Sesión 04

PROGRAMACION ORIENTADA A OBJETOS

Curso Virtual

Fundamentos de POO, Funcional y Reactivo

Ing. Aristedes Novoa

anovoa@galaxy.edu.pe

GALAXY TRAINING

ÍNDICE

Análisis

Paradigmas de Programación

Es un paradigma de programación que usa objetos en sus interacciones.

Busca que nuestra forma de programar sea mas cercana a las cosas de la vida real.

Clase:

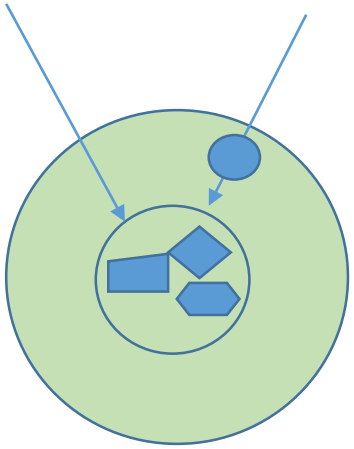
Es una plantilla que especifica atributos y comportamientos, es un prototipo para crear objetos. Los atributos se especifican mediante variables y el comportamiento, mediante métodos.

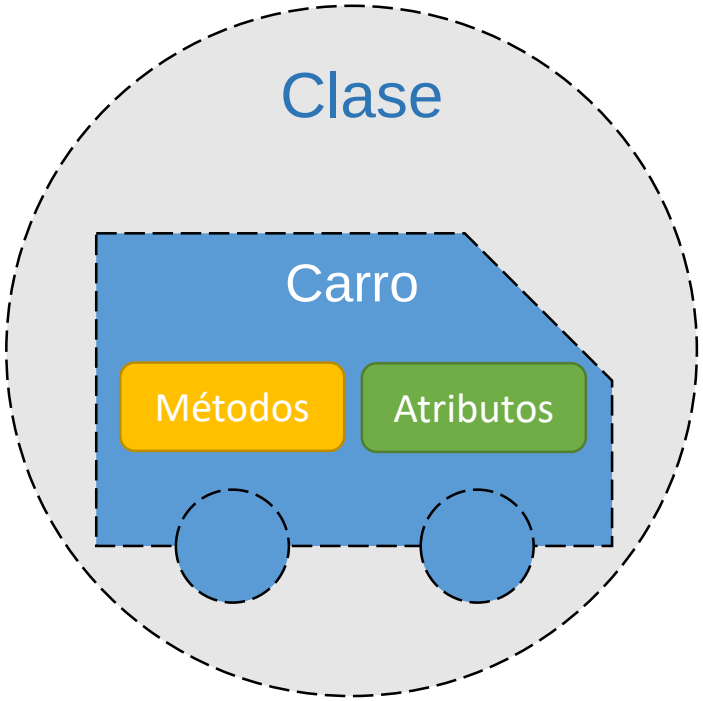
Objeto:

Es una abstracción de algún hecho o ente del mundo real, con atributos (características o propiedades) y métodos que emulan su comportamiento.

El objeto es una instancia de la clase.

Conceptos fundamentales de POO





Conceptos Fundamentales

Abstracción

Permite seleccionar las características relevantes dentro de un conjunto e identificar comportamientos comunes para modelar un conjunto de clases para la solución.

Encapsulamiento

Significa reunir todos los elementos pertenecientes a una misma entidad, al mismo nivel de abstracción.

Polimorfismo

Comportamientos diferentes asociados a objetos diferentes del mismo tipo instanciados de diferentes clases.

Herencia

Permite definir una clase (subclase) sobre la base de otra clase (superclase) heredando los atributos y métodos que sean accesibles. La herencia organiza y facilita el polimorfismo y el encapsulamiento.

Conceptos Fundamentales (Cont…)

Clase

Vehiculo

-color
-marca
-cantidadPuertas

+Vehiculo()
+acelerar()
+frenar()

objeto1

-color = “Rojo”
-marca = “Lamborghini”
-cantidadPuertas = 2

+Vehiculo()
+acelerar()
+frenar()

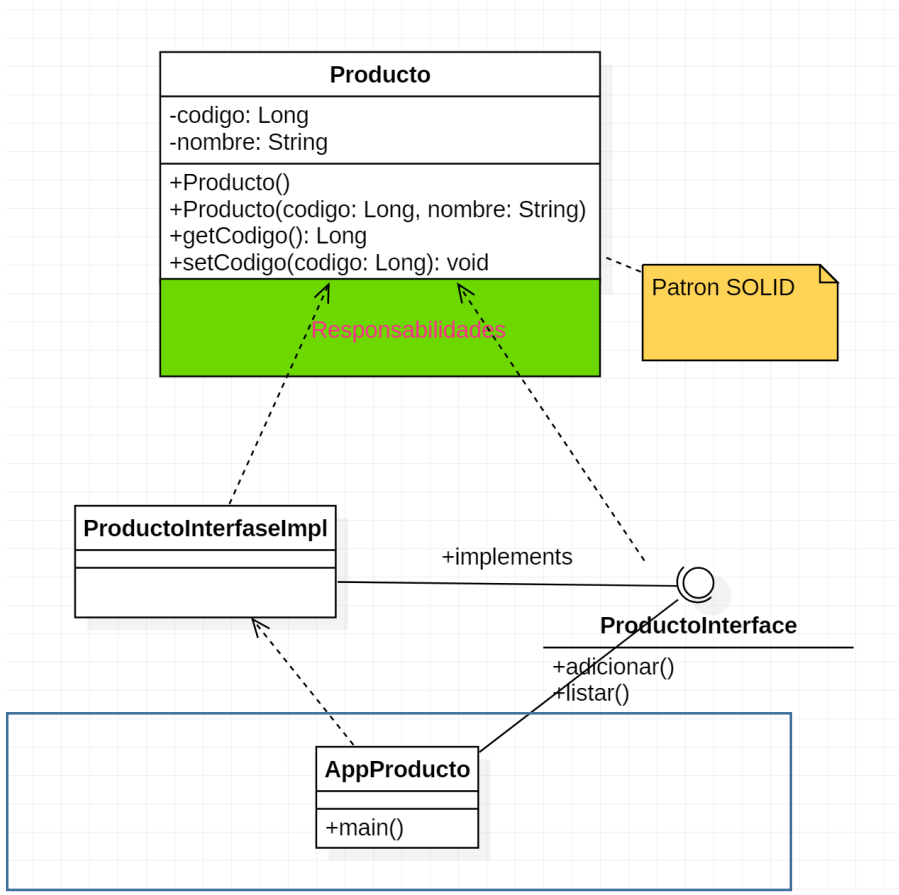
objeto2

-color = “Azul”
-marca = “Ferrari”
-cantidadPuertas = 2

+Vehiculo()
+acelerar()
+frenar()

Objetos

Representación gráfica



Nota: El rectángulo de color verde no es oficial de UML. Es un concepto que permite diseñar componentes especializados, simples y fáciles de mantener (mas detalles leer SOLID).

Ej.: En la clase Producto no debería existir un método para registrar su proveedor; de lo contrario debería de existir una clase Proveedor.

Modelamiento en UML

Creación de Clases y Objetos:

```java
public class Producto {

    // Atributos
    private String nombre;

    // Metodos

    // Constructor
    public Producto(){

    }

    // Getters/Setters
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getNombre() {
        return nombre;
    }

}
```

```java
public static void main(String[] args) {

    Producto producto = new Producto();
    producto.setNombre("Java desde Cero");

}
```

```java
//Interface     coleccion     Clase
List<Producto> productos= new ArrayList<Producto>();

//Calse  objeto        Constructor
Producto producto= new Producto();

producto.setCodigo(100L);
producto.setNombre("Impresora EPSON L555");
producto.setPrecioCompra(750);
producto.setStock(4);
productos.add(producto);
```

Implementando en Java

https://docs.oracle.com/javase/tutorial/java/concepts/index.html

Mas información


GALAXY
TRAINING