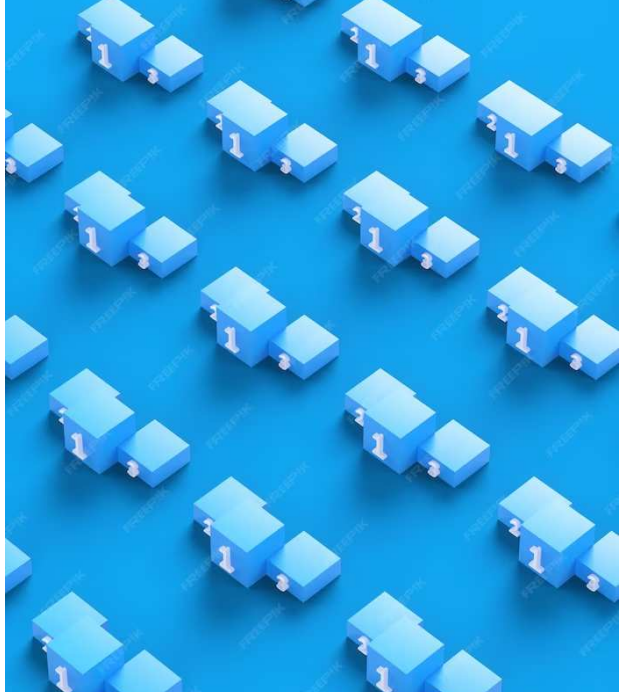




Cars moved

Bem-vindos ao CarsMoved

Sejam bem-vindos ao CarsMoved, o sistema de gerenciamento de veículos que simplifica e otimiza a gestão de frotas. Este projeto visa fornecer uma solução eficiente para organizar e controlar sua frota de maneira inteligente e inovadora





Desafios na Gestão de Frotas

Enfrentar os desafios da gestão de frotas pode ser complexo. Desde o acompanhamento de **múltiplos** veículos até a manutenção eficaz, o CarsMoved surge como uma resposta a esses desafios, oferecendo funcionalidades poderosas para **simplificar** e aprimorar o gerenciamento de frotas.

Os Benefícios do CarsMoved

O CarsMoved não é apenas um sistema, é uma solução completa. **Desfrute** de benefícios como rastreamento eficiente, manutenção preventiva, relatórios detalhados e muito mais. Nossa plataforma proporciona **economia de tempo** e recursos, permitindo que você se concentre no que realmente importa: **otimizar sua frota**.





Inovação com Tecnologias Emergentes

Nosso sistema está alinhado com as mais recentes **tecnologias emergentes**. Com recursos como Internet das Coisas (IoT), análise de dados avançada e integração móvel, o CarsMoved está na vanguarda da inovação. Mantenha-se à **frente** da concorrência e transforme a **gestão da sua frota com tecnologia de ponta**

O código

```
@property
def velocidade_atual(self):
    return self.__velocidade_atual

def acelerar(self, incremento):
    self.__velocidade_atual += incremento

def desacelerar(self, decremento):
    self.__velocidade_atual -= decremento

def to_dict(self):
    """Converte os atributos do veículo em um dicionário."""
    return {
        "modelo": self._modelo,
        "cor": self._cor,
        "velocidade_atual": self.__velocidade_atual
    }

@classmethod
def from_dict(cls, veiculo_dict):
    """Cria uma instância de VeiculoBase a partir de um dicionário."""
    return cls(veiculo_dict["modelo"], veiculo_dict["cor"])

class CarroNovoFrota:
    def __init__(self, nome_arquivo="frota.json"):
        self.veiculos = []
        self.nome_arquivo = nome_arquivo

    def adicionar_veiculo(self, veiculo):
        """Adiciona um veículo à frota e salva as alterações."""
        self.veiculos.append(veiculo)
        self._salvar_frota()

    def mostrar_informacoes(self):
        """Exibe informações detalhadas sobre cada veículo na frota."""
        for veiculo in self.veiculos:
            print(f"{type(veiculo).__name__} - Modelo: {veiculo.modelo}, Cor: {veiculo.cor}, Velocidade Atual: {veiculo.velocidade_atual}")

    def gerar_relatorio(self):
        """Gera um relatório em formato de lista de dicionários."""
        relatorio = []
        for veiculo in self.veiculos:
            relatorio.append(veiculo.to_dict())
        return relatorio

    def _salvar_frota(self):
        """Salva os dados da frota em um arquivo JSON."""
        with open(self.nome_arquivo, "w") as arquivo:
            json.dump(self.gerar_relatorio(), arquivo)

    def _carregar_frota(self):
        """Carrega a frota a partir de um arquivo JSON, se existir."""
        try:
            with open(self.nome_arquivo, "r") as arquivo:
                dados = json.load(arquivo)
                for veiculo_dict in dados:
                    veiculo = VeiculoBase.from_dict(veiculo_dict)
                    self.veiculos.append(veiculo)
        except FileNotFoundError:
            print("Arquivo de frota não encontrado. Inicializando uma nova frota.")

# Exemplo de uso
frota = CarroNovoFrota()
```

- Herança e Polimorfismo
- Hierarquia de classes: VeiculoBase, Carro, e Moto. Métodos acelerar e desacelerar implementados de forma polimórfica em cada classe.

- Encapsulamento:
- Atributos privados acessados por meio de métodos getters e setters. Controle seguro e abstrato do estado interno dos veículos.

- Construtores
- Construtores inicializam atributos específicos de cada veículo.

- Listas e Métodos
- Lista veiculos armazena a frota. Métodos para adicionar veículos, mostrar informações e persistir dados.

- Persistência de Dados
- Dados da frota persistem em arquivo JSON entre execuções.

- Implementação Eficiente
- Estrutura modular facilita adição de veículos e manutenção do sistema.



Implementação Eficiente

A implementação do CarsMoved é simples e eficiente. Com uma **interface intuitiva**, a integração do sistema na sua operação é **rápida e fácil**. Nosso suporte técnico está pronto para garantir que você tire o máximo proveito de todas as funcionalidades, proporcionando uma **transição suave e eficiente**

Conclusão

Em conclusão, o **CarsMoved** é mais do que um sistema de gerenciamento de veículos. É uma ferramenta que impulsiona a eficiência, reduz custos e **aumenta a produtividade**. O compromisso é fornecer soluções que facilitam a gestão de frotas.

Transforme sua operação hoje com o **CarsMoved**!