

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

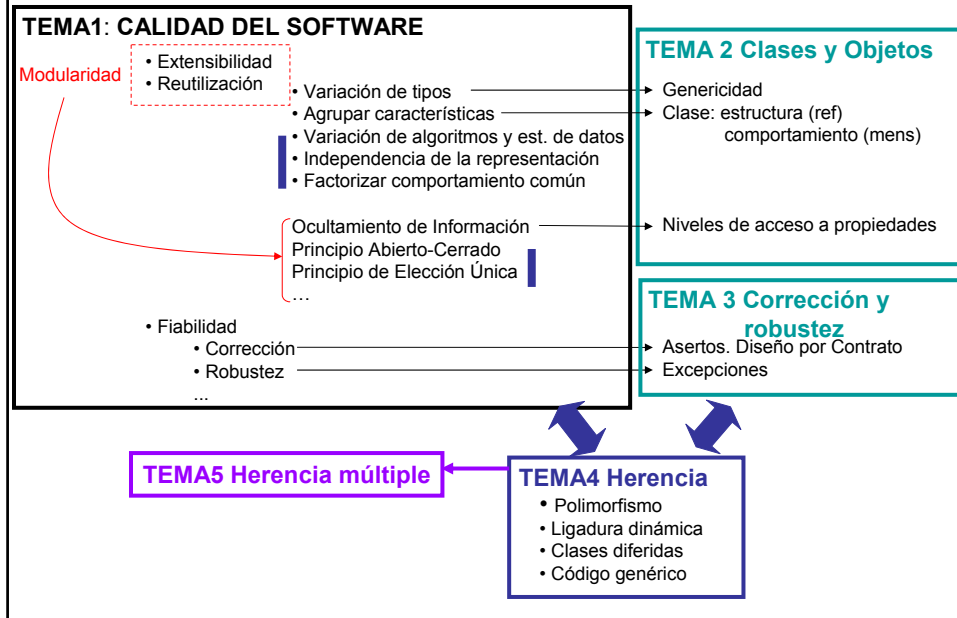
TEORÍA

Programación Orientada a Objetos
Curso 2006 / 2007

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 de la asignatura

- Exámenes:
 - 13 febrero
 - 29 junio
 - 7 septiembre
- $NOTA = Teoría * 0'6 + Prácticas * 0'4$
- El aprobado del examen o las prácticas sólo se guardará durante un año

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 (2ª planta): 2.37



- Martes: 16:30 a 18:30
- Miércoles: 16:30 a 18:30
- Jueves: 11:00 a 13:00

PRÁCTICAS

Programación Orientada a Objetos
Curso 2006 / 2007

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
- Mecanismo de excepciones
- Cadenas y Entrada/Salida
- Herencia, clases abstractas
- Genericidad
- Interfaces
- Colecciones e Iteradores

Metodología

- Seminarios semanales.
- Prácticas laboratorio abierto.
- Ejercicios semanales.
- Entrevistas de control y revisión.
- Dos correcciones generales de prácticas.
- Defensa de prácticas al azar dentro del grupo.

Modalidad de prácticas

- **S3** (9 octubre):
 - Práctica laboratorio cerrado: **IDE Eclipse**
 - Comienza el seminario en clase de teoría
- A partir de la **S4** (16 octubre)
 - Prácticas de laboratorio abierto

Práctica

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

Evaluación de las prácticas

- **Penalización:**
 - Por faltar entrega programadas
 - No hacer las correcciones de las entrevistas
- ***No realizar la primera entrega general
pérdida de convocatoria de febrero***
- **Defensa de prácticas**

Horario

- **Seminario:**
 - Miércoles de 12:00 – 13:00
- **Laboratorio 1.3:**
 - Grupo 1: Martes de 16:30 – 18:30
 - Grupo 2: Miércoles de 16:30 – 18:30
- **Tutorías Marcos** (despacho 2.33):
 - Martes: 9:30 – 13:30
 - Miércoles: 18:30 - 20:30
 - Horas de laboratorio

Bibliografía Básica

- Gratuitos (versiones en PDF):
 - *Curso de Java en español*. Iván Párraga García
 - *Tutorial de Eclipse*. <http://eclipsutorial.forge.os4os.org/in1.htm>
 - *Thinking in Java*, 3ª edición (inglés), en español en la biblioteca
- Biblioteca:
 - *Java 2. Vol I. Fundamentos*. Cay S. Horstmann, Gary Cornell
 - *Java 2. Vol II. Características Avanzadas*. Cay S. Horstmann, Gary Cornell
 - *El lenguaje de programación Java*, de J. Gosling
 - *Effective Java. Programming Language Guide*.

INTRODUCCIÓN

Programación Orientada a Objetos
Curso 2006 / 2007

¿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

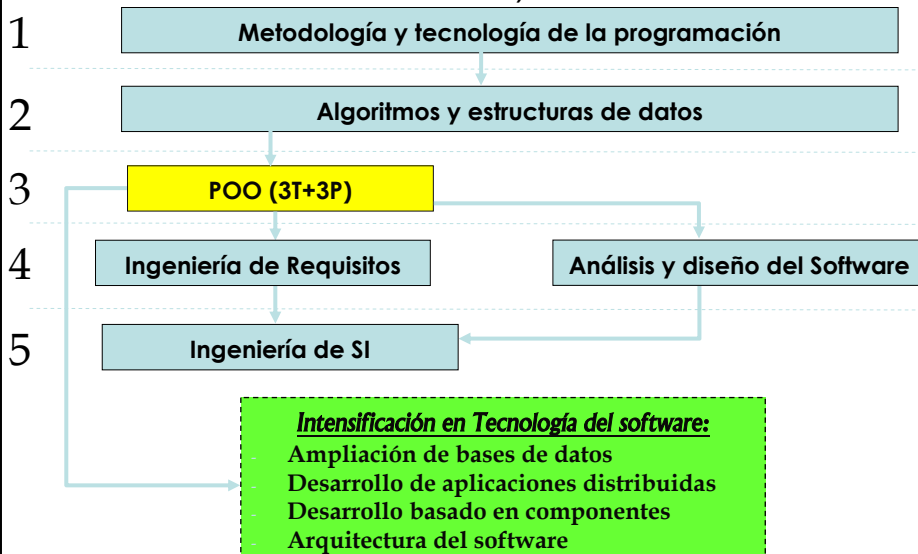
El futuro del paradigma OO

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):**

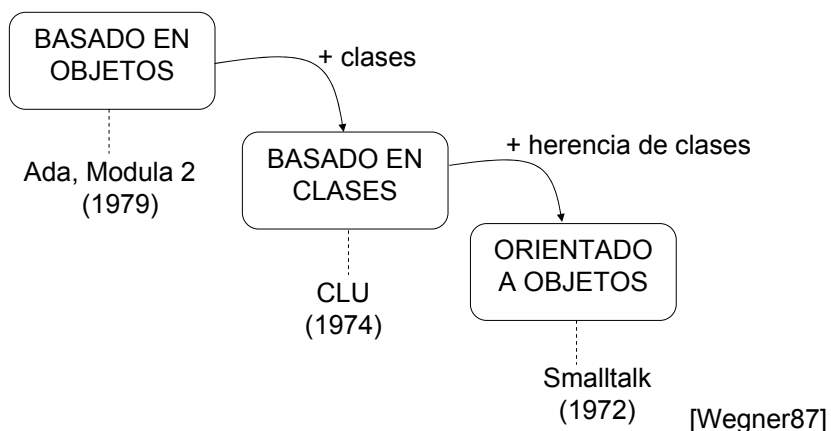
- comienzo de la OO
- Se identificaba como lenguaje de simulación
- Ha influido en el desarrollo de otros LPOO

- **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 más de 20 lenguajes los que están en proceso de migrarse a la plataforma .NET