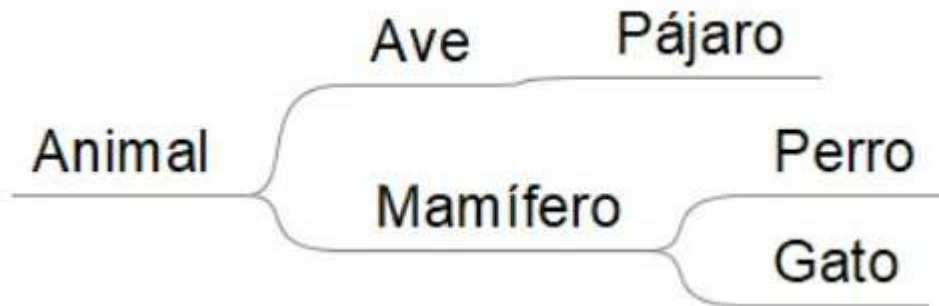


Se pide crear una jerarquía de clases de animales que tendrán en común la capacidad de "hablar" y de "comer". Además, los animales tendrán en cada momento más o menos energía (o "vitalidad"); la energía se reducirá ligeramente cuando hablen y aumentará cuando coman. Hablar consistirá en imprimir en pantalla mediante un sysout determinado texto, acorde con el animal de que se trate. Comer consistirá en aumentar la energía. La jerarquía será así:



Algunas de estas clases serán abstractas. Habrá métodos abstractos en **Animal**:

- `habla()`
- `come(int cantidadComida)`

Animal tendrá un campo: energía que irá de 0 a 100, nunca bajará de 0 ni pasará de 100
Se pide:

- constructor(es) que reciban la energía inicial del animal o si no se indica nada, será 100
- implementación del `.habla()`, en función de la energía. Por ejemplo, el gato:
 - `energía < 30` → miauuuuuu...
 - `energía >= 30 && <70` → miau
 - `energía >= 70` → ¡¡MIAU!!
- Y los mismos valores para pájaro y perro.
- Al hablar las aves reducirán su energía en 3 unidades y los mamíferos en 5 unidades.
- pista: implementar esto en concreto en **ANIMAL**, y llamar a `super.habla()` desde las subclases para hacer este parte, y luego añadir lo que corresponda
- implementación del `.come(int cantidadComida)`, en función de la cantidad de comida y del tipo de animal:
 - un pájaro aumentará 5 de vitalidad por cada unidad de comida,
 - un perro aumentará 7
 - un gato 12

Clase principal, la lista de animales debe salir ordenada de menor a mayor.

```

public static void main(String[] args) {
    ArrayList<Animal> listaAnimal = new ArrayList<Animal>();
    listaAnimal.add(new Pajaro(paramEnergia: 30));
    listaAnimal.add(new Perro(paramEnergia: 60));
    listaAnimal.add(new Gato(paramEnergia: 20));
    listaAnimal.add(new Gato(paramEnergia: 90));
    listaAnimal.add(new Pajaro(paramEnergia: 70));
    Collections.sort(listaAnimal);
    System.out.println("Lista animales inicial="+listaAnimal);
    for (int ronda=1; ronda<=10; ronda++) {
        for (Animal animalActual: listaAnimal){
            animalActual.habla();
            animalActual.come( (int) (Math.floor(Math.random()*3)) );
        }
    }
    //Volvemos a ordenar
    Collections.sort(listaAnimal);
    System.out.println("\nLista animales final="+listaAnimal);
    System.out.println(x: "-----FIN-----");
}

```

Resultado de la ejecución

```

.lista animales inicial=[Gato [energia=20], Pajaro [energia=30], Perro [energia=60], Pajaro [energia=70], Gato [energia=90]]
miauuuuu pio guau ¡¡PIO PIO!!¡¡MIAOU!! miauuuuu pio guau ¡¡PIO PIO!!¡¡MIAOU!! miau pio guau ¡¡PIO PIO!! miauuuuu miau pio ¡¡GUAU!!¡¡PIO PIO!! miauuuuu miau pio ¡¡G
IAU!!¡¡PIO PIO!! miauuuuu miau pio ¡¡GUAU!!¡¡PIO PIO!! miauuuuu miau pio ¡¡GUAU!!¡¡PIO PIO!! miau ¡¡MIAOU!! pio ¡¡GUAU!!¡¡PIO PIO!! miau ¡¡MIAOU!! pio ¡¡GUAU!!¡¡PIO PIO!! m
iau ¡¡MIAOU!! pio ¡¡GUAU!!¡¡PIO PIO!! miau
.lista animales final=[Gato [energia=0], Pajaro [energia=63], Gato [energia=64], Perro [energia=92], Pajaro [energia=100]]
-----FIN-----

```