um jogo entre dois jogadores, esta carta tem exatamente o mesmo efeito da carta “pular”.
- “curinga”: na sua vez, o jogador pode jogar esta carta sobre qualquer outra carta, independentemente de cor e número. ~~Se esta carta for aberta no início do jogo, o primeiro jogador pode jogar qualquer uma de suas cartas (independentemente de cor e número).~~ O jogador que jogar esta carta deverá escolher a cor para a próxima jogada. Se esta carta for aberta no começo do jogo, ela é descartada (vai para a pilha de descarte) e outra carta é aberta.
- “curinga +4”: esta é a melhor carta do jogo. Além de poder ser jogada sobre qualquer cor ou

número e do jogador escolher a cor para a próxima jogada, o próximo jogador deverá comprar 4 cartas da pilha de compra, sem perder a vez. ~~Esta carta somente poderá ser jogada se o jogador não tiver nenhuma carta da mesma cor ativa no monte de descarte.~~ Se esta carta for aberta no começo do jogo, ela é descartada (vai para a pilha de descarte) e outra carta é aberta.

Caso um jogador consiga descartar todas as suas cartas, ele vence a partida. ~~E recebe o número de pontos correspondente à soma dos valores das cartas que estavam na mão do outro jogador.~~

O jogo também pode terminar quando a pilha de compra tiver se esgotado ~~e o próximo jogador não tiver mais nenhuma opção de jogo possível~~. Neste caso, calcula-se a pontuação de cada jogador, somando-se os valores das cartas de cada jogador. ~~Os pontos de um jogador correspondem à soma das cartas do outro jogador.~~ Vence quem terminar o jogo totalizando menos pontos em sua mão. Ou se a pontuação for a mesma, é decretado empate.

Os pontos das cartas são definidos da seguinte forma:

- cartas com número valem o respectivo valor do número;
- cartas "+2", "pular" e "inverter" valem 20 pontos;
- cartas de curinga valem 50 pontos.

Na versão distribuída não é necessário avisar que está com uma única carta gritando "Uno!". Também não há penalidades.

Objetivos

Os objetivos deste trabalho são:

- exercitar a programação distribuída usando ***Web Services*** na linguagem **Java**;
- desenvolver uma aplicação formada por dois programas, um servidor (executado por um processo) e um cliente (eventualmente executado por vários processos) que interagem entre si para a solução de determinado problema;
- desenvolver uma aplicação servidora capaz de receber acessos concorrentes, gerenciando vários jogos simultaneamente, sem apresentar falhas de consistência.

Definição

Neste trabalho deverá ser implementada uma aplicação distribuída em **Java *Web Services*** (usando SOAP – *Simple Object Access Protocol*) que permita o gerenciamento de várias partidas simultâneas entre 2 jogadores do jogo Uno (conforme regras definidas na seção Introdução).

A aplicação deverá ser formada por dois programas: um servidor (executado por um processo) e um cliente (eventualmente executado por vários processos).

Quando o servidor for iniciado, ele deverá ser criado de forma que se possa disputar até 500 partidas simultâneas. Este servidor deverá funcionar de forma muito parecida com o servidor implementado como Trabalho 1 desta disciplina. No entanto, antes do registro normal dos jogadores para iniciar uma

partida, o servidor deverá aceitar o "pré-registro" dos jogadores, onde informa-se o nome do primeiro jogador, o identificador que este primeiro jogador receberá quando ele fizer o registro, o nome do segundo jogador e o identificador que este segundo jogador receberá quando ele fizer o registro. Esta especificação de nomes e seus respectivos identificadores servirá apenas como ferramenta de teste automatizado do servidor. Se for feito o pré-registro envolvendo 2 jogadores, eles deverão ser associados na mesma partida, mesmo que entre o registro deles ocorra o registro de outros jogadores.

O programa cliente, por sua vez, terá uma estrutura diferente da estrutura do cliente tradicional para execução de uma aplicação interativa (portanto, diferente da estrutura do cliente sugerida para o Trabalho 1 desta disciplina). A nova estrutura do cliente lerá de um arquivo (com a extensão ".in") a especificação de um conjunto de chamadas remotas que devem ser executadas no servidor. E deverá salvar em outro arquivo (com a extensão ".out") o resultado da execução de cada uma destas chamadas.

Para iniciar uma partida de Uno no servidor remoto, é preciso fazer o registro de dois jogadores, cada um recebendo como resposta um identificador único (que será usado nas demais chamadas). O uso de determinado nome de jogador deve ser feito de forma única, sem permitir dois jogadores com o mesmo nome e reservando-se o direito de uso de determinado nome a quem registrar-se antes no servidor. Com o pré-registro, será possível prever qual o identificador que determinado jogador receberá no seu registro, e assim predefinir uma série de operações.

Especificação de Entradas e Saídas de um Cliente

O programa cliente deverá ser capaz de ler a especificação de um conjunto de chamadas remotas de um arquivo com a extensão ".in", e deverá ser capaz de escrever as respectivas respostas em um arquivo com a extensão ".out". Entradas e saídas serão sempre fornecidas em formato texto (codificação ASCII, sem acentos).

A especificação da entrada começa com o número total de operações remotas que o cliente deve enviar para o servidor. Na sequência aparece uma linha para cada operação remota. Cada uma destas linhas contém o código da operação seguido de um caractere de tabulação e dos parâmetros da operação remota separados por ":".

Devem ser usados os seguintes códigos para as operações:

- Operação 0[1] – `preRegistro` (usada para viabilizar o teste)
  Informa ao servidor o nome de um jogador (o primeiro da dupla), o identificador que o servidor deverá utilizar para este primeiro jogador, o nome de outro jogador (o segundo da dupla) e o respectivo identificador que o servidor deverá utilizar para este segundo jogador. Esta operação retorna sempre 0 e não haverá nenhuma inconsistência nas entradas referente às operações de pré-registro (ou seja, elas serão sempre consistentes).
- Operação 1 – **registraJogador**
  Recebe: *string* com o nome do usuário/jogador.
  Retorna: id (valor inteiro) do usuário (que corresponde a um número de identificação único para este

1 Esta operação serve para viabilizar os testes, de forma que as demais operações (que necessitam especificar os identificadores dos jogadores) possam ser previamente especificadas com os identificadores que cada jogador receberá após o registro. Nas entradas de teste, garante-se apenas que os registros dos jogadores de uma dupla (especificados em determinado pré-registro) ocorrerão sempre na mesma ordem deste pré-registro. Após o registro, os dados do respectivo pré-registro não serão mais necessários e podem ser excluídos.

usuário durante uma partida), -1 se este usuário já está cadastrado ou -2 se o número máximo de jogadores tiver sido atingido.

- Operação 2 – **encerraPartida**
  Recebe: id do usuário (obtido através da chamada **registraJogador**).
  Retorna: -1 (erro), 0 (ok).
- Operação 3 – **temPartida**
  Recebe: id do usuário (obtido através da chamada **registraJogador**).
  Retorna: -2 (tempo de espera esgotado), -1 (erro), 0 (ainda não há partida), 1 (sim, há partida e o jogador inicia jogando) ou 2 (sim, há partida e o jogador é o segundo a jogar).
- Operação 4 – **obtemOponente**
  Recebe: id do usuário (obtido através da chamada **registraJogador**).
  Retorna: *string* vazio para erro ou *string* com o nome do oponente.
- Operação 5 – **ehMinhaVez**
  Recebe: id do usuário (obtido através da chamada **registraJogador**).
  Retorna: -2 (erro: ainda não há 2 jogadores registrados na partida), -1 (erro), 0 (não), 1 (sim), 2 (é o vencedor), 3 (é o perdedor), 4 (houve empate), 5 (vencedor por WO), 6 (perdedor por WO).
- Operação 6 – **obtemNumCartasBaralho**
  Recebe: id do usuário (obtido através da chamada **registraJogador**).
  Retorna: -2 (erro: ainda não há 2 jogadores registrados na partida), -1 (erro) ou um valor inteiro com o número de cartas do baralho de compra.
- Operação 7 – **obtemNumCartas**
  Recebe: id do usuário (obtido através da chamada **registraJogador**).
  Retorna: -2 (erro: ainda não há 2 jogadores registrados na partida), -1 (erro) ou um valor inteiro com o número de cartas do próprio jogador.
- Operação 8 – **obtemNumCartasOponente**
  Recebe: id do usuário (obtido através da chamada **registraJogador**).
  Retorna: -2 (erro: ainda não há 2 jogadores registrados na partida), -1 (erro) ou um valor inteiro com o número de cartas do oponente.
- Operação 9 – **mostraMao**
  Recebe: id do usuário (obtido através da chamada **registraJogador**).
  Retorna: *string* vazio em caso de erro ou *string* representando o conjunto de cartas que um jogador tem na sua mão, conforme definido na seção Representação das Cartas.
- Operação 10 – **obtemCartaMesa**
  Recebe: id do usuário (obtido através da chamada **registraJogador**).
  Retorna: *string* vazio em caso de erro ou *string* representando a carta que está no topo da pilha de descarte (usando o mesmo padrão definido pela operação mostraMao).
- Operação 11 – **obtemCorAtiva**
  Recebe: id do usuário (obtido através da chamada **registraJogador**).
  Retorna: -2 (erro: ainda não há 2 jogadores registrados na partida), -1 (erro) ou valor inteiro correspondendo à cor que está ativa no topo do baralho de descarte (0 = azul; 1 = amarelo; 2 = verde; 3 = vermelho) – esta informação é necessária nos casos onde um jogador descartou um curinga e escolheu determinada cor.
- Operação 12 – **compraCarta**
  Recebe: id do usuário (obtido através da chamada **registraJogador**).
  Retorna: 1 (código de sucesso), -1 (jogador não encontrado), -2 (partida não iniciada: ainda não há

dois jogadores registrados na partida), -3 (não é a vez do jogador).
Observações: (a) sempre que uma carta é comprada ou acrescentada na mão do jogador (em qualquer situação), ela é colocada no final da sua pilha, sem qualquer ordenação; (b) depois de comprar uma carta, o jogador sempre perde a vez; (c) não há problema algum em um jogador comprar uma carta, mesmo quando já possuir uma carta que poderia ser jogada; (d) um jogador somente poderá comprar uma carta na sua vez de jogar.

- Operação 13 – **jogaCarta**
  Recebe: id do usuário (obtido através da chamada **registraJogador**), índice da carta da mão que deve ser jogada (de 0 até o número máximo de cartas do jogador menos 1) e cor da carta (0 = azul; 1 = amarelo; 2 = verde; 3 = vermelho), no caso da carta jogada ser um curinga.
  Retorna: 1 (tudo certo), 0 (jogada inválida: por exemplo, a carta não corresponde à cor que está na mesa), -1 (jogador não encontrado), -2 (partida não iniciada: ainda não há dois jogadores registrados na partida), -3 (não é a vez do jogador), -4 (parâmetros inválidos). O parâmetro cor da carta somente deve ser validado e/ou utilizado se for jogada uma carta de curinga.
  Observação: ao jogar uma carta válida o jogador que jogou a carta passa a vez para o outro jogador, a não ser que tenha sido jogada uma carta "+2", "pular" ou "inverter".
- Operação 14 – **obtemPontos**
  Recebe: id do usuário (obtido através da chamada **registraJogador**).
  Retorna: valor inteiro com o número de pontos conquistado no jogo, -1 (jogador não encontrado), -2 (partida não iniciada: ainda não há dois jogadores registrados na partida)~~, -3 (a partida ainda não foi concluída)~~.
- Operação 15 – **obtemPontosOponente**
  Recebe: id do usuário (obtido através da chamada **registraJogador**).
  Retorna: valor inteiro com o número de pontos conquistado no jogo pelo adversário, -1 (jogador não encontrado), -2 (partida não iniciada: ainda não há dois jogadores registrados na partida)~~, -3 (a partida ainda não foi concluída)~~.

A sequência a seguir corresponde a um exemplo de entrada que mostra 2 pré-registros sendo feitos. A primeira partida será entre os jogadores "J1" (identificador 1) e "J2" (identificador 2). E a outra será entre os dois jogadores "J3" (identificador 3) e "J4" (identificador 4). Esta segunda partida, no entanto, será encerrada logo em seguida. A primeira partida é desenvolvida até o final, sendo vencida pelo primeiro jogador ("J1"). Ao longo desta partida são feitos vários testes para verificar se as operações remotas estão gerando os resultados esperados.

Entrada com a especificação das operações		Resultado esperado para cada operação
628		
0	J1:1:J2:2	0
0	J3:3:J4:4	0
1	J1	1
3	1	0
5	1	-2
4	1	
1	J3	3
3	3	0
5	3	-2
4	3	
1	J2	2
1	J4	4
1	J1	-1
3	4	2
5	4	1
4	4	J3
3	3	1
5	3	0
4	3	J4
2	3	0
2	4	0

3	2	2
6	2	93
9	2	3/Vd|6/Vm|2/Am|4/Vm|6/Vm|In/Am|9/Am
7	2	7
14	2	50
4	2	J1
8	2	7
15	2	108
10	2	3/Vd
11	2	2
5	2	0
3	1	1
6	1	93
9	1	+2/Vd|C4/*|8/Az|2/Vd|7/Am|1/Az|+2/Am
7	1	7
14	1	108
4	1	J2
8	1	7
15	1	50
10	1	3/Vd
11	1	2
5	1	1
13	2:0:-1	-3
13	1:7:-1	-4
13	1:2:-1	0
13	1:4:-1	0
13	1:5:-1	0
13	1:6:-1	0
13	1:3:-1	1
6	1	93
9	1	+2/Vd|C4/*|8/Az|7/Am|1/Az|+2/Am
7	1	6
14	1	106
4	1	J2
8	1	7
15	1	50
10	1	2/Vd
11	1	2
5	1	0
6	2	93
9	2	3/Vd|6/Vm|2/Am|4/Vm|6/Vm|In/Am|9/Am
7	2	7
14	2	50
4	2	J1
8	2	6
15	2	106
10	2	2/Vd
11	2	2
5	2	1
13	1:0:-1	-3
13	2:7:-1	-4
13	2:1:-1	0
13	2:3:-1	0
13	2:4:-1	0
13	2:5:-1	0
13	2:6:-1	0
13	2:0:-1	1
6	2	93
9	2	6/Vm|2/Am|4/Vm|6/Vm|In/Am|9/Am
7	2	6
14	2	47
4	2	J1
8	2	6
15	2	106
10	2	3/Vd
11	2	2
5	2	0
6	1	93
9	1	+2/Vd|C4/*|8/Az|7/Am|1/Az|+2/Am
7	1	6
14	1	106
4	1	J2
8	1	6
15	1	47
10	1	3/Vd
11	1	2
5	1	1
13	2:0:-1	-3
13	1:6:-1	-4
13	1:2:-1	0
13	1:3:-1	0
13	1:4:-1	0
13	1:5:-1	0
13	1:0:-1	1
6	1	91
9	1	C4/*|8/Az|7/Am|1/Az|+2/Am
7	1	5
14	1	86
4	1	J2
8	1	8
15	1	55
10	1	+2/Vd
11	1	2

5	1	1
13	2:0:-1	-3
13	1:5:-1	-4
13	1:1:-1	0
13	1:2:-1	0
13	1:3:-1	0
13	1:4:-1	1
6	1	89
9	1	C4/*|8/Az|7/Am|1/Az
7	1	4
14	1	66
4	1	J2
8	1	10
15	1	79
10	1	+2/Am
11	1	1
5	1	1
13	2:0:-1	-3
13	1:4:-1	-4
13	1:1:-1	0
13	1:3:-1	0
13	1:2:-1	1
6	1	89
9	1	C4/*|8/Az|1/Az
7	1	3
14	1	59
4	1	J2
8	1	10
15	1	79
10	1	7/Am
11	1	1
5	1	0
6	2	89
9	2	6/Vm|2/Am|4/Vm|6/Vm|In/Am|9/Am|8/Am|0/Vd|+2/Az|4/Vd
7	2	10
14	2	79
4	2	J1
8	2	3
15	2	59
10	2	7/Am
11	2	1
5	2	1
13	1:0:-1	-3
13	2:10:-1	-4
13	2:0:-1	0
13	2:2:-1	0
13	2:3:-1	0
13	2:7:-1	0
13	2:8:-1	0
13	2:9:-1	0
13	2:4:-1	1
6	2	89
9	2	6/Vm|2/Am|4/Vm|6/Vm|9/Am|8/Am|0/Vd|+2/Az|4/Vd
7	2	9
14	2	59
4	2	J1
8	2	3
15	2	59
10	2	In/Am
11	2	1
5	2	1
13	1:0:-1	-3
13	2:9:-1	-4
13	2:0:-1	0
13	2:2:-1	0
13	2:3:-1	0
13	2:6:-1	0
13	2:7:-1	0
13	2:8:-1	0
13	2:1:-1	1
6	2	89
9	2	6/Vm|4/Vm|6/Vm|9/Am|8/Am|0/Vd|+2/Az|4/Vd
7	2	8
14	2	57
4	2	J1
8	2	3
15	2	59
10	2	2/Am
11	2	1
5	2	0
6	1	89
9	1	C4/*|8/Az|1/Az
7	1	3
14	1	59
4	1	J2
8	1	8
15	1	57
10	1	2/Am
11	1	1
5	1	1
13	2:0:-1	-3
13	1:3:-1	-4
13	1:1:-1	0

13	1:2:-1	0
13	1:0:-1	-4
13	1:0:0	1
6	1	85
9	1	8/Az|1/Az
7	1	2
14	1	9
4	1	J2
8	1	12
15	1	84
10	1	C4/*
11	1	0
5	1	0
6	2	85
9	2	6/Vm|4/Vm|6/Vm|9/Am|8/Am|0/Vd|+2/Az|4/Vd|2/Vd|+2/Vd|0/Vm|5/Am
7	2	12
14	2	84
4	2	J1
8	2	2
15	2	9
10	2	C4/*
11	2	0
5	2	1
13	1:0:-1	-3
13	2:12:-1	-4
13	2:0:-1	0
13	2:1:-1	0
13	2:2:-1	0
13	2:3:-1	0
13	2:4:-1	0
13	2:5:-1	0
13	2:7:-1	0
13	2:8:-1	0
13	2:9:-1	0
13	2:10:-1	0
13	2:11:-1	0
13	2:6:-1	1
6	2	83
9	2	6/Vm|4/Vm|6/Vm|9/Am|8/Am|0/Vd|4/Vd|2/Vd|+2/Vd|0/Vm|5/Am
7	2	11
14	2	64
4	2	J1
8	2	4
15	2	63
10	2	+2/Az
11	2	0
5	2	1
13	1:0:-1	-3
13	2:11:-1	-4
13	2:0:-1	0
13	2:1:-1	0
13	2:2:-1	0
13	2:3:-1	0
13	2:4:-1	0
13	2:5:-1	0
13	2:6:-1	0
13	2:7:-1	0
13	2:9:-1	0
13	2:10:-1	0
13	2:8:-1	1
6	2	81
9	2	6/Vm|4/Vm|6/Vm|9/Am|8/Am|0/Vd|4/Vd|2/Vd|0/Vm|5/Am
7	2	10
14	2	44
4	2	J1
8	2	6
15	2	69
10	2	+2/Vd
11	2	2
5	2	1
13	1:0:-1	-3
13	2:10:-1	-4
13	2:0:-1	0
13	2:1:-1	0
13	2:2:-1	0
13	2:3:-1	0
13	2:4:-1	0
13	2:8:-1	0
13	2:9:-1	0
13	2:5:-1	1
6	2	81
9	2	6/Vm|4/Vm|6/Vm|9/Am|8/Am|4/Vd|2/Vd|0/Vm|5/Am
7	2	9
14	2	44
4	2	J1
8	2	6
15	2	69
10	2	0/Vd
11	2	2
5	2	0
6	1	81
9	1	8/Az|1/Az|C4/*|4/Am|1/Am|5/Az
7	1	6

14	1	69
4	1	J2
8	1	9
15	1	44
10	1	0/Vd
11	1	2
5	1	1
13	2:0:-1	-3
13	1:6:-1	-4
13	1:0:-1	0
13	1:1:-1	0
13	1:3:-1	0
13	1:4:-1	0
13	1:5:-1	0
13	1:2:0	1
6	1	77
9	1	8/Az|1/Az|4/Am|1/Am|5/Az
7	1	5
14	1	19
4	1	J2
8	1	13
15	1	84
10	1	C4/*
11	1	0
5	1	0
6	2	77
9	2	6/Vm|4/Vm|6/Vm|9/Am|8/Am|4/Vd|2/Vd|0/Vm|5/Am|6/Am|7/Vm|7/Am|Pu/Vd
7	2	13
14	2	84
4	2	J1
8	2	5
15	2	19
10	2	C4/*
11	2	0
5	2	1
13	1:0:-1	-3
13	2:13:-1	-4
13	2:0:-1	0
13	2:1:-1	0
13	2:2:-1	0
13	2:3:-1	0
13	2:4:-1	0
13	2:5:-1	0
13	2:6:-1	0
13	2:7:-1	0
13	2:8:-1	0
13	2:9:-1	0
13	2:10:-1	0
13	2:11:-1	0
13	2:12:-1	0
12	1	-3
12	2	1
6	2	76
9	2	6/Vm|4/Vm|6/Vm|9/Am|8/Am|4/Vd|2/Vd|0/Vm|5/Am|6/Am|7/Vm|7/Am|Pu/Vd|1/Vd
7	2	14
14	2	85
4	2	J1
8	2	5
15	2	19
10	2	C4/*
11	2	0
5	2	0
6	1	76
9	1	8/Az|1/Az|4/Am|1/Am|5/Az
7	1	5
14	1	19
4	1	J2
8	1	14
15	1	85
10	1	C4/*
11	1	0
5	1	1
13	2:0:-1	-3
13	1:5:-1	-4
13	1:2:-1	0
13	1:3:-1	0
13	1:0:-1	1
6	1	76
9	1	1/Az|4/Am|1/Am|5/Az
7	1	4
14	1	11
4	1	J2
8	1	14
15	1	85
10	1	8/Az
11	1	0
5	1	0
6	2	76
9	2	6/Vm|4/Vm|6/Vm|9/Am|8/Am|4/Vd|2/Vd|0/Vm|5/Am|6/Am|7/Vm|7/Am|Pu/Vd|1/Vd
7	2	14
14	2	85
4	2	J1
8	2	4

15	2	11
10	2	8/Az
11	2	0
5	2	1
13	1:0:-1	-3
13	2:14:-1	-4
13	2:0:-1	0
13	2:1:-1	0
13	2:2:-1	0
13	2:3:-1	0
13	2:5:-1	0
13	2:6:-1	0
13	2:7:-1	0
13	2:8:-1	0
13	2:9:-1	0
13	2:10:-1	0
13	2:11:-1	0
13	2:12:-1	0
13	2:13:-1	0
13	2:4:-1	1
6	2	76
9	2	6/Vm|4/Vm|6/Vm|9/Am|4/Vd|2/Vd|0/Vm|5/Am|6/Am|7/Vm|7/Am|Pu/Vd|1/Vd
7	2	13
14	2	77
4	2	J1
8	2	4
15	2	11
10	2	8/Am
11	2	1
5	2	0
6	1	76
9	1	1/Az|4/Am|1/Am|5/Az
7	1	4
14	1	11
4	1	J2
8	1	13
15	1	77
10	1	8/Am
11	1	1
5	1	1
13	2:0:-1	-3
13	1:4:-1	-4
13	1:0:-1	0
13	1:3:-1	0
13	1:2:-1	1
6	1	76
9	1	1/Az|4/Am|5/Az
7	1	3
14	1	10
4	1	J2
8	1	13
15	1	77
10	1	1/Am
11	1	1
5	1	0
6	2	76
9	2	6/Vm|4/Vm|6/Vm|9/Am|4/Vd|2/Vd|0/Vm|5/Am|6/Am|7/Vm|7/Am|Pu/Vd|1/Vd
7	2	13
14	2	77
4	2	J1
8	2	3
15	2	10
10	2	1/Am
11	2	1
5	2	1
13	1:0:-1	-3
13	2:13:-1	-4
13	2:0:-1	0
13	2:1:-1	0
13	2:2:-1	0
13	2:4:-1	0
13	2:5:-1	0
13	2:6:-1	0
13	2:9:-1	0
13	2:11:-1	0
13	2:8:-1	1
6	2	76
9	2	6/Vm|4/Vm|6/Vm|9/Am|4/Vd|2/Vd|0/Vm|5/Am|7/Vm|7/Am|Pu/Vd|1/Vd
7	2	12
14	2	71
4	2	J1
8	2	3
15	2	10
10	2	6/Am
11	2	1
5	2	0
6	1	76
9	1	1/Az|4/Am|5/Az
7	1	3
14	1	10
4	1	J2
8	1	12
15	1	71

10	1	6/Am
11	1	1
5	1	1
13	2:0:-1	-3
13	1:3:-1	-4
13	1:0:-1	0
13	1:2:-1	0
13	1:1:-1	1
6	1	76
9	1	1/Az|5/Az
7	1	2
14	1	6
4	1	J2
8	1	12
15	1	71
10	1	4/Am
11	1	1
5	1	0
6	2	76
9	2	6/Vm|4/Vm|6/Vm|9/Am|4/Vd|2/Vd|0/Vm|5/Am|7/Vm|7/Am|Pu/Vd|1/Vd
7	2	12
14	2	71
4	2	J1
8	2	2
15	2	6
10	2	4/Am
11	2	1
5	2	1
13	1:0:-1	-3
13	2:12:-1	-4
13	2:0:-1	0
13	2:2:-1	0
13	2:5:-1	0
13	2:6:-1	0
13	2:8:-1	0
13	2:10:-1	0
13	2:11:-1	0
13	2:7:-1	1
6	2	76
9	2	6/Vm|4/Vm|6/Vm|9/Am|4/Vd|2/Vd|0/Vm|7/Vm|7/Am|Pu/Vd|1/Vd
7	2	11
14	2	66
4	2	J1
8	2	2
15	2	6
10	2	5/Am
11	2	1
5	2	0
6	1	76
9	1	1/Az|5/Az
7	1	2
14	1	6
4	1	J2
8	1	11
15	1	66
10	1	5/Am
11	1	1
5	1	1
13	2:0:-1	-3
13	1:2:-1	-4
13	1:0:-1	0
13	1:1:-1	1
6	1	76
9	1	1/Az
7	1	1
14	1	1
4	1	J2
8	1	11
15	1	66
10	1	5/Az
11	1	0
5	1	0
6	2	76
9	2	6/Vm|4/Vm|6/Vm|9/Am|4/Vd|2/Vd|0/Vm|7/Vm|7/Am|Pu/Vd|1/Vd
7	2	11
14	2	66
4	2	J1
8	2	1
15	2	1
10	2	5/Az
11	2	0
5	2	1
13	1:0:-1	-3
13	2:11:-1	-4
13	2:0:-1	0
13	2:1:-1	0
13	2:2:-1	0
13	2:3:-1	0
13	2:4:-1	0
13	2:5:-1	0
13	2:6:-1	0
13	2:7:-1	0
13	2:8:-1	0

13	2:9:-1	0
13	2:10:-1	0
12	1	-3
12	2	1
6	2	75
9	2	6/Vm|4/Vm|6/Vm|9/Am|4/Vd|2/Vd|0/Vm|7/Vm|7/Am|Pu/Vd|1/Vd|6/Vd
7	2	12
14	2	72
4	2	J1
8	2	1
15	2	1
10	2	5/Az
11	2	0
5	2	0
6	1	75
9	1	1/Az
7	1	1
14	1	1
4	1	J2
8	1	12
15	1	72
10	1	5/Az
11	1	0
5	1	1
13	2:0:-1	-3
13	1:-4:-1	-4
13	1:0:-1	1
6	1	75
9	1	
7	1	0
14	1	0
4	1	J2
8	1	12
15	1	72
10	1	1/Az
11	1	0
5	1	2
6	2	75
9	2	6/Vm|4/Vm|6/Vm|9/Am|4/Vd|2/Vd|0/Vm|7/Vm|7/Am|Pu/Vd|1/Vd|6/Vd
7	2	12
14	2	72
4	2	J1
8	2	0
15	2	0
10	2	1/Az
11	2	0
5	2	3
2	1	0
2	2	0

Nesta entrada há 628 operações remotas especificadas. Na primeira coluna há o código da operação (conforme a definição de operações citada anteriormente) separado dos parâmetros da operação por um caractere de tabulação. Na coluna da direita é apresentada a saída (resultado gerado pela execução da operação) correspondente à operação da coluna da esquerda.

Quando há mais de um parâmetro para a operação, estes parâmetros estarão separados pelo caractere ":". Inicialmente usa-se a operação de número 0 para pré-registrar os jogadores de duas partidas. A primeira será formada pelos jogadores "J1" (com identificador 1) e "J2" (com identificador 2). Isto significa que eles que receberão respectivamente os identificadores 1 e 2 e **serão alocados na mesma partida** assim que fizerem o seu registro. A segunda operação de pré-registro é para os jogadores "J3" (identificador 3) e "J4" (identificador 4), que serão alocados na mesma partida. Na sequência, aparecem operações de registro (número 1) para "J1", "J3", "J4" e "J2" (nesta ordem), sendo que, mesmo que os registros dos jogadores das partidas estejam intercalados, eles serão corretamente alocados nas partidas, conforme o pré-registro. Nos arquivos com as definições de operações, apenas se garante que o primeiro jodador citado no pré-registro será registrado antes do que o segundo deste mesmo pré-registro. Na sequência, os jogadores de identificadores 3 e 4 executam a operação de número 2 (para abandonar o jogo, ou seja, encerrar sua partida). Seguem-se várias chamadas alternadas que executam uma partida entre os dois jogadores da primeira partida. O jogador "J1" (que inicia jogando) vence a partida contra o jogador "J2". Por fim, ambos os jogadores executam a chamada para encerramento da partida (operação 2).

Representação das Cartas

Para identificar e representar todas as cartas e principalmente as cartas que fazem parte da mão do jogador deve-se a atribuir a cada uma das 108 cartas do jogo um valor numérico de 0 a 107 e também um rótulo que permita identificar a carta e suas características (cor e valor, por exemplo) de uma forma mais amigável. O Quadro 1 mostra as equivalências entre valores numéricos e rótulos a serem usados na representação das cartas. A operação **mostraMao** poderia, por exemplo, retornar a seguinte cadeia de caracteres "3/Az|1/Vd|0/Az|+2/Vm|In/Am|Cg/*|3/Am", representando as 7 cartas iniciais para determinado jogador (3 azul, 1 verde, 0 azul, "+2" vermelho, "inverter" amarelo, curinga, 3 amarelo), separadas pelo caractere "|".

Quadro 1 – Representação das Cartas

valor	carta	valor	carta	valor	carta	valor	carta	valor	carta
0	0/Az	25	0/Am	50	0/Vd	75	0/Vm	100	Cg/*
1	1/Az	26	1/Am	51	1/Vd	76	1/Vm	101	Cg/*
2	1/Az	27	1/Am	52	1/Vd	77	1/Vm	102	Cg/*
3	2/Az	28	2/Am	53	2/Vd	78	2/Vm	103	Cg/*
4	2/Az	29	2/Am	54	2/Vd	79	2/Vm	104	C4/*
5	3/Az	30	3/Am	55	3/Vd	80	3/Vm	105	C4/*
6	3/Az	31	3/Am	56	3/Vd	81	3/Vm	106	C4/*
7	4/Az	32	4/Am	57	4/Vd	82	4/Vm	107	C4/*
8	4/Az	33	4/Am	58	4/Vd	83	4/Vm		
9	5/Az	34	5/Am	59	5/Vd	84	5/Vm		
10	5/Az	35	5/Am	60	5/Vd	85	5/Vm		
11	6/Az	36	6/Am	61	6/Vd	86	6/Vm		
12	6/Az	37	6/Am	62	6/Vd	87	6/Vm		
13	7/Az	38	7/Am	63	7/Vd	88	7/Vm		
14	7/Az	39	7/Am	64	7/Vd	89	7/Vm		
15	8/Az	40	8/Am	65	8/Vd	90	8/Vm		
16	8/Az	41	8/Am	66	8/Vd	91	8/Vm		
17	9/Az	42	9/Am	67	9/Vd	92	9/Vm		
18	9/Az	43	9/Am	68	9/Vd	93	9/Vm		
19	Pu/Az	44	Pu/Am	69	Pu/Vd	94	Pu/Vm		
20	Pu/Az	45	Pu/Am	70	Pu/Vd	95	Pu/Vm		
21	In/Az	46	In/Am	71	In/Vd	96	In/Vm		
22	In/Az	47	In/Am	72	In/Vd	97	In/Vm		
23	+2/Az	48	+2/Am	73	+2/Vd	98	+2/Vm		
24	+2/Az	49	+2/Am	74	+2/Vd	99	+2/Vm		

Observação: uma pilha de cartas ou um baralho pode ser representado internamente de várias formas. O exemplo de código a seguir mostra o uso de uma estrutura de dados baseada em um vetor. Inicialmente, para viabilizar a realização dos testes, será preciso criar **para cada partida** um gerador de números aleatórios com semente definida a partir da soma dos identificadores dos dois jogadores envolvidos na partida (`id1` e `id2`):

```
// Inicializacao do gerador de numeros aleatorios
Random gerador = new Random(id1+id2);
```

Os valores obtidos a partir deste gerador serão usados exclusivamente no embaralhamento das cartas. Para criar o baralho ordenado usa-se então:

```
// Criacao do baralho com as 108 cartas
final int totalCartas = 108;
int[] baralho = new int[totalCartas];
for (int i=0;i<totalCartas;++i)
    baralho[i] = i;
```

Finalmente pode-se embaralhar as 108 cartas em 2 etapas:

```
// Embaralhamento
for (int c=0;c<totalCartas;++c) {
    int outra = gerador.nextInt(totalCartas);
    int aux = baralho[c];
    baralho[c] = baralho[outra];
    baralho[outra] = aux;
```

```
}
for (int c=0;c<totalCartas*totalCartas;c++) {
    int c1 = gerador.nextInt(totalCartas);
    int c2 = gerador.nextInt(totalCartas);
    int aux = baralho[c1];
    baralho[c1] = baralho[c2];
    baralho[c2] = aux;
}
int numCartas = totalCartas;
```

Para comprar uma carta deste baralho, deve-se remover sempre a carta da última posição ocupada do baralho:

```
// O baralho sera o monte de compras
int carta = baralho[--numCartas];
```

A distribuição de cartas deverá seguir também uma ordem predefinida. Começar comprando uma carta e atribuindo ela ao primeiro jogador, depois ao segundo e assim alternadamente até que ambos tenham recebido as 7 cartas iniciais. Cada jogador deve ter também a sua pilha que deverá ser gerenciada como uma lista sem ordenação: cartas são colocadas no final da lista e podem ser removidas de qualquer posição.

Avaliação

O programa cliente será usado para submeter vários conjuntos de entradas e suas respectivas operações ao processo servidor. O requisito mínimo para entrega e apresentação corresponde a apresentar corretamente os resultados esperados para o exemplo de entrada apresentado nesta definição. Nos demais casos será avaliado tanto o número de entradas acertadas quanto o nível de erro apresentado.

Outras Especificações:

- O trabalho deverá ser realizado individualmente;
- Não incluir nenhum código-fonte copiado. Em caso de uso de código-fonte não desenvolvido pelos alunos, será atribuída a nota 0 (ZERO) ao trabalho.

Data de Entrega

• 21h15min do dia 26 de junho de 2018. **Não há opção de entrega com atraso.**

Formato de Entrega

• Entregar os arquivos referentes ao código-fonte. Apresentar e descrever verbalmente o funcionamento da aplicação ao professor.

Referências

BARALHO-DO-JOGO-UNO.JPG. Altura: 558 *pixels*. Largura: 900 *pixels*. 300 dpi. 24 bits RGB. 159 KiB. Formato JPEG bitmap. Compactado. [s.d.]. Disponível em: <http://www.metropolybar.com.br/wp-content/uploads/2017/09/baralho-do-jogo-uno.jpg>. Acesso em: 22 mar. 2018.

UNO Jogo de Cartas. Cajamar/SP: Mattel, 2003. Manual de Intruções.