

Necesidad energetica en perros:

$$70 \times (\text{peso en kg})^{0.75}$$

En gatos:

Ejercicio 4. (20 pts.) Si se tiene la siguiente implementación:

```
1 public interface AutoModificable{
2     void aumentar(int n);
3     void disminuir(int n);
4 }
5 public class Fraccion implements AutoModificable{
6     private int numerador, denominador;
7     public void aumentar(int n){
8         .....
9     }
10    }
11    public void disminuir(int n){
12        .....
13    }
14 }
15
16 public class Rectangulo extends AutoModificable{
17     private int base, altura;
18     public void aumentar(int n){
19         .....
20     }
21    }
22    public void disminuir(int n){
23        .....
24    }
25 }
26
27 ArrayList<AutoModificable> lista;
28 lista = new ArrayList<AutoModificable>();
29 lista.add(new Fraccion(6, 8));
30 lista.add(new AutoModificable());
31 lista.add(new Rectangulo(0, 5));
32 for(AutoModificable a: lista){
33     a.aumentar(10);
34 }
```

Se te pide realizar el modelo, identificar línea a línea las correcciones y errores, ¿puedes decir cual es la contenedora polimórfica?; ¿qué es lo que hace la instrucción de la línea 55?

public private
TODOS SOLO YO
protected?

✓m

Ejercicio 3. (30 ptos.) Indica que aseveraciones son verdaderas y porque.

1. Se tiene la interfaz: *Rastreable*

```
public interface Rastreable {  
    int calcularPosicion();  
    String direccion(int p);  
}
```

¿Cuál pedazo de código compilaría? (Seleccionar todas las que se aplican)

- a) interface Lozalizable ~~implements~~ Rastreable {} ?X
- b) abstract class Misil ~~extends~~ Rastreable {} X
- c) abstract class Avion ~~implements~~ Rastreable {} ✓
- d) abstract class Nave implements Rastreable {
 public int calcularPosicion() { return 0; } ✓
- e) class Class2 implements Rastreable {
 int calcularPosicion() { return 0; }
 String direccion(int p) { return "Posicion " + p; } X

2. Si se tiene la siguiente jerarquía de clases:

```
public interface A {  
    int metA1();  
    String metA2();  
}  
public class B implements A{  
    public int metA1(){...}  
    public String emtA2(){...}  
    public double metB1(){...}  
}
```

¿Cómo ejecutaría cada pedazo de código?

- a) A obj;
- b) B obj1;
- c) obj instanceof B
- d) obj1 = (B)obj
- e) obj = new B();
- f) obj1.metB1();

1 a) No compila, porque la palabra reservada para "heredar" de una interface a otra es EXTENDS

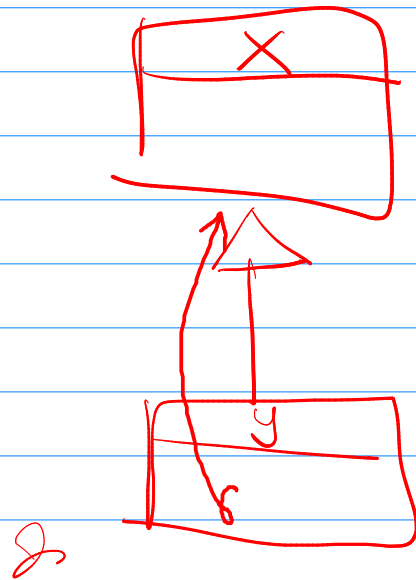
1 b) No compila, porque la palabra reservada para "heredar" (implementar) de una clase abstracta a una interface es IMPLEMENTS

1 c) Si compila, porque si usa la palabra reservada correcta para implementar de la interface, y APESAR de que Avion no implemente los metodos calcularPosicion y direccion, no esta obligado a hacerlo, ya que es una clase ABSTRACTA

1 d) Si compila, porque si usa la palabra reservada correcta y ademas, implementa el metodo calcularPosicion, pero no implementa el metodo direccion, eso no importa porque la clase es abstracta y NO esta obligafa a implementarla

1 e) NO compila, porque a pesar de implementar los dos metodos de la interface, NO les pone el modificador de acceso PUBLIC de manera especifica, y en este caso, es NECESARIO declarar que los metodos SON publicos

	Concrete	Abstract	Interface
Concrete	✓ extends	✓ extends	✓ implements
Abstract	✓ extends	extends	implements
Interface	X	X	extends



```

public class Y extends X {
}
  
```