

Ampliaciones de una clase

Ampliar una clase es dotarla de funcionalidades que no están en su especificación original. Hay varias maneras:

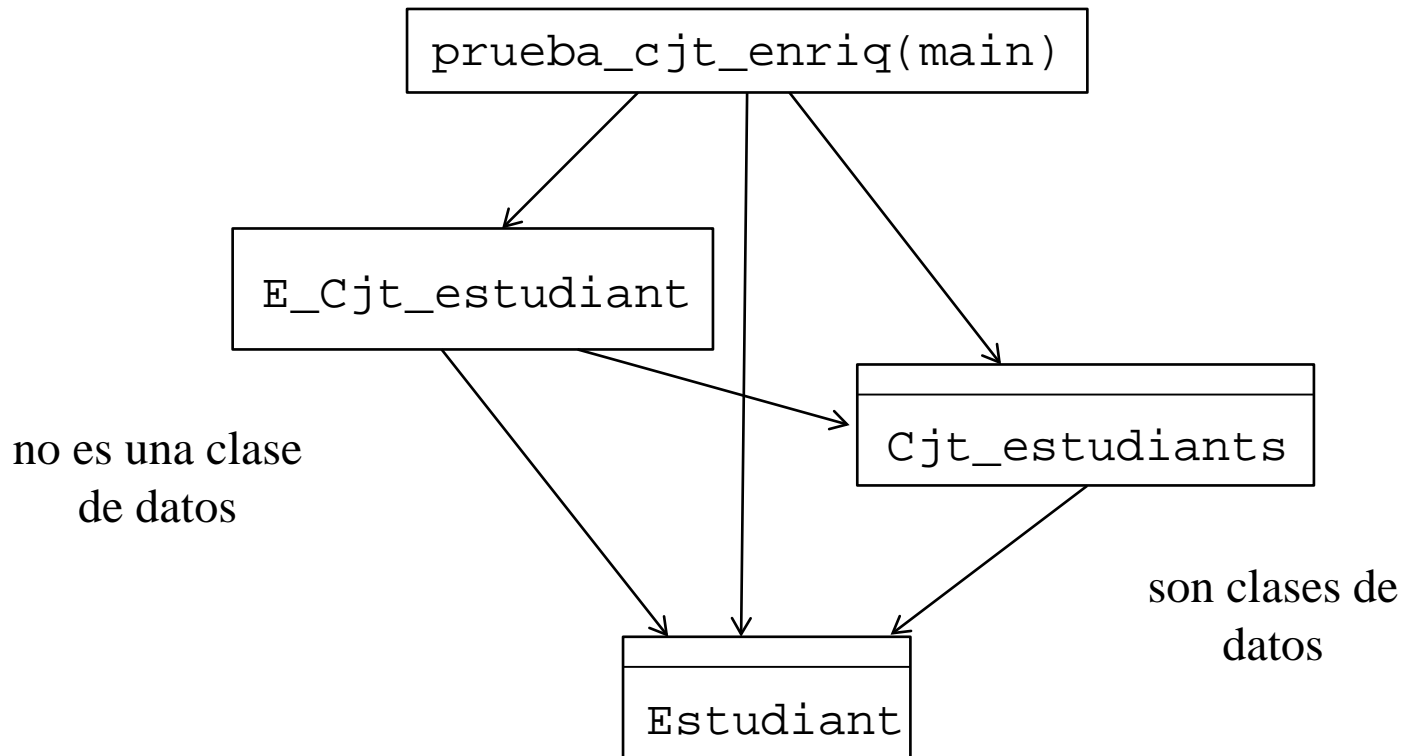
- Programar las nuevas funcionalidades en el lugar donde se usen (`arrodonir`, `presentados`, etc)
- Reunir las nuevas funcionalidades en un módulo funcional o enriquecimiento: `E_Cjt_estudiants`
- Añadir las nuevas funcionalidades a la propia clase. **Esto suele ser lo mejor**, pero solo se puede hacer si se tiene acceso al `.cc` de la clase.
- Crear una nueva clase mediante herencia de la anterior (no forma parte del contenido de PRO2)

Ejemplo de enriquecimiento

- `E_Cjt_estudiants (.hh y .cc)`: contiene nuevas operaciones de conjuntos de estudiantes que **no pertenecen** a la clase `Cjt_estudiants` y **permite reutilizarlas** mediante `#include` y linkando el `.o` tras compilar el `.cc`
 - `esborrar_estudiant(c, dni)`
 - `estudiant_nota_max(c)`

Uso: ver `prueba_cjt_enriq.cc`

Diagrama modular: uso de E_Cjt_estudiant



Nuevas operaciones dentro de la clase

La otra manera de ampliar Cjt_estudiants consiste en añadir las nuevas operaciones a sus `.hh` y `.cc`.

- `void esborrar_estudiant(int dni);`
- `Estudiant estudiant_nota_max() const;`

Comprobad que el código puede quedar mucho mas limpio y eficiente!

Uso: ver `prueba_cjt_ampliado.cc`

Nuevas operaciones dentro de la clase

Puede aprovecharse para optimizar las nuevas operaciones:

- `esborrar_estudiant`: solo requiere desplazar los elementos posteriores en `vest` al elemento borrado y restar 1 al tamaño (mucho mejor que la versión anterior)
- `estudiant_nota_max`: se puede añadir un nuevo campo `imax` para tener siempre guardado el estudiante de nota máxima
- (privada) `recalcular_posicio_imax`: para cuando `imax` cambie y no tengamos pistas sobre el nuevo valor

Nuevas operaciones dentro de la clase

Ejercicios:

- Implementar `recalcular_posicio_imax`
- Todas las operaciones originales han de retocarse para que tengan en cuenta el nuevo campo `imax`
- También ha de actualizarse el invariante de la representación
- **Ineficiencia típica:** usar `recalcular_posicio_imax` en situaciones donde no es necesario. Ejemplos
 - si borramos un elemento posterior a `imax`, no hay que modificarlo
 - si borramos un elemento anterior, solo hay que restar 1 a `imax`
 - si añadimos un elemento menor que `v[imax]`, no hay que modificarlo
 - si añadimos un elemento mayor que `v[imax]`, basta con una asignación