

El Patrón State

Manuel Alejandro Bacallado López

Nombre del Patrón y su Clasificación

- Nombre
 - Estado - State
- Clasificación
 - Comportamiento de Objetos
- También conocido como
 - Objetos para los Estados - Objects for States

Propósito

- ¿Qué hace el patrón Estado?
 - Define una familia de clases.
 - Encapsula cada una de ellas y las hace intercambiables.
 - Permite a un objeto cambiar su comportamiento según el estado interno en el que se encuentre.

Motivación

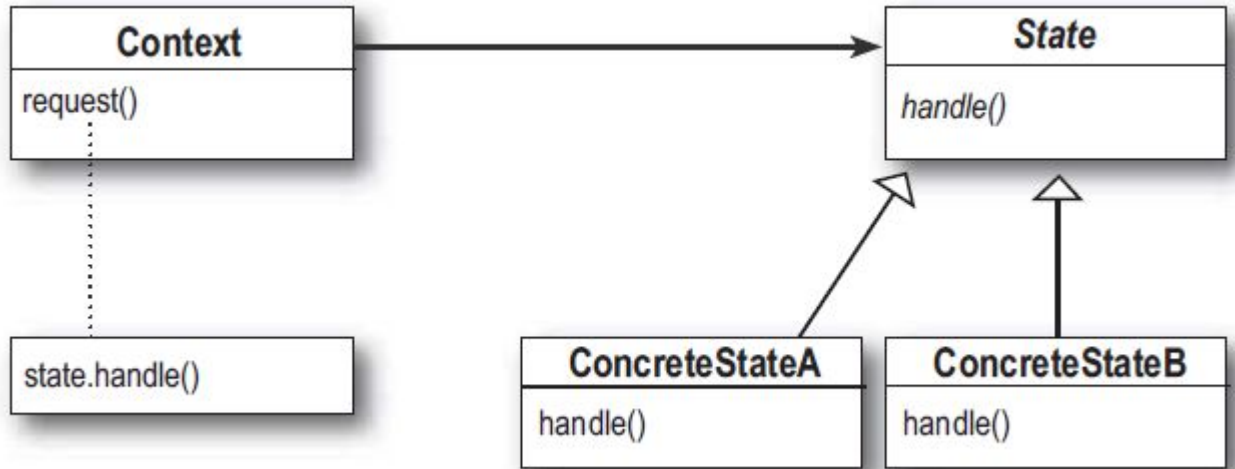
- ¿Qué problema de diseño resuelve el patrón de diseño State Pattern?
 - Duplicidad del código.

Aplicabilidad - Use el patrón estado cuando...

- El comportamiento de un objeto depende de su estado y tiene que cambiarlo en tiempo de ejecución.
- Una clase define muchos estados, y estos a su vez se presentan como múltiples sentencias condicionales en sus operaciones.
 - En vez de tener muchos métodos y atributos que representen los estados, se mueven a su propia clase Estado.

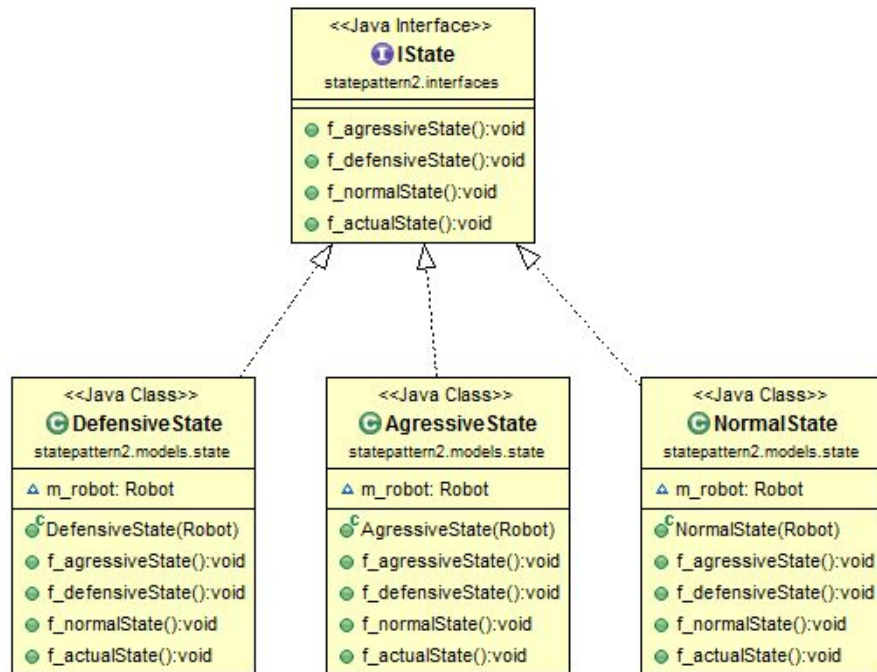
Estructura - I

- Una representación gráfica de las clases



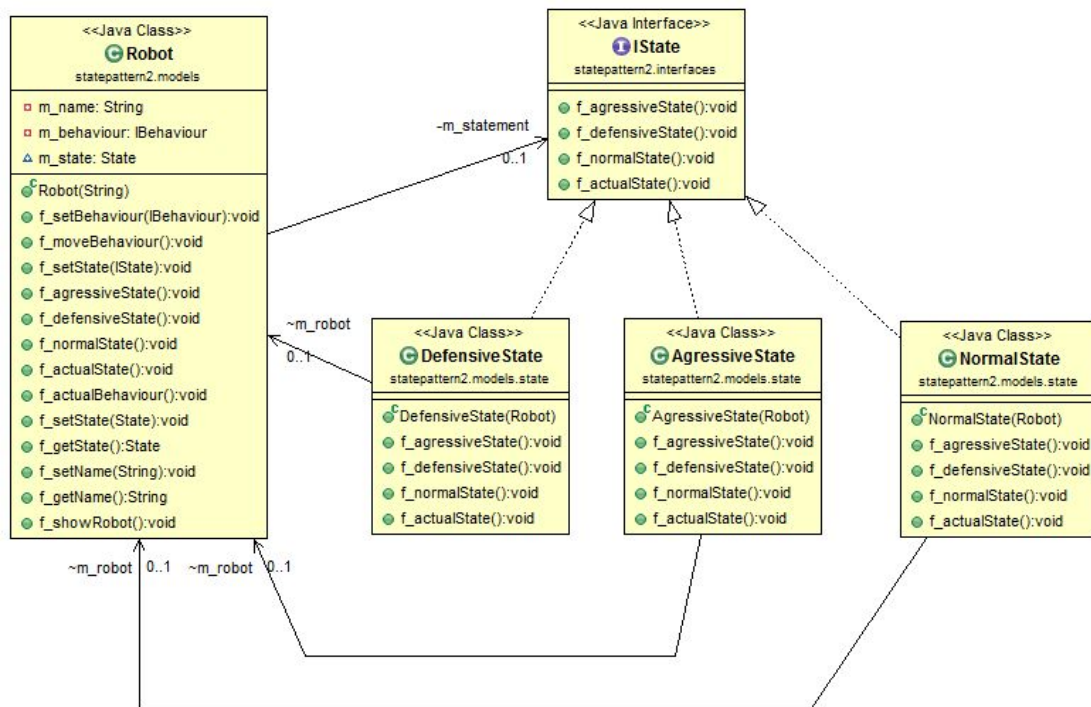
Estructura - II

- Una representación gráfica de las clases



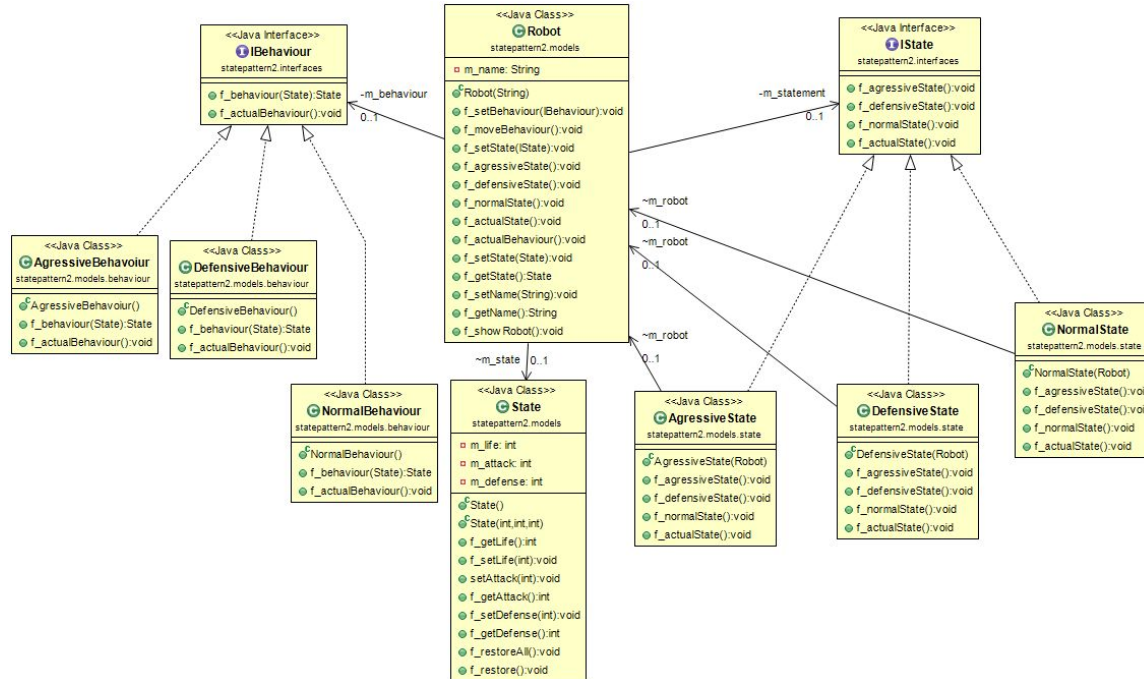
Estructura - III

- Una representación gráfica de las clases



Estructura - IV

- Una representación gráfica de las clases



Participantes

- Estado
 - Define una interfaz para el encapsulamiento de las responsabilidades asociadas con el estado particular del contexto.
- Estado Concreto
 - Implementa el comportamiento de un estado del contexto.
- Contexto
 - Define la interfaz con el cliente.
 - Mantiene una instancia del Estado Concreto que define cuál es el estado actual.

Colaboraciones

- El Contexto delega al Estado Concreto sus peticiones y puede pasarse a sí mismo como parámetro del Estado Concreto.
- Los clientes pueden configurar un Contexto con un Estado Concreto y no tiene que acceder directamente con el estado.
- Tanto el Contexto como Estado Concreto pueden decidir que estado sucede a otro y bajo qué circunstancias.

Consecuencias

● Ventajas

- Fácil localización de las responsabilidades de los estados específicos, facilitando la ampliación de estados.
- Cambios de estados explícitos.
- Los estados permiten eliminar sentencias condicionales.
- Selección de estados. El mismo estado puede admitir distintas implementaciones.

● Desventajas

- Las clases tienen que conocer los diferentes estados.
- Costes de comunicación entre Estado y Contexto.
- Mayor número de objetos. Los estados incrementan el número de objetos. (Subclases).

Patrones relacionados

- ¿Qué patrones están relacionados con este?
 - Flyweight: Explica cuando y como los objetos pueden ser compartido.
 - State Pattern a menudo se utilizan en Singleton Pattern.

Bibliografía

- E. Freeman, E. Freeman. Head First Design Pattern, O'Reilly, 2004, Chapter 10
- Wikipedia
 - [https://es.wikipedia.org/wiki/State_\(patr%C3%B3n_de_dise%C3%B1o\)](https://es.wikipedia.org/wiki/State_(patr%C3%B3n_de_dise%C3%B1o))
- Design Patterns with UML
 - <http://design-patterns-with-uml.blogspot.com.ar/2013/02/state-pattern.html>

Fin

Código de ejemplo: <https://github.com/MBacallado/statepattern.git>

Gracias por su atención.

Autor: Manuel Alejandro Bacallado López

E-mail: manuelbacallado89@gmail.com