



INSTITUTO FEDERAL DA PARAÍBA
CAMPUS CAMPINA GRANDE
BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO
DISCIPLINA DE POO e LAB. POO
PROF. VICTOR ANDRÉ PINHO DE OLIVEIRA

Atividade Unidade I - 4 - Introdução à OO

Instruções

Responda às questões abaixo. Pode usar este próprio documento.

Questões teóricas valem 0,5 ponto. Questões práticas valem 2 pontos.

Questões

1. Explique a diferença entre classes e objetos.

Classes são os tipos de dados criados pelo programador, as classes irão conter atributos e métodos. A partir das classes os objetos são instanciados. O objeto é algo concreto, onde os elementos estarão presentes ali.

2. Qual a diferença entre uma variável local e um membro de dados (atributo)?

Atributos estarão contidos nas classes, são as características, podemos estendê-las para outros objetos através da herança. Variáveis locais estarão apenas dentro daquele objeto.

3. Considere o objeto Hamster, um animal fofinho que parece um rato. Descreva 3 atributos e 3 comportamentos.

Atributos: Cor do pelo, tamanho, peso.

Comportamento: Roer, comer, escalar.

4. Considere o objeto mouse (periférico). Descreva 3 atributos e 2 comportamentos.

Atributos: Tamanho, DPI, marca.

Comportamento: Clicar, scrollar.

5. Qual a unidade básica de programação de C++?

As classes.

6. O que é instanciar?

A partir da classe podemos instanciar um objeto, ou seja, cria-lo, inserindo seus atributos e métodos.

7. De que maneira uma Classe oferece serviços para outrem?

Através da herança podemos reutilizar métodos em outras classes

8. O que é encapsulamento?

É ocultar dados que estão contidos nas classes, para que o usuário não consiga alterar os atributos de forma direta.

9. O que é UML?

É uma linguagem utilizada em processos de software para modelar, através de diagramas, sistemas Orientados a Objetos

10. Como que um objeto envia uma mensagem para outro objeto?

Por meio de interface.

11. Para que serve o especificador de acesso public? E o private?

Public: Podemos reutilizar os métodos ou atributos ao longo do código.

Private: Fará com que aquele método ou atributo esteja apenas dentro da classe.

12. O que são métodos set? E métodos get?

Set: Irá setar o valor do dado ou modificar.

Get: Irá obter o dado em si.

13. O que é um construtor-padrão?

Quando não criamos um método construtor, o próprio compilador cria um método construtor padrão.

14. Quando um construtor é invocado?

Na função main, onde criamos o objeto.

15. Por que é importante separar a Interface da Implementação? Como podemos fazer isso em C++?

Além de deixar o código mais organizado, o código ficará mais seguro.

16. Modifique a última versão da **Classe Carro** desenvolvida na “sala” (presente no slide) de modo que:
- Inclua um segundo membro de dados **int** que representa o ano do carro;
 - Forneça uma função **set** para alterar o ano e uma função **get** para recuperá-lo;
 - Modifique o construtor para aceitar dois parâmetros: um para a marca e outro para o ano;
 - Modifique o método **displayMessage** para que ele apresente na tela a marca e o ano do carro.
17. Crie uma Classe chamada **ContaBanco** que um banco poderia utilizar para representar contas bancárias dos clientes. Sua classe deve incluir um membro de dados do tipo **double** para representar o **saldo da conta**. Sua Classe deve fornecer um **construtor** que recebe um saldo inicial e o utiliza para inicializar o membro de dados. O construtor deve validar o saldo inicial para assegurar que ele seja maior ou igual a zero. Caso seja menor que zero, o construtor simplesmente deverá setar o saldo para zero. A classe deve fornecer três **funções-membro**. A função-membro *creditar* deve adicionar uma quantia passada como argumento ao saldo atual. A função-membro *debitar* deve retirar a quantia passada como argumento da conta. Se houver tentativa de retirar um valor acima do saldo, então o saldo deverá permanecer inalterado e a função-membro deverá exibir uma mensagem de erro na tela. A função-membro *getSaldo* deve retornar o saldo atual. Escreva um programa que cria dois objetos **ContaBanco** e testa suas funções-membro.
18. Crie uma Classe chamada **Empregado** que possui 3 **membros de dados** - um **nome**, um **sobrenome** e um **salário mensal**. Sua Classe deve ter um **construtor** que inicialize os 3 membros de dados. Forneça uma função **set** e uma função **get** para cada membro de dados. Se o salário mensal não for positivo, configure-o como 0. Escreva um programa de teste que demonstre as capacidades da classe **Empregado**. Crie dois objetos **Empregado** e exiba seu salário mensal. Em seguida, dê um aumento de 10% para cada um dos empregados e exiba novamente o salário mensal.
19. Crie uma Classe **Data** que inclua 3 partes de informações como atributos - dia (**int**), mes (**int**) e ano (**int**). Sua Classe deve ter um construtor com três parâmetros que utilize os parâmetros para inicializar os 3 membros de dados. Para o propósito deste exercício, assumo que os valores fornecidos para o dia e ano estão corretos, mas certifique-se de que o valor do mês esteja no intervalo 1-12; se não estiver, configure o mês como 1. Forneça uma função **set** e uma função **get** para cada membro de dados. Forneça também uma função-membro **mostrarData** que exiba o dia, o mês e o ano separados por barras (/). Escreva um programa que demonstre as capacidades da sua Classe **Date**.
20. Crie uma Classe chamada **Cupom** que uma loja de suprimentos de informática possa utilizar para representar um cupom de um item vendido na loja. Um **Cupom** deve incluir quatro membros de dados - **id** (**string**), **descrição** (**string**), **quantidade**

(int) e o preço do item (float). Sua Classe deve ter um construtor que inicializa os 4 membros de dados. Forneça uma função set e uma função get para cada membro de dados. Além disso, forneça uma função-membro chamada de calculaCupom que calcula o valor total da nota e depois retorna esse valor. Se a quantidade não for positiva, ela deve ser configurada como 0. Se o preço do item não for positivo, ele deve ser configurado com 0. Escreva um programa para ilustrar as capacidades de sua Classe Cupom.