

Cómo Diseñar Clases – Receta Básica¹

Cómo diseñar una Clase²

1 del Análisis de Problema al Diagrama de Clase

Lea la declaración del problema. Busque declaraciones que mencionen o enumeren los atributos de los objetos en el espacio del problema. Estas dicen cuántos atributos (campos de datos) necesita para la definición de clases y qué información van a representar en el espacio de la solución.

Registre sus hallazgos como un diagrama de clases porque proporciona una vistazo general de las clases, especialmente cuando se trata con muchas clases o cuando se vuelve al problema en un dentro de unos meses.

2 del Diagrama de Clase a las Definición de la Clase

Para cada clase del diagrama de clases, agregue una definición de clase donde conste una declaración de propósito de esa clase. La declaración de propósito debe explicar qué información es la que se desea representar (¡y cómo!) en una instancia cualquiera de esa clase.

Esta traducción es principalmente mecánica. La única parte que necesita su plena atención es decidir para cada atributo si su valor va a ser el mismo para todas las instancias de esa clase, o si cada instancia puede tener un valor distinto.

- Si el valor del atributo va a ser el mismo para todas las instancias de la clase, inicialícelo al declarar el atributo.
- Si cada instancia va a poder asignarle un valor distinto a ese atributo, agregue el atributo como parámetro al constructor y añadir una inicialización de la forma `"this.field = campo"` al constructor.

3 de la Definición de la Clase a Ejemplos Funcionales

Por un lado, obtenga ejemplos de información y representelos como instancias de las clases. Por el otro lado, invente instancias de las clases e interpretelas como información.

Tenga presente que algunos ejemplos de datos no tienen explicación en el espacio del problema. Advierta de tales casos con comentarios en el código. (¡Recuerde que Ud. es uno de los posibles futuros lectores de su propio código!)

4 [Plantilla, Pruebas, y Definición]

(Ver receta básica de programas.)

¹ Adaptado de: **"How to Design Classes - Data: Structure and Organization (June 15 2012 draft)"** (Cómo Diseñar Clases – Datos: Estructura y Organización, Borrador del 15 de Junio de 2012). *Matthias Felleisen, Matthew Flatt, Robert Bruce Findler, Kathryn E. Gray, Shriram Krishnamurthi & Viera K. Proulx.* Versión digital disponible gratuitamente en <https://felleisen.org/matthias/htdc.html>
Traducción y adaptación al castellano: Leandro Doctors <ldoctors@untref.edu.ar>

² Los primeros tres pasos corresponden a los primeros tres pasos de la receta básica de programas.

Cómo Diseñar Clases – Receta Avanzada³

Cómo diseñar una Jerarquía de Clases⁴

1 del Análisis de Problema al Diagrama de Clases

Lea la declaración del problema, cuidadosamente.

Determine con qué tipos de información debe lidiar el programa. Es importante identificar *clases* de información, y no sólo atributos sueltos. El resultado de este paso es una lista de nombres de las clases relevantes y una breve descripción de cada una.

Para cada una de ellas, busque declaraciones que mencionen o enumeren los atributos de los objetos en el espacio del problema. Estas dicen cuántos atributos (campos de datos) necesita para la definición de clases y qué información van a representar en el espacio de la solución. Recuerde que pueden haber casos en los que algún atributo no pertenezca a ninguna clase.

Registre sus hallazgos como un diagrama de clases porque proporciona una vistazo general de las clases, especialmente cuando se trata con muchas clases o cuando se vuelve al problema en un dentro de unos meses.

2 del Diagrama de Clases a las Definiciones de Clases

2.a Definiciones de Clases (Responsabilidades)

Para cada clase del diagrama de clases, agregue una definición de clase donde conste una declaración de propósito de esa clase. La declaración de propósito debe explicar qué información es la que se desea representar (¡y cómo!) en una instancia cualquiera de esa clase.

Esta traducción es principalmente mecánica. La única parte que necesita su plena atención es decidir para cada atributo si su valor va a ser el mismo para todas las instancias de esa clase, o si cada instancia puede tener un valor distinto.

- Si el valor del atributo va a ser el mismo para todas las instancias de la clase, inicialícelo al declarar el atributo.
- Si cada instancia va a poder asignarle un valor distinto a ese atributo, agregue el atributo como parámetro al constructor y añadir una inicialización de la forma `"this.field = campo"` al constructor.

2.b Colaboraciones entre Clases (Interacciones)

- a) Utilice un tipo primitivo si hay una correspondencia obvia entre la información y un tipo atómico que el lenguaje ya soporta. Ejemplos: mediciones numéricas, nombres, interruptores de encendido/apagado.

³ Adaptado de: **"How to Design Classes - Data: Structure and Organization (June 15 2012 draft)"** (Cómo Diseñar Clases – Datos: Estructura y Organización, Borrador del 15 de Junio de 2012). *Matthias Felleisen, Matthew Flatt, Robert Bruce Findler, Kathryn E. Gray, Shriram Krishnamurthi & Viera K. Proulx*. Versión digital disponible gratuitamente en <https://felleisen.org/matthias/htdc.html>

⁴ Traducción y adaptación al castellano: Leandro Doctors <ldoctors@untref.edu.ar>
Los primeros tres pasos corresponden a los primeros tres pasos de la receta básica de programas.

b) Agregue una nueva clase si necesita representar cierta información que consiste en varias otras piezas de información. Esto es típicamente el caso con cosas que tienen varias propiedades. Cada propiedad se convierte en un campo de la clase. Ejemplos: posiciones en el espacio, direcciones.

c) Composición: Pida a una clase A que haga referencia a otra clase B si algún componente de una pieza de información es en sí misma la composición de varias piezas de Información. Es decir, si al crear una instancia de la clase A, ésta contiene una instancia de la clase B. Ejemplos: una entrada de registro contiene una fecha; un tren se refiere a un horario.

d) Herencia: Utilice una unión de clases si una recopilación de información consiste en varias subclases distintas. La unión representa a toda la colección de objetos frente al resto del programa (como un tipo). Ejemplos: trenes locales y expresos, formas geométricas.

e) ¡Favorezca la Composición! ¡Evite a toda costa la Herencia!

f) El diagrama de clases es autorreferencial si necesita representar piezas de información que consisten en un número desconocido e ilimitado de otras piezas de información. Ejemplos: listas de lectura, árboles genealógicos, sistemas fluviales, carpetas de archivos.

3 de las Definiciones de Clase a Ejemplos Funcionales

Por un lado, obtenga ejemplos de información y representelos como instancias de las clases. Por el otro lado, invente instancias de las clases e interpretelas como información. Tenga presente que algunos ejemplos de datos no tienen explicación en el espacio del problema. Advierta de tales casos con comentarios en el código. (¡Recuerde que Ud. es uno de los posibles futuros lectores de su propio código!)

4 [Plantilla, Pruebas, y Definición]

(Ver receta básica de programas.)