

Guião Prático - Polimorfismo

Introdução

O nosso objectivo é criar um modelo que permita implementar um simulador onde existem vários tipos de veículos.

Para o fazer queremos tirar partido do **polimorfismo**.

Nesta ficha está **proibido** o uso de *instanceof* e de outros testes de tipo equivalentes.

Modelo

Considere as seguintes entidades:

- Veículo
- Avião
- Barco a Remos
- Bicicleta
- Carro
- Mota

Comece por criar uma classe para representar cada uma destas entidades.

A classe Veículo deverá ser abstracta. Todas as outras devem herdar de Veículo.

Crie também a classe Main e a função main.

Na classe mais genérica (a que representa o Veículo) adicione o atributo **boolean ligado** que vai ser true caso o veículo tenha o motor ligado. Deve também adicionar 2 métodos que devem afectar o valor desse atributo:

- void ligar()
- void desligar()

Métodos

Pretende-se que todos veículos saibam responder a um método

- **void mover()**

O método deverá ter o seguinte comportamento:

- no caso das Bicicletas e dos Barcos a Remos (que não têm motor) deve ser sempre apresentada a mensagem:
 - “A mover a bicicleta” ou “A mover o barco a remos”
- no caso dos restantes sub-tipos:
 - Se o veículo estiver **ligado**, deverá apresentar a mensagem
 - “A mover o veículo X”
 - Se o veículo estiver desligado, deverá apresentar a mensagem
 - “Não posso mover um X desligado”

(Onde X é o nome do tipo do Veículo - p.e. “carro”).

(No caso da Mota, coloque as mensagens em concordância (i.e. “uma mota desligada”).

Criação de objectos

Vamos agora criar, na classe Main, uma função cujo objectivo é devolver um conjunto de objectos das classes indicadas. A assinatura esperada é a seguinte:

```
static _____ getVehicles()
```

O conjunto devolvido deve conter um objecto de cada sub-tipo (um objecto “Bicicleta”, um objecto “Mota”, etc).

Qual poderá ser o tipo de retorno da sua função?

Use o método **ligar()** para garantir que o objecto que representa o Carro está “ligado”.

Funcionalidade

Na função `main()` pretende-se simular o movimento dos vários tipos de veículos, recorrendo aos objectos criados na função `getVehicles()`.

Invoke então essa função para obter o conjunto de Veículos.

De seguida, usando um ciclo, aplique o método `mover()` a cada veículo.

O *output* esperado deve conter as seguintes linhas (eventualmente por outra ordem):

```
Não posso mover um avião desligado
A mover o barco a remos
A mover a bicicleta
A mover o carro
Não posso mover uma mota desligada
```

Execute o seu programa e verifique se o output respectivo tem as mesmas linhas que

o indicado acima.

Revisão do modelo

Pela funcionalidade anterior percebemos que nas classes que representam a Bicicleta e o Barco a Remos não existe a noção de ligado / desligado.

Como podemos alterar o nosso modelo de classes para melhor reflectir esta realidade?

FIM