

Universidade Federal de Campina Grande – UFCG

Centro de Engenharia Elétrica e Informática – CEEI

Departamento de Sistemas e Computação – DSC

Disciplina: Laboratório de Programação 2

Período: 2013.2

Professores: Gustavo Soares (turma1)

Kyller Gorgônio (turma 2)

Livia Campos (turma 3)

Raquel Lopes (turma 4)

Laboratório 04

Neste laboratório, experimentaremos a edição e compilação de um programa em Java que use objetos de classes pertencentes a bibliotecas que não façam parte da API de Java. Nesse caso, classes do pacote `p1`, escrito pelo prof. Jacques Sauvé e disponibilizadas em <https://sites.google.com/site/prog2ufcg/recursos>.

INSTRUÇÕES

Este laboratório terá duração de 1 semana, sendo que a primeira parte não deve ser entregue e a segunda parte será entregue seguindo as datas abaixo:

- Data de entrega: 19/11/2013, até 23:59h;
- **Será tolerado o atraso de no máximo 24hs** para a entrega do laboratório, contado de acordo com a turma de cada aluno. Laboratórios entregues com atraso terão uma **penalização de 50%** na nota;
- Crie o projeto lab04 no eclipse e programe todos os exercícios dentro do pacote `lp2.lab04`;
- Salve o pacote criado, com as classes implementadas, em um arquivo zip chamado `turmaX-lab04-<seuNome>.zip` (ou `.tgz`) e o submeta usando o formulário:
 - <http://goo.gl/Ug5oij>
- Certifique-se de que seus programas não têm erros de compilação;
- Antes da definição de cada classe escreva o seu nome em comentário

```
/*  
  
* Aluno: <seu nome>  
  
* Matrícula: <sua matrícula>  
  
*/
```

Já cansados de jogar o jogo-da-velha, os alunos do curso de computação resolveram que era hora de variar um pouco. Depois de uma discussão bastante animada eles decidiram que era

hora de partir para um carteador. Eles queriam jogar o popular blackjack, mais conhecido por essas bandas como vinte-e-um. Disseram que seria difícil, mas nessa hora alguém lembrou do pacote p1 que foi feito pelo professor Jacques que é um conhecido apreciador de jogos de cartas.

SOBRE O JOGO VINTE-E-UM

O vinte-e-um é um jogo de cartas de origem inglesa, também conhecido como Blackjack. O objetivo do jogo é acumular pontos, aproximando-se ao máximo de 21, sem estourar, ou seja, sem ultrapassar este valor.

Um dos jogadores é o carteador. Inicialmente, o carteador entrega duas cartas abertas para cada jogador e uma carta aberta e outra fechada para si mesmo. Em seguida, iniciam-se rodadas em que cada jogador deve decidir se quer ou não mais cartas. Se o jogador estourar, perde. Quando ninguém quiser mais cartas, verifica-se quem se aproximou mais do total de 21 pontos para identificar o ganhador.

ALGUMAS OBSERVAÇÕES

- Quando a mesa fizer 21 pontos a rodada deve terminar e ela é declarada vencedora;
- Quando o jogador fizer 21 pontos ele não deve mais pegar cartas;
- Cartas com figuras (valetes, damas, reis) valem todas 10 pontos;
- A carta “ás” vale 1 ponto;
- Em jogos de cartas, logicamente, o baralho deve ser embaralhado antes de começar.

PARTE 1: FAMILIARIZANDO-SE COM O PACOTE P1

1. Para usar as classes do pacote p1 é preciso configurar seu projeto no eclipse para considerar estas classes no seu Java Build Path;
 - a. Copie o pacote p1 (p1.jar) para sua máquina. Este arquivo está disponível em <https://sites.google.com/site/prog2ufcg/recursos>

Nota: Um arquivo JAR (Java Archive) é um arquivo compactado usado para distribuir um conjunto de classes Java. É usado para armazenar classes compiladas e metadados associados que podem constituir um programa. Arquivos jar podem ser criados e extraídos usando o utilitário "jar" da JDK. Ferramentas de compressão (e.g. Winzip) também podem criar arquivos jar.

- b. Abra o menu *Project* → *Properties*;
 - c. Clique na opção *Java Build Path* e depois na aba *Libraries* e finalmente no botão *Add External Jars*;
 - d. Selecione o arquivo p1.jar usando a caixa de diálogo.
2. No arquivo VinteUmSimples.java¹ contém uma versão bastante simplificada do jogo. Baixe esse arquivo para seu computador e importe-o para dentro do seu projeto como mostrado na aula passada. Execute esse programa no eclipse para se certificar de que as bibliotecas foram devidamente configuradas;
 3. Agora modifique o programa VinteUmSimples para, em vez distribuir cartas até que haja um vencedor, perguntar ao um usuário se o jogador irá pedir mais uma carta ou não. Nesse caso, o nosso jogo será usado por duas pessoas que querem jogar vinte-

¹ Código fonte: <http://sdrv.ms/HZaZnZ>

um mas não possuem um baralho físico disponível; A cada iteração, seu programa deverá:

- a. Exibir a mão de cada jogador;
- b. Adicionar uma nova carta a mão do jogador enquanto o ele quiser;
- c. O jogador poderá responder com 't' para terminar, ou com qualquer outro string (inclusive vazio) para pedir uma nova carta. A entrada de dados será através do teclado. Note que nesse caso, a condição de término do jogo deve ser modificada.

Observação: Lembre-se que existe apenas um baralho disponível e depois que todas as suas cartas forem entregues ao jogador o programa deve terminar.

PARTE 2: IMPLEMENTANDO O JOGO VINTE-E-UM

Um dos problemas com nosso jogo VinteUmSimples é que ou o computador joga sozinho, ou sempre precisamos de alguém pra jogar com a gente. Vamos resolver esse problema.

1. Implemente o programa VinteUmJogo que representará o nosso jogo (uma versão simplificada do jogo vinte-e-um). Para jogar precisaremos tratar duas mãos de cartas simultaneamente, assim, teremos a `maoJogador` (correspondendo à mão de cartas do jogador) e a `maoMesa` (correspondendo à mão de cartas da mesa; o papel do carteador). Seu programa deve apresentar o comportamento a seguir:
 - a. O jogo começa com o jogador e a mesa recebendo uma carta (sem necessidade de perguntar ao jogador se ele deseja alguma carta);
 - b. O valor da carta é usado para acumular pontos ao longo do jogo;
 - c. Uma partida do jogo é composta de várias rodadas. A cada rodada o jogador e a mesa recebem uma carta; deve-se exibir a mão de cartas do jogador juntamente com sua pontuação, e a quantidade de cartas da mesa;
 - d. Quando algum jogador completar 21, a partida será finalizada e este jogador será declarado vencedor;
 - e. Quando o jogador não quiser mais cartas, a mesa deve continuar recebendo cartas enquanto o total de pontos for menor que 17 (a mesa também pode parar de receber cartas antes do jogador, caso esse total de pontos seja alcançado);
 - f. Se a mesa (ou o jogador) estourar, ou seja, acumular mais de 21 pontos, o programa deve declarar o jogador (ou a mesa) como vencedor. Neste ponto, não existe mais entrega de cartas para evitar que a mesa e o jogador estourem ao mesmo tempo;
 - g. Se nem a mesa nem o jogador estourarem, ao final do jogo (quando o jogador e a mesa não desejarem mais cartas) deve ser declarado vencedor aquele que tiver feito o maior número de pontos. Em caso de empate, a mesa será o vencedor;
 - h. Ao final da partida deve-se mostrar na tela a situação final do jogo (quem venceu, quem estourou, se houve empate, etc.), incluindo as cartas na mão do jogador e da mesa e suas respectivas pontuações.
2. Modifique o seu programa VinteUmJogo para permitir que o usuário jogue mais de uma partida em sequência. O programa deve perguntar se o usuário deseja jogar uma nova partida e caso a resposta seja positiva, o programa iniciará uma nova partida. Atente para os seguintes comportamentos do seu jogo:

- a. Quando as cartas do baralho acabarem, o programa deve exibir uma mensagem indicando que o baralho está vazio;
- b. A cada rodada o programa deve exibir na tela o número de cartas restante no baralho. Caso o baralho acabe, deve-se sair do jogo corrente e perguntar ao jogador se deseja continuar no sistema iniciando um novo jogo.

Mais observações

- É importante lembrar o conceito de NÚMEROS MÁGICOS e utilizar isso corretamente;
- Se for necessário realizar comparação entre Strings isso deve ser feito usando o método `equals(String variavel)` que retorna true se as strings forem iguais ou false se forem diferentes.

PARA EXERCITAR MAIS

O problema que temos agora é que o jogo fez um sucesso enorme e todos os alunos da UFCG estão querendo jogá-lo. Modifique seu programa para que até 4 jogares possam disputar uma partida tendo sempre o computador representando a banca. Os jogadores poderão disputar diversas partidas, e ao final seu programa deverá exibir um ranking dos vencedores.