



TEMA 1

Introducción a la Programación Orientada a Objetos

V1.1

Manuel Pereira González

Agenda



- **Introducción**
- Objetos y Clases
- Modelo de Objetos
 - Abstracción
 - Encapsulamiento
 - Herencia
 - Polimorfismo
- Relaciones entre Objetos
 - Asociación
 - Agregación
 - Generalización / Especialización
- Resumen



Introducción: Paradigma POO



- POO surge como necesidad de Reusabilidad de los Componentes
- Función -> Objeto (Objeto=Concepto)
- Ventajas
 - Más parecido a la **forma de pensar** de las personas
 - Desarrollo **rápido**
 - Mejora en **calidad y legibilidad del código**
 - Mejora en **mantenibilidad**. Definición clara de interfaces, **Niveles de Abstracción**
 - Reusabilidad. Reutilización de código.
Encapsulamiento

Introducción: Historia POO



- Simula: 1967 -> Introduce el concepto de Objeto
- Smalltalk: 1980 -> POO puro, desarrollado en Xerox Palo Alto Research Center Learning Research Group
- C++: 1983-1990 -> por Bjarne Stroustrup (C con clases)
- Object Pascal: 1986 -> Esqueleto simplificado de lenguaje POO por desarrolladores de Apple Computer y Niklaus Wirth
- LISP/CLOS: 1988 -> Diseñado por un comité presidido por Daniel Bobrow a partir de la ACM Lisp and Functional Programming Conference de 1986
- Java: 1995, 1997, 1998 -> Sun Microsystems, estándar de facto en el desarrollo de aplicaciones actuales
- C# (C Sharp)-> Microsoft (Scott Wiltamuth y Anders Hejlsberg). Plataforma ".Net"

Agenda



- Introducción
- **Objetos y Clases**
- Modelo de Objetos
 - Abstracción
 - Encapsulamiento
 - Herencia
 - Polimorfismo
- Relaciones entre Objetos
 - Asociación
 - Agregación
 - Generalización / Especialización
- Resumen



Objetos y Clases



- Objeto = Concepto
- Clase -> Tipo de Objetos
- Objeto -> Instancia (ejemplar) de una Clase

Clase	Objeto
Persona	José Menendez López
Coche	Citroen Xsara matrícula 4545BBS
Perro	"popi", el perro de José
Montaña	Mont Blanc
Teléfono Móvil	El móvil de José: Nokia 3310 rojo

Objetos y Clases

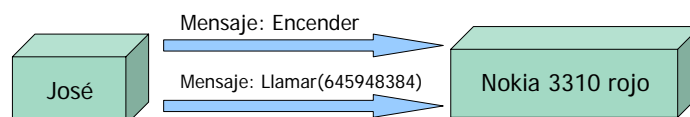


- Clase: Patrón que define atributos y métodos comunes a los objetos de cierto tipo.
- Componentes de un Objeto
 - Operaciones (métodos)
 - Estado (atributos)
- Ej:
 - Clase: Teléfono Móvil
 - Atributos: Marca, Modelo, Color, Dimensiones, ...
 - Operaciones: Encender, Apagar, Llamar,...
 - Objeto: Teléfono Móvil de José
 - Atributos: Marca=Nokia, Modelo=3310, Color=Rojo, ...
 - Operaciones: Encender, Apagar, Llamar,...

Objetos y Clases



- Comunicación entre Objetos: Mensajes
- Componentes de un mensaje
 - Objeto destinatario
 - Nombre del método
 - Parámetros (si existen)



Agenda



- Introducción
- Objetos y Clases
- **Modelo de Objetos**
 - Abstracción
 - Encapsulamiento
 - Herencia
 - Polimorfismo
- Relaciones entre Objetos
 - Asociación
 - Agregación
 - Generalización / Especialización
- Resumen



Modelo de Objetos: Abstracción



- Abstracción
 - Conocimiento humano basado en niveles de abstracción
 - Átomo -> Molécula -> Persona -> Planeta -> Galaxia
 - “Abstraerse” de la complejidad interna para poder construir sobre un nivel de abstracción estable
 - Extraer la información fundamental de una entidad ignorando los detalles accidentales
 - Definir una clase = Realizar una abstracción de un concepto. Ej: Clase Teléfono Móvil
 - Atributos: Marca, Modelo, Color, Dimensiones, ...
 - Operaciones: Encender, Apagar, Llamar,...

Modelo de Objetos: Encapsulamiento



- Encapsulamiento
 - “Caja Negra”: Métodos y Atributos, da igual la implementación específica
 - Ignorar los detalles internos
 - Definir una **Interfaz**: Especificar una abstracción en forma de métodos y atributos
 - Sólo es necesario conocer la interfaz de una clase, no es necesario conocer los detalles de su funcionamiento interno
 - Ventaja del Encapsulamiento: Piezas intercambiables mientras se siga cumpliendo la interfaz -> Reutilización de Código

Modelo de Objetos: Herencia



- Herencia
 - Una clase (subclase ó clase hija) **hereda** la estructura de otra (superclase o clase padre), y **concretiza** ciertos aspectos de ésta.
 - La clase hija comprende un **subconjunto** de los objetos de la clase padre.
 - Ej:
 - Clase: **Animal**
 - **Ser vivo que nace, crece, se reproduce y muere**
 - Subclase: **Perro** (subconjunto de animales)
 - **Ser vivo que nace, crece, se reproduce, muere, tiene cuatro patas, dos ojos, ladra, ...**

Modelo de Objetos: Polimorfismo



- Polimorfismo
 - Tratar objeto de una clase más general independientemente de que sea de una clase concreta
 - Ej: Si tengo un perro, un elefante y una jirafa, puedo tratarlos a todos como animales
 - Ej: Polígono -> Método para calcular el perímetro
 - Círculo: $2 * \pi * R$
 - Rectángulo: $Base * Altura$
 - Triángulo: $Base * Altura / 2$
 - Lista de polígonos, cada uno sabe calcular su área pero se tratan de igual manera sin saber de qué tipo de polígono concreto se trata

Agenda



- Introducción
- Objetos y Clases
- Modelo de Objetos
 - Abstracción
 - Encapsulamiento
 - Herencia
 - Polimorfismo
- **Relaciones entre Objetos**
 - Asociación
 - Agregación
 - Generalización / Especialización
- Resumen



Relaciones entre objetos: Asociación



- Asociación
 - Describe vínculos entre objetos
 - Multiplicidad:
 - Uno a Muchos: Ej: Una ciudad pertenece a un país, un país tiene muchas ciudades.
 - Fija: Un perro tiene cuatro patas
 - Variable: Una casa tiene una o más habitaciones
 - Muchos a Muchos: Ej: Una empresa tiene varios trabajadores, una persona puede trabajar en varias empresas a la vez

Relaciones entre objetos: Agregación

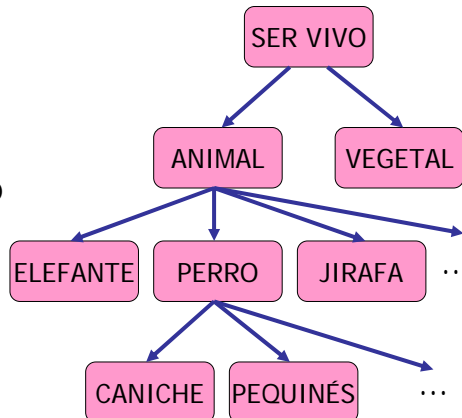


- Agregación
 - Relación del tipo Todo / Parte
 - Ej:
 - La Tierra se compone de 5 continentes
 - Un continente se compone de varios países
 - Un país se compone de varias ciudades
 - Elección Agregación / Asociación: En manos del analista

Relaciones entre objetos: Generalización/Especialización



- Generalización / Especialización
- Jerarquía definida a través de la **Herencia**:
 - Un CANICHE **es un** PERRO
 - Una PERRO **es un** ANIMAL
 - Un ANIMAL **es un** SER VIVO



Agenda



- Introducción
- Objetos y Clases
- Modelo de Objetos
 - Abstracción
 - Encapsulamiento
 - Jerarquía
 - Polimorfismo
- Relaciones entre Objetos
 - Asociación
 - Agregación
 - Generalización / Especialización
- **Resumen**



Resumen: Para más información



- <http://www.lsi.uned.es/lp/IntroPOO.pdf>
- <http://pisuerga.inf.ubu.es/lsi/Invest/Java/Tuto/Index.htm>
- http://eees.ii.uam.es/alfonso/web_poo_04/teoria/material/material_temas.html
- <http://www.eps.uam.es/%7Ecastells/docencia/>
- <http://www.monografias.com/trabajos15/lenguajes-programacion/lenguajes-programacion.shtml>
- http://ieee.udistrital.edu.co/concurso/programacion_orientada_objetos/poo/historia.html