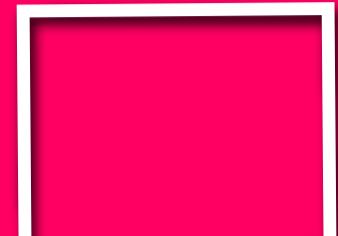


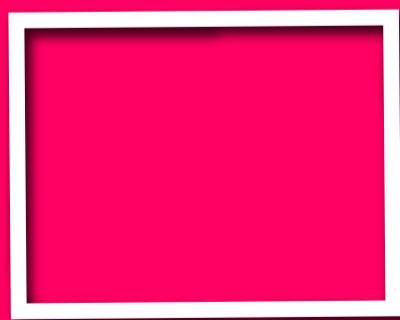
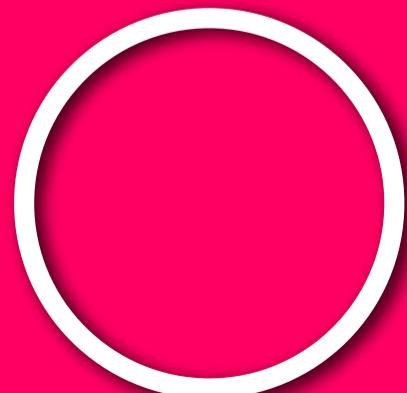


El futuro digital
es de todos

MinTIC



Polimor
fismo
Puro



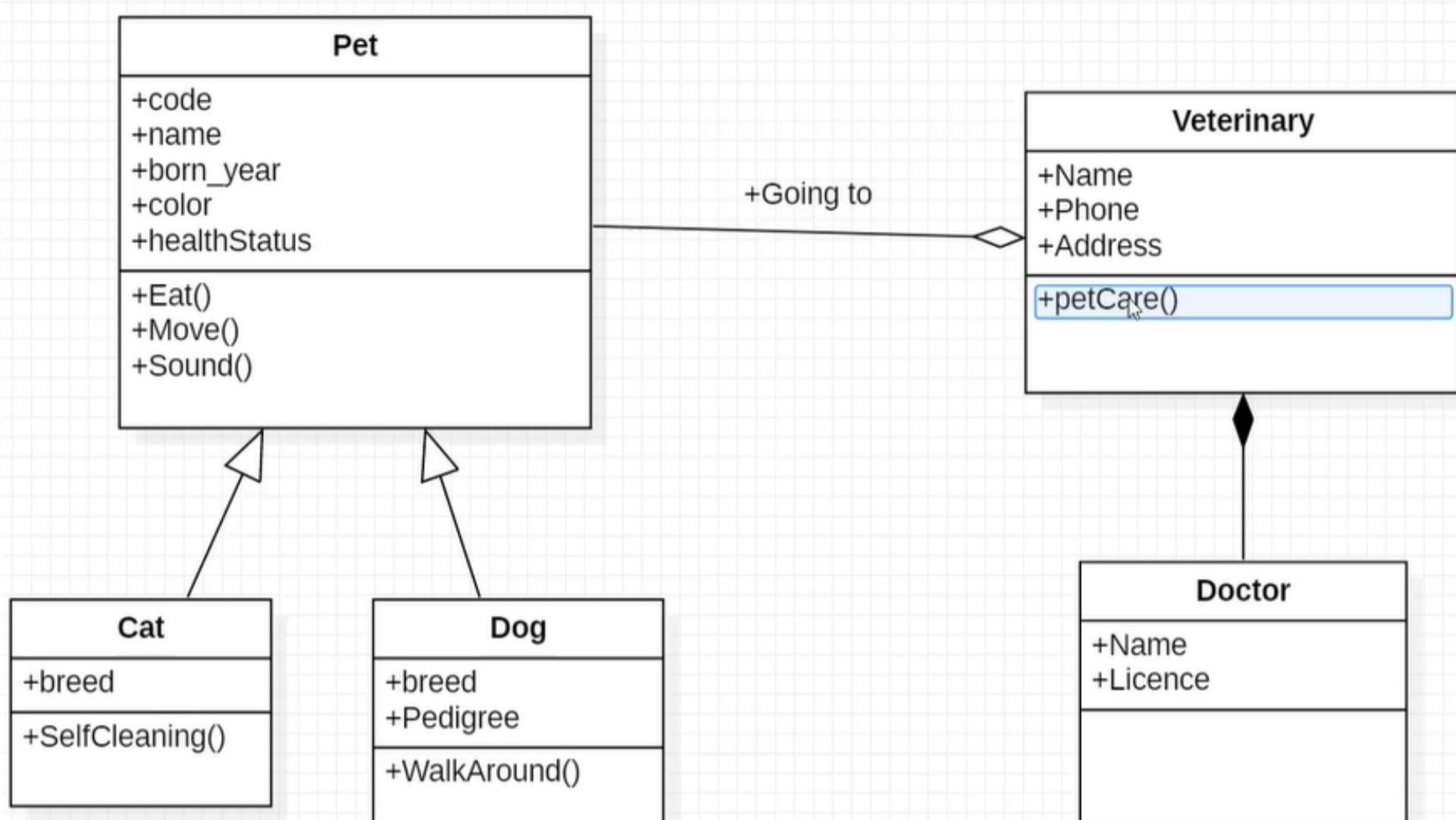
Universidad de Caldas

Hola:

Teniendo en cuenta que ya aprendimos sobre las relaciones que existen entre clases, conoceremos el concepto llamado **Polimorfismo**, que a su vez lo podemos ver desde diferentes perspectivas y en este caso les explicaremos tres (3) tipos de polimorfismo que en nuestro concepto son los más importantes en la programación orientada a objetos. Vamos a detallar cada uno de ellos teóricamente y luego realizaremos su aplicación.

El primer concepto que veremos se llama **Polimorfismo Puro** que se basa en la herencia que ya les hemos explicado, ya que un método o función puede recibir un parámetro de un tipo de objeto de una clase padre y desde la invocación a dicho método o función se puede proveer como parámetro a un objeto de cualquier clase hija. Para aclarar esta definición veremos un ejemplo práctico.

Video con el ejemplo de polimorfismo puro (ejemplo de llevar mascota al veterinario que recibe un objeto mascota y se le puede pasar un gato o un perro)



The screenshot shows the NetBeans IDE interface with the following details:

- Project Structure:** The project is named "PetManager". The **Source Packages** node contains sub-packages **ds** and **petmanager**. The **ds** package contains classes **clsCat.java**, **clsDoctor.java**, **clsDog.java**, **clsPet.java**, and **clsVeterinary.java**. The **petmanager** package contains the class **PetManager.java**.
- Code Editor:** The file **dsVeterinary.java** is open. The code defines a class **clsVeterinary** with a constructor and a method **PetCare**. The constructor initializes attributes **name**, **phone**, **address**, and **doctor**. The **PetCare** method prints a message and returns the pet's health status.
- Output:** The output window shows the message "Atendiendo a la mascota " followed by the pet's name.
- Navigator:** The **PetCare - Navigator** panel shows the members of the **clsVeterinary** class, including the constructor and the **PetCare** method.

The screenshot shows the NetBeans IDE interface with the following details:

- Project Explorer:** Shows the "PetManager" project structure under "JavaInIntroduction".
- Code Editor:** Displays the content of the "Polymorphism.java" file. The code defines a main method that creates instances of `clsDog`, `clsCat`, `clsDoctor`, and `clsVeterinary` classes. It then calls the `PetCare` method on each instance and prints the results.
- Output Window:** Shows the run output for the "petmanager.Polymorphism > main > careDog" run. The output is:

```
run:
Atendiendo a la mascota Firulais
El estado de salud de Firulais es Sano
Atendiendo a la mascota Minino
El estado de salud de Minino es Enfermo
BUILD SUCCESSFUL (total time: 0 seconds)
```

Ahora con el **polimorfismo puro** comprendido, los invitamos a ponerlo en práctica.



**Mision
TIC2022**

The logo features the text "Mision TIC2022" in a bold, sans-serif font. The word "Mision" is positioned above "TIC2022". A thin, curved line starts from the top of the letter "i" in "Mision" and sweeps down to the bottom of the letter "c" in "TIC2022", ending with a small circular dot. The entire logo is set against a white circular background, which is itself centered within a larger, semi-transparent gray circle.



Universidad de Caldas