

PATRON FLYWEIGHT (Peso Ligero)

- Propósito
 - Usa compartimiento para permitir un gran numero de objetos de grano fino de forma eficiente.

PATRON FLYWEIGHT (Peso Ligero)

- Aplicabilidad, solo si se cumple todo esto:
 - Una aplicación utiliza un gran numero de objetos.
 - Los costes de almacenamiento son elevados debido a la gran cantidad de objetos.
 - La mayor parte del estado del objeto puede hacerse extrínseco.
 - Muchos grupos de objetos pueden reemplazarse por relativamente pocos objetos compartidos.
 - La aplicación no depende de la identidad de un objeto.

PATRON FLYWEIGHT (Peso Ligero)

- Consecuencias:
 - Pueden introducir costes en tiempo de ejecución asociados con la transferencia, búsqueda y calculo del estado extrínseco.
 - Estos costos se ven compensados por el ahorro de espacio de almacenamiento, que se incrementa a medida que se comparten más objetos.

PATRON FLYWEIGHT (Peso Ligero)

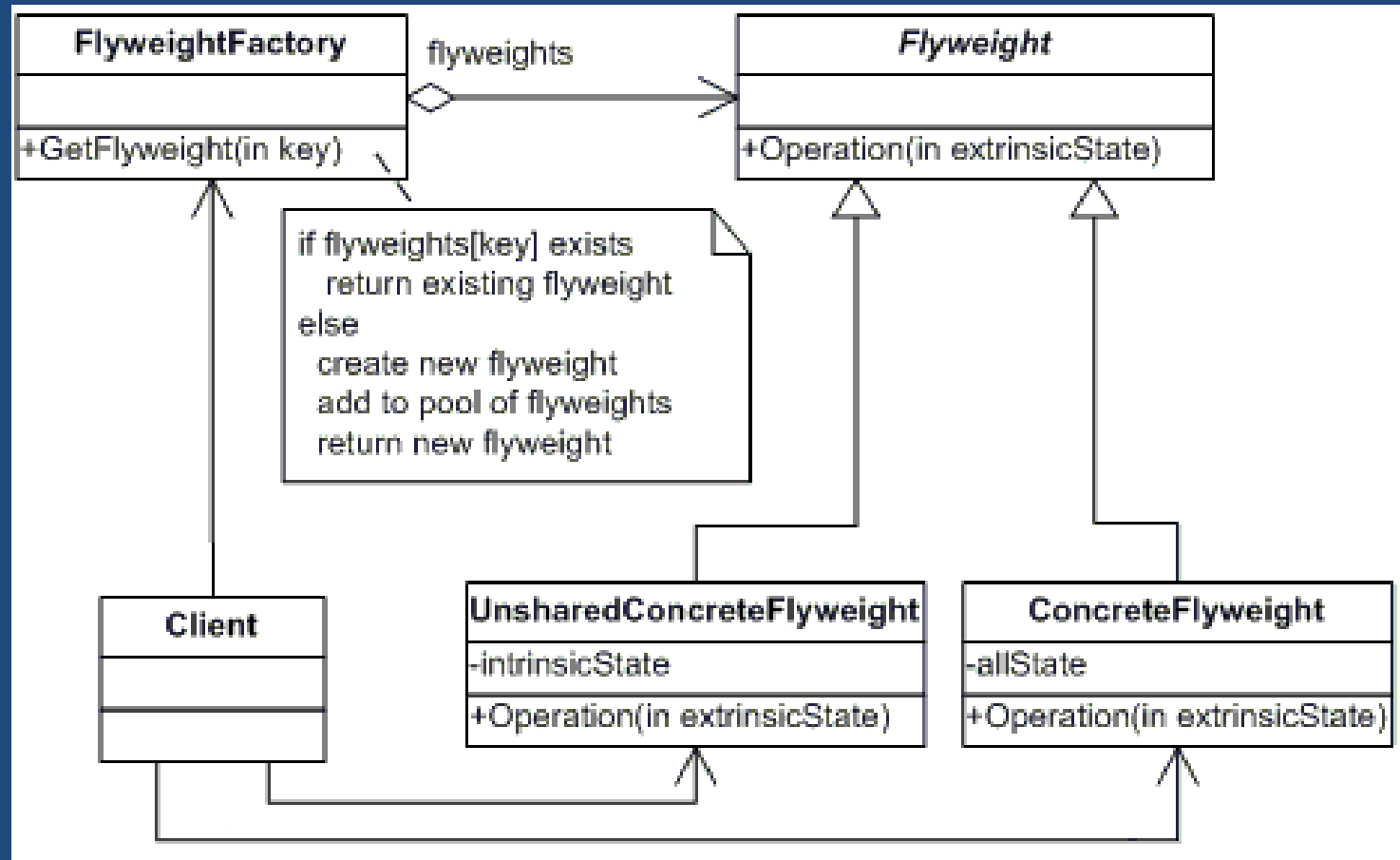
- Forma de uso
 - Dividir el objetivo principal en estados: Estado Intrínseco (elementos que se puedan compartir o son comunes) Estado Extrínseco (elementos particulares a cada tipo)
 - Retirar los elementos con estado extrínseco de los atributos de la clase, y añádale más bien una llamada a métodos
 - Crear una fábrica que pueda almacenar y reutilizar las instancias existentes de clases
 - El cliente debe usar la fábrica en vez de utilizar el operador new si requiere de creación de objetos

PATRON FLYWEIGHT (Peso Ligero)

- Forma de uso
 - ¿Qué datos son factorizables?
 - ¿Es posible aglutinar en una parte del objeto un conjunto de información redundante que no es probable que vaya a cambiar? Esa información ¿ocupa mucho espacio en memoria?

PATRON FLYWEIGHT (Peso Ligero)

- Estructura



PATRON FLYWEIGHT (Peso Ligero)

