

PROGRAMAÇÃO ORIENTADA A OBJETOS I – SISTEMAS DE INFORMAÇÃO

Lista de exercícios 4 – Funções

Prof. Jean Eduardo Glazar

1. Escreva uma função que calcule a distância entre dois pontos, sabendo que a distância entre dois pontos é dada pela expressão:

$$\text{DISTANCIA} = ((X_2 - X_1)^2 + (Y_2 - Y_1)^2)^{1/2}$$

O primeiro ponto é (X_1, Y_1) e o segundo ponto é dado por (X_2, Y_2) .

Observação: Para essa questão poderão ser utilizadas as funções **pow** e/ou **sqrt** da Classe Math.

2. Sabendo-se que um triângulo é formado por três pontos:

$$A = (X_1, Y_1)$$

$$B = (X_2, Y_2)$$

$$C = (X_3, Y_3)$$

E que o perímetro de um triângulo é a soma das distâncias dos lados do triângulo. Podendo ser escrito da seguinte forma:

$$\text{Perímetro} = \text{DISTANCIA}(A,B) + \text{DISTANCIA}(A,C) + \text{DISTANCIA}(B,C)$$

Faça um algoritmo que calcule o perímetro de um triângulo.

Observação 1: Use a função distância da questão anterior.

Observação 2: Crie apenas uma função para a leitura dos pontos, com passagem de parâmetros por referência.

3. Faça uma função que dado dois valores (X, Y) retorne $X * Y$, entretanto não é permitido usar o operador de produto. $(*)$.

Sugestão: Use o operador de soma com um comando de repetição.

4. Faça uma função que dado dois valores (X, Y) retorne X^Y , entretanto não é permitido usar a função **pow**.

Observação 1: Use a função multiplicação da questão anterior com um comando de repetição.

Observação 2: Essa questão deverá ser implementada considerando Y como do tipo **int**. Se Y for float, a implementação de potência é mais complexa, pois precisaríamos trabalhar com potência de números fracionários.

5. Dado dois valores (X, Y) , faça uma função para o operador de raiz $X^{1/Y}$, entretanto não é permitido utilizar a função **sqrt**.

Observação: Use a função de potência da questão anterior.

6. Implemente um algoritmo de calculadora que faça as operações de:

- Adição
- Subtração
- Divisão
- Multiplicação (usando a função da questão 3)
- Potência (usando a função da questão 4)
- Raiz (usando a função da questão 5)

- 7. Jogo de Craps.** Faça um programa que implemente um jogo de Craps. O jogador lança um par de dados, obtendo a soma entre 2 e 12. Se na primeira jogada você tirar 7 ou 11, você ganhou. Se você tirar 2, 3 ou 12 na primeira jogada, isto é chamado de "craps" e você perdeu. Se na primeira jogada você somou 4, 5, 6, 8, 9 ou 10, este é seu "Ponto". Seu objetivo agora é continuar jogando os dados até tirar este número novamente. Você perde, no entanto, se tirar um 7 antes de tirar este "Ponto" novamente.

A função `lançarDado()` abaixo retorna um número aleatório entre 1 e 6 e simula o lançamento de um dado.

A função `jogarDados()` simula uma jogada com os dois dados e retorna a soma.

```
import java.io.IOException;
import java.util.Random;

public class Exercício07 {

    static Random aleatorio = new Random();

    public static int lançarDado() {
        return aleatorio.nextInt(6) + 1;
    }

    public static int jogarDados() throws IOException {
        int d1, d2, soma;
        System.out.println("Pressione <enter> para jogar os dados");
        System.in.read();

        d1 = lançarDado();
        d2 = lançarDado();
        soma = d1 + d2;

        System.out.println("Dado 1: " + d1);
        System.out.println("Dado 2: " + d2);
        System.out.println("SOMA: " + soma);
        System.out.println("-----");
        return soma;
    }

    public static void main(String[] args) throws IOException {

    }

}
```

- 8. DESAFIO: Cassino.** Altere o Jogo de Craps da questão anterior para que o jogador possa apostar. Ele começa com R\$ 100,00. No início de cada rodada ele informa quanto quer apostar. Se perder, perde esse valor. Se ganhar, ganha o dobro do que apostou. No final de cada rodada informe o total que ele tem em dinheiro e pergunte se quer continuar jogando ou se quer parar.