

## Parte 1

1.1 Crear una clase abstracta denominada Animal, de la cual derivan las clases Perro, Gato y Vaca. Todas las subclases deben redefinir el método abstracto hablar(), de la clase Animal, según como “hable” cada uno. Para ello puede declarar el método como de tipo de retorno void y abstracto y luego utilizar el método println de System.out en cada implementación del mismo.

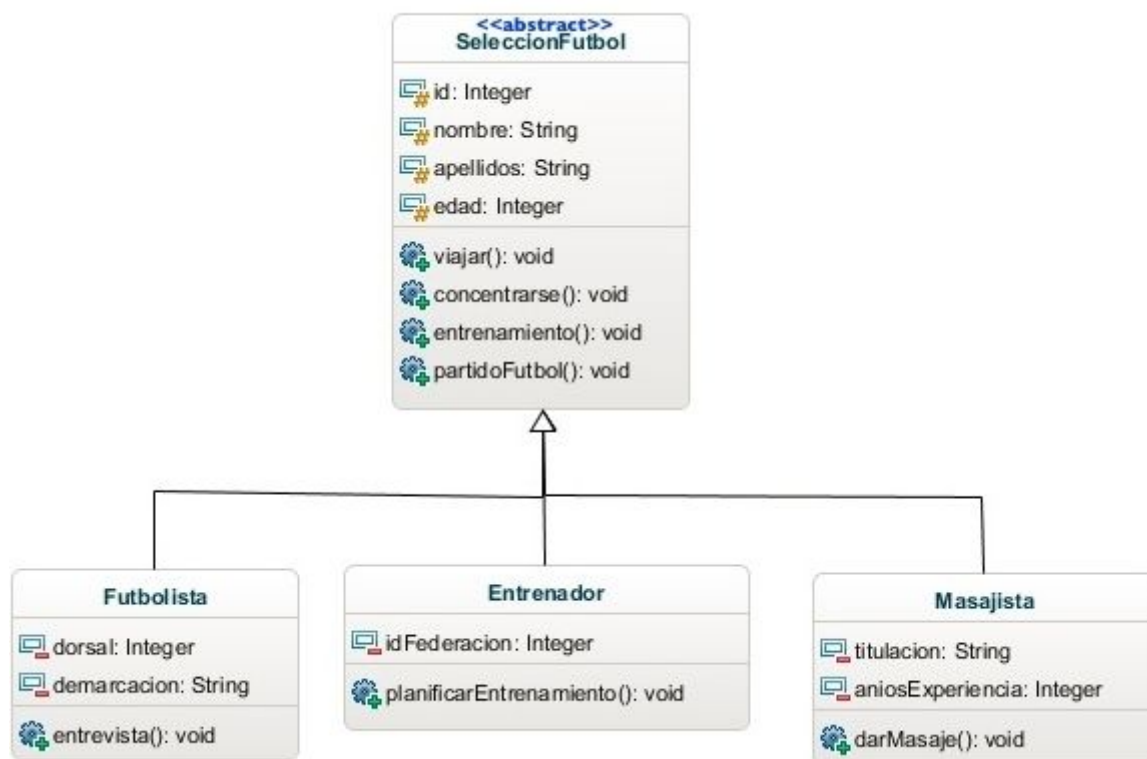
Además realice una clase que contenga el método main() denominada Principal que invoque a un método denominado hablarAnimal(Animal a) que reciba un “Animal” como parámetro, y que le permita “hablar”, polimórficamente, según sea Gato, Perro o Vaca.

1.2 Modifique el método main utilizado en el ejercicio anterior, invocando a otro método denominado comerAnimal que reciba como parámetro a un Animal e invoque al correspondiente método para comer (comerCarne o comerHierba) según el tipo de animal que sea.

1.3 Crear un ArrayList y almacenar diferentes cantidades de objetos: perros, gatos y vacas, desarrollar un método que se llame contarAnimales, en donde permita pasarle el arraylist generado previamente y que devuelva la cantidad de perros, gatos y vacas almacenados.

## Parte 2.

Dado el siguiente diagrama de clases:



Donde el método entrenamiento es abstracto. En cada método colocar solo un System.out.println con alguna cadena describiendo el método.

Implementar una clase principal con un método main, en donde podamos crear tres objetos de la clase SeleccionFutbol que adoptan una de las tres formas que pueden adaptar (Entrenador, Futbolista y Masajista) y los almacenamos en un "ArrayList" de objetos de la clase "SeleccionFutbol".

Métodos a implementar:

arrancarConcentracion, donde le pasemos el arraylist con los integrantes de la selección de fútbol, y para cada integrante ejecute su método concentración.

Luego realizar otro método donde todos los integrantes viajan para jugar un partido.

arrancarEntrenamiento: Todos los integrantes tienen su función en un entrenamiento. Es decir implementar un método que según el tipo de integrante llame al método de entrenamiento correspondiente. (polimorfismo)

JugarPartido: Todos los integrantes tienen su función en un partido. (sobreescripción de método).

Parte 3

Revisa el apunte de Java Iterator vs ForEach y encuentra cuales son las diferencias entre ambos.