**Programação Orientada por Objetos  
Professor Marcelo Eustáquio**

**Questão 01**

Descreva, no espaço a seguir, o problema que seu sistema se propõe a resolver.

|  |
| --- |
| O sistema propõe resolver o problema de gerenciamento de tarefas, permitindo que os usuários organizem suas atividades diárias, definindo títulos, descrições, datas de vencimento e status de conclusão para cada tarefa. Isso ajuda a aumentar a produtividade e a manter as atividades em dia. |

**Questão 02**

Apresente o conceito de CLASSE e mostre exemplos de classes definidas no seu projeto.

|  |
| --- |
| Classe é uma estrutura de programação que define um tipo de objeto, especificando seus atributos (variáveis) e comportamentos (métodos). No projeto de gerenciamento de tarefas, exemplos de classes seriam:  Tarefa: Essa classe representa uma tarefa e possui os atributos como título, descrição, data de vencimento e status de conclusão. Além disso, pode ter métodos para definir e obter os valores desses atributos.  ListaDeTarefas: Essa classe gerencia uma lista de tarefas e contém métodos para adicionar, remover, atualizar e exibir as tarefas da lista. Pode ter atributos como um array ou lista para armazenar as tarefas.  Usuário: Essa classe representa um usuário do sistema e pode ter atributos como nome de usuário, senha, e-mail, entre outros. Ela pode lidar com a autenticação e autorização dos usuários. |

**Questão 03**

Apresente o conceito de OBJETO e indique exemplos de objetos utilizados no seu projeto.

|  |
| --- |
| Um objeto é uma instância de uma classe. No projeto de gerenciamento de tarefas, exemplos de objetos utilizados seriam:  tarefa1: Um objeto da classe Tarefa que representa uma tarefa específica com seus atributos preenchidos, como título "Fazer compras", descrição "Comprar itens do mercado", data de vencimento "2023-06-20" e status "A fazer".  listaTarefas: Um objeto da classe ListaDeTarefas que contém várias tarefas adicionadas a ele. Por exemplo, pode ter as tarefas tarefa1, tarefa2, tarefa3, etc.  usuário1: Um objeto da classe Usuário que representa um usuário registrado no sistema, com seus atributos preenchidos, como nome de usuário "user123", senha "senha123" e e-mail "user@example.com". |

**Questão 04**

Descreva o conceito de ENCAPSULAMENTO e mostre como esse conceito foi utilizado no seu projeto.

|  |
| --- |
| O encapsulamento é um princípio da POO que envolve ocultar os detalhes internos de implementação de uma classe e fornecer uma interface pública para interagir com os objetos. No projeto de gerenciamento de tarefas, o encapsulamento pode ser utilizado definindo os atributos da classe Tarefa como privados (ou protegidos) e fornecendo métodos getter e setter para acessá-los e modificá-los.  Por exemplo, a classe Tarefa pode ter um atributo privado "titulo" e métodos públicos como "getTitulo()" e "setTitulo()" para obter e definir o valor do título da tarefa. Isso garante que o acesso e a modificação dos atributos sejam controlados e evita que sejam alterados diretamente sem a devida validação. |

**Questão 05**

O que é um método CONSTRUTOR e mostre como esse método foi utilizado no seu projeto.

|  |
| --- |
| Um método construtor é um método especial em uma classe que é chamado automaticamente quando um objeto é criado a partir dessa classe. Ele é usado para inicializar os atributos do objeto e configurar seu estado inicial.  No projeto de gerenciamento de tarefas, pode haver um método construtor na classe Tarefa que recebe os parâmetros necessários, como título, descrição, data de vencimento e status, e os atribui aos atributos correspondentes da classe. Por exemplo:  public class Tarefa {  private String titulo;  private String descricao;  private LocalDate dataVencimento;  private boolean concluida;  public Tarefa(String titulo, String descricao, LocalDate dataVencimento) {  this.titulo = titulo;  this.descricao = descricao;  this.dataVencimento = dataVencimento;  this.concluida = false;  }  } |

**Questão 06**

Teste exaustivamente o sistema desenvolvido no seu projeto. Transcreva, para o espaço a seguir, alguns dos testes realizados, explicando-os.

|  |
| --- |
| Os testes realizados no sistema de gerenciamento de tarefas podem incluir:  Adicionar uma nova tarefa: Criar um objeto da classe Tarefa e adicioná-lo à lista de tarefas para garantir que a tarefa seja armazenada corretamente.  Atualizar o status de uma tarefa: Acessar uma tarefa existente na lista de tarefas e atualizar seu status de "A fazer" para "Concluída" para verificar se o status foi alterado corretamente.  Exibir lista de tarefas: Verificar se todas as tarefas da lista são exibidas corretamente, incluindo os atributos como título, descrição e data de vencimento.  Remover uma tarefa: Remover uma tarefa da lista de tarefas e garantir que ela seja removida corretamente, verificando se a lista é atualizada adequadamente.  Esses são apenas alguns exemplos de testes que podem ser realizados. É importante testar diferentes cenários e validar o comportamento esperado do sistema em várias situações. |

**Questão 07**

Para versões futuras, indique pelo menos 4 (quatro) melhorias que poderiam ser implementadas no sistema que você desenvolveu. Justifique suas escolhas.

|  |  |  |
| --- | --- | --- |
| **N°** | **Melhoria** | **Justificativa** |
| **1** | Adicionar prioridades às tarefas | Permitir que os usuários definam prioridades para suas tarefas, como alta, média e baixa, para ajudar na organização e destacar as tarefas mais importantes |
| **2** | Implementar categorias de tarefas | Permitir que os usuários criem categorias personalizadas para agrupar suas tarefas relacionadas, como trabalho, estudos, pessoal, etc. Isso facilitaria a filtragem e a visualização das tarefas por categoria. |
| **3** | Notificações e lembretes | Adicionar recursos de notificação para informar os usuários sobre prazos de tarefas, enviando lembretes por e-mail, notificações push ou mensagens no aplicativo |
| **4** | Compartilhamento de tarefas | Permitir que os usuários compartilhem tarefas com outros usuários, possibilitando a colaboração em projetos e atividades em equipe |