

Նախագծման Ձևանմուշներ: Mediator

Հրաչյա Թանդիլյան

2020

Mediator

Նպատակը

Սահմանել օբյեկտ, որը կինկապսուլացնի այն, թե ինչպես են փոխհամագործակցում մի խումբ օբյեկտներ:

Այս Ն.Ձ. նպաստում է փոխկապակցվածության թուլացմանը, կանխելով օբյեկտների միմյանց ուղղակի հղվելու անհրաժեշտությունը և թույլ է տալիս անկախ կերպով փոփոխել նրանցով փոխազդեցությունը:

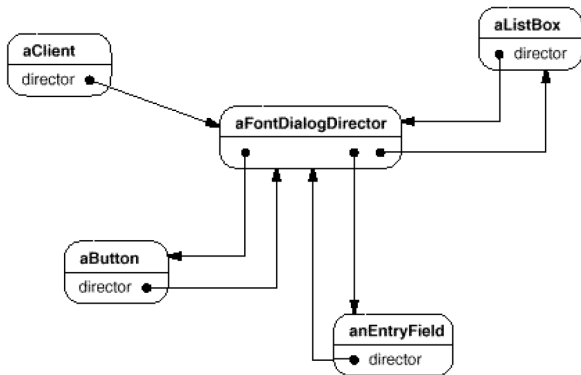
Նաև հայտնի է որպես

- Այլ լայնորեն կիրառվող անուններ չկան:

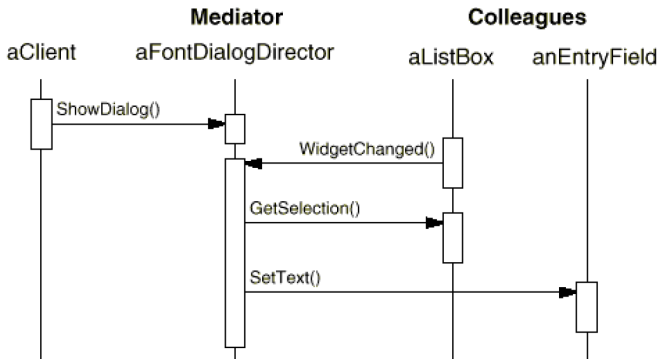
Մոտիվացիան



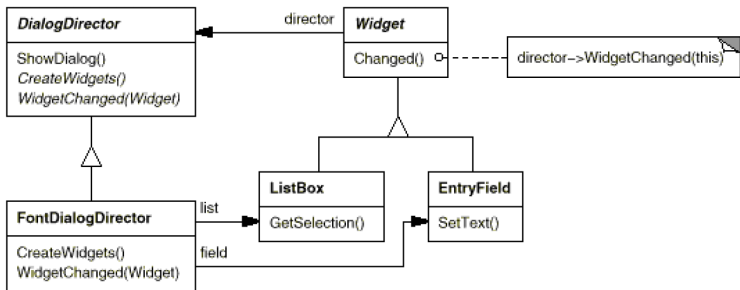
Մոտիվացիան



Մոտիվացիան



Մոտիվացիան

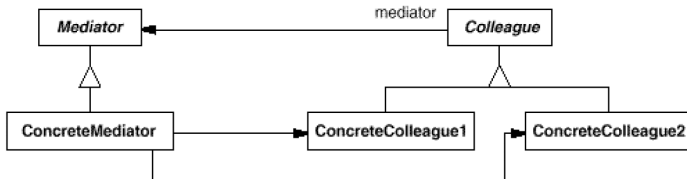


Կիրառելիությունը

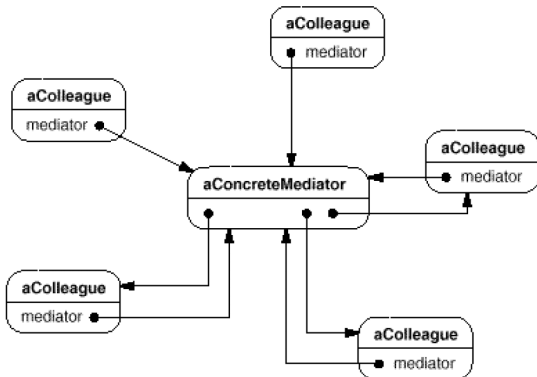
Այս Ն.Ձ. պետք է օգտագործել երբ.

- Ա** Մի խումբ օբյեկտներ փոխազդում են լավ սահմանված և կոմպլեքս ձևով:
- Բ** Օբյեկտի վերօգտագործումը դժվար է նրա շատ օբյեկտների հետ փոխազդեցության հետևանքով:
- Գ** Շատ օբյեկտների մեջ բաժանված վարվելակերպը պետք է փոփոխելի լինի առանց մեծ քանակով ենթադասեր ստեղծելու:

Կառուցվածքը



Կառուցվածքը



Հետևանքները

Այս Ն.Ձ. ունի հետևյալ առավելություններն ու թերությունները.

- Ա Նվազեցնում է ժառանգության անհրաժեշտությունը:
- Բ Բաժանում է գործընկեր դասերին:
- Գ Պարզեցնում է օբյեկտների հաղորդակցումը:
- Դ Աբստրակտացնում է օբյեկտների փոխհամագործակցությունը:
- Ե Կենտրոնացնում է դեկլարումը:

Իրականացումը

Ա Արստրալտ Mediator դասի բացակայություն:

Բ Colleague-Mediator հաղորդակցություն:

Օրինակ

```
class DialogDirector {  
  
    public:  
        virtual ~DialogDirector();  
        virtual void ShowDialog();  
        virtual void WidgetChanged(Widget*) = 0;  
  
    protected:  
        DialogDirector();  
        virtual void CreateWidgets() = 0;  
};
```

Օրինակ

```
class Widget {  
  
    public:  
        Widget(DialogDirector*);  
        virtual void Changed();  
        virtual void HandleMouse(MouseEvent& event);  
  
    private:  
        DialogDirector* director;  
};  
  
void Widget::Changed () {  
    director->WidgetChanged(this);  
}
```

Օրինակ

```
class ListBox : public Widget {  
  
public:  
    ListBox(DialogDirector*);  
  
    virtual const char* GetSelection();  
    virtual void SetList(List<char*>* listItems);  
    virtual void HandleMouse(MouseEvent& event);  
  
    // Other methods  
};
```

Օրինակ

```
class EntryField : public Widget {  
  
public:  
    EntryField(DialogDirector*);  
  
    virtual void SetText(const char* text);  
    virtual const char* GetText();  
    virtual void HandleMouse(MouseEvent& event);  
  
    // Other methods  
};
```

Օրինակ

```
class Button : public Widget {  
  
    public:  
        Button(DialogDirector*);  
  
        virtual void SetText(const char* text);  
        virtual void HandleMouse(MouseEvent& event);  
  
        // Other methods  
};  
  
void Button::HandleMouse(MouseEvent& event) {  
    // Local event handling  
    Changed();  
}
```


Օրինակ

```
class FontDialogDirector : public DialogDirector {  
  
public:  
    FontDialogDirector();  
    virtual ~FontDialogDirector();  
    virtual void WidgetChanged(Widget*);  
  
protected:  
    virtual void CreateWidgets();  
  
private:  
    Button*      ok;  
    Button*      cancel;  
    ListBox*     fontList;  
    EntryField*  fontName;  
};
```

Օրինակ

```
void FontDialogDirector::CreateWidgets () {
```

```
    ok = new Button(this);
```

```
    cancel = new Button(this);
```

```
    fontList = new ListBox(this);
```

```
    fontName = new EntryField(this);
```

```
    // Fill the listBox with the available font names
```

```
    // Assemble the widgets in the dialog
```

```
}
```

Օրինակ

```
void FontDialogDirector::WidgetChanged(Widget* theChangedWidget) {  
    if (theChangedWidget == fontList) {  
        fontName->SetText(fontList->GetSelection());  
    } else if (theChangedWidget == ok) {  
        // apply font change and dismiss dialog  
    } else if (theChangedWidget == cancel) {  
        // dismiss dialog  
    }  
}
```

Առնչվող Նախագծման Ձևանմուշները

- Facade

- Observer