

Entregue todas as classes e métodos em um arquivo chamado `lab3.py` e escreva o seu nome como um comentário na primeira linha do arquivo. Critérios de avaliação: (50%) o código funciona e os métodos fazem o que foi pedido, (50%) a herança de classes foi corretamente usada, o código é legível e eficaz.

Veja a classe `VeiculoAutomotor` definida abaixo. Vamos criar as subclasses `Automovel` e `Caminhao` dessa classe, então copie esse código no seu arquivo `lab3.py`.

```
class VeiculoAutomotor():
    def __init__(self, dono, placa, combustivel):
        self.dono = dono
        self.placa = placa
        self.combustivel = combustivel

    def __str__(self):
        return "Dono: {}, placa: {}, combustivel: {}".format(self.dono, self.placa, \
                                                            self.combustivel)
```

Veja um exemplo de uma instância da classe:

```
>>> auto = VeiculoAutomotor("Emile", "ABC123", "gás")
>>> print(auto)
Dono: Emile, placa: ABC123, combustivel: gás
```

1. Crie a subclasse chamada `Automovel` da classe `VeiculoAutomotor` da seguinte forma:

- (a) (3 pontos) Defina o construtor da classe que deve corretamente usar o construtor da superclasse. A classe `Automovel` tem 6 atributos: `dono`, `placa`, `combustivel`, `lugares`, `portas`, `ano`. Veja um exemplo de uma instância da classe abaixo:

```
>>> fusca = Automovel("Lea", "XYZ111", "eletricidade", 5, 3, 1940)
```

- (b) (2 pontos) Defina método especial `__str__` que retorna uma string com os dados de um objeto da classe como no exemplo abaixo:

```
>>> print(fusca)
Dono: Lea, placa: XYZ111, combustivel: eletricidade, lugares: 5, portas: 3, ano: 1940
```

- (c) (1 ponto) Implemente o método `trocarDono` que recebe o nome do novo dono e então muda o atributo `dono` do objeto. Veja como deve funcionar:

```
>>> fusca.trocarDono("Mia")
>>> print(fusca)
Dono: Mia, placa: XYZ111, combustivel: eletricidade, lugares: 5, portas: 3, ano: 1940
```

2. Crie a subclasse chamada `Caminhao` da classe `VeiculoAutomotor`:

- (a) (3 pontos) Defina o construtor da classe que deve corretamente usar o construtor da superclasse. A classe `Caminhao` tem 4 atributos: `dono`, `placa`, `combustivel`, `cargaMax`. Veja um exemplo de um objeto da classe abaixo:

```
>>> camionete = Caminhao("Emile", "ABC123", "gás", 4)
```

- (b) (1 ponto) Defina método especial `__str__` que retorna uma string com os dados de um objeto da classe como no exemplo abaixo:

```
>>> print(camionete)
Dono: Emile, placa: ABC123, combustivel: gás, carga máxima: 4 toneladas
```