

Plan 258 Ing. Tec. en Informática de Sist.

Asignatura 16541 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 y calidad en el diseño modular

A continuación se introducen ideas fundamentales de Programación Orientada a Objetos.

Por último se estudian técnicas de pruebas de programas, verificación formal de algoritmos y programación bajo contrato.

### Programa Básico

Modularidad y Calidad del Software. Introducción a la Programación Orientada a Objetos. Prueba de Programas y Verificación de Algoritmos. Introducción a la Programación bajo Contratos.

### Objetivos

#### Objetivo general

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

#### Objetivos específicos

- . Conocer diferentes modelos de ciclo de vida del software
- . Reconocer las características fundamentales de calidad de los diseños modulares
- . Conocer la aportación que la Orientación al Objeto hace al desarrollo del software
- . Conocer los conceptos de clase y objeto
- . Conocer los mecanismos de herencia y genericidad
- . Saber diseñar programas utilizando el paradigma orientad a objetos y sus mecanismos de herencia y genericidad.
- . Conocer el concepto de programación bajo contrato
- . Abordar la prueba y depuración de programas siguiendo las estrategias más adecuadas, de una forma sistemática.
- . Saber verificar formalmente algoritmos, incluidos los recursivos.
- . Conocer el lenguaje Eiffel y saber manejar la biblioteca de clases del entorno de SmartEiffel.

### Programa de Teoría

#### 1. Calidad del Software y Modularidad

##### 1.1 Introducción.

- Factores internos y externos de calidad.
- Ciclo de vida y calidad del software.
- Fiabilidad y corrección.
- Documentación

##### 1.2 Modularidad.

- Aspectos de la calidad de un diseño modular: Cohesión y Acoplamiento.
- Modularidad y orientación a objetos.

#### 2. Introducción a la OO

##### 2.1 La 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.

##### 2.2 La estructura dinámica: 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 y mecanismo de 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.

### 3.3 Contratos software

- Aserciones: Pre y postcondiciones.
  - Invariantes de clase.
- 

## Programa Práctico

Las prácticas de la asignatura consistirán en la aplicación de los conceptos expuestos en el programa de teoría, utilizando el lenguaje Eiffel y el compilador SmartEiffel.

---

## Evaluación

El 85% de la nota se obtendrá mediante un examen escrito de cuestiones teóricas y problemas, tanto en la convocatoria ordinaria como en la extraordinaria.

Un10% corresponderá a la evaluación de las prácticas efectuadas durante el curso.

El 5% restante se obtendrá en el aula realizando dos pruebas de tipo test sobre la materia nueva de la asignatura vista hasta ese momento.

---

## 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
-