

ATIVIDADE 3 - ESOFT - ANÁLISE E PROJETO ORIENTADO A OBJETOS - 51_2025**Período:** 24/03/2025 08:00 a 27/04/2025 23:59 (Horário de Brasília)**Status:** ABERTO**Nota máxima:** 0,50**Gabarito:** Gabarito será liberado no dia 28/04/2025 00:00 (Horário de Brasília)**Nota obtida:****1ª QUESTÃO**

O diagrama de casos de uso é utilizado na análise de requisitos. Este diagrama acompanha o software desde o seu início até a conclusão (Rando; Freitas, 2020).

Fonte: RANDO, D. R.; FREITAS, J. A. de. **Análise e Projeto Orientado a Objetos**. Maringá: UniCesumar, 2020.

Sobre diagrama de caso de uso, considere:

I. O diagrama de caso de uso é um diagrama comportamental.

PORQUE

II. modela o comportamento dos atores no sistema.

A respeito dessas asserções, assinale a opção correta:

ALTERNATIVAS

- As asserções I e II são proposições verdadeiras, e a II é uma justificativa correta da I.
- As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa correta da I.
- A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.
- A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.
- As asserções I e II são proposições falsas.

2ª QUESTÃO

O diagrama de sequência mostra as trocas de mensagens entre os objetos, considerando o tempo (Rando; Freitas, 2020).

Fonte: RANDO, D. R.; FREITAS, J. A. de. **Análise e Projeto Orientado a Objetos**. Maringá: UniCesumar, 2020.

Considerando a situação apresentada sobre o que podemos afirmar no processo de construção de modelos, assinale a afirmativa correta:

ALTERNATIVAS

- Alterações no diagrama de classes são comuns após a elaboração do diagrama de sequência, pois o diagrama de classe só é construído no final.
 - Alterações no diagrama de classes não são comuns após a elaboração do diagrama de sequência, pois o diagrama de sequência é construído primeiro.
 - Alterações no diagrama de classes são comuns após a elaboração do diagrama de sequência, pois o diagrama de classe não é construído no final.
 - Alterações no diagrama de classes não são comuns após a elaboração do diagrama de sequência, pois os diagramas de classes são elaborados antes do diagrama de sequência.
 - Alterações no diagrama de classes não são comuns após a elaboração do diagrama de sequência, pois os diagramas de classes são reformados antes do diagrama de sequência.
-

3ª QUESTÃO

Um diagrama de classes é uma representação visual da estrutura e das interações entre classes em um sistema, detalhando atributos, métodos e os tipos de relacionamentos entre elas.

Fonte: Elaborado pelo professor.

A seguir, é apresentado um exemplo de diagrama de classes para um sistema de controle bancário.

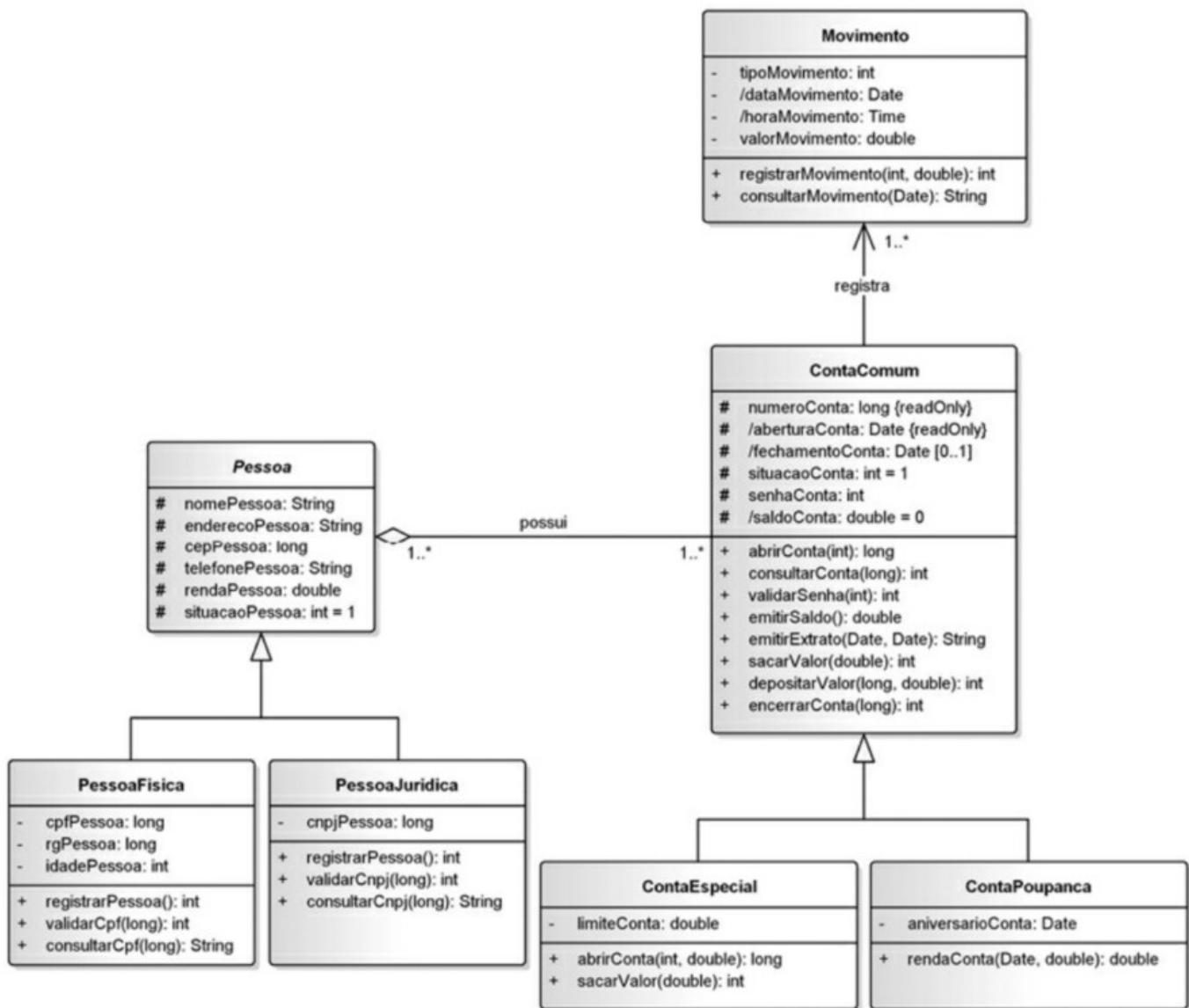


Figura 1 - Diagrama de Classes

Fonte: GUEDES, G. T. A. **UML 2 - Uma abordagem prática**. Novatec Editora, 2018.

Conforme o diagrama apresentado, analise as seguintes afirmações:

- I. O relacionamento entre as classes Pessoa e ContaComum é de composição, indicando que as instâncias de Pessoa não podem existir independentemente de ContaComum.
- II. A classe ContaComum possui um relacionamento de generalização com as classes ContaEspecial e ContaPoupança, permitindo herança de atributos e métodos.
- III. A classe Movimento possui um relacionamento de agregação com a classe ContaComum, o que significa que Movimento não pode existir independente da classe ContaComum.
- IV. A visibilidade dos atributos na classe ContaComum é marcada como privada (private), isso significa que somente a classe ContaComum poderão acessá-los.
- V. Os atributos da classe Movimento são privados, enquanto seus métodos são públicos. Visibilidade pública indica que qualquer objeto pode ter acesso a estes métodos, e privada implica que somente os objetos da classe podem ter acesso a estes atributos.

É correto o que se afirma em:

ALTERNATIVAS

- I e III, apenas.
- II, apenas.
- II e V, apenas.
- III e V, apenas.
- IV e V, apenas.

4ª QUESTÃO

Na década de 1980, houve preponderância dos métodos estruturados para o desenvolvimento de software. Atualmente, o paradigma OO é mais forte e, por isso, é importante diferenciar entre análise e projeto estruturado e orientado a objetos (Rando; Freitas, 2020).

Fonte: RANDO, D. R.; FREITAS, J. A. de. **Análise e Projeto Orientado a Objetos**. Maringá: UniCesumar, 2020.

Sobre o paradigma de análise e projeto orientado a objetos, analise as seguintes afirmações:

- I. Visualizamos classes responsáveis por atributos;
- II. Possui uma visão sequencial e bem dividida;
- III. A execução das atividades dos sistemas passa a depender da interação dessas classes;
- IV. O agrupamento é feito por funcionalidade.

É correto o que se afirma em:

ALTERNATIVAS

- I e II, apenas.
- I e III, apenas.
- I, II, III e IV.
- I, III e IV, apenas.
- II e III, apenas.

5ª QUESTÃO

O diagrama de objetos na UML é usado para representar uma visão estática de um sistema em um momento específico, mostrando instâncias de classes e os valores de seus atributos. Ele é frequentemente usado para complementar o diagrama de classes, oferecendo uma representação prática de como os objetos interagem e se relacionam em uma situação específica.

Fonte: Elaborado pelo professor, 2025.

Com base no texto e em relação ao diagrama de objetos na UML, analise as seguintes afirmativas:

- I. Um diagrama de objetos representa instâncias específicas de classes e os valores de seus atributos em um momento particular da execução do sistema.
- II. Os objetos em um diagrama de objetos são representados por retângulos com o nome da classe sublinhado, indicando ser instâncias.
- III. O diagrama de objetos é usado principalmente para representar a hierarquia de classes em um sistema.
- IV. As associações entre objetos no diagrama de objetos representam relações dinâmicas, que podem mudar durante a execução do sistema.
- V. Um diagrama de objetos é útil para validar o design representado no diagrama de classes, ajudando a verificar cenários específicos do sistema.

É correto o que se afirma em:

ALTERNATIVAS

- I, II e V, apenas.
- I, III e IV, apenas.
- I, IV e V, apenas.
- II, IV e V, apenas.
- III, IV e V, apenas.

6ª QUESTÃO

O diagrama de classes é um dos artefatos mais importantes da UML, ao representar a estrutura estática do sistema, exibindo as classes e os relacionamentos entre elas. Esses relacionamentos incluem associações, generalizações, composições e agregações, cada um com uma semântica e propósito específicos para representar diferentes tipos de conexões entre os elementos do sistema.

Fonte: Elaborado pelo professor.

Com base no texto sobre o Diagrama de Classes, analise as seguintes afirmativas:

- I. O Diagrama de Classes representa a estrutura estática de um sistema, mostrando as classes, seus atributos e operações, bem como os relacionamentos entre essas classes.
- II. Os atributos em um Diagrama de Classes representam as características ou propriedades de uma classe, enquanto as operações representam os comportamentos ou ações que os objetos dessa classe podem executar.
- III. Os relacionamentos em um Diagrama de Classes descrevem como as classes interagem entre si, incluindo associações, herança, agregação e composição.
- IV. O Diagrama de Classes é uma ferramenta exclusiva para a fase de implementação do software, sendo dispensável nas fases de análise e projeto.
- V. O Diagrama de Classes é utilizado principalmente para modelar o comportamento dinâmico do sistema, descrevendo a sequência de mensagens trocadas entre os objetos durante a execução.

É correto o que se afirma em:

ALTERNATIVAS

- I, II e III, apenas.
 - I, II e IV, apenas.
 - I, III e V, apenas.
 - II, III e IV, apenas.
 - III, IV e V, apenas.
-

7ª QUESTÃO

A programação orientada a objetos é fundamentada em quatro pilares essenciais: abstração, encapsulamento, herança e polimorfismo. Cada um desses pilares desempenha um papel importante na criação de sistemas organizados, reutilizáveis e flexíveis. Esses princípios definem como os objetos se relacionam, interagem e compartilham funcionalidades no desenvolvimento de software orientado a objetos.

Fonte: Elaborado pelo professor.

Com relação aos quatro pilares da orientação a objetos, analise as seguintes sentenças:

- I. A abstração permite simplificar a complexidade do sistema, destacando apenas os aspectos essenciais de um objeto, sem expor detalhes internos irrelevantes.
- II. O encapsulamento permite que os atributos e métodos de uma classe sejam acessados diretamente por outros objetos, promovendo a flexibilidade do sistema.
- III. A herança possibilita que uma classe reutilize atributos e métodos de outra classe, facilitando a criação de hierarquias e promovendo o reaproveitamento de código.
- IV. O polimorfismo permite que métodos com o mesmo nome sejam utilizados de formas diferentes, dependendo do objeto que os invoca, promovendo flexibilidade no comportamento do sistema.
- V. A composição é um dos quatro pilares, permitindo que objetos sejam formados pela combinação de outros objetos para alcançar uma funcionalidade específica.

É correto o que se afirma em:

ALTERNATIVAS

- I, II e III, apenas.
 - I, IV e V, apenas.
 - I, III e IV, apenas.
 - II, III e V, apenas.
 - III, IV e V, apenas.
-

8ª QUESTÃO

O diagrama de classes é um dos mais importantes da UML, ao representar a estrutura estática do sistema, exibindo as classes e os relacionamentos entre elas. Esses relacionamentos incluem associações, generalizações, composições e agregações, cada um com uma semântica e propósito específicos para representar diferentes tipos de conexões entre os elementos do sistema.

Fonte: Elaborado pelo professor, 2025.

Com relação aos relacionamentos no diagrama de classes, analise as seguintes sentenças:

- I. A associação é um dos relacionamentos mais comuns no diagrama de classes, representando um vínculo estrutural entre classes. A direcionalidade indica a naveabilidade entre as classes, e a multiplicidade define quantas instâncias de uma classe estão conectadas a outra.
- II. Na agregação, as partes não podem existir independentemente do todo. O ciclo de vida das partes é controlado pelo todo, diferentemente da composição.
- III. A composição é uma forma forte de agregação, donde as partes não podem existir independentemente do todo, identificando que os objetos-parte precisam estar associados a um objeto-todo.
- IV. A generalização define uma relação hierárquica entre classes, permitindo que uma classe filha herde somente os atributos de uma classe pai, devendo implementar os seus próprios métodos.
- V. A dependência é um relacionamento transitório, indicado por uma linha tracejada, que demonstra que uma classe utiliza outra para realizar uma operação.

É correto o que se afirma em:

ALTERNATIVAS

- I, III e V, apenas.
- I, III e IV, apenas.
- I, II e V, apenas.
- II, IV e V, apenas.
- II, III e IV, apenas.

9ª QUESTÃO

A coleção de casos de uso representa, do ponto de vista do usuário, todos os modos de execução do sistema (Rando; Freitas, 2020).

Fonte: RANDO, D. R.; FREITAS, J. A. de. **Análise e Projeto Orientado a Objetos**. Maringá: UniCesumar, 2020.

Considerando a situação apresentada sobre o caso de uso, podemos afirmar que:

ALTERNATIVAS

- Um caso de uso é uma sequência de ações que produz algum resultado para um ator.
- Um caso de uso é uma sequência de ações que produz qualquer resultado para um ator.
- Um caso de uso é uma sequência de ações que produz um erro significativo para um ator.
- Um caso de uso é uma sequência de ações que produz um resultado significativo para um ator.
- Um caso de uso é uma sequência de ações que produz nenhum resultado significativo para um ator.

10ª QUESTÃO

Diagramas de implantação são utilizados para modelar a arquitetura física de distribuição em que o *software* será executado. Esse diagrama mostra os nós e os relacionamentos de comunicação.

Fonte: RANDO, D. R.; FREITAS, J. A. de. **Análise e Projeto Orientado a Objetos**. Maringá - PR.: UniCesumar, 2020.

Pensando sobre o texto, o que é uma associação de conectores em um Diagrama de Implantação?

ALTERNATIVAS

- Um estereótipo aplicado a um nó.
 - Uma ligação de dados entre classes.
 - Uma relação de herança entre dispositivos.
 - Uma conexão física entre nós de dispositivo.
 - Uma relação de dependência entre componentes de software.
-