C#

```
using System

public class CalculatorApp
{
  Protected static double number;

  public static void          (double n)
  { double count = 0;
    number = n;          //los espacios corresponden a la operacion a hacer
    if (n != 0)
      Console.WriteLine( number      count)
  }

  public static void ShowResults (double n)
  {
    suma (n)
    resta (n)
    division (n)
    multiplicacion (n)
  }
}

public class          : CalculatorApp  //Esto con cada clase que se
{                                      hereda
  Public static void ShowResults()
  {
    ShowResults (4);
    CalculatorApp. ShowResults ()
  }
}

public class MainClass
{
  public static void loader() //de nuevo lo blanco es por cada clase
  {
    Console.WriteLine ("los resultados dados por...");
    claseObjeto.ShowResults();
  }

  public static void main (string[] args)
  {
    loader();
  }
}
```

Java

```
public class calculatorApp{
  private static double number;

  public calculatorApp(){

  public static void      (double n){
      double count=0;
      number = n;          //en los espacios va la operacion a hacer
      if (n != 0){
      Sout.println(number   count);
      }

  public static void showResults(double n){
     resta(n);
     suma(n);
     multiplicacion(n);
     division(n);
  }
}

public class          extends calculatorApp{
   private calculatorApp c = new calculatorApp();

     public       (){
     }

     public static void showResults(){
        showResults(4)
        calculatoApp.showResults(36);
     }
}

public class main{
   public static void louder(){ " nuevamente aca se llama a cal. dos
      sout.println("los resultados dados por...");
      claseObjeto.ShowResults();

   public static void main(String[] args){
      louder();
   }
}
```

GO

```
package main
import "fmt"
type CalculatorApp struct{}

Func (CalculatorApp) operacion(n Float64){
    IF n != 0{
        fmt.PrintLn(n*0) //de nuevo, esto dependiendo la operacion.
    }
}

Func (CalculatorApp) ShowResults(n Float64){
    c := CalculatorApp{}
    c.suma(n)
    c.resta(n)
    c.division(n)
    c.multiplicacion(n)
}

type          struct{
    CalculatorApp    //en el espacio vacio va la clase creada.
}

Func (          ) ShowResults(){ // lo mismo.
    c := CalculatorApp{}
    c.ShowResults(4)
    c.ShowResults(36)

Func loadder(){
    Fmt.Println("los resultados operados por...") //aca pos la clase
    claseObjeto{}.ShowResults()

}

Func main(){
    loader()
}
```

JavaScript

```
class CalculatorApp {
    static suma(n){
        if (n !==0) console.log(n+0);
    }
    static resta(b){
        if (n !==0) console.log(0-b);
    }
    static division(n){
        if (n !==0) console.log(n/n);
    }
    static multiplicacion(n){
        if (n !==0) console.log(n*n);
    }
    static results(n){
        this.suma(n);
        this.resta(n)
            .division(n)
            .multiplicacion(n)
    }
}

class  extends CalculatorApp // lo blanco es la clase
    static showResults(){
        super.showResults(9);
        CalculatorApp.showresults(36);
    }
}

Function loader(){ // aca se muestran las clases
    console.log("los resultados dados por...")
    claseObjeto.showResults();
}

loader();
```