

Programación Orientada a Objetos

Teoría: **Begoña Moros Valle**

Prácticas: **Marcos Menárguez Tortosa**

Contenido

- Presentación contenido teórico
- Presentación contenido práctico
- Introducción a la Programación Orientada a Objetos

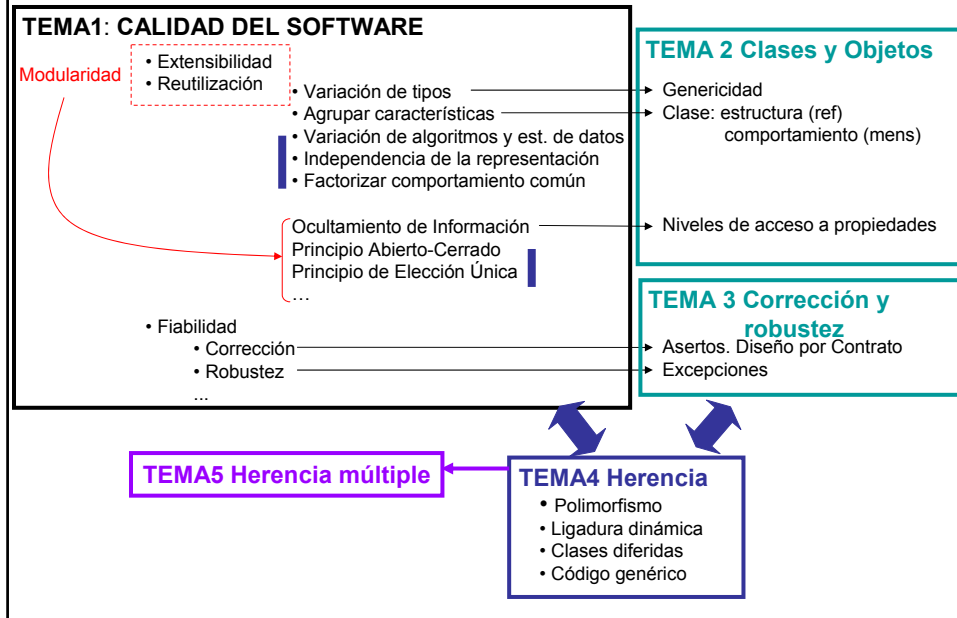
TEORÍA

Programación Orientada a Objetos
Curso 2005 / 2006

Objetivos

- Describir los conceptos que caracterizan el modelo OO
- Valorar en qué medida las técnicas OO favorecen la calidad del software
- Contrastar cómo diferentes lenguajes (Eiffel, Java, C++, C#) reflejan los conceptos del paradigma OO

Programa de teoría



Evaluación

- Exámenes:
 - 2 diciembre
 - 14 febrero
 - 30 junio
 - 6 septiembre
- $NOTA = Teoría * 0'6 + Prácticas * 0'4$

Bibliografía básica

	TEMA 1	TEMA 2	TEMA 3	TEMA 4	TEMA 5
		X	X	X	X
		X		X	X
 		X	X	X	X
	X	X	X	X	X
		X	X	X	X

Horario de atención alumnos Teoría

Facultad de Informática (3ª planta): E-20

- Martes: 18:30 a 20:30
- Miércoles: 18:30 a 20:30
- Jueves: 18:30 a 20:30

PRÁCTICAS

Programación Orientada a Objetos
Curso 2005 / 2006

Objetivos

- Aprender el lenguaje de programación Java
- Dominar las librerías básicas del lenguaje
- Practicar los conceptos de orientación a objetos en Java
- Manejar un entorno de programación Java

Programa de Prácticas

- Introducción a Java
- Entorno de programación
- Sintaxis del lenguaje
- Clases y objetos
- Cadenas y Entrada/Salida
- Herencia, clases abstractas
- Genericidad e Interfaces
- Colecciones e Iteradores. Clases Anidadas.
- Hilos

Metodología

- Seminarios semanales.
- Prácticas laboratorio abierto.
- Ejercicios semanales.
- Entrevistas de control y revisión.
- Dos correcciones generales de prácticas.
- Defensa de prácticas.

Planificación 1/2

- **S2** (3-October):
 - Comienzo de los seminarios
 - Práctica laboratorio cerrado: entorno de programación
- De **S7** (7-N) a **S9** (21-N):
 - **Entrevistas de control**
- **S10**: viernes 2 de diciembre
 - **Primera entrega de prácticas**
- **S12 – S13** (del 12 D al 21 D)
 - **Entrevistas de revisión**

Planificación 2/2

- **S15**: 16 de Enero
 - **Segunda entrega de prácticas**
- Del 30 Enero al 10 de Febrero
 - **Entrevistas de revisión**
- 20 de Febrero:
 - **Entrega final corregida**
- Del 20, 21 y 22 de Febrero:
 - **Defensa de prácticas**

Práctica

- Desarrollo de un **videojuego**
- Grupos de 1 ó 2 alumnos
- Propuesta de otra práctica para Junio/Septiembre
- Entrega de prácticas en SUMA y en papel
- Entorno: **JBuilder X**

Evaluación

- **Penalización:**
 - Por faltar entrega semanal:- 5 % por semana y práctica
 - No hacer las correcciones de las entrevistas.
 - ***No realizar la primera entrega general***
- **Defensa de prácticas:**
 - Determinar trabajo cada componente del grupo
- **No hay convocatoria de diciembre:**
 - Febrero / Junio o Septiembre

Horario

- **Seminario:**
 - Lunes de 12:00 – 13:00
- **Laboratorio:**
 - Grupo 1: Martes de 18:30 – 20:30
 - Grupo 2: Miércoles de 18:30 – 20:30
- **Tutorías, sótano Facultad de Informática:**
 - Lunes 18:30 – 20:30
 - Martes 16:30 – 18:30
 - Jueves 16:30 – 18:30

Bibliografía Básica

- Gratuitos (versiones en PDF):
 - **Core Java 2**, volumen 1 y 2 (inglés)
 - 4 Manuales de *JBuilder X* en (español)
 - *Thinking in Java, 3ª edición* (inglés), en español en la biblioteca
 - *Effective Java. Programming Language Guide.*
- Biblioteca:
 - *El lenguaje de programación Java*, de J. Gosling
 - *Java 1.2 al descubierto*, de J. Jaworski

INTRODUCCIÓN

Programación Orientada a Objetos
Curso 2005 / 2006

¿Qué significa **Orientación a Objetos**?

- El software se organiza como una **colección de objetos** que contienen tanto **estructura** como **comportamiento**.

¿Qué es el **desarrollo OO**?

- Una nueva forma de pensar acerca del software basándose en abstracciones que **existen** en el **mundo real**.



Bombilla
color
encender() apagar()

Paradigma de programación

Colección de **conceptos** que **guían** el proceso de **construcción** de un programa, determinando su **estructura**. Estos conceptos controlan la forma en que pensamos y formulamos los programas.

- Un **lenguaje de programación** refleja un paradigma.

PARADIGMA

LENGUAJE

- | | |
|-----------------------|---|
| • Imperativo | C, Pascal, Cobol, ... |
| • Funcional | Lisp, Hope, Miranda, ... |
| • Lógico | Prolog, Parlog, ... |
| • Orientado a Objetos | Smalltalk, C++ , Eiffel , Java , ... |

Marco Conceptual del paradigma OO

Programación Orientada a Objetos

Métodos de Análisis y Diseño OO

Modelo de Objetos

*Abstracción
Encapsulación
Modularidad
Herencia
Polimorfismo*

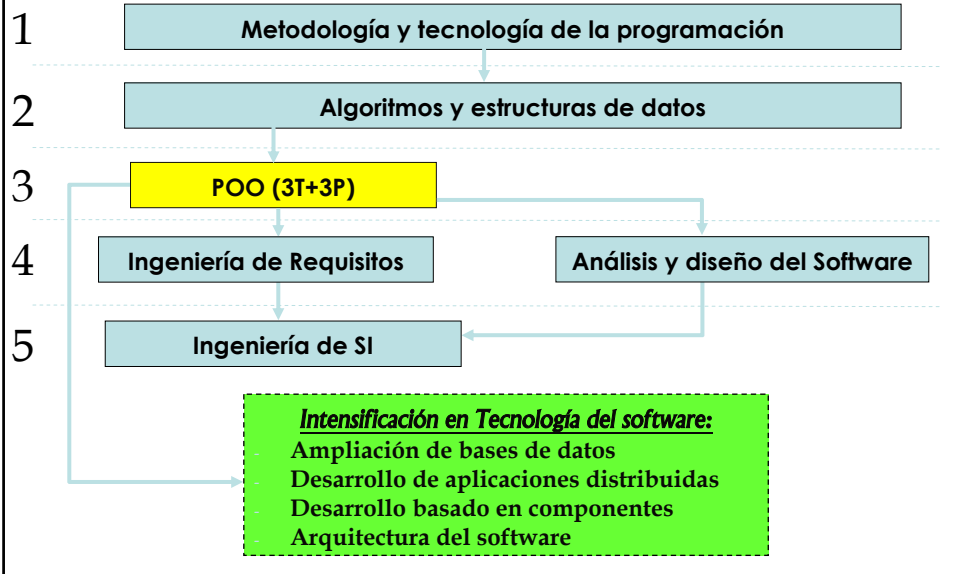
El futuro del paradigma OO

Bases de Datos OO

Desarrollo basado en componentes (CORBA COM, EJB, COM)

Tecnología de objetos distribuidos (CORBA, RMI, Plataforma .NET)

Titulación Ingeniero en Informática (Plan 2002)



Problemas en la creación del software

- A finales de los 60 se acuñó el término **crisis del software**:
 - Los proyectos no cumplían los plazos y presupuestos.
- **Dificultades inherentes a la naturaleza del software**:
 - **Complejidad**
 - dificultad de enumerar todos los estados posibles del programa
 - dificultad de comunicación entre los miembros del equipo
 - **Conforme** a otras interfaces
 - Sujeto a continuos **cambios**
 - El software es **intangibile**, no tiene representación gráfica
- Especificación de **requisitos** completa, precisa y correcta.

Problemas en la creación del software

“La construcción de software siempre será una tarea difícil. No hay bala de plata”

[Brooks, 1987]

- **Soluciones:**

- Reutilizar componentes (Comprar y no construir)
- Prototipado
- Buenos programadores/diseñadores

Historia de los LPOO

- **1964- SIMULA (Dahl y Nygaard):**

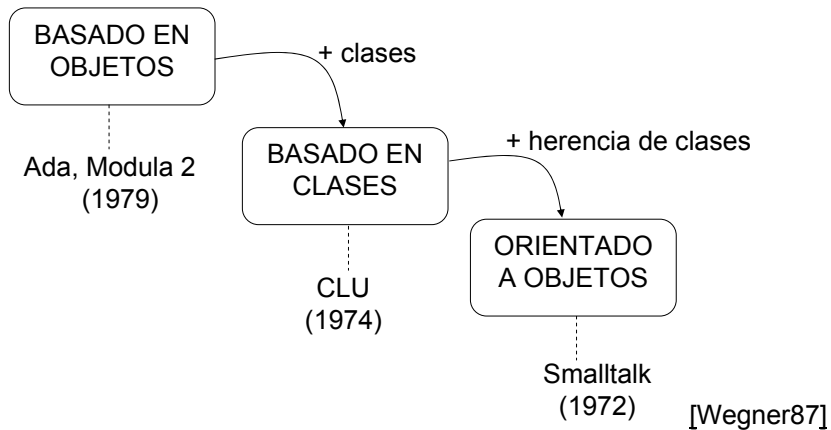
- comienzo de la OO
- Se identificaba como lenguaje de simulación
- Ha influido en el desarrollo de otros LPOO

- **1972- SMALLTALK (Kay, Goldberg, Ingalls)**

- Simula + LISP (sin tipos) -> LPOO puro
- Características de un LPOO [Byte81]:
 - Todo es un **objeto**.
 - Programa = cjo de objetos que se comunican mediante **mensajes**
 - Todo objeto es **instancia** de una clase (tiene un tipo).
 - La **clase** es el repositorio de comportamiento asociado con un objeto
 - Las clases se organizan en **jerarquías de herencia**

Historia de los LPOO – Años 70-80

- “orientado a objetos” era sinónimo de “bueno”.
- Soportar el concepto de objeto no es suficiente.



Historia de los LPOO - Años 80

- Proliferación de LPOO
- Clasificación atendiendo al origen:

- **Híbridos:**

- Basados en C:

- **C++** 1985 B. Stroustrup
- Objective-C 1983 Brad Cox y Tom Love

- Basados en Pascal:

- Object Pascal 1985 Apple con Wirth
- Modula-3 1988 Digital y Olivetti

- Basados en Lisp:

- CLOS 1988

- **Púros:**

- Smalltalk 1972 Alan Kay
- **Eiffel** 1985 Bertrand Meyer

Historia de los LPOO – Años 90

- Aplicaciones centradas en el **web**
- **1995 JAVA- SUN**
 - “comportamiento” en páginas HTML
 - Tecnología de implementación: bytecode+MV
- **2000 C# - Microsoft**
 - Plataforma .NET
 - Máquina virtual para conseguir la portabilidad
 - Combina Java y C++
 - Son muchos los lenguajes que se están migrando a la plataforma .NET