

## Boletín de clases e implementación

Implementa este boletín con Eclipse. Podes emplegar un único proxecto. Utiliza, en xeral, un paquete para cada exercicio (`com.acarballeira.exercicios.clases. ejercicio1`)

Para facer as probas emprega o paquete `com.acarballeira.exercicios.drivers`

1. Implementa a clase **Arbore** que conteña as propiedades/atributos seguintes:
  - a. Altura en metros (pode haber decimais)
  - b. Ano de nacemento
  - c. Concello de localizacion
  - d. Nome común
  - e. Nome latino
  - f. Tipo de folla que será un enumerado cos posibles valores: caduco, perenne
  - g. Os atributos serán todos públicos
  - h. O constructor será o que teña por defecto
  - i. Créalle un método `toString` para poder obter información da árbore
  - j. Crea unha clase driver asociada ao proxecto no paquete indicado e dá de alta 3 árbores. Amosa a información das árbores.

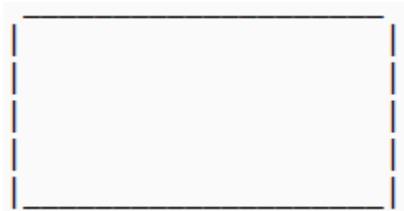
Exemplo de saída:

```
-----
|Nome: Piñeiro
|Nome latino: Pinus Pinaster
|Tipo folla: PERENNE
|Altura: 23,00
|Concello: Ramirás
|Ano nacemento: 2000
-----
-----
|Nome: Castiñeiro
|Nome latino: Castanae Sativa
|Tipo folla: CADUCA
|Altura: 23,00
|Concello: Ramirás
|Ano nacemento: 1990
-----
-----
|Nome: Amieiro
|Nome latino: Alnus Glutinosa
|Tipo folla: PERENNE
|Altura: 23,00
|Concello: Celanova
|Ano nacemento: 2003
```

2. Efectúa as seguintes evolucións sobre a clase anterior
  - a. Engádelle unha propiedade estática, e privada chamada **numArbores**, que incremente o seu valor cada vez que se cree unha árbore.
  - b. Crea un constructor que acepte todos os atributos.
  - c. No método `toString` emprega o método estático **format** da clase `String` para formatar a saída. Resolve este método nunha so aliña. É similar a traballar con `printf` (<https://dzone.com/articles/java-string-format-examples>)
  - d. Emprega o operador **this** para traballar a nivel interno na clase.
  - e. Proba o programa cunha nova clase driver e que finalmente amose o número de árbores creadas.
3. Representa a clase **Arbore** dos exercicios anteriores mediante diagramas con umlet en Eclipse.

4. Crear una clase **Rectangulo**, con atributos double base y altura. Crear también el constructor de la clase y los métodos necesarios para calcular el área y el perímetro. Crea unha clase denominada **PruebaRectangulo** no paquete de drivers para probar o cálculo de áreas e perímetros de varios rectángulos. Implementa o método `toString` para pintar o rectángulo (redondea os decimais).

Exemplo resultado impresión dun rectángulo de 10x5:



5. Crear una clase **Tiempo**, con atributos hora, minuto y segundo, que pueda ser construida indicando los tres atributos, sólo la hora y minuto, sólo la hora o ninguno. Crear además los métodos necesarios para modicar la hora, minuto y segundo en cualquier momento de forma manual. Mantenga la integridad de los datos en todo momento.

Todos los atributos deben ser privados a la clase. Construye para ello métodos getters y setters para dichos atributos.

Implementa el método `toString` que devuelva la hora en formato HH:MM:SS (p.e: 09:55:02)

Crear una clase **PruebaTiempo** dentro del paquete de drivers que prueba una hora concreta y la modique a su gusto mostrándola por pantalla.

6. Crear una clase **PruebaTiempoCorrido** dentro del paquete de drivers que defina una hora de inicio y una hora de fin y que cada segundo actualice la hora, y la muestre por pantalla hasta la hora de fin. Implementa los métodos necesarios en `Tiempo` para extender esta funcionalidad (entre ellos el método `equals`).

```
14:55:00
14:55:01
14:55:02
14:55:03
14:55:04
14:55:05
14:55:06
14:55:07
```

7. Crea un diagrama de clases en Eclipse con umlet que dea cobertura ás seguintes premisas e posteriormente impleméntao:

- Haberá unha clase denominada **Coordenada** que albergará as propiedades x e y que serán valores enteiros. Crea dous construtores.
- Haberá un clase denominada **Parcela** que albergará as propiedades puntoRef1 e puntoRef2 (de tipo Coordenada). Tamén terá unha propiedade metros que será un doublé e almacenará os metros cadrados. Crea un único construtor que reciba valores para todas as propiedades da clase.
- Haberá unha clase **Finca** que terá as propiedades parcela (de tipo Parcela), topónimo, cultivo (enumerado TipoCultivo que pode ser *monte baixo*, *arbustivo*, *cultivo* ou *viñedo*) e un indicador de se é de regadío ou non. Tamén terá un atributo chamado metrosLimpos, que indicará a cantidade de m<sup>2</sup> que están fóra de perigo como medida de protección contra os incendios. Inicialmente os m<sup>2</sup> da

finca estarán todos limpos. Crea o constructor por defecto e outro que requira todos os atributos menos metrosLimpos.

- d. Todas as propiedades serán privadas e terás que implementar os getters e setters.
- e. Terás que implementar o método `toString` só da clase Finca. *Emprega String.format*
- f. Terás que implementar o método `incrementarSupALimpar` que indicará a necesidade de limpar os m<sup>2</sup> que se pasen como parámetro (esta cantidad restarase á cantidad xa almacenada na propiedade metrosLimpos).
- g. Terás que implementar o método `decrementarSupALimpar` que indicará que se limpian os m<sup>2</sup> que se pasen como parámetro. (esta cantidad sumarase á cantidad xa almacenada na propiedade metrosLimpos).
  - i. Mantén a integridade dos datos. Non pode haber m<sup>2</sup> negativos, nin que superen a superficie da finca.
- h. Crea unha clase chamada ProbaFinca en onde trastees con dúas fincas e fagas probaturas.

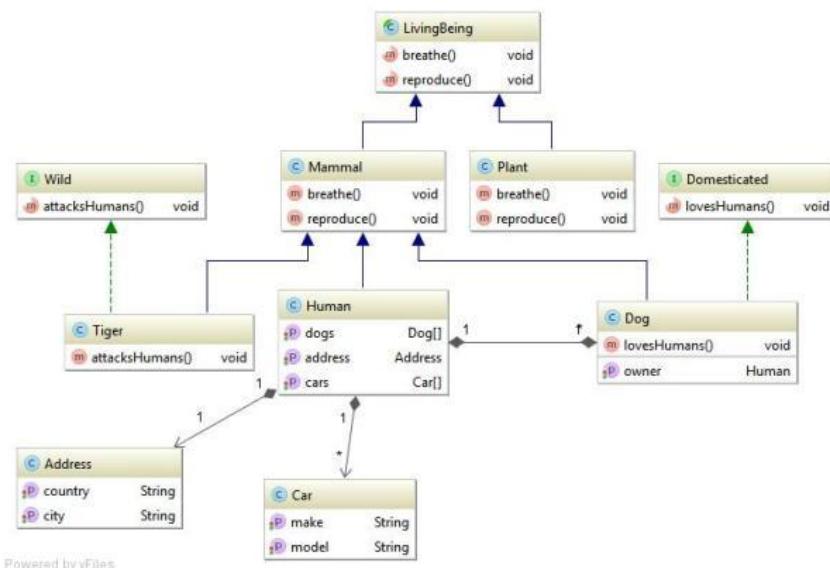
```
*** FINCA *** AS REGADAS
Tipo de cultivo: CULTIVO
Regadio: true
Coordenadas de referencia: 4,5 - 8,9
Metros totais: 500,000
Metros limpios: 400,000
```

- 8. Dota de información á clase Arbore para que teña unha localización mediante a clase Coordenada e implementa o método `trasplantar` de dúas formas distintas (sobrecarga)
  - a. indicando a nova posición da árbore
  - b. indicando a nova posición da árbore e a data de trasplantado (dálle voltas ao asunto).

*Para este exercicio traballa coa paquete java.time e a clase LocalDate.*

*Todos os atributos que engadas deberán ser privados. Non modifiques os constructores.*

- 9. Lista os atributos da clase Alumno. Sería nome un dos atributos da clase?
- 10. Cales serían as propiedades e métodos da clase Coche? Indica polo menos tres instancias da clase coche.
- 11. Pensa nunha vivenda/casa e en polo menos 5 clases que poidan relacionarse coa casa.
- 12. Indica algúns atributos/propiedades da clase vehículo. Razoa tamén a existencia de subclases asociadas.
  
- 13. Cales serían os atributos da clase Ventana de computadora? E os seus métodos?
- 14. Pensa na axenda de contactos do teu móvil. Poderías organizar as seguintes clases (Axenda, Categoria, Contacto, Amigo, Familiar) nun diagrama?
  
- 15. Interpreta do seguinte diagrama



Powered by yFiles.