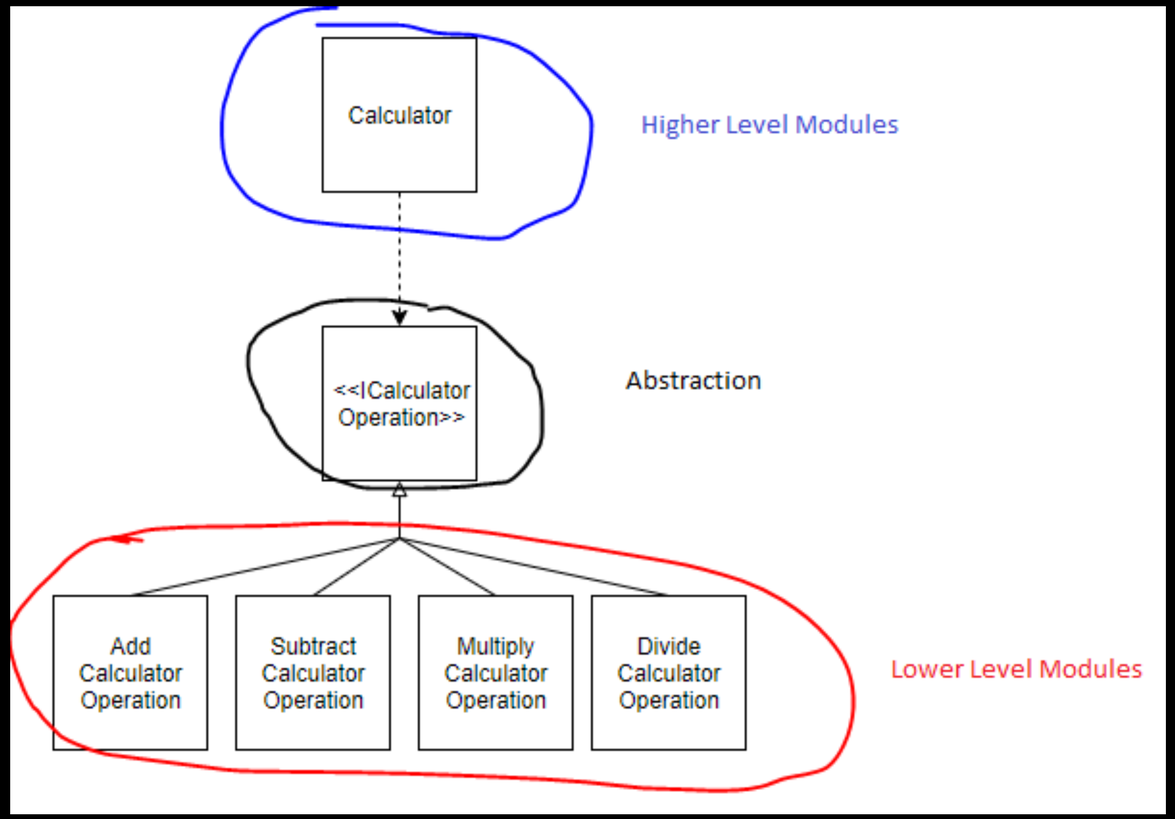
Principio de Inversión de Dependencias – SOLID

Robert C Martin

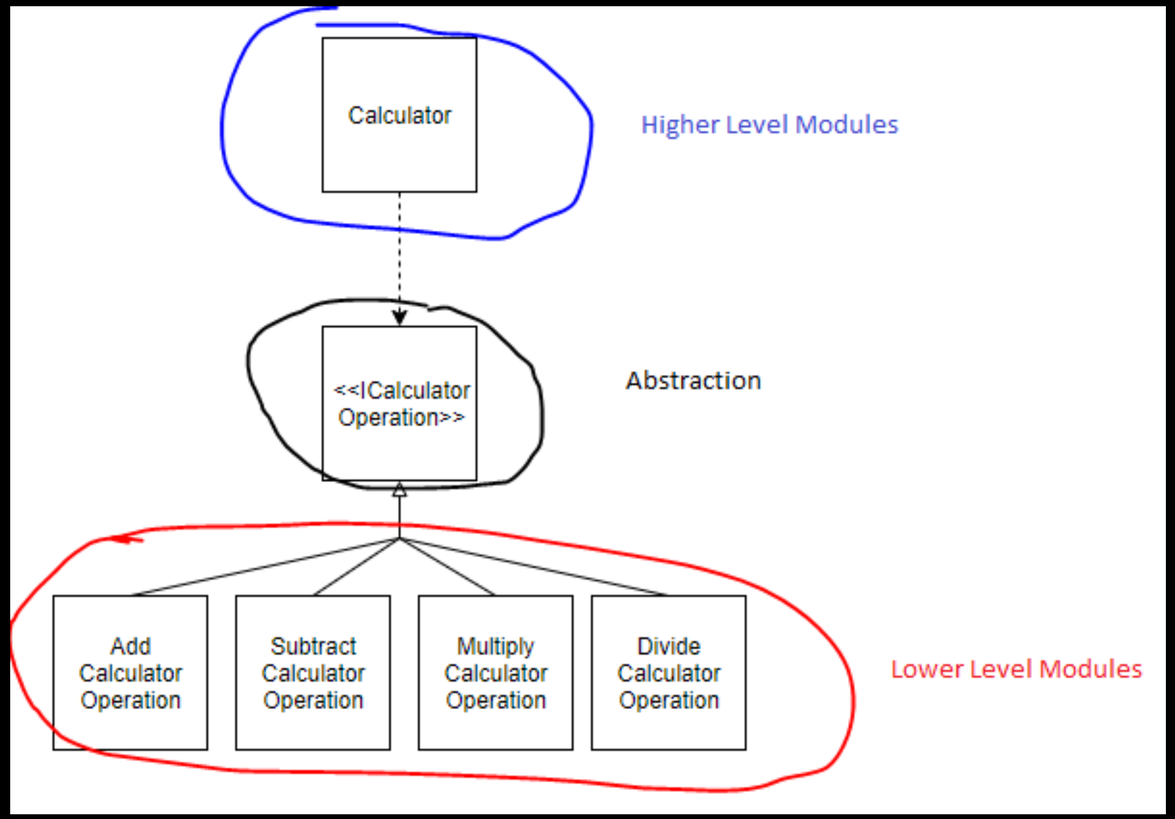
La definición que dio originalmente Uncle Bob es:

Los **módulos de alto nivel** no deberían depender de los **módulos de bajo nivel**. Ambos deberían depender de **abstracciones**.

Las **abstracciones** no deberían depender de los **detalles**. Los **detalles** deben depender de **abstracciones**.

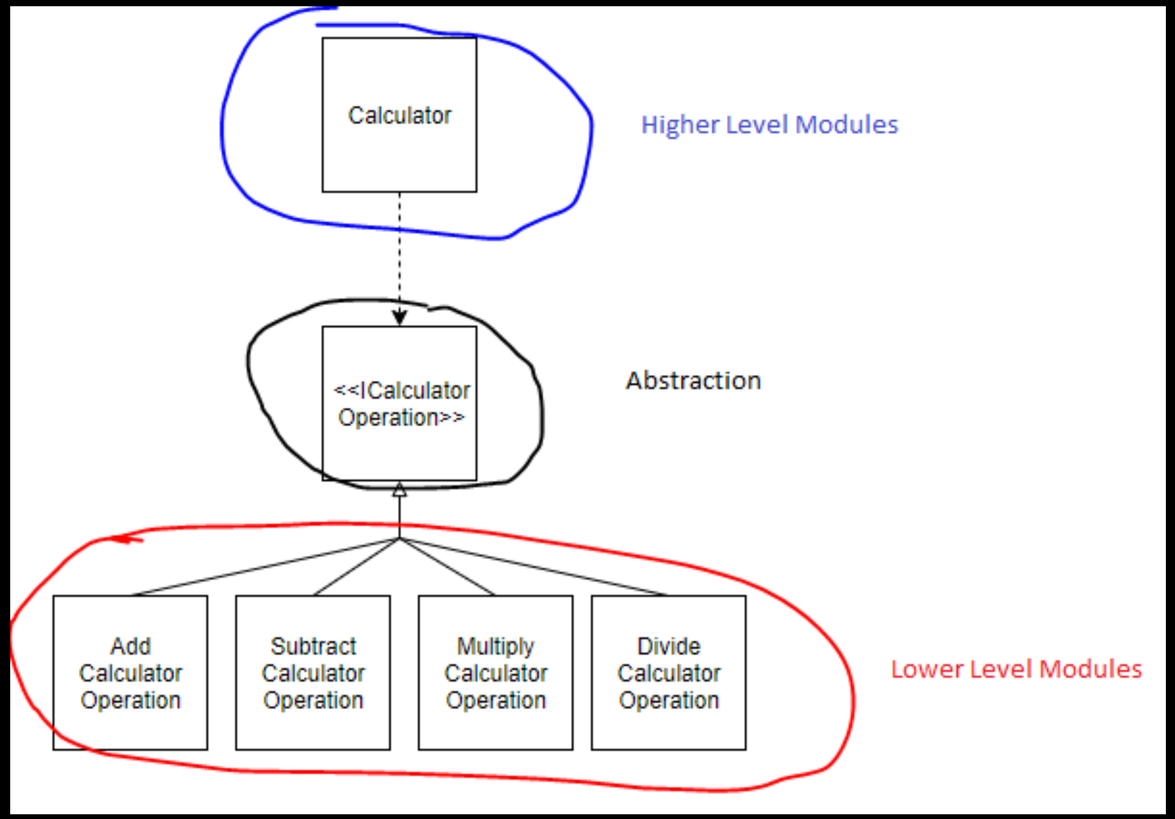


definamos ahora algunos conceptos básicos



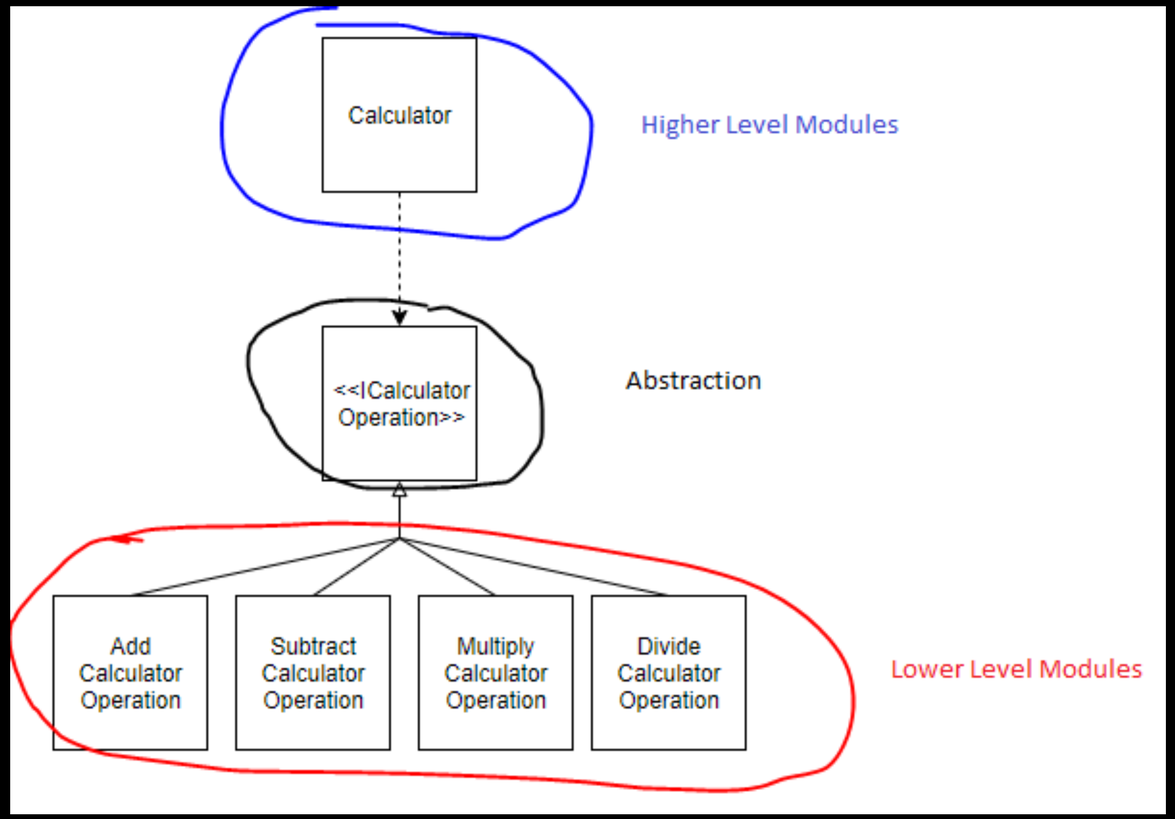
- **Módulos de alto nivel:** se refieren a los objetos que definen **el qué es** y **qué hace** tu programa. Aquellos que contienen la lógica de negocio y cómo interactúa el software entre sí. **Son los objetos más importantes del programa**.

- Por ejemplo, en una aplicación de un banco, podría ser el objeto que se encarga de devolver la lista de cuentas bancarias.



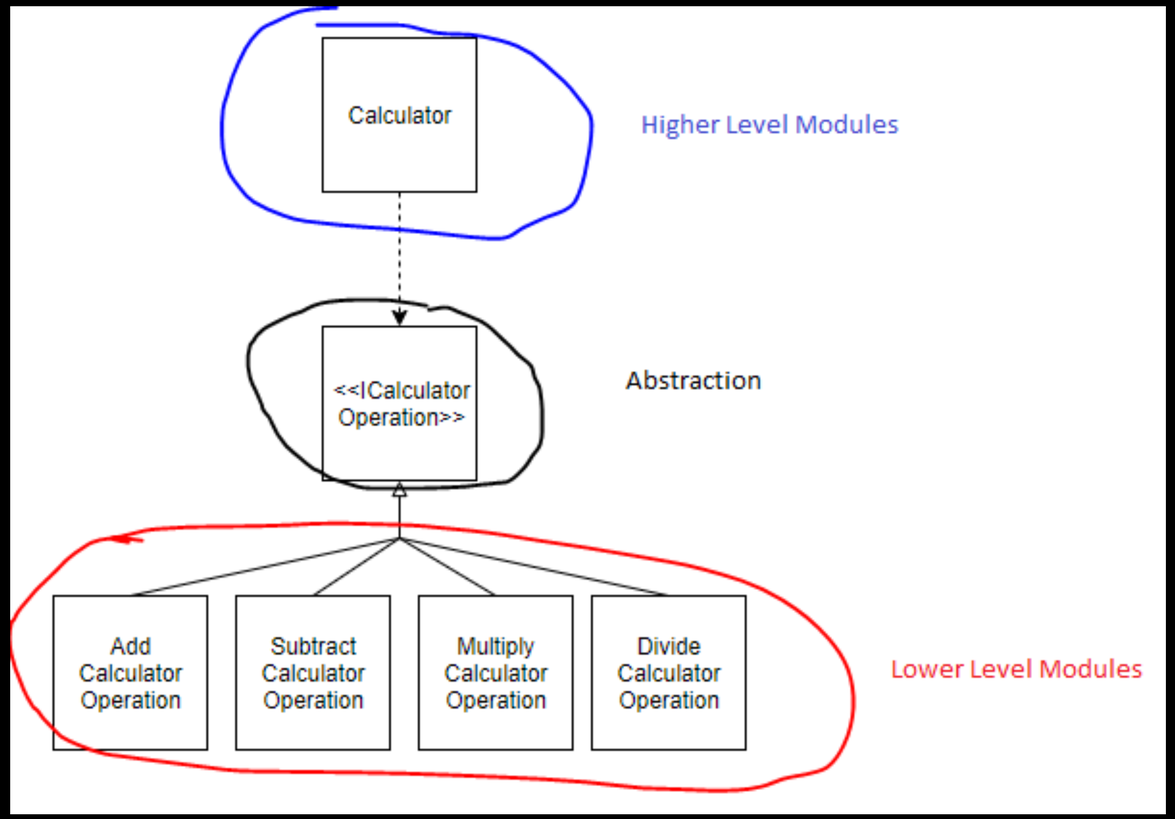
- **Módulos de bajo nivel:** son aquellos objetos que no están directamente relacionados con la lógica de negocio del programa.

- Son objetos menos importantes, de los cuales no depende la lógica de negocio.



- **Abstracciones:** se refieren a Tipos de Datos que no son las implementaciones concretas, si no los que definen la interfaz pública.

- Serán, por tanto, protocolos (o interfaces) o clases abstractas



Los problemas que pueden resultar de no cumplir con el **Principio de Inversión de Dependencias** son:

- **Añades Acoplamiento** entre módulos de alto nivel hacia módulos de bajo nivel.

O lo que es lo mismo, tu lógica de negocio dependerá de detalles de implementación. Si alguna vez quieres cambiar algún detalle de implementación, tendrás que modificar también los objetos más importantes del programa.

- **Las dependencias entre objetos no quedan claras:** a simple vista, no puedes saber fácilmente de qué otros objetos depende un módulo.
- **Tu programa no es testeable:** si tu módulo depende de un objeto que no puedes intercambiar por un *mock* en tus tests no puedes testarla de manera unitaria. Si el test falla, no sabrás realmente qué objeto es el culpable del error.