



UNIVERSIDADE  
FEDERAL DO CEARÁ

Universidade Federal do Ceará  
Sistemas de Informação/Ciência da Computação  
Disciplina: **Programação Orientada a Objetos**  
Profa. **Amanda Pires**  
Aluno(a): \_\_\_\_\_

## Lista de Exercícios N° 01

1. **Pedra-Papel-Tesoura** é um jogo simples para ser jogado apenas com as mãos e que não requer nenhuma habilidade especial ou equipamentos específicos. Os jogadores devem simultaneamente esticar a mão, formando um símbolo (que significa pedra, papel ou tesoura) e comparam os símbolos para decidir quem ganhou, da seguinte forma:
  - Pedra ganha da tesoura (amassando-a ou quebrando-a);
  - tesoura ganha do papel (cortando-o);
  - papel ganha da pedra (embrulhando-a).

A pedra é simbolizada por um punho fechado; a tesoura, por dois dedos esticados; e o papel, pela mão aberta. Caso dois jogadores façam o mesmo gesto, ocorre um empate, e geralmente se joga de novo até desempatar.

**Passo1:** Implemente um programa chamado PedraPapelTesoura.java para simular o jogo descrito acima. 2 jogadores com seus respectivos nomes. O valor do jogador 1 deve ser lido do teclado usando a classe [java.util.Scanner](#). O valor do Jogador 2 deve ser gerado pela máquina. *Uma dica é usar um gerador de números randômicos que escolha o valor do computador.*

`Math.random() * 3 + 1` //cria um gerador de números randômicos no intervalo de 1-3

Seu programa deve informar o jogador vencedor da rodada. *No jogo quando se tem empate, automaticamente inicia-se uma nova partida. Modifique seu programa para permitir isso.*

Um exemplo de execução do seu programa:

Pedra (1), Papel (2), Tesoura (3)? 3

Pedra (1), Papel (2), Tesoura (3)? 2

Jogador 1 Ganhou!

2. Escreva um programa em Java que dentro do pacote `jogo` contenha duas classes: `Guerreiro` e `Principal`.
  - a) A classe `Guerreiro` deve conter como atributos: nome, arma, energia (20 pontos) e armadura. Um construtor parametrizado, todos os métodos assessores e o método `ataca(Guerreiro guerreiro)`, que verifica se o guerreiro que atacou está armado e se o guerreiro que foi atacado não possui armadura, pois caso não possua, deverá perder energia (5 pontos).
  - b) Na classe `Principal` dois guerreiros devem ser instanciados e receber nomes. O guerreiro 2 recebe sua armadura, o guerreiro 1 ataca o guerreiro 2, o guerreiro 2 ataca o guerreiro 1, o guerreiro 1 recebe sua armadura e também uma arma, o guerreiro 1 ataca o guerreiro 2, o guerreiro 2 não encontra sua arma e ataca o guerreiro 1, quais são as energias dos dois guerreiros?
3. Escreva um programa em Java que leia N ( $N \geq 3$ ) inteiros e imprima o número que mais se repete entre os números lidos, bem como a quantidade de vezes que ele aparece. Por exemplo, se forem fornecidos os números "1, 2, 4, 1, 4 e 4", a saída do programa deve ser:

Número que mais aparece: 4

Quantidade de vezes que aparece: 3

**Obs.:** Se houver mais de um número que mais se repete, qualquer um deles pode ser impresso. Por exemplo, para a entrada “1, 2, 4, 1, 4, 4 e 1”, pode ser impresso o número “1” ou o número “4” já que ambos aparecem 3 vezes.

4. Escreva uma classe Contato que possui dois atributos: nome e email do tipo String. Crie outra classe, chamada Agenda, que possui um atributo contatos do tipo vetor de Contato. A classe Agenda deve conter um método para adicionar um novo contato em uma posição vazia do vetor, outro método para buscar um contato (retorna uma instância de Contato) através do nome e, por fim, um método para excluir um contato através do nome.

5. **Jogo da velha.** Seu sistema deve garantir os seguintes requisitos:

- Os jogadores não poderão jogar em posições que já foram “ocupadas em jogadas anteriores”.
- O mesmo jogador não pode jogar duas vezes seguidas.
- Se um dos jogadores ganhar a partida ela deve ser finalizada, mesmo que o tabuleiro ainda não se encontre completamente preenchido.
- Se ao “ocupar” todas as casas do tabuleiro, nenhum dos jogadores ganhar o jogo deve ser decretado como empate.
- A cada jogada o tabuleiro deve ser exibido para mostrar quais posições encontram-se ocupadas e por qual dos jogadores.

6. Implemente a classe Funcionario e a classe Gerente.

- crie a classe Assistente, que também é um funcionário, e que possui um número de matrícula (faça o método GET). Sobrescreva o método exibeDados().
- sabendo que os Assistentes Técnicos possuem um bônus salarial e que os Assistentes Administrativos possuem um turno (dia ou noite) e um adicional noturno, crie as classes Técnico e Administrativo.
- crie um assistente administrativo e um técnico. Imprima o número de matrícula e o nome de cada um deles.

7. Implemente os diagramas de classe abaixo:

- crie um animal do tipo cachorro e faça-o latir. Crie um gato e faça-o miar. Faça os dois animais caminharem.

