

POO Antes ES16

Questão 1. Selecione uma classe que você criou nos exercícios anteriores e converta para o objeto literal, a classe deve ter construtor, herança e ampliar os métodos.

Questão 2. Selecione uma classe que você criou nos exercícios anteriores e converta para função objeto, a classe deve ter construtor, herança e ampliar os métodos.

Questão 3. Você foi encarregado de desenvolver um sistema para gerenciar os usuários de uma aplicação. Para cada usuário, o sistema deve registrar o **name**, o **email**, a **creationDate** e o **status** (ativo ou inativo).

O sistema precisa oferecer as seguintes funcionalidades:

1. **Add a new user:** O usuário será adicionado fornecendo seu **name**, **email** e a **creationDate** será registrada automaticamente.
2. **Change a user's status:** O sistema deve permitir alternar o **status** do usuário entre **active** e **inactive**.
3. **View a user's data:** Deve ser possível exibir o **name**, **email**, **creationDate** e **status** (ativo ou inativo) de um usuário.
4. **List all users:** O sistema deve permitir listar todos os usuários cadastrados, com seus dados completos.
5. **Iterate over users:** Utilizando um **iterator**, deve ser possível percorrer a lista de usuários e realizar operações personalizadas, como verificar quais usuários estão **active** ou listar apenas os **emails**.

Com base nos requisitos descritos, utilizando os conceitos de **Programação Orientada a Objetos (POO)**, implemente as classes **User** e **UserManager**, sendo que a classe **UserManager** deve ser capaz de gerenciar a lista de usuários e suportar iteração por meio de um **iterator**.

Questão 4. Você foi solicitado a criar um sistema para gerenciar veículos em uma locadora. Cada veículo deve ter métodos básicos para **start** e **stop**, que imprimem "Vehicle started" e "Vehicle stopped", respectivamente. No entanto, alguns tipos de veículos, como **Car** e **Motorcycle**, possuem funcionalidades extras, como **accelerate** ou **brake**.

Para implementar essas funcionalidades extras, crie um mixin chamado **MovementControlMixin** que adicione os métodos **accelerate** e **brake** aos veículos. Em seguida, utilize esse mixin nas classes **Car** e **Motorcycle**.

Por fim, crie uma função **testVehicle** que receba um veículo e, dependendo de sua classe, execute e exiba as ações de **start**, **stop**, **accelerate** ou **brake**, conforme apropriado.