



USAC
TRICENTENARIA
Universidad de San Carlos de Guatemala

CLASE 6

PATRONES DE DISEÑO


Análisis y Diseño de Sistemas 2

Aux. Kenny Miguel Eguizabal

Patrones de Estructura

- Los patrones estructurales son los que se preocupan por la composición de las clases y objetos. Ellos utilizan la herencia para componer las interfaces y definir maneras para componer objetos para obtener nuevas funcionalidades. Afectando la manera en que los objetos se conectan unos con otros y para asegurar que los cambios en el sistema no requieran cambiar esas relaciones (conexiones).

Patrones de Estructura

- **Adapter:** permite clases con interfaces incompatibles trabajar juntas envolviendo su propia interface con una clase existente.
 - **Bridge:** desacopla una abstracción de su implementación para que las dos puedan variar independientemente.
 - **Composite:** compone cero o mas objetos similares que pueden ser manipulados como solo un objeto.
- 

Patrones de Estructura

- **Decorator:** dinámicamente agrega o sobrescribe (overrides) comportamiento en un método existente del objeto.
- **Facade:** Fachada provee una interfaz simplificada o unificada para un largo trozo de codificación, o múltiples interfaces.
- **Flyweight:** reduce el costo de crear y manipular grandes grupos de objetos.
- **Proxy:** provee un lugar para colocar otro objeto que se encarga del control del acceso, reduciendo el costo y la complejidad.

Referencias Patrones de Estructura

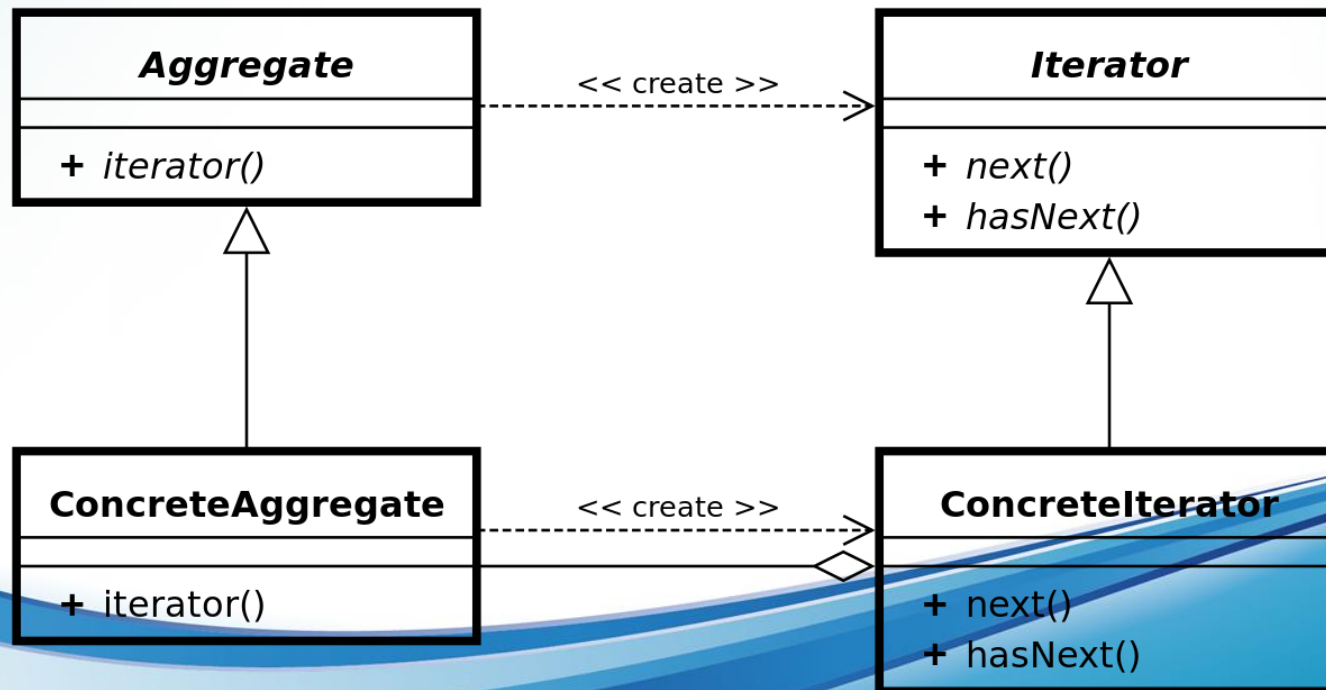
- <https://es.wikipedia.org/wiki/Adapter> (patr%C3%B3n de dise%C3%B1o)
- <https://es.wikipedia.org/wiki/Bridge> (patr%C3%B3n de dise%C3%B1o)
- <https://es.wikipedia.org/wiki/Bridge> (patrón de diseño)
- <https://es.wikipedia.org/wiki/Composite> (patr%C3%B3n de dise%C3%B1o)
- <https://es.wikipedia.org/wiki/Decorator> (patr%C3%B3n de dise%C3%B1o)
- <https://es.wikipedia.org/wiki/Facade> (patr%C3%B3n de dise%C3%B1o)
- <https://es.wikipedia.org/wiki/Flyweight> (patr%C3%B3n de dise%C3%B1o)
- <https://es.wikipedia.org/wiki/Proxy> (patr%C3%B3n de dise%C3%B1o)

Patrones de Comportamiento

- Los patrones de comportamiento o composición, se definen como patrones de diseño de software que ofrecen soluciones respecto a la interacción y responsabilidades entre clases y objetos, así como los algoritmos que encapsulan.

Patrones de Comportamiento

- **Iterator:** Permite realizar recorridos sobre objetos compuestos independientemente de la implementación de estos.

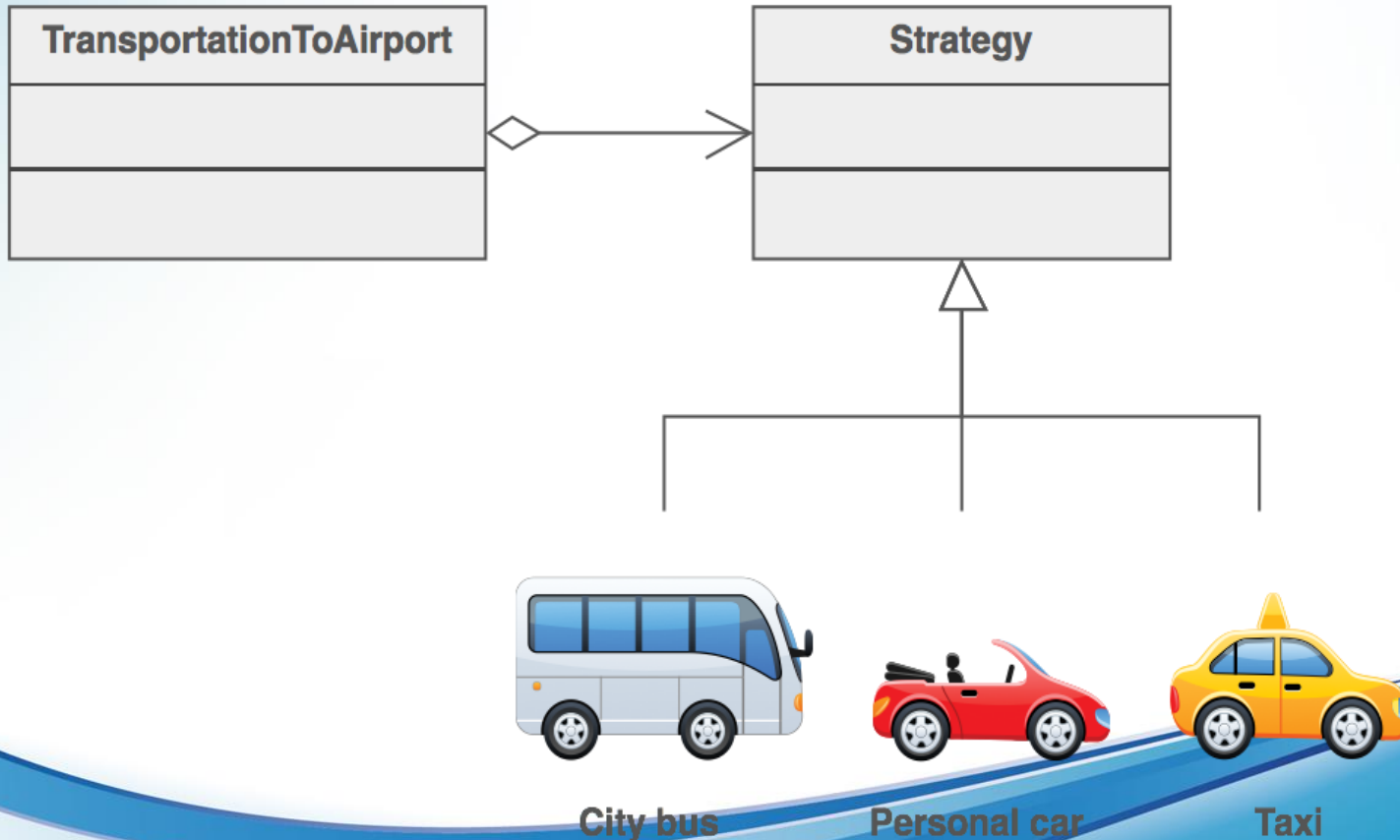


Patrones de Comportamiento

- **Strategy:** El patrón Strategy es un patrón de diseño para el desarrollo de software. Se clasifica como patrón de comportamiento porque determina cómo se debe de realizar el intercambio de mensajes entre diferentes objetos para resolver una tarea.
- El patrón *estrategia* permite mantener un conjunto de algoritmos de entre los cuales el objeto cliente puede elegir aquel que el conviene e intercambiarlo dinámicamente según sus necesidades.

Patrones de Comportamiento

- STRATEGY

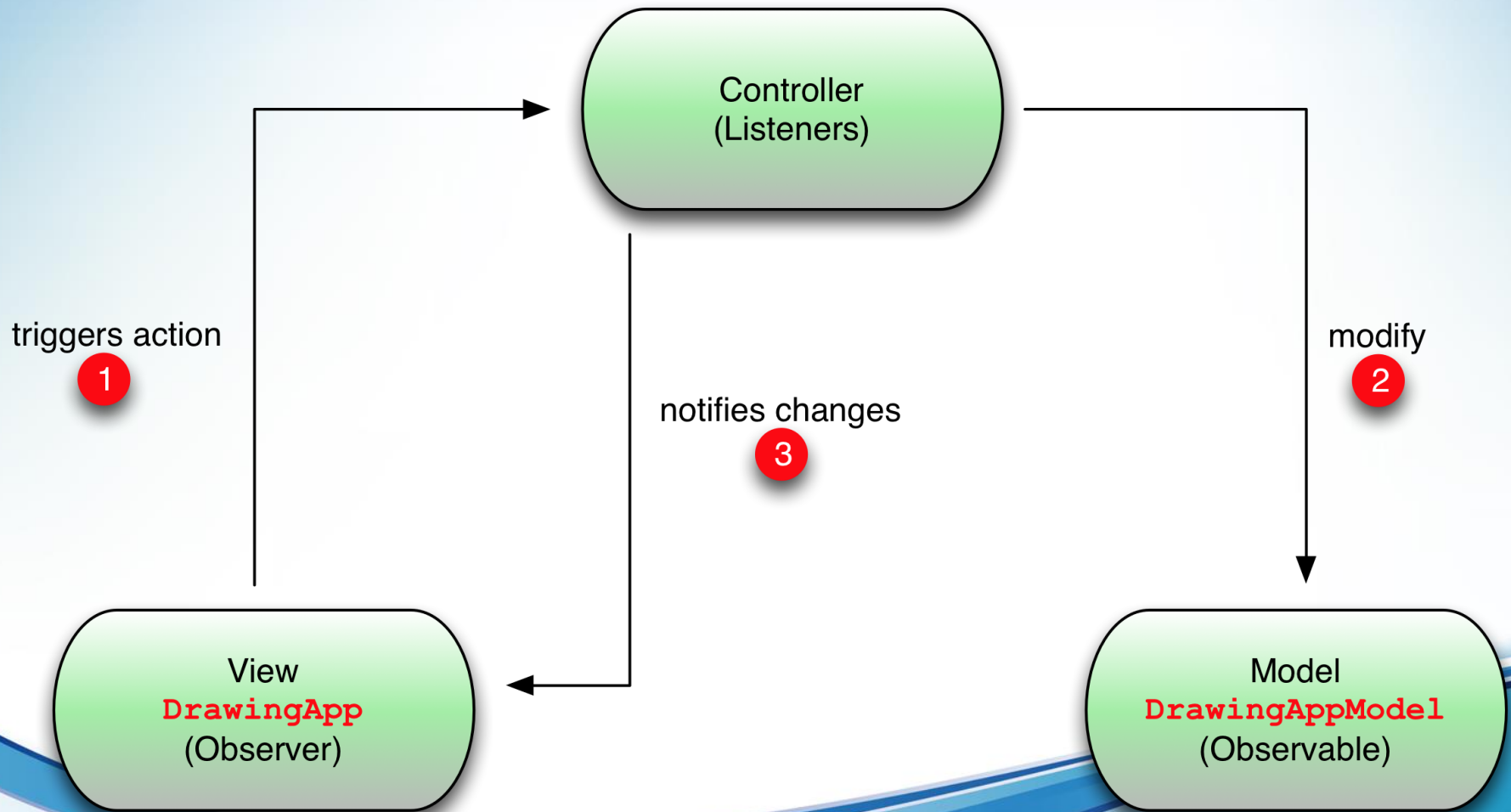


Concrete strategies (options)

Patrones de Comportamiento

- **OBSERVER:** El patrón observador es un patrón de diseño que define una dependencia del tipo uno-muchos entre objetos, de manera que cuando uno de los objetos cambia su estado, notifica este cambio a todos los dependientes.
- La variación de la encapsulación es la base de muchos patrones de comportamiento, por lo que cuando un aspecto de un programa cambia frecuentemente, estos patrones definen un objeto que encapsula dicho aspecto. Los patrones definen una clase abstracta que describe la encapsulación del objeto.

Observer



Observer

- Este patrón también se conoce como el patrón de *publicación-inscripción*. El objeto inicial, que se le puede llamar «Sujeto» contiene atributos mediante los cuales cualquier objeto Observador o vista se puede suscribir a él pasándole una referencia a sí mismo. El sujeto mantiene así una lista de las referencias a sus observadores. Los observadores a su vez están obligados a implementar unos métodos determinados mediante los cuales el Sujeto, es capaz de notificar a sus observadores suscritos, los cambios que sufre para que todos ellos tengan la oportunidad de refrescar el contenido representado.

¡GRACIAS!