

Folla 4.2.- Interfaces e Clases abstractas 1

1. Fai un programa que cree as seguintes clase e métodos relacionados coa seguinte xerarquía de animais:
 - x Os **mamíferos** son unha clase abstracta cun método abstracto **amamantarCrias()**, que non devolve nada.
 - x As **baleas** forman unha clase, son mamíferos e implementan o método abstracto herdado: mostra simplemente unha mensaxe indicando que esa balea amamanta ás súas crías.
 - x Os animais ovíparos son unha interface co método **porOvos()**.
 - x O **Ornitorrinco** é un mamífero que pon ovos, polo que será unha clase filla de **Mamífero**, e implementará a interfaz **Ovíparo**. O método herdado da clase nai debe amosar que o ornitorrinco amamanta ás súas crías, e o método implementado desde a interface amosará que o ornitorrinco pon ovos.

Crea unha clase **AppMamífero**, na que crees 2 obxectos baleas, e 2 ornitorrinco e chames aos métodos herdados/implementados.

2. Completa o exercicio anterior cunha interfaz **Voador** que representa un animal que voa, cun método **voa()**. Agrega unha clase **Morcego** que implementa a interfaz **Voador**, amosando por pantalla que o animal pode voar. Crea agora unha clase **Gaivota**, e implementa as dúas interfaces. Finalmente, crea na clase **AppMamífero** do exercicio anterior 2 morcegos e 2 gaivotas que voan, e comproba que os métodos funcionan.

Lembra que unha clase pode implementar varias interfaces separadas por comas:

```
public class Elefante extends Mamifero implements Selvatico, Salvaxe
```

3. Define unha clase **Vehículo**, con atributos *velocidadeActual*, e *velocidadeMáxima*. Terá 2 métodos abstractos:
 - *acelerar()*: permitirá incrementar a velocidade actual sumando a velocidade pasada como parámetro. A velocidade actualizada non poderá superar a velocidade máxima.
 - O método *frear()* permite baixar a velocidade restando a cantidade pasada como parámetro. A velocidade actualizada non pode ser negativa.

O método *imprimir* mostra en pantalla a velocidade actual e a velocidade máxima do vehículo.

Teremos 2 tipos de vehículos: **terrestres** e **acuáticos**. Os terrestres teñen tamén cantidade de rodas, e uso do vehículo (militar ou civil). Os acuáticos terán como atributos o *tipo* (superficie ou submarinos) e *capacidade* de viaxeiros.

Existen tamén 2 interfaces: **Motor** e **Vela**. A clase Vehículo terrestre implementa a interface **Motor**. A clase **VehiculoAcuatico** implementa as dúas interfaces. Estas son do seguinte modo:

- A interface **Motor** ten o método *calculaRevolucionesMotor(int forza, int radio)*: o número de revolucións vén calculado pola multiplicacións de ambos.
- A interface **Vela** ten o método *recomendarVelocidade(int velocidadeVento)*, se a velocidade do vento é superior a 60 km/h non se recomenda saír, polo que a velocidade actual actualízase a 0. Se é menor que 10 km/h tampouco se recomenda saír, polo que fará o mesmo.

Define unha clase **AppVehiculos** que cree 2 camiós e dúas barcas (as dúas con vela e motor), que chaman aos métodos de cada clase e mostren os resultados por pantalla. Debes comprobar que funcionan correctamente todos os métodos.