

Lógica de programação II 2023/1

4ª lista de exercícios

A lista é composta por questões teóricas dissertativas, questões de múltipla escolha e exercícios de programação. Nas questões de múltipla escolha, podem haver mais de um item correto dentro de uma mesma Questão, assim como pode não ter nenhum item correto ou todos itens corretos. As questões de programação devem ser implementadas em linguagem Java. A lista é individual e as respostas devem ser entregues através do moodle no prazo estipulado.

Questão 1: Qual a diferença entre os laços WHILE, DO WHILE e FOR?

WHILE executa repetidamente um bloco de código enquanto uma condição especificada for verdadeira;

DO WHILE é semelhante ao WHILE, mas a diferença é que a condição é verificada após a execução do bloco de código. Portanto, o bloco de código é executado pelo menos uma vez, mesmo que a condição seja inicialmente falsa.

FOR é uma estrutura de controle mais estruturada, especialmente útil quando se sabe o número exato de iterações.

Questão 2: Após a execução da linha de código a seguir, quais afirmativas são verdadeiras? Justifique. `int[] x = new int[25];`

a(X) O valor de `x[24]` é 0.

Ao criar um novo array de inteiros com o tamanho 25, todos os elementos são inicializados com o valor padrão para o tipo de dado, que no caso de inteiros é 0. Portanto, o valor de `x[24]` é 0.

b() O valor de `x[24]` é indefinido.

c() O valor de `x[25]` é 0.

d() O valor de `x[0]` é null.

e() O valor de `x[0]` é 1.

f(X) O valor de `x.length` é 25.

A afirmativa f é verdadeira. Ao criar o array com "`new int[25]`", seu tamanho é definido como 25. Portanto, o valor de `x.length` é 25, representando o número de elementos no array.

Questão 3: Implemente um programa que auxilie os profissionais de química a estimarem a quantidade de substância radioativa que permanece no decorrer do tempo. O usuário deve entrar a quantidade inicial da substância, a taxa de desintegração da substância e o tempo que se passou; e deve apresentar ao usuário a quantidade que ainda sobrou da substância. Utilize a seguinte expressão

$$Q = Q_0 \times e^{-rt}$$

,onde Q é a massa final, Q₀ é a massa inicial, r é a taxa e t é o tempo em anos.

```
import java.text.DecimalFormat;
```

```
import javax.swing.*;
```

```
public class JavaApplication5 {  
    public static void main(String[] args) {  
        // TODO code application logic here  
        DecimalFormat df = new DecimalFormat("0.0#");  
        double q0 = Double.parseDouble(JOptionPane.showInputDialog("Digite a quantidade inicial da substância: "));  
        double r = Double.parseDouble(JOptionPane.showInputDialog("Digite a taxa de desintegração da substância: "));  
        double t = Double.parseDouble(JOptionPane.showInputDialog("Digite o tempo que se passou em anos: "));  
        double q = q0 * Math. exp (-r*t);
```

```

        JOptionPane.showMessageDialog(null, "sobraram"+df.format(q)+ "gramas" );
    }
}

```

Questão 4: Escreva um programa que auxilie geógrafos a estimarem o tempo que uma determinada população leva para chegar a uma determinada quantidade de habitantes, dado um crescimento populacional. O programa deve receber do usuário a população inicial da cidade (P_0), a taxa de crescimento populacional (t) e o tamanho da população que se espera chegar (P_x). E então deve calcular quantos anos ele levará para chegar a este tamanho da população (x). Para realizar esta tarefa, é sugerido usar a fórmula:

$$x = \frac{\log(\frac{P_x}{P_0})}{\log(1+t)}$$

```

import javax.swing.JOptionPane;
public class LancamentoProjeto {
    public static void main(String[] args) {
        double velocidade = Double.parseDouble(JOptionPane.showInputDialog("Digite a velocidade de lançamento (m/s): "));
        double angulo = Double.parseDouble(JOptionPane.showInputDialog("Digite o ângulo de lançamento (graus): "));
        double distanciaMaxima = calcularDistanciaMaxima(velocidade, angulo);
        double alturaMaxima = calcularAlturaMaxima(velocidade, angulo);
        JOptionPane.showMessageDialog(null, "Distância máxima alcançada: " + distanciaMaxima + " metros" +
            "\nAltura máxima atingida: " + alturaMaxima + " metros");
    }
    // Função para calcular a distância máxima alcançada
    public static double calcularDistanciaMaxima(double velocidade, double angulo) {
        double radianos = Math.toRadians(angulo);
        double distancia = (velocidade * velocidade * Math.sin(2 * radianos)) / 10;
        return distancia;
    }
    // Função para calcular a altura máxima atingida
    public static double calcularAlturaMaxima(double velocidade, double angulo) {
        double radianos = Math.toRadians(angulo);
        double altura = (velocidade * velocidade * Math.pow(Math.sin(radianos), 2)) / (2 * 10);
        return altura;
    }
}

```

Questão 5: Implemente um programa para o cálculo de lançamento de projéteis. O usuário deve inserir a velocidade de lançamento do projétil e o ângulo de lançamento, e o programa deve apresentar a altura máxima atingida e a distância máxima alcançada. Considere a aceleração da gravidade como sendo $10m/s^2$. Onde D é a distância máxima alcançada, H é a altura máxima atingida, V_0 é a velocidade de lançamento, θ é o ângulo de lançamento e g é a aceleração da gravidade. Use as seguintes fórmulas:

$$D = \frac{V_0^2 \times \sin(2\theta)}{g}$$

$$H = \frac{V_0^2 \times \sin^2(\theta)}{2g}$$

```

public static void main(String[] args) {
    double v0 = Double.parseDouble(JOptionPane.showInputDialog(
        "Digite a velocidade de lançamento."));
    double a = Double.parseDouble(JOptionPane.showInputDialog(
        "Digite o ângulo de lançamento em graus."));
    double t = Math.toRadians(a);
    double g = 10;
}

```

```

        double d = v0*v0*Math.sin(2*t)/g;
        double h = Math.pow(v0,2)*Math.pow(Math.sin(t),2)/(2*g);
        JOptionPane.showMessageDialog(null, "A distancia maxima de :"+d+
            "\nA altura maxima de :"+h);
    }
}

```

Questão 6: Escreva um programa que gere uma senha aleatória composta por 6 dígitos.

```

import java.util.Random;

public class qst6 {
    public static void main(String[] args) {
        Random geraraleatorio = new Random (5);
        for (int i = 0; i < 6; i++) {
            int aleatorio = geraraleatorio.nextInt(10); // Gera um número aleatório entre 0 e 9
            System.out.print(aleatorio);
        }
    }
}

```

Questão 7: Escreva um programa que receba 10 números e armazene-os em um array. Depois calcule qual o maior e qual o menor destes valores.

```

public static void main(String[] args) {
    int[] numeros = new int[10];
    // 10 n p/ usuario
    for (int i = 0; i < 10; i++) {
        numeros[i] = Integer.parseInt(JOptionPane.showInputDialog("Digite o " + (1 + i) + " numero: "));
    }
    // maior e o menor valor
    int maior = numeros[0];
    int menor = numeros[0];

    for (int i = 1; i < 10; i++) {
        if (numeros[i] > maior)
            maior = numeros[i];
        if (numeros[i] < menor)
            menor = numeros[i];
    }
    JOptionPane.showMessageDialog(null, "O maior numero e: " + maior + "\nO menor numero e: " +
    menor);
}
}
}

```

Questão 8: Escreva um programa que calcule o fatorial de um número.

```

import javax.swing.JOptionPane;

public class qst8 {
    public static void main(String[] args) {
        int numero = Integer.parseInt(JOptionPane.showInputDialog("Digite um número inteiro: "));

        long fatorial = 1;

        for (int i = 1; i <= numero; i++) {
            fatorial *= i;
        }

        JOptionPane.showMessageDialog(null, "O fatorial de " + numero + " é: " + fatorial);
    }
}

```

Questão 9: Escreva um programa que receba do usuario um número n e outro p e calcule o

arranjo n, p a p .

$$A_p^n = \frac{n!}{(n-p)!}$$

Questão 10: Escreva um programa que receba do usuário um número n e outro p e calcule combinação n, p a p .

$$C_p^n = \frac{n!}{p! \cdot (n-p)!}$$