

Aula XII - Desenvolvimento de Aplicativo I - Estudo Dirigido

Valor 6,0 - N1

- O que for de parte teórica é para salvar em um arquivo com o nome: teoria + seu nome completo.
- Códigos: criar um projeto com nome de EstudoDirigido, publicar no github e postar o link na atividade juntamente com o arquivo teórico.
- O trabalho de Estudo Dirigido é individual, você pode usar a IDE que desejar desde que dê suporte para a plataforma Java.

Leia com atenção cada tópico e execute o que foi solicitado.

1. Analise os códigos abaixo e elabore um diagrama de classe para este projeto. Para relembrar o que é um diagrama de classe e como fazer, você pode consultar [O que é um diagrama de classe UML?](#) ou ainda [Entendendo o Diagrama de Classes da UML](#) <https://plleon.wordpress.com/2009/02/14/uml-diagrama-de-classes/>

```
public class Aluno {
    //Atributos - dados de entrada:
    public String nome;
    public double nota1;
    public double nota2;
    //Métodos - cálculos:
    public double calcularMedia() {
        return (this.nota1 + this.nota2)/2;
    }
}

import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner ler = new Scanner(System.in);
        //Instancia a classe - criar um objeto:
        Aluno a1 = new Aluno();
        //Pede e lê:
        System.out.println("Digite o nome do aluno: ");
        a1.nome = ler.nextLine();
        System.out.println("Digite a nota 1: ");
        a1.nota1 = ler.nextDouble();
        System.out.println("Digite a nota 2: ");
        a1.nota2 = ler.nextDouble();
        //Respostas:
        System.out.println("\nRESPOSTAS:");
        System.out.println("Aluno: "+a1.nome);
        System.out.println("Nota 1: "+a1.nota1);
        System.out.println("Nota 2: "+a1.nota2);
        System.out.println("Média: "+a1.calcularMedia());
    }
}
```

2. Analisando o seguinte diagrama de classes, qual o código fonte representa da linguagem Java sabendo que o cálculo de desconto do produto é de 2,5% sobre o valor do mesmo. **Crie a classe especificada no diagrama;**

Produto
- nome : String - valor : double
+ calcularDesconto() : double

3. Analise o seguinte código desenvolvido na linguagem Java:

```

1 public class Produto{
2     private String nome;
3     private double valor;
4     public Produto (String nome, double valor){
5         this.nome = nome;
6         this.valor = valor;
7     }
8     public String getNome(){
9         return nome;
10    }
11    public void setNome(String nome){
12        this.nome = nome;
13    }
14    public double getValor(){
15        return valor;
16    }
17    public void setValor(double valor){
18        this.valor = valor;
19    }
20    public double calcularPrecoVenda(){
21        if (valor > 1000){
22            return valor * 0.97;
23        }else{
24            return valor;
25        }
26    }
27 }

```

Agora, analise as seguintes afirmações:

- I. O método criado na linha 4 é considerado o método construtor da classe.
- II. O método criado na linha 17 permite a leitura do atributo valor da classe Produto
- III. Caso o produto possua um valor superior a 1000, será apresentado o preço de venda com 9% de desconto.
- IV. Os métodos criados nas linhas 8, 11, 14 e 17 encapsulam os atributos da classe Produto.

São corretas as afirmativas:

- a. () I e II
- b. () I, III e IV
- c. () II e III
- d. () I, II e III
- e. () I, II, III e IV

4. Observe o código abaixo (na utilização do laço while). **O laço de repetição while caracteriza-se por ter seu teste de execução antes de iniciar o loop. Neste tipo de laço nem sempre temos a execução dos comandos, ou seja, nem sempre ele entra no loop, traduzindo, o While é utilizado para construir uma estrutura de repetição que executa, repetidamente, uma única instrução ou um bloco delas “enquanto” uma expressão booleana for verdadeira. Descreva** qual parte do código está definindo enquanto a condição for verdadeira e qual a parte do código é responsável por executar as instruções:

```

while (vsal[(cont-1)]!= -1 ){
    System.out.println ("Entre com o valor de salarios:" +1);
    vsal[cont] = new Scanner(System.in).nextDouble();
    cont= cont+1;
}

```

5. Observe o código abaixo (na utilização do laço for). **O for pode conter apenas uma instrução no seu corpo. Neste caso não é necessário abrir um bloco. Isso é assim porque o “for” já implementa alguns comandos na sua assinatura, ou seja, no seu cabeçalho, como a inicialização da variável e o passo da repetição, ou seja, o incremento/decremento da variável (executado sempre no fim de cada ciclo). O laço for é uma estrutura de repetição compacta. Seus elementos de inicialização, condição e iteração são reunidos na forma de um cabeçalho e o corpo é disposto em seguida. O laço for e o laço while são apenas formas diferentes de uma mesma estrutura básica de repetição.**

Descreva qual o comando inicial, qual a condição e qual o comando de loop. E para aproveitar, explique as condições que estão estabelecidas no código.

```

for (int i = 0; i < quantos; i++){
    System.out.println("Insira o numero da posição do vetor: " + i);
    numeros[i] = new Scanner(System.in).nextInt();
    if (sequencia.length() > 0) sequencia = sequencia + ", ";
    sequencia = sequencia + numeros[i];
}

```

6. Observe o código abaixo e responda as questões:

```

public class Treino {
    private String nome;
    private double[] marcas;
    public Treino() {
        this.marcas = new double[10];
    }
    public String getNome() {
        return this.nome;
    }
    public double[] getMarcas() {
        return this.marcas;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
    public void setMarcas(double[] marcas) {
        this.marcas = marcas;
    }
    public void armazenarMarca(int numero, double marca) {
        if(numero >= 0 && numero < this.marcas.length) {
            this.marcas[numero] = marca;
        }
    }
}

```

- Quais são os atributos do código?
- Qual o tipo de visibilidade destes atributos?
- Que tipo de dado o método construtor está retornando?
- Quais os métodos acessores do código?
- Que tipo de método encontramos no código que não faz parte dos acessores?

7. Analise as imagem abaixo e descreva o significado de suas sintaxes e a função destes métodos:

<pre> public int getValor() { return valor; } </pre>	
<pre> public void setValor(int valor) { this.valor = valor; } </pre>	

<pre>public Numero (int valor){ this.valor= valor; }</pre>	
<pre>public String getNome() { return nome; }</pre>	

8. Comportamento é uma ação executada por um objeto quando passada uma mensagem ou em resposta a uma mudança de estado: é algo que o objeto faz. Um objeto pode disparar o comportamento de outro, executando uma operação sobre esse objeto. Você pode ver os termos chamada de método, chamada de função ou passar uma mensagem usados em vez de “executar uma operação”. (SINTES, A. Aprenda Programação Orientada a Objetos em 21 Dias. São Paulo: Pearson Education do Brasil, 2002.). Considere a seguinte declaração de um comportamento/método para a classe **Questao3**:

```
public class Questao3 {
    public double calcular(int n, int x, double valor) {
        if(n = 0) {
            return (valor + valor * x);
        } else {
            return (valor - n * x);
        }
    }
}
```

Em algum ponto do programa as seguintes mensagens são passadas:

```
Questao3 q1 = new Questao3();
Questao3 q2 = new Questao3();

System.out.print(q1.calcular(2,5,100));
System.out.print(q2.calcular(0,5,100));
```

Que valores serão impressos em tela para o usuário?

- a. () 1000 e 100
- b. () 90 e 600
- c. () 450 e 150
- d. () 600 e 250
- e. () 1000 e 500

9. Em Java podemos criar variáveis de tipos primitivos e de tipos de referência. As variáveis de referência a objetos não contêm os objetos propriamente. Na verdade elas indicam a posição (endereço) onde o objeto encontra-se na memória, e o Java procura nestes endereços os objetos. Considere o seguinte programa, que declara e usa as variáveis de referência locais “doc1” e “doc2”:

```
public class Programa {  
    public static void main(String[] args) {  
        Documento doc1 = new Documento();  
        doc1.nome = "Alfredo";  
        Documento doc2 = doc1;  
        doc2.nome = "Juliana";  
        System.out.println("doc1: " + doc1.nome);  
        System.out.println("doc2: " + doc2.nome);  
    }  
}
```

O que será impresso para o usuário em doc1 e doc2?

10. As estruturas de repetição também são conhecidas como laços (loops) e são utilizados para executar, repetidamente, uma instrução ou bloco de instrução enquanto determinada condição estiver sendo satisfeita.

Qualquer que seja a estrutura de repetição, ela contém quatro elementos fundamentais: inicialização, condição, corpo e iteração. A inicialização compõe-se de todo código que determina a condição inicial da repetição. A condição é uma expressão booleana avaliada após cada leitura do corpo e determina se uma nova leitura deve ser feita ou se a estrutura de repetição deve ser encerrada. O corpo compõe-se de todas as instruções que são executadas repetidamente. A iteração é a instrução que deve ser executada depois do corpo e antes de uma nova repetição.

A respeito dos laços de repetição utilizados na linguagem Java, assinale a alternativa correta.

- a. () O comando associado ao laço do-while é executado no mínimo uma vez.
- b. () O comando associado ao laço while é executado no mínimo uma vez.
- c. () O comando associado ao laço do-while é executado no mínimo uma vez desde que a condição do laço seja verdadeira.
- d. () O comando associado ao laço do-while é executado no mínimo uma vez desde que a condição do laço seja falsa.

11. Criar um sistema em Java 0.0 **partindo da base da UML abaixo:**

