

CC51H Programación Orientada al Objeto

10 UD

1 Vigencia

A partir del otoño 2005

2 Requisitos

CC31A, CC31B, CC41A. Es recomendable tener dominio en programación estructurada, tipos de datos abstractos y nociones de ingeniería de software.

3 Objetivos

Al final del curso se espera que el alumno entienda los conceptos más importantes de las etapas de análisis, diseño y programación orientada objetos, y sea capaz de aplicarlos correctamente en las distintas etapas del desarrollo de software. En particular, en este curso se dá énfasis a diseñar buenos objetos, a usar herencia solo cuando provee claras ventajas, y en general, a conocer en profundidad los aspectos de diseño y programación que permiten el desarrollo de componentes reusables, y a los que facilitan la extensión y mantención del software.

4 Programa

4.1 Introducción

Complejidad del software. Enfrentando la complejidad: descomposición, abstracción y jerarquías. Factores de calidad del software. Modularidad. Reutilización. Notación.

4.2 Conceptos básicos

Cómo definir buenos objetos?. Diseño de buenos tipos de datos abstractos. Contratos. De tipos de datos abstractos a clases. Características de c++ que

permiten una buena implementación de un TDA. Objetos: estado, comportamiento e identidad. Herencia. Polimorfismo y enlace dinámico. Contraste entre una solución con y sin herencia. Ejemplos.

4.3 Herencia y subtipos

Tipos y subtipos. Contraste entre herencia y subtipos. Qué tipos de formas de heredar existen?. Contratos a través de la herencia. Herencia múltiple. Ejemplos.

4.4 Aspectos importantes en el desarrollo de software orientado a objetos

Relaciones entre clases. Contraste entre composición y herencia. Cuándo es conveniente heredar? Cuándo es mejor usar composición? Diseño de buenas clases. Patrones de diseño y aplicaciones. Evaluación de diseños: métricas. Patrones de análisis.

4.5 Análisis y diseño orientado a objetos

Etapas en el desarrollo de software orientado a objetos. Proceso de desarrollo recomendado (RUP). Lenguaje de modelamiento: UML. Ejemplos

4.6 Validación de software orientado a objetos

Métodos de validación de software. Ambiente de prueba de programas orientados a objetos.

5 Evaluación

El curso se evaluará a través de controles y examen (con apuntes), controles de lectura (con la lectura) y tareas computacionales.

- 2 controles y examen
 - C1: Fecha probable: 26 o 28 de abril
 - C2: Fecha probable: 7 o 9 de junio

- 4 lecturas (no se elimina ninguna y su promedio vale como un tercer control)
- 3 tareas computacionales (no se elimina ninguna). (C++, rational o ArgoUml)

6 Lecturas

- Bjarne Stroustrup. What is object oriented programming? IEEE Software. 1988. Fecha probable: 22 o 24 de marzo
- Bertrand Meyer. Applying design by contract. Computer. 1990. Fecha probable: 12 o 14 de abril
- Daniel Halbert y Patrick O'Brien. Using types and inheritance in object oriented programming. 1987. Fecha probable: 10 o 12 de Mayo
- Por definir. Fecha probable: 31 de Mayo o 2 de junio.

References

- [Jac2000] Ivar Jacobson, Grady Booch, James Rumbaugh. *The unified software development process* Addison Wesley, 2000.
- [Eli95] A. Eliëns. *Principles od Object-Oriented Software Development*. Addison Wesley, 1995.
- [Mey97] Bertrand Meyer. *Object-Oriented Software Construction*. Second Edition. Prentice Hall, 1997.
- [Str2002] Bjarne Stroustrup. *The C++ Programming Language*. Addison-Wesley, 2002. Edicion especial.(Disponible tambien en castellano)
- [Kho90] Setrag Khoshafian and Razmik Abnous. *Object Orientation - Concepts, Languages, Databases, User Interfaces*. John Wiley & Sons, Inc., 1990.
- [Gam95] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides *Design Patterns: Element of Reusable Object Oriented Software*. Addison-Wesley, 1995.

- [Cop92] J. Coplien. *Advanced C++. Programming styles and idioms*. Addison-Wesley, 1992.
- [Jac92] Ivar Jacobson *Object Oriented Software Engineering: A use case driven approach* Addison-Wesley, 1992
- [Poh89] I. Pohl. *C++ for C Programmers*. Benjamin/Cummings, 1989.
- [Wie90] Richard Wiener and Lewis Pinson. *The C++ Workbook*. Addison-Wesley, 1990.
- [Cor97] G. Cornell, Cay S. Horstmann *Java Core* SunSoft Pres, 1997

Todos estos libros esta'n en biblioteca. Otras fuentes son: Journal of Object-Oriented Programming, ACM OOPS Messenger, Communications of ACM (Sep 90), actas de OOPSLA.