

Diseño modular

- Técnica de descomposición/construcción de programas, basada en unidades lógicas de datos (módulos) y sus funcionalidades
- Necesario para programas grandes
- Caso particular: programación orientada a objetos
módulo = clase; variable = objeto; operación = método
- Ejemplo de módulo/clase standard: `vector`

Clasificación de módulos

- Módulos de **datos** : tipo + operaciones
 - Módulos **funcionales** : sólo operaciones
- (apenas se usarán en este curso)

Roles en el diseño modular

- **Especificación:** qué contiene el módulo y qué se puede hacer con ello
- **Uso:** integración del módulo en un diseño completo
- **Implementación:** cómo se representa el tipo y cómo se codifican las operaciones en un lenguaje de programación

Especificación de Módulos

- Información sobre el contenido del módulo y cómo utilizarlo
- Elementos públicos: nombre, descripción, operaciones. Esto es lo que ve el usuario. **NO** se puede dar información sobre cómo está implementado (abstracción de datos: se explica lo que se puede hacer con la clase, **NO** cómo se hace)
- Independencia respecto de la implementación

Especificación Pre/Post de operaciones

- **Cabecera:** Acción (`void`)/función, nombre, lista de parámetros y/o resultados
- **Precondición:** propiedades que han de cumplir los parámetros para que la operación se pueda ejecutar
- **Postcondición:** propiedades que han de cumplir los resultados y parámetros por referencia al acabar la ejecución
- **Abstracción funcional:** se explican los efectos de la operación, no cómo los consigue

Operaciones de una clase

En programación orientada a objetos, las operaciones de una clase se consideran propiedad de los objetos de la misma.

Las llamadas se realizan de la forma

`objeto.operacion(resto de parámetros)`

El objeto sobre el que se aplica la operación no sale en la cabecera de la misma, por eso recibe el nombre de **parámetro implícito**.

Clasificación de las operaciones de una clase

- Constructoras y destructoras
- Modificadoras
- Consultoras
- Lectura y escritura

Ejemplo: clase `Estudiant`