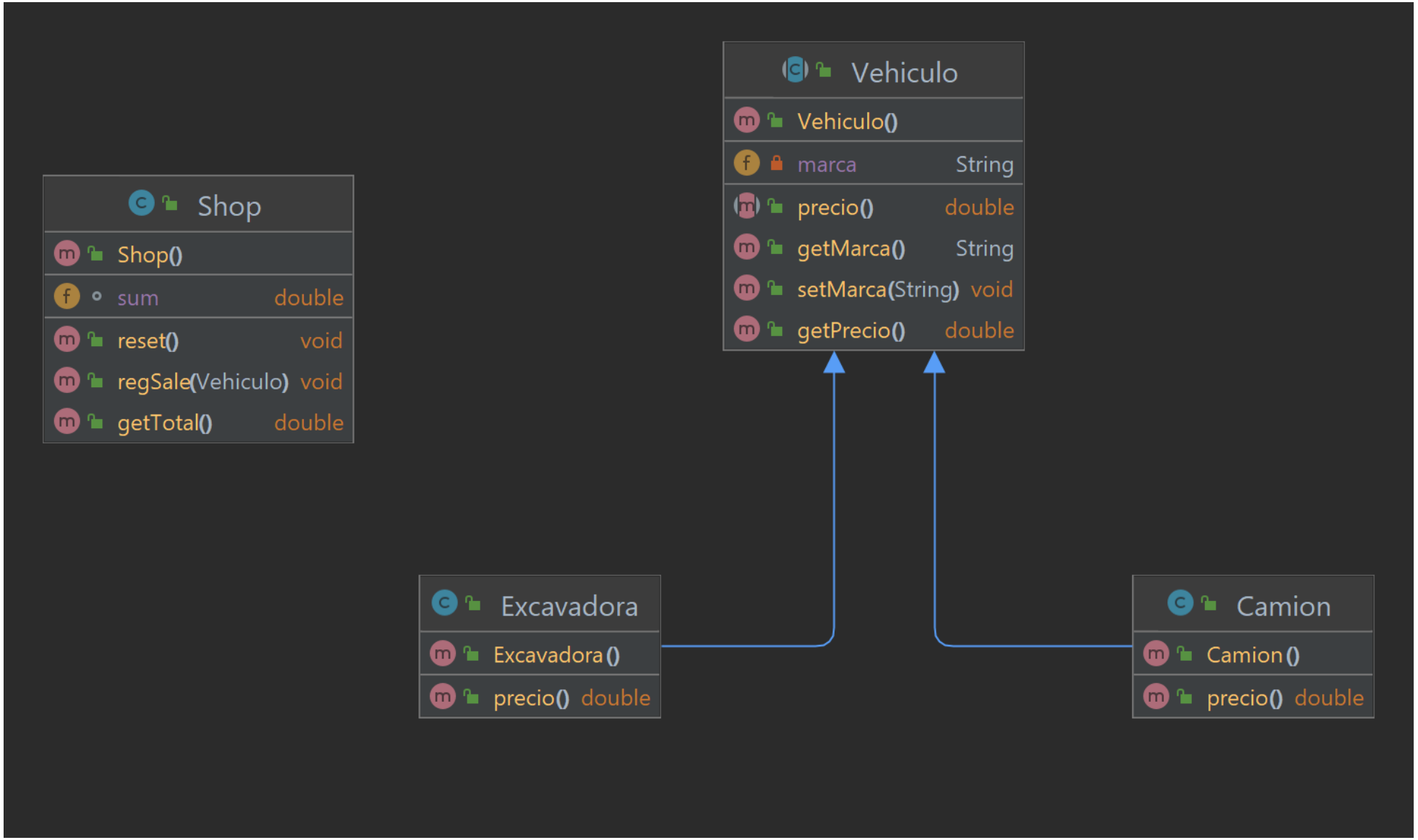
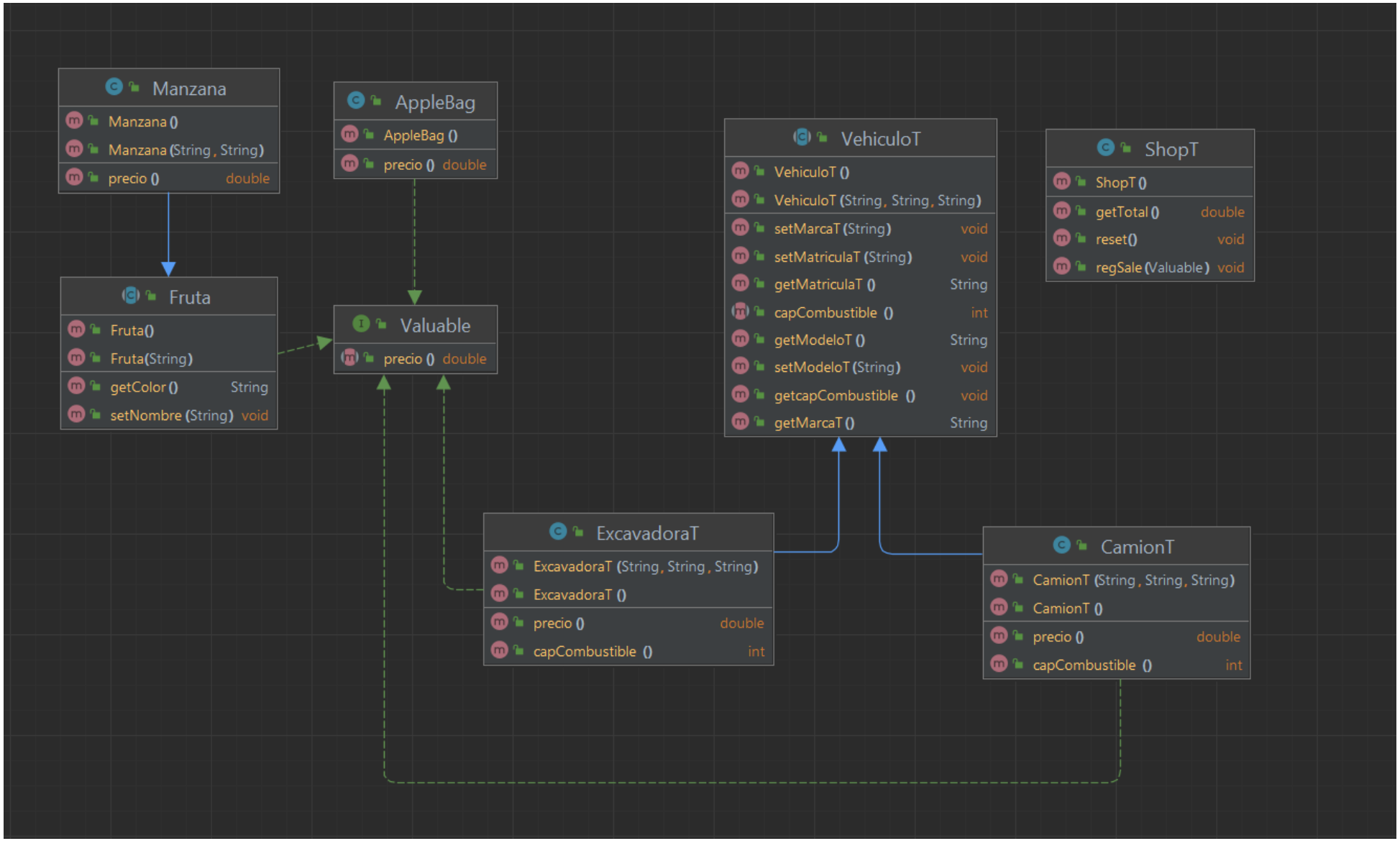
HERENCIA DE CLASE Y HERENCIA DE TIPO
Java , C++ , Smalltalk

Jose Miguel Cano Vilcapaza

Herencia de Clase en Java



Herencia de Tipo en Java



Herencia en Java

```
package herencia.tipoInterfaces;

public class ShopT {
    double sum = 0.0;
    public void reset () {
        sum = 0.0;
    }
    public double getTotal() {
        return sum;
    }
    public void regSale( Valuable
itemSold) {
       sum += itemSold.precio();
    }
}
```

```
package herencia.Clase;

public class Shop {
    double sum = 0.0;
    public void reset () {
        sum = 0.0;
    }
    public double getTotal() {
        return sum;
    }
        public void regSale(
Vehiculo itemSold) {
       sum += itemSold.precio();
    }
}
```

Herencia en C++

```
#include <iostream>
#include <string>
using namespace std;

class Vehiculo
{
private:
  string marca;
public:
  Vehiculo(){marca = "No
tiene";}
  Vehiculo(string
x):marca(x){}
  virtual string info()
{return marca;};
  virtual double precio() =
0;
};
```

```
class Camion : public Vehiculo
{
private:
  int carga;
public:
  Camion(){carga = 0;};
  Camion(int _carga,string
x):carga(_carga),Vehiculo(x){}
  string info() { return Vehiculo::info() +
to_string (carga);};
  double precio()
  {
    return 12.23;
  }
};
```

Herencia en C++

```
class Excabadora : public Vehiculo
{
private:
  string tamano;
public:
  Excabadora(){tamano = "No tiene";};
  Excabadora(string _tamano,string x)
       :tamano(_tamano),Vehiculo(x){}
  string info() {  return Vehiculo::info()
+ tamano;};
  double precio()
  {
    return 13.2;
  }
};
```

```
class Shop
{
private:
  double caja = 0.0;

public:
  void regSale(Vehiculo *obj) { caja +=
obj->precio(); };
  double getTotal() { return caja; };
};

int main()
{
  Shop a;
  a.regSale(new Excabadora);
  a.regSale(new Camion);
  std::cout << "caja : " << a.getTotal();
  return 0;
}
```

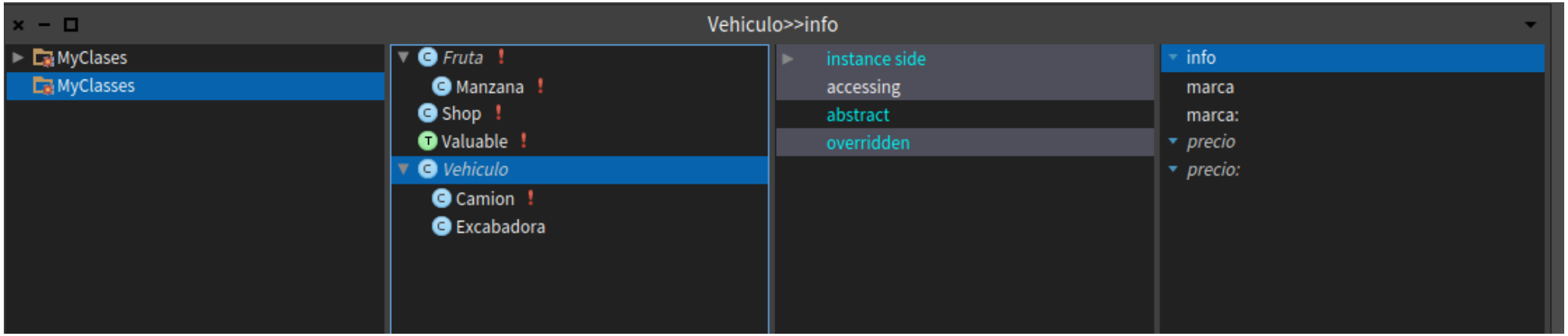
Herencia en C++

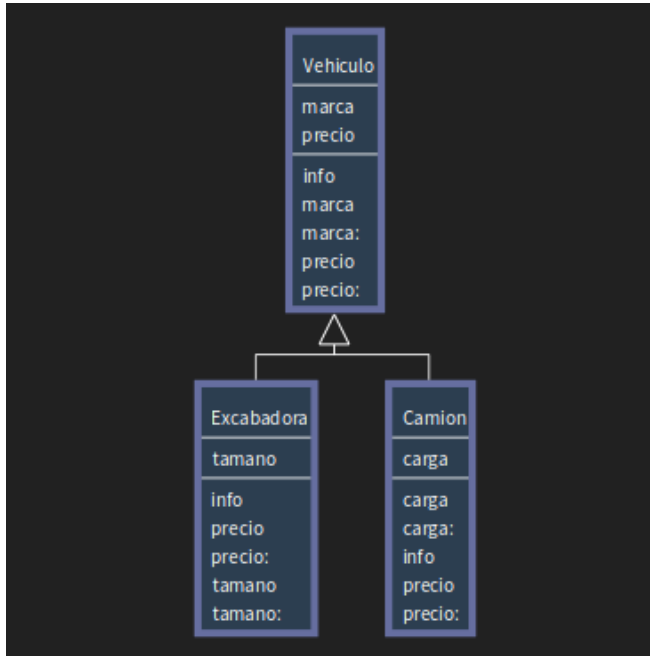
```
class Fruta
{
private:
  string color;
public:
  double precio() { return 3.2; }
};
```

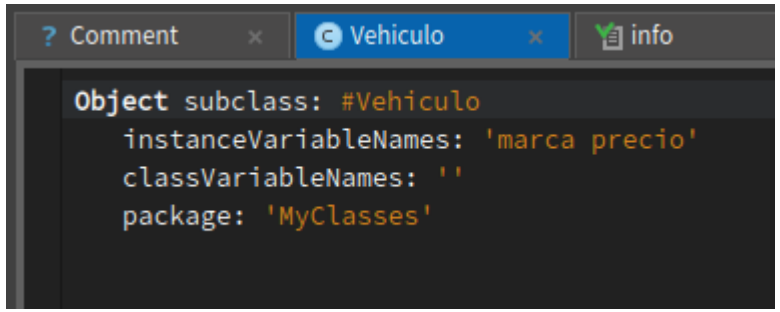
```
argument of type "Fruta *" is incompatible
with parameter of type "Vehiculo *"
```

```
int main()
{
  Shop a;
  a.regSale(new Excabadora);
  a.regSale(new Camion);
  a.regSale(new Fruta);
  std::cout << "caja : " << a.getTotal();
  return 0;
}
```

Herencia Clase en Smalltalk



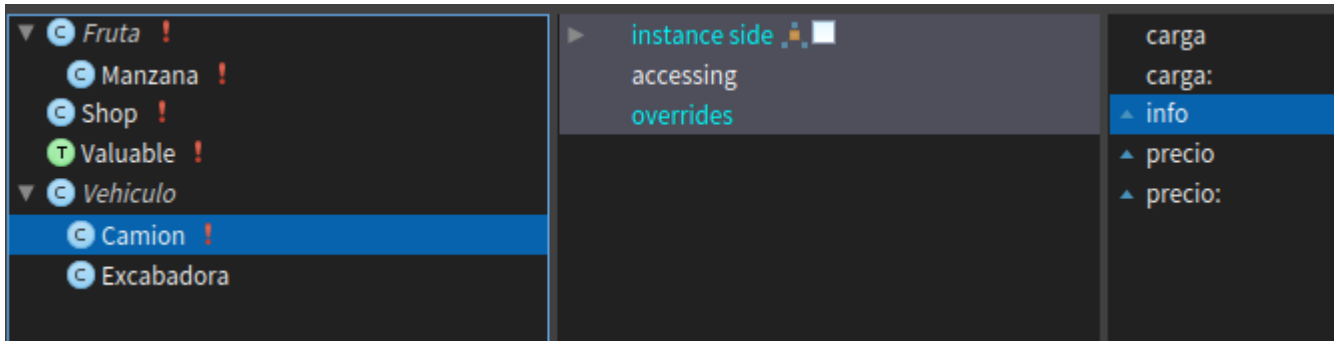




Herencia Clase en Smalltalk

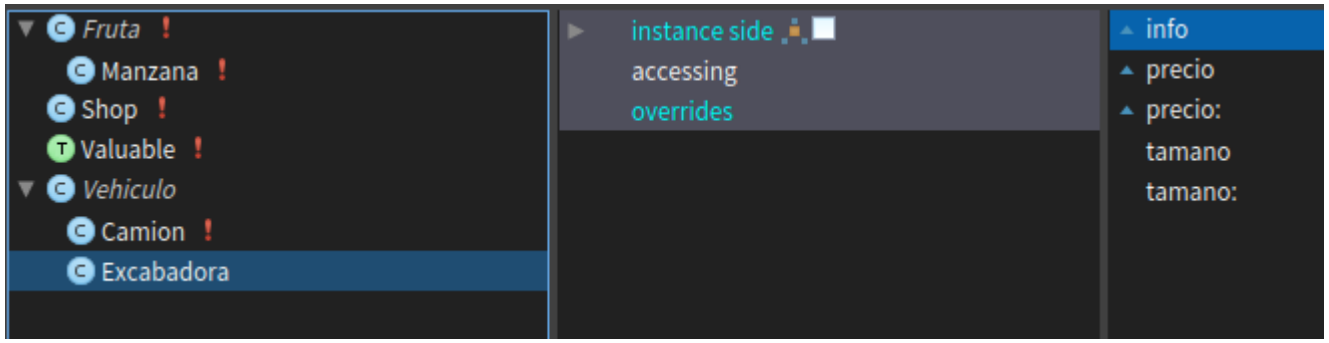
Comment | Camion | info

```
Vehiculo subclass: #Camion
    instanceVariableNames: 'carga'
    classVariableNames: ''
    package: 'MyClasses'
```



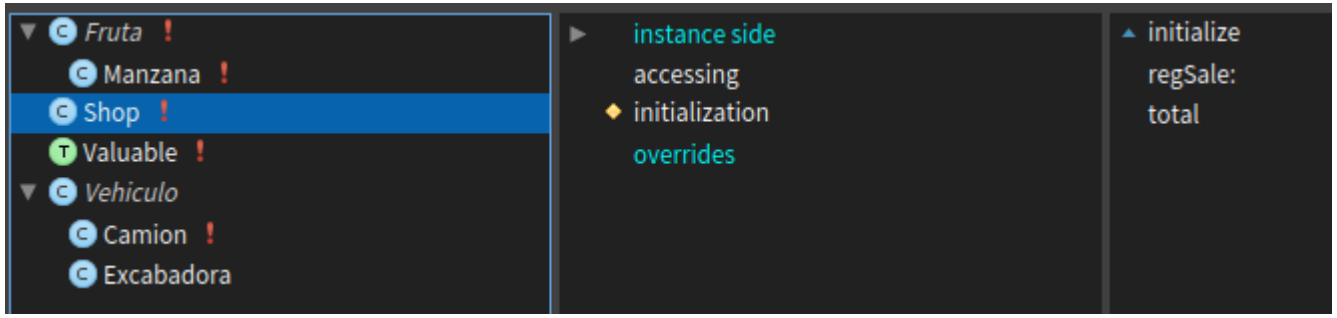
Comment | Excabadora | info

```
Vehiculo subclass: #Excabadora
    instanceVariableNames: 'tamano'
    classVariableNames: ''
    package: 'MyClasses'
```

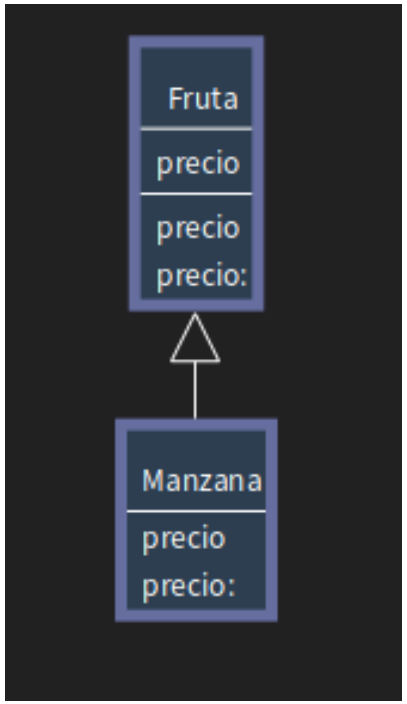


Comment | Shop | UML-Class

```
Object subclass: #Shop
    instanceVariableNames: 'caja'
    classVariableNames: ''
    package: 'MyClasses'
```



Herencia Tipo en Smalltalk



Comment | Fruta

```
Object subclass: #Fruta
    instanceVariableNames: 'precio'
    classVariableNames: ''
    package: 'MyClasses'
```

Comment | Manzana

```
Fruta subclass: #Manzana
    instanceVariableNames: ''
    classVariableNames: ''
    package: 'MyClasses'
```

Comment | Shop | regSale:

```
regSale: unObjeto
    caja := caja + unObjeto precio.
```

Playground

Do it | Publish | Bindings | Versions | Pages | a SmallInteger (81)

```
|s f c e c1|
s := Shop new.
e := Excabadora new.
e precio: 23.
c := Camion new.
c precio: 32.
f := Manzana new.
f precio: 3.

s regSale: e.
s regSale: c.
s regSale: f.
s regSale: (Excabadora precio: 23).
s total.
```

Integer | Raw | Breakpoints | Meta

key	value
decimal	81
hex	51
octal	121
binary	1010001