



Programación II

Clase 02

Programación Orientada
a Objetos

Introducción a P00

- Paradigma
- Características

Clases

- Definición
- Tipos
- Constructores

Objetos

- Real
- Abstracto
- Composición
- Métodos y Mensajes



01.

Introducción a P00

Concepto

- **Práctica de desarrollo de software**

Adopción popular.

Plantea solución a un problema de una determinada manera.

- **Clasificación**

Lenguaje de programación.

Programación orientada a objetos

- **Datos**

Información relevante de un modelo conceptual.

- **Estados e Interacción**

Cambios en los datos producto de la interacción entre los componentes.

Características

- **Abstracción**
Características principales.
- **Encapsulación**
Visibilizar datos no implementaciones.
- **Herencia**
Jerarquía.
- **Polimorfismo**
Interacción entre objetos.



02.

Objetos

¿Qué es un objeto?

- **Objeto real**
Interactivo en el mundo real.
- **Objeto abstracto**
Réplica.
Modelo con la información “interesante”
para el sistema.
- **Entidad mínima y básica**

ABSTRACCIÓN

Objetos



ABSTRACCIÓN



PASAJEROS
CONSUMO
BAUL



VELOCIDAD
ACELERACIÓN
POTENCIA



RODADO
MARCA
TRACCIÓN

Composición

- **Estructura**
Atributos.
Información, datos y estados.
- **Comportamiento**
Métodos.
Definiciones de cómo interactuar.

ESTRUCTURA

MARCA: "CHEVROLET"

MODELO: "CRUZE"

ASIENTOS: 4

CON_BAUL: TRUE

KM: 128024

EN_VIAJE: FALSE



AUTO_1

INTERFAZ O PROTOCOLO

COMPORTAMIENTO

INICIAR_VIAJE()

TERMINAR_VIAJE()

KM_RECORRIDOS()

TIENE_BAUL()

Interacciones entre objetos

- **Mensaje [usuario]**
Solicitud de acción.
Tengo que conocer el objeto. Referencia.
- **Método [objeto]**
Definición dónde está las instrucciones a ejecutar.

Interacciones entre objetos

- **Interfaz o Protocolo [usuario]**
Define qué acciones pueden realizarse con dicho objeto.
Público.
- **Implementación [objeto]**
Cómo se realiza la acción.
Ejecución del método.
Reservado. Evita “acople”.

Objetos

AUTO_1



EN_VIAJE: FALSE

EN_VIAJE: TRUE

IMPLEMENTACIÓN

INICIAR_VIAJE()

CONOCE A **AUTO_1** Y SU
INTERFAZ PUBLICA

AUTO_1. INICIAR_VIAJE()



ENVÍA MENSAJE

INICIAR_VIAJE()

EJECUTA MÉTODO



PASAJERO_1

Colaboración

- **Objeto parte del estado de otro**
Forma parte del comportamiento.
Se le delegan responsabilidades.

Objetos

ESTRUCTURA

NOMBRE: "NOMBRE"
APELLIDO: "APELLIDO"
ID: 123456789



CHOFER_1



ESTRUCTURA

MARCA: "CHEVROLET"
MODELO: "CRUZE"
ASIENTOS: 4
BAUL: TRUE
KM: 128024
EN_VIAJE: FALSE
CHOFER: CHOFER_1



AUTO_1



03.

Classes

Plantillas de Objetos

- **Definen Estructura**
Qué atributos tienen.
- **Definen Comportamiento**
Qué mensajes pueden responder.
Qué acciones pueden realizar.
- **Definen Creación**
Qué parámetros necesita.
De qué **tipo** es el objeto.

Tipos de Clases

- **Estática**

No puede ser instanciada.

Acceso a los atributos de manera directa [nombre].

- **Abstracta**

No puede ser instanciada y sus atributos no son accesibles.

- **No Estática**

Se instancian objetos.

Puede contener atributos estáticos.

Ciclo de vida del objeto

- **Creación**

La palabra reservada **new** instancia un objeto.
Siempre hay un constructor por defecto.
Se crea en memoria **Heap**.

- **Destrucción**

No puede hacerse de manera explícita.
Se encarga el garbage collector.

NO EN TODOS LOS
LENGUAJES

OBJETOS HUÉRFANOS
MEMORIA COMPROMETIDA

Constructores

POR DEFECTO

```
1 class Auto{  
2     string marca;  
3     string modelo;  
4     int asientos;  
5     bool conBaul;  
6     bool enViaje;  
7 }
```

INICIALIZA CON ATRIBUTOS EN **NULL**,
CERO O FALSE

Constructores

DECLARADO

```
1 class Auto{
2     string marca;
3     string modelo;
4     int asientos;
5     bool conBaul;
6     bool enViaje;
7
8     public class Auto(bool viajando){
9         this.enViaje = viajando;
10    }
11 }
```

ENVIAJE TOMA EL ARGUMENTO
VIAJANDO

INICIALIZA RESTO DE ATRIBUTOS EN
NULL, CERO O FALSE

Constructores

ESTÁTICO

```
1  class Auto{  
2      string marca;  
3      string modelo;  
4      static int asientos;  
5      bool conBaul;  
6      bool enViaje;  
7  
8      static Auto(){  
9          asientos = 4;  
10     }  
11 }
```

ASIENTOS TOMA EL VALOR 4

INICIALIZA RESTO DE ATRIBUTOS EN
NULL, CERO O FALSE

Accesibilidad de atributos

- **public**
Accesible para todas las clases.
- **private**
Accesible dentro de la misma clase.
- **protected**
Accesible dentro de la misma clase y subclases.^{HERENCIA}