Patrones de Diseño I

¿Qué son los patrones de diseño?

- Son formas bien conocidas y probadas de resolver problemas recurrentes.
- Ampliamente utilizados en las disciplinas creativas y técnicas. Como un guionista que crea guiones a partir de patrones argumentales (comedia, drama, ciencia-ficción).
- Se presentan bajo contextos determinados
- Ayudan a reducir el tiempo de diseño.

Elementos Básicos del Diseño

- Tipos, normalmente clases, aunque es válido para lenguajes que no soportan clases.
- Métodos, operaciones que se pueden realizar sobre un tipo.
- Campos, que pueden ser variables o atributos.
- Objetos, instancias de un tipo determinado y que tiene entidad por sí mismo.

Estructura:

- Nombre, nombre corto, bien definido y fácil de manejar.
- Problema y contexto, el patrón debe resolver un problema en concreto y el contexto describe las precondiciones que deben ser cumplidas.
- Solución, la solución que proporciona un patrón se describe genéricamente y nunca ligada a ninguna implementación.
- Ventajas y desventajas, aplicar un patrón implica evaluar los beneficios y las posibles complicaciones.

Patrones Elementales

Algunos patrones elementales emergidos de la relación <method-method>, es decir, «cuando un método A llama desde su implementación a un método B».

Esta relación es transitiva, es decir, no importa lo largo que sea la cadena de invocaciones entre un método y otro para decir que el método A depende de B. Por ejemplo, en el siguiente fragmento de código a.x() depende de b.y()

Patrones Elementales

```
1  class B {
2      void y() {
3          // ...
4      }
5  }
6
7  class A {
8      B b;
9      void x() {
10         b.y();
11         // ...
12     }
13 }
14
15 main () {
16     A a;
17     a.x();
18 }
```

```
1  class B {
2      void y() {
3          // ...
4      }
5  }
6
7  class C {
8      B b;
9      void z() {
10         b.y();
11     }
12 }
13
14 class A {
15     C c;
16     void x() {
17         c.z();
18         // ...
19     }
20 }
21
22 main () {
23     A a;
24     a.x();
25 }
```

Patrones Elementales

- **Recursión,** la recursión podemos definirla como la relación method-method en el que los objetos y los métodos implicados son los mismos.

- **Conglomeración,** si los objetos son el mismo pero se invoca a un método diferente. Esto ocurre con bastante frecuencia dentro de una clase en la que diferentes métodos se invocan entre sí.

Patrones Elementales

Recursión

```
1   class A {
2       void x() {
3           if ( condition ) {
4               x();
5           }
6           else {
7               return;
8           }
9       }
10  }
11
```

Conglomeración

```
1   class A {
2       void x() {
3           ...
4       }
5
6       void y() {
7           x();
8           ...
9       }
10  }
```

Patrones Elementales

- **Redirección,** una redirección se produce cuando un objeto recibe una invocación a un método y éste invoca «el mismo» método de otro objeto. Cuando se habla de «el mismo» nos referimos a un método que «hace lo mismo» en términos de *diseño* aunque su implementación no sea exactamente la misma. Por ejemplo, la redirección es común en sistemas en donde el trabajo se divide en diferentes objetos (instancias) y un manager reparte el trabajo

.

- **Delegación,** decimos que un método delega funcionalidad en otro cuando se invoca a un método diferente del mismo objeto. Un ejemplo típico son las clases que tienen referencias a objetos que hacen diferentes trabajos por ella.

Patrones Elementales

Listado 4.3: *Delegation* (ejemplo)

```
1  class WheelMaker {
2      Wheel makeWheel() {
3          // do the wheel and return it
4      }
5  }
6
7  class CarMaker {
8      WheelMaker w;
9      Car makeCar() {
10         Wheel wheel1 = w.makeWheel();
11         // ... do more stuff
12     }
13 }
14
15 main() {
16     WheelMaker w;
17     w.makeCar();
18 }
```

Listado 4.4: *Redirection*(ejemplo)

```
1  class Printer {
2      void print(Document d) {
3          //...
4      }
5  }
6
7  class PrinterManager {
8      Printer p;
9      void print(Document d) {
10         p.print(d);
11         //...
12     }
13 }
14
15 main() {
16     Document d;
17     PrinterManager m;
18     m.print(d);
19 }
```

Patrones Elementales

	Method Similar	Method Dissimilar
Object similar	*Recursion*	*Conglomeration*
Object dissimilar	*Redirection*	*Delegation*

Referencias

- Desarrollo de Videojuegos. Un enfoque Práctico.: Volumen 1. Arquitectura del Motor: Volume 1 (http://www.cedv.es/) (https://drive.google.com/file/d/1zbYTO9wH5bxIyfsdVgQ_0QmPr8IAY1ca/view?usp=sharing)

- Elemental Design Patterns 1st Edition by Jason McC. Smith (https://drive.google.com/file/d/1k6Eg43abqqlEG0HGyK_CQhBZsprWKp52/view?usp=sharing)