

Exercício 4: Sistema de Gestão de Tarefas

Descrição do Estudo de Caso

Uma equipe de desenvolvimento utiliza um sistema para gerenciar suas tarefas e projetos. O sistema deve permitir aos membros da equipe:

- **Adicionar** novas tarefas.
- **Atualizar** o status de tarefas existentes.
- **Consultar** as tarefas de um projeto específico.

Você foi encarregado de desenvolver uma aplicação em TypeScript que atenda a essas necessidades.

Tarefas

1. **Crie uma classe `Tarefa`** que possua:
 - Propriedades:
 - `id` (number): identificador único da tarefa.
 - `descricao` (string): descrição da tarefa.
 - `status` (string): status atual da tarefa (e.g., "Pendente", "Em Andamento", "Concluída").
 - `projeto` (string): nome do projeto ao qual a tarefa pertence.
 - Construtor que inicializa todas as propriedades acima.
2. **Crie uma classe `GestorTarefas`** que gerencie as tarefas:
 - Propriedade privada:
 - `tarefas` (array de `Tarefa`): lista de tarefas da equipe.
 - Métodos:
 - `adicionarTarefa(tarefa: Tarefa): void` - Adiciona uma nova tarefa ao sistema.
 - `atualizarStatus(id: number, status: string): void` - Atualiza o status da tarefa com o ID especificado.
 - `consultarTarefasPorProjeto(projeto: string): Tarefa[]` - Retorna todas as tarefas pertencentes ao projeto especificado.
3. **Implemente funções** para testar o sistema:
 - **Função para adicionar tarefas:** Crie pelo menos três instâncias de `Tarefa` e adicione-as ao sistema usando o método `adicionarTarefa`.
 - **Função para atualizar status:** Atualize o status de uma das tarefas existentes.

- **Função para consultar tarefas por projeto:** Consulte as tarefas de um projeto específico e imprima as informações no console.

Requisitos Técnicos

- Utilize os conceitos de **classe** e **função** em TypeScript.
- Use modificadores de acesso apropriados (**public**, **private**) para encapsulamento.
- Tipifique todas as variáveis e retornos de funções explicitamente.
- Trate possíveis erros, como tentativas de atualizar ou consultar tarefas inexistentes.

Objetivos de Aprendizado

- Praticar a definição e utilização de **classes** em TypeScript.
- Implementar e invocar **funções** para manipular objetos e classes.
- Aplicar conceitos de encapsulamento e tipagem estática.
- Simular um cenário real de gestão de tarefas em uma aplicação.

Entrega

- Código-fonte em TypeScript com a implementação das classes e funções solicitadas.
- Comentários no código explicando o funcionamento de cada parte.
- Um arquivo README explicando como executar o programa e os testes realizados.