AUX6

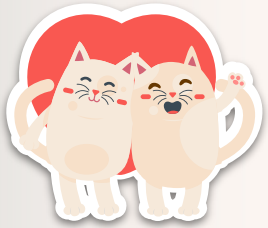
DESIGN PATTERNS

Sofía Bobadilla Ponce
sofia.bobadilla@ug.uchile.cl
@xpanpan en Telegram

TABLE OF CONTENTS

01

Design Patterns



Design Patterns

Suelen ser la mejor solución a un tipo de problema
(Re pensadas y testeadas)

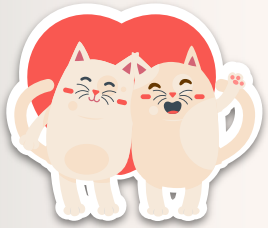
Dan respuesta a problemas típicos en el desarrollo de software

NUNCA HACER COPY/PASTE DE UN PATRÓN DE DISEÑO

Design Patterns- Clasificación

Se clasifican en 3 grandes categorías:

- **Creacionales:** Se encargan de la creación de objetos
- **Estructurales:** Se encargan de componer las clases y objetos manteniendo flexibilidad
- **De comportamiento:** Se encargan de generar comunicación efectiva y asignar responsabilidades



Design Patterns

Creacionales:

- Builder
- Abstract factory

Estructurales:

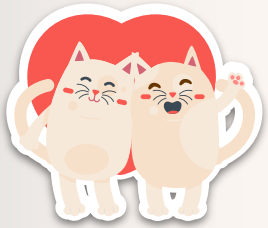
- Adapter
- Facade

De comportamiento:

- Template
- Observer

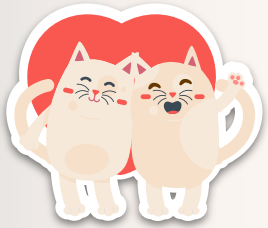
02

STATE PATTERN



STATE PATTERN

Veamos un ejemplo de los estados en un juego y su conexión con el controlador



STATE PATTERN

Exception y se queda en la fase

Jugador apreta botón de mover

Llamado a controler.trytomove()

SI NO se puede mover

Revisar la fase

SI se puede mover

Controler.movePlayer() y se actualizan las fases si corresponde

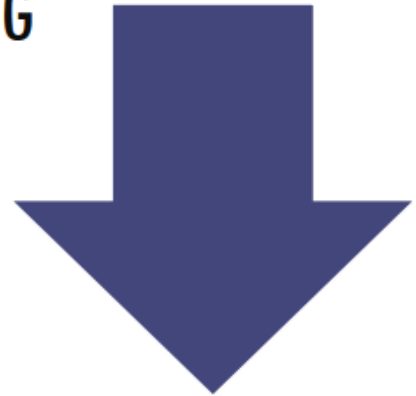
03

OBSERVER

MY LASERS TRACE EVERYTHING YOU DO

Observer pattern

- **Problema:**
  - Necesito saber cuando ocurre algún evento en específico sin tener que estar preguntando constantemente para saber su estado
- **Solución:**
  - Hacer que el evento nos notifique cuando ocurre
- Para poder aprovechar realmente este patrón las notificaciones deben hacerse en paralelo a todos los suscriptores que esperan por el evento



De pana

- Cumple el principio **Open/Closed**
- Se pueden agregar **nuevos suscriptores en tiempo de ejecución**

De perro

- El paralelismo hace que **no se pueda saber el orden** en el que se harán las notificaciones
- **Ineficiente** si hay demasiados o si hay observadores que son observados

THE EYE OF THE BEHOLDER

```
public class Player extends Observable {
  public void playCard(ICard card) {
    ...
    setChanged();
    notifyObservers(card);
  }
}
```


```
public class Controller implements Observer {
  private Player player;
  public Controller() {
    player = new Player();
    player.addObserver(this);
  }
  @Override
  public void update(final Observable o,
      final Object arg) {
    if (arg instanceof ICard) {
      onCardPlayed((ICard) arg);
    }
  }
  private void onCardPlayed(ICard arg) { ... }
}
```

IS THERE ANYBODY LISTENING TO ME?

```java
public class Player {
  private PropertyChangeSupport cardPlayedNotification =
    new PropertyChangeSupport(this);

  public void addPlayCardListener(
      PropertyChangeListener listener) {

    cardPlayedNotification.addPropertyChangeListener(
      listener);
  }
}
```


```java
public class Controller
    implements PropertyChangeListener {

  private Player player;

  public Controller() {
    player = new Player();
    player.addPlayCardListener(this);
  }

  @Override
  public void propertyChange(PropertyChangeEvent evt) {
    if (evt.getNewValue() instanceof ICard) {
      onCardPlayed((ICard) evt.getNewValue());
    }
  }

  private void onCardPlayed(ICard arg) { ... }
}
```

‘CAUSE I CAN ONLY HANDLE ONE AT A TIME


```
public class Controller {

  private Player player;

  private PlayedCardHandler
    playedCardHandler = new PlayedCardHandler(this);

  public Controller() {
    player = new Player();
    player.addPlayCardListener(playedCardHandler);
  }

  public void onCardPlayed(ICard arg) { ... }
}
```

```
public class PlayedCardHandler
    implements PropertyChangeListener {

  private Controller controller;

  public PlayedCardHandler(Controller controller) {
    this.controller = controller;
  }

  @Override
  public void propertyChange(PropertyChangeEvent evt) {
    controller.onCardPlayed((ICard) evt.getNewValue());
  }
}
```

04

Consultas

Link al repo:

https://github.com/Sofi1410/TareaMetodologia

Thank U for coming

THANKS!

Do you have any questions?

@xpanpan en Telegram

Please keep this slide for attribution


slidesgo