

Plan 257 Ing.Tec.Informática de Gestión

Asignatura 16495 PROGRAMACION II

Grupo 1

### Presentación

La asignatura se inicia con una introducción al concepto de Ingeniería del Software, Ciclo de Vida y Calidad del Software. Se introducirán a continuación conceptos fundamentales de modularidad, metodología de diseño modular y prueba de módulos. A continuación se introducen ideas de programación orientada a objetos, programación bajo contrato y la verificación formal de algoritmos.

RECOMENDACIONES: Para cursar adecuadamente esta asignatura es muy recomendable que los alumnos hayan comprendido los conceptos básicos que se abordan en Programación 1.

En la página web del departamento los alumnos encontrarán una dirección en la que se encuentra información de interés sobre esta asignatura: transparencias que se usarán en clase, ejercicios, prácticas, etc.

### Programa Básico

### Objetivos

#### OBJETIVO GENERAL.

Establecer los conocimientos generales sobre calidad del software, modularidad, programación orientada a objetos y verificación y validación de algoritmos.

#### OBJETIVOS ESPECIFICOS.

- 1.- Conocer diferentes modelos de ciclo de vida del software.
- 2.- Reconocer las características fundamentales de calidad de los diseños modulares.
- 3.- Conocer la aportación que la Orientación al Objeto hace al desarrollo del software.
- 4.- Conocer los conceptos de clase y objeto.
- 5.- Conocer los mecanismos de herencia y genericidad.
- 6.- Saber diseñar programas utilizando el paradigma orientado a objetos y sus mecanismos de herencia y genericidad.
- 7.- Conocer el concepto de programación bajo contrato.
- 8.- Abordar la prueba y depuración de programas siguiendo las estrategias más adecuadas, de una forma sistemática.
- 9.- Saber verificar formalmente algoritmos, incluidos los recursivos.
- 10.- Conocer el lenguaje Smart Eiffel y manejar la biblioteca de clases del mismo.

### Programa de Teoría

#### 1.Programación modular

##### 1.1Introducción

Factores internos y externos de calidad.

Ciclo de vida y calidad del software.

Fiabilidad y corrección.

Documentación

##### 1.2Modularidad

Aspectos de la calidad de un diseño modular: Cohesión y Acoplamiento.

Modularidad y orientación a objetos.

#### 2.Introducción a la OO

##### 2.1La estructura estática: Clases.

El papel de las clases en orientación a objetos.

Un sistema de tipos uniforme.

El estilo de programación orientado a objetos.

Ocultación de la información.

##### 2.2La estructura dinámica: Objetos.

Objetos.

- 
- Objetos y referencias.
  - Procedimientos de creación.
  - Operaciones sobre referencias.
  - Objetos compuestos y tipos expandidos.
  - Un ejemplo especial: la clase STRING.
  - Ligadura por referencia y por valor.
  - Trabajando con referencias, ventajas e inconvenientes.
  - La gestión de la memoria.
  - 2.3 Genericidad y Herencia
    - Generalización horizontal y vertical.
    - La necesidad de parametrización de tipos.
    - Clases genéricas.
    - El mecanismo de la herencia
  - 3. Programación bajo contrato
    - 3.1 Prueba de programas.
      - Principios de prueba.
      - Pruebas indirectas.
      - Pruebas directas: estrategias de caja blanca y caja negra.
      - Pruebas de nivel superior.
      - Depuración.
    - 3.2 Verificación de algoritmos.
      - Reglas de verificación.
      - Corrección parcial de un algoritmo.
      - Corrección total.
      - Algoritmos recursivos.
    - 3.3 Contratos software
      - Aserciones: Pre y Postcondiciones.
      - Invariantes de clase.
      - Invariantes y variantes de bucles.
      - Manejo de excepciones.
- 

## Programa Práctico

Las prácticas de la asignatura son individuales. Para su realización los alumnos deberán disponer de una cuenta personal en el ordenador del departamento habilitado a tal efecto.

El calendario y la asignación de grupos para las prácticas pueden ser consultadas desde aquí.

Las primeras sesiones tendrán como objetivo el aprendizaje del lenguaje Eiffel. Se impartirán algunas clases sobre las nociones básicas del lenguaje, poniendo a disposición de los alumnos un conjunto de problemas cuya solución implique la utilización de estructuras básicas de control, pruebas de rutinas etc..

Habrà una práctica evaluable en la que se deberán utilizar los conceptos de POO y diseño bajo contrato, cuyas especificaciones estarán disponibles en noviembre.

---

## Evaluación

Para la evaluación de la asignatura se considerará tanto el aspecto teórico como el práctico de la misma.

La nota final se obtendrá de la nota de prácticas y la nota del examen ordinario (o extraordinario en su caso) mediante la fórmula:  $0.9 \times \text{nota del examen} + 0.1 \times \text{nota de las prácticas}$ .

En los grupos de teoría que hayan hecho evaluaciones en el aula, la nota final se modificará de la siguiente forma:  $0.85 \times \text{nota del examen} + 0.1 \times \text{nota de las prácticas} + 0.05 \times \text{nota del aula}$

Para obtener la nota de prácticas los alumnos deberán depositar el código fuente de los programas que resuelvan los problemas planteados a tal efecto en el directorio que oportunamente se indicará antes de las 8 de la tarde del día del examen final (o extraordinario en su caso) y entregar la documentación necesaria al comenzar dicho examen.

En caso necesario se guardará la nota de prácticas para la convocatoria extraordinaria.

---

## Bibliografía

- \* Meyer B., "Construcción de software orientado a objetos" Prentice Hall, 1999.
  - \* Myers G.J., "El arte de probar software" El Ateneo, 1979
  - \* Page-Jones M. , "The Practical Guide to Software System Design" Prentice Hall, 1988
  - \* Pressman R. S., "Ingeniería del software" McGraw-Hill, 1993
  - \* Sommerville I., "Ingeniería de software" Addison Wesley Iberoamericana, 1989
  - \* Wirth N., "Introducción a la programación sistemática" El Ateneo, 1982
-

---

Nota: consultar la página web de la asignatura.

---