

Item 19: Treat class design as type design.

Date:

Defining a new class, defines a new Type
overloading functions & operators,
controlling memory allocation & deallocation
defining object initialization & finalization
is all done by class / type designer.

Issues faced to create good type → Ask below questions

① How object of new type be created & destroyed?

⇒ influences design of class's
ctor, dtor, memory allocation &
deallocation (operator new, new[],
delete, delete[])

② How should object initialization differs from object assignment.

⇒ difference b/w ctor & assignment
operator. Initialization & assignment
correspond to different function calls.
(Item 4)

③ What does it mean for objects of your new type to be passed by value?

⇒ copy ctor defines how pass by
value is implemented for a type.

④ What are restriction on legal value for your new type

⇒ Invariants determine error checking inside ctors, assignment operators & setter functions.

⑤ Does your new type fit into an inheritance graph.

→ If you inherit from other class, then it depends on the function declared as virtual or non-virtual, & if you want other classes to inherit from your type, then it depends on the functions which you make virtual specially destructor.

⑥ What type kind of type conversion are allowed for new type?

If we want to ~~write~~ allow objects of Type T1 to be ~~or~~ implicitly converted into objects of type T2, we will have to write ⇒ type conversion function in class T1
⇒ non-explicit ctor in class T2.

if we want explicit conversions only avoid

⇒ type conversion operators
⇒ non-explicit ctor.

- ⑦ What operators & functions make sense for new type.
 ⇒ which function to declare as member function
- ⑧ Standard function to be disallowed?
 ⇒ which ones to declare private
- ⑨ Who should have access to members of your new type?
 → which members are public, protected or private.
 → which classes / function should be friends
 → whether to nest one class inside other.
- ⑩ What is 'undeclared interface' of your new type?
 → guarantee of worst performance except safety & resource usage (eg: locks & dynamic memory)
 → Above imposes constraint on class implementation.
- ⑪ How general is new type?
 → You may be defining a whole family of types
 → like new class template not new class

⑩ Is new type really what u need?

→ If derived class is defined to add functionality to existing class, ~~for~~

→ Above can be achieved by non-member funcon / template.