**Universidade Vila Velha**

**Atividade em Sala de Aula Máximo de 3 componentes por grupo:**

**As respostas devem ter justificativas.**

1. Cite 3 motivos por que utilizamos modelagem.

- Para auxiliar no desenvolvimento do código posteriormente.

- Para ajudar na organização da estrutura do sistema como um todo.

-

1. Desenhe o elemento de modelagem de acordo com seu nome:
   * Classe
   * Associação binária
   * Generalização / herança
   * Dependência
   * Agregação
   * Composição
   * Comentário
   * Multiplicidade
2. Para uma linguagem de programação ser considerada no Paradigma Orientada a Objetos, são necessários 4 pilares importantes. Quais são eles?
   1. Estrutura de dados, Interface, Pacotes, Polimorfismo.
   2. Mensagem, Interface, Pacotes, Atributo.
   3. Polimorfismo, Instância, Objeto, Classe.
   4. Interface, Classe, Objeto, Atributo.
   5. Abstração, Encapsulamento, Herança,

Polimorfismo.

1. Diante dos paradigmas principais de desenvolvimento de software no mercado, tem-se o paradigma estruturado e o da orientação a objetos como opções.

Um objeto, dentro do contexto do paradigma OO, é

* 1. um método.
  2. uma instância de uma classe.
  3. uma variável.
  4. array.
  5. função.

1. Em programas orientados a objetos, é necessário que a associação de uma solicitação a um objeto e a uma de suas operações seja realizada somente em tempo de execução. Esse fenômeno, conhecido como ligação dinâmica, é importante porque permite a implementação do seguinte conceito de:
   1. acoplamento
   2. polimorfismo
   3. composição
   4. encapsulamento
   5. abstração
2. Qual é a técnica que faz com que os detalhes internos a respeito do funcionamento dos métodos de uma classe permaneçam ocultos para outros objetos?

- encapsulamento

1. Em POO, dizer que a classe A é superclasse de B é o mesmo que dizer que:
   1. A é derivada de B;
   2. A estende B;
   3. B é derivada de A;
   4. B implementa A;
   5. A implementa B.
2. A herança é uma característica poderosa e importantes da orientação a objetos, pois permite o reaproveitamento de atributos e métodos. Em

aplicações que utilizam herança,

* 1. é possível a implementação de herança múltipla, em todas as linguagens orientadas a objetos.
  2. não é possível sobrescrever em uma subclasse, um método de sua superclasse.
  3. cada superclasse pode ter apenas uma subclasse.
  4. somente as superclasses poderão ter métodos ou construtores sobrecarregados.
  5. é possível se obter polimorfismo.

1. Dê exemplos de 5 classes em um sistema de academia de ginástica. Para cada classe pense em dois atributos e duas operações.
2. Identifique as classes, atributos e métodos necessários para modelar os seguintes requisitos. Utilize o draw.io [(https://app.diagrams.net/)](https://app.diagrams.net/) para montar o diagrama.
   * Uma conta corrente que possui número, saldo, status que informa se é ou não especial, um limite e um conjunto de movimentações;
   * Uma movimentação que possui uma descrição, um valor e uma informação se ela é uma movimentação de crédito ou débito;
   * Um banco que armazene um conjunto de contas e forneça métodos que permitam que sejam feitos criações de conta, exclusão de contas, saque (uma conta corrente só pode fazer saques desde que o valor não exceda o limite de saque-limite + saldo negativo), depósitos, emissão de saldo e extrato e transferência entre contas;

**Vale lembrar:**

Passos para criar um Diagrama de Classes:

1. Identificar os nomes das classes

a. Identifique os objetos do sistema

2. Definir relações

a. Determine como as classes estão relacionadas entre si

3. Criar a estrutura

* + 1. Dê nomes às classes
    2. Associe as classes com os conectores apropriados
    3. Inclua atributos e operações