

Traits en Scala

S4N Campus

3 de marzo de 2021

1. Preliminares

Este taller tiene como objetivo poner en practica los elementos de *traits* en Scala.

2. Traits

Ejercicio 1. Vamos a modelar un comportamiento común a gatos, leones, tigres, panteras, jaguares, etc. Vamos a definir una interface común (*trait*) **Felino**. En la figura 1. Defina el *trait* Felino y las diferentes sub-especies de la super clase ella.

Tenga en cuenta que:

- En Felino se encuentra el color.
- En Felino se encuentra el sonido que hace cada uno de los diferentes tipos.
- Únicamente Gato tiene una comida favorita.
- Un León tiene un tamaño de melena.
- Recuerde que todos los atributos tiene sus correspondientes métodos *getters*.
- Los constructores se hacen en las clases no en los *traits*.

□

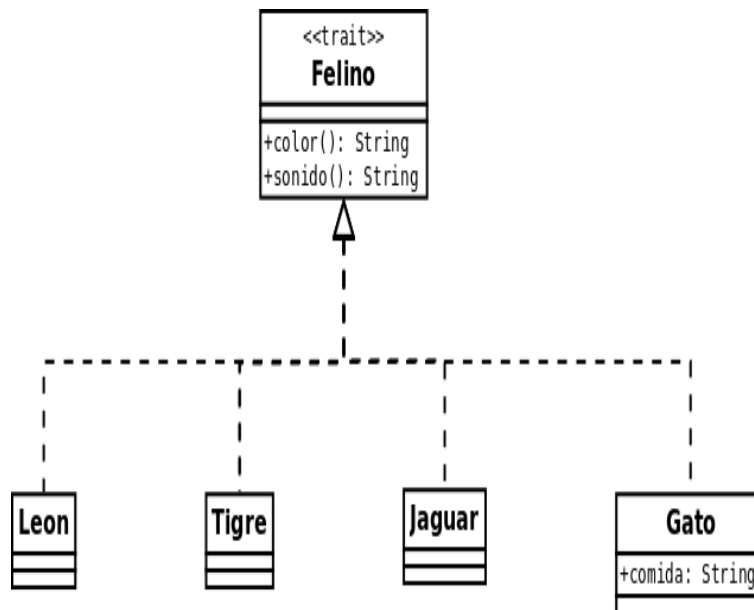


Figura 1: Jerarquía de gatos

Ejercicio 2. Define un *trait* llamado Forma y tiene tres métodos abstractos:

- tamaño que retorna el número de lados.
- perímetro que retorna la longitud total de sus lados.
- área que retorna el área de la forma en cuestión.

Implemente Forma con tres clases: *Círculo*, *Rectángulo* y *Cuadrado*. En cada caso implemente los tres métodos. Asegure un constructor principal con los parámetros para cada forma y que los campos son accesibles (*getters*).

□

Ejercicio 3. La solución del ejercicio anterior produjo tres tipos diferentes de formas. Sin embargo, este no modela la relación entre los elementos correctamente. Un *Cuadrado* no es solamente una forma también es una forma de tipo *Rectángulo* donde la longitud y la altura son iguales.

Refactorize la solución del anterior ejercicio así el *Cuadrado* y *Rectángulo* son subtipos de un tipo común llamado *Rectangular*.

Note: Un *trait* puede extender otro *trait*.

□

3. Sealed Traits

Ejercicio 4. Revisemos la forma implementada en la anterior seccion 2. En primer lugar haga `Forma` en un `sealed trait`. Entonces, escriba un objeto único (*singleton object*) llamado `Draw` con un método `apply` que tomar una `Forma` como argumento y retorna un descripción de él en la terminal.

Por ejemplo:

```
scala> Draw(Circulo(10))
res0: String = Un circulo de radio 10.0cm
scala> Draw(Rectangulo(3,4))
res1: String = Un rectangulo de ancho 3.0cm y largo 4.0cm
```

Finalmente, verifique que el compilador se queja cuando se comente una clausula `case` en el método `Draw`.

□

Ejercicio 5. Escriba una clase `Color`

1. Da al color tres propiedades: `RGB` (`Red`, `Green`, `Blue`).
2. Cree tres colores predefinidos: `Rojos`, `Amarillos` y `Rosa`.
3. Suministre un medio para que la gente produzca sus propios colores personalizados de `Color` con sus propios valores de `RGB`.

Este ejercicio es abierto a interpretación. □