

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: 2011.1

Professores: Livia Campos (turma1)
Reinaldo Gomes (turma2)
Nazareno Andrade (turma 3)

Laboratório 03

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 deve ser entregue no dia de aplicação do laboratório e a segunda parte após 7 dias (na primeira aula após o carnaval).

- Data de entrega (Parte 1): 03/03/2011 até 12h (Turmas 1 e 2) ou 04/03/2011 (Turma 3) até 10h;
- Data de entrega (Parte 2): 10/03/2011 até 10h (Turmas 1 e 2) ou até 12h (Turma 3);
- Crie o projeto lab03 no eclipse e programe todos os exercícios dentro do pacote lp2.lab03;
- Salve as classes criadas em um arquivo zip chamado lab03-<seuNome>-<parte1 ou parte2>.zip (ou .tgz) e o envie para juiz.lp2@gmail.com Use como *subject* do email lab03-<seuNome>-<parte1 ou parte2> (turma 1, 2 ou 3);
- 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>*/*);
- Não copie o programa do seu vizinho. Se tiver dúvida converse com o professor ou com um monitor.

As classes do pacote p1 serão usadas para implementar uma versão simplificada do jogo de cartas Vinte-e-Um (blackjack).

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.
- É importante lembrar o conceito de NÚMEROS MÁGICOS e utilizar isso corretamente.
- Em jogos de cartas, logicamente, o baralho deve ser embaralhado antes de começar.
- A comparação entre Strings é feita usando o método equals(String variavel) que retorna true se as strings forem iguais ou false se forem diferentes.

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. Isso é feito através do menu Project -> Properties -> Java Build Path -> Add External Jars. Copie o Java Archive do pacote p1 (p1.jar) para sua máquina; este arquivo está disponível em <https://sites.google.com/site/prog2ufcg/recursos>). Java Archive (JAR) 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 (como o Winzip) também podem criar arquivos jar.
2. Implemente o programa VinteUmPreparacao que deve criar uma mão com uma carta e exibi-la na tela (a mão de cartas deve ser denominada maoJogador). Lembre-se que você precisará também de um baralho e este baralho deve ser embaralhado antes do jogo começar. Execute o seu programa usando o eclipse para se certificar de que seu projeto está com as bibliotecas devidamente configuradas.
3. Modifique seu programa VinteUmPreparacao para simular o ato de entregar cartas ao jogador enquanto ele quiser. Os seguintes passos devem ser executados pelo seu programa:
 - a. Exibir maoJogador inicialmente vazia;
 - b. Adicionar uma nova carta em maoJogador enquanto o jogador quiser;
 - c. A cada iteração, a nova mão de cartas deve ser exibida na tela;
 - d. A cada iteração, 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.

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

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 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);
 - e. 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.
 - f. 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;
 - g. 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.