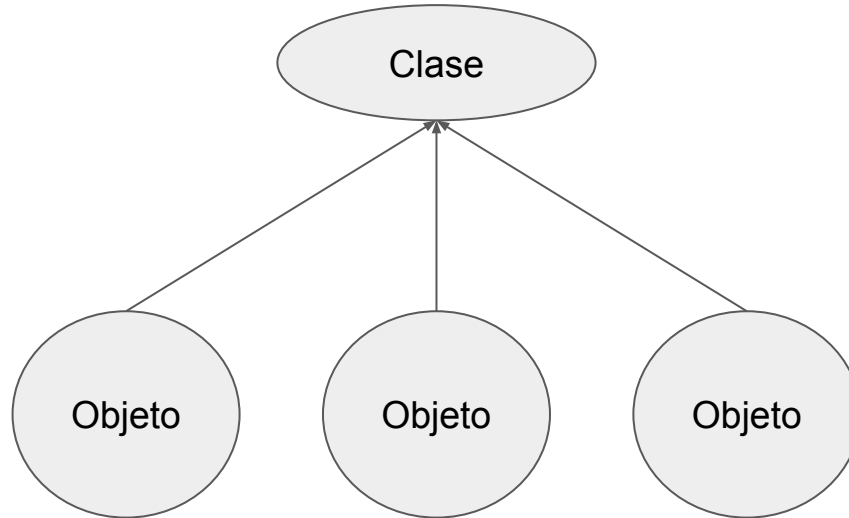


# **Programación Orientada a Objetos**

# Programación Orientada a Objetos

La programación orientada a objetos se basa en la programación de clases



# Programación Orientada a Objetos

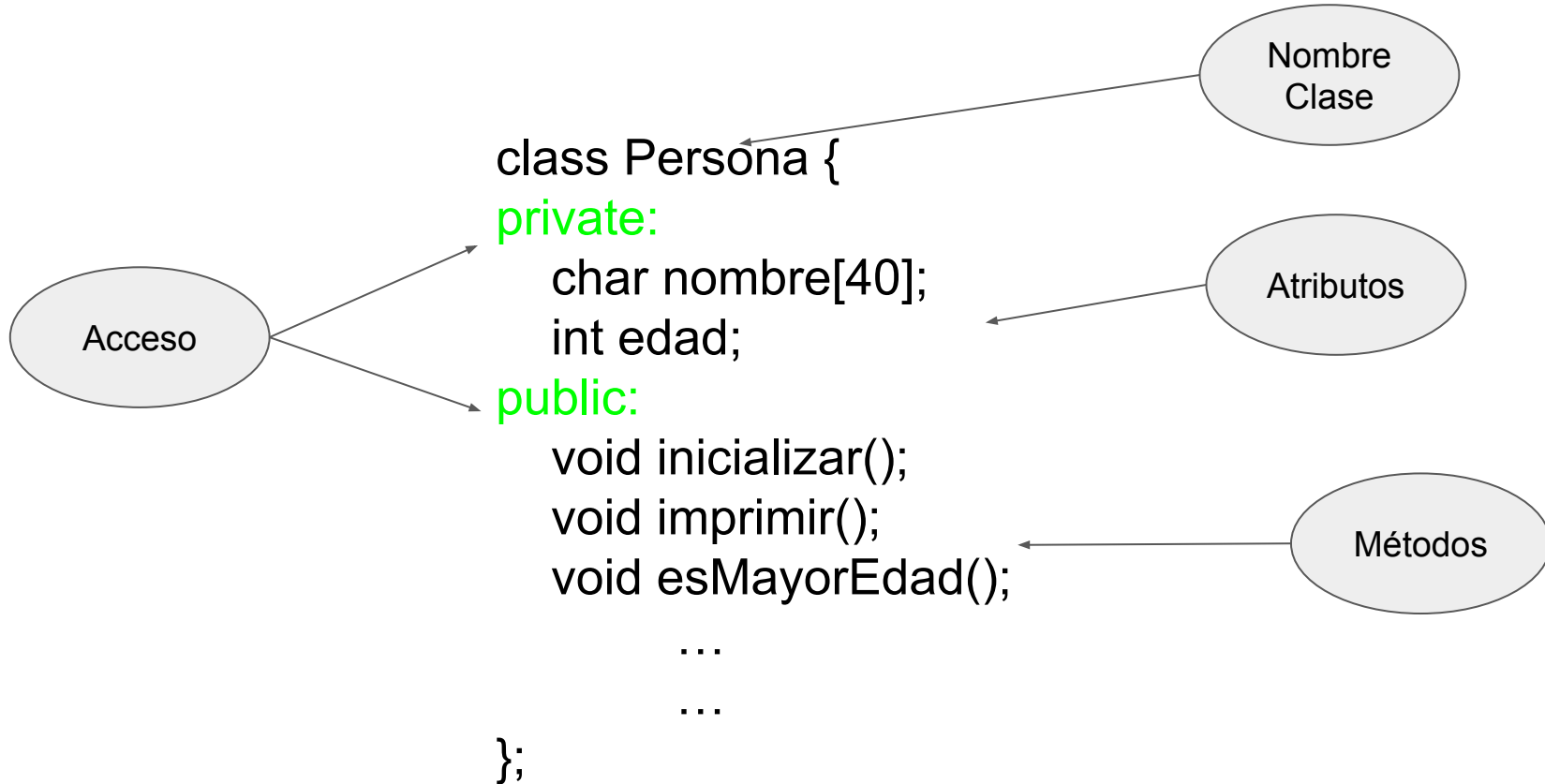
Clase define Estructura:

- Atributos
- Métodos

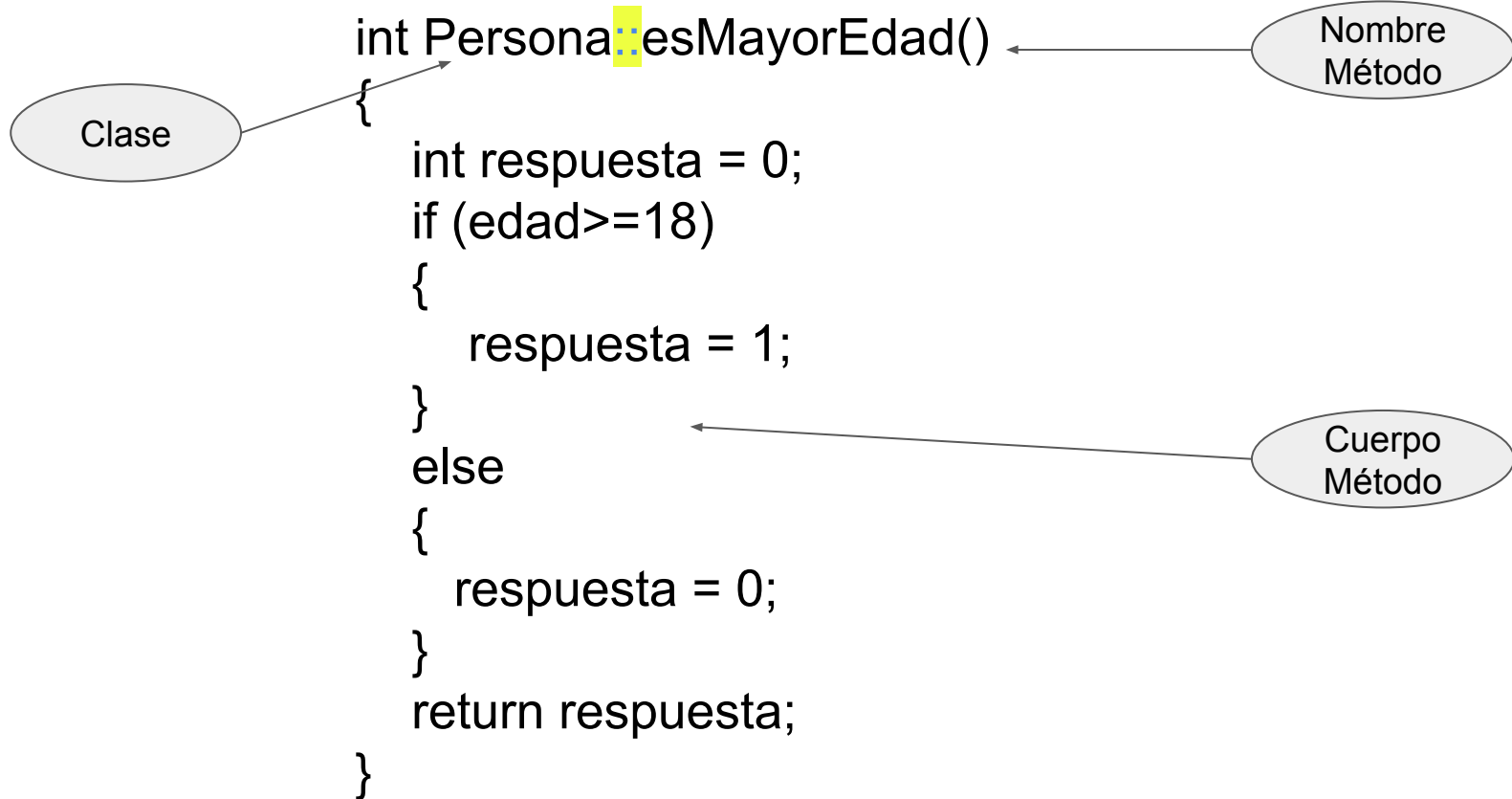
**Objeto:** Tiene valores propios. Comparte los mismos Métodos.

Crear un objeto de una clase, implica crear una instancia de la clase.

# Programación Orientada a Objetos



# Programación Orientada a Objetos



# Programación Orientada a Objetos

Generalmente se desarrollan métodos para:

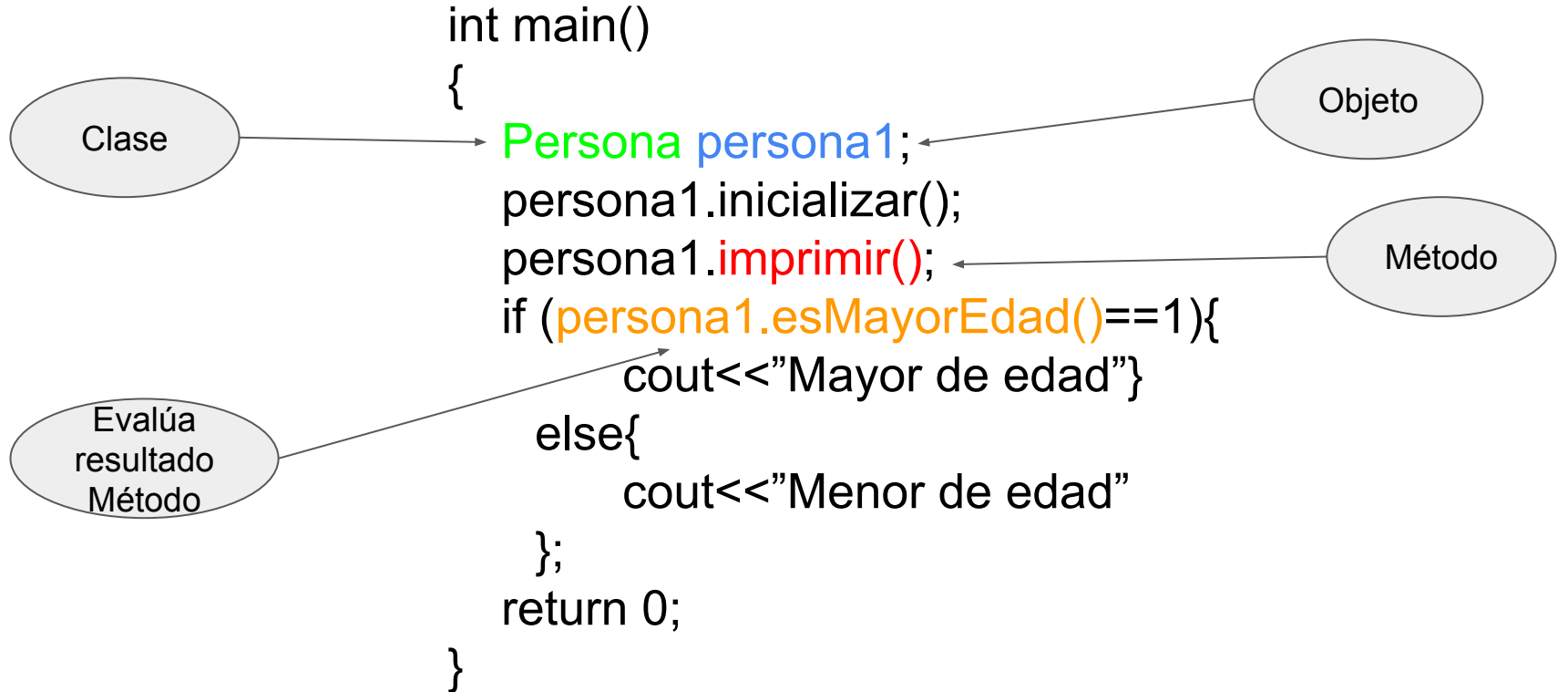
Inicializar atributos:

```
void setNombreAtributo( Parámetro );
```

Extraer valor de Atributo:

```
tipoDato getNombreAtributo( );
```

# Programación Orientada a Objetos



# **Programación Orientada a Objetos**

## **Ejemplo:**

**Crear un programa Orientado a Objetos que cree una clase que permita cargar el nombre y la edad de una persona.**

**La clase debe permitir mostrar los datos cargados y determinar si la persona es mayor de edad ( $\text{edad} \geq 18$ ).**



# Programación Orientada a Objetos

## Problemas propuestos

- Desarrollar un programa que cargue los lados de un triángulo e implemente los siguientes métodos: inicializar los atributos, imprimir el valor del lado mayor y otro método que muestre si es equilátero o no.
- Desarrollar una clase que represente un punto en el plano y tenga los siguientes métodos: cargar los valores de x e y, imprimir en que cuadrante se encuentra dicho punto (concepto matemático, primer cuadrante si x e y son positivas, si  $x < 0$  e  $y > 0$  segundo cuadrante, etc.)

# **Programación Orientada a Objetos**

- **Desarrollar una clase que represente un Cuadrado y tenga los siguientes métodos: cargar el valor de su lado, imprimir su perímetro y su superficie.**
- **Implementar la clase operaciones. Se deben cargar dos valores enteros, calcular su suma, resta, multiplicación y división, cada una en un método, imprimir dichos resultados.**