

Նախագծման Ձևանմուշներ: Bridge

Հրաչյա Թանդիլյան

2020

Bridge

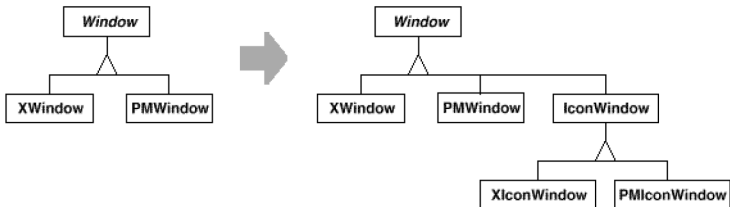
Նպատակը

Առանձնացնում է արստրակցիան (ինտերֆեյսը) իրականացումից այնպես, որ այդ երկուսը կարող են փոփոխվել միմյանցից անկախ:

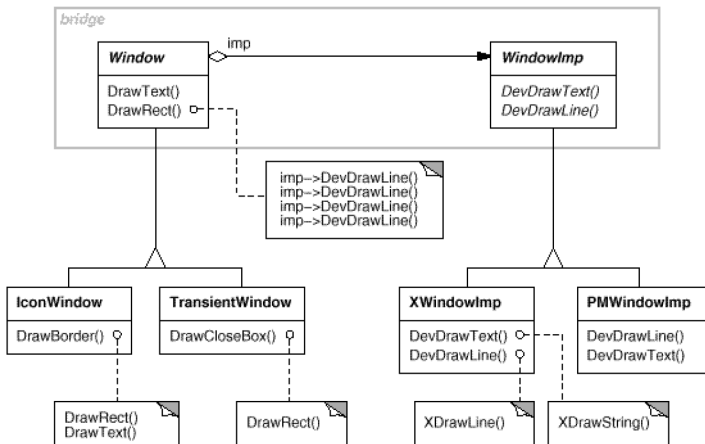
Նաև հայտնի է որպես

- Handle / Body

Մոտիվացիան



Մոտիվացիան



Կիրառելիությունը

Այս Ն.Ձ. պետք է օգտագործել երբ.

- Ա Անհրաժեշտ է արստրակցիաի և իրականացման մեջ մշտական կապ չստեղծել:

Կիրառելիությունը

Այս Ն.Ձ. պետք է օգտագործել երբ.

- Ա** Անհրաժեշտ է արստրակցիաի և իրականացման մեջ մշտական կապ չստեղծել:
- Բ** և՛ արստրակցիան, և՛ իրականացումը պետք է ընդլայնելի լինեն ժառանգության միջոցով:

Կիրառելիությունը

Այս Ն.Ձ. պետք է օգտագործել երբ.

- Ա** Անհրաժեշտ է արստրակցիաի և իրականացման մեջ մշտական կապ չստեղծել:
- Բ** Ա՛ արստրակցիան, ա՛ իրականացումը պետք է ընդլայնելի լինեն ժառանգության միջոցով:
- Գ** Արստրակցիաի իրականացման մեջ փոփոխությունները օգտագործողների վրա չպետք է ազդեցություն ունենան, այսինքն նրանց կողմը չպետք է վերակոմպիլիացիաի ենթարկվի:

Կիրառելիությունը

Այս Ն.Ձ. պետք է օգտագործել երբ.

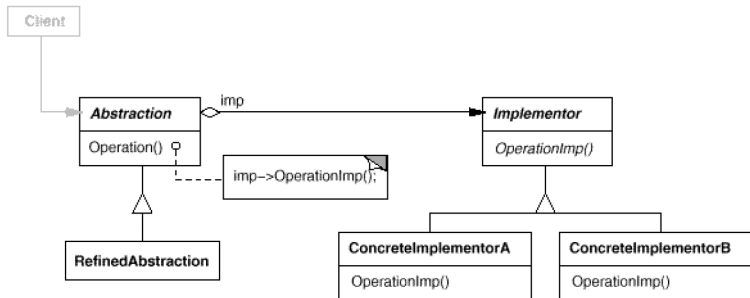
- Ա** Անհրաժեշտ է արստրակցիաի և իրականացման մեջ մշտական կապ չստեղծել:
- Բ** Ա՛ արստրակցիան, և՛ իրականացումը պետք է ընդլայնելի լինեն ժառանգության միջոցով:
- Գ** Արստրակցիաի իրականացման մեջ փոփոխությունները օգտագործողների վրա չպետք է ազդեցություն ունենան, այսինքն նրանց կողմը չպետք է վերակոմպիլիացիաի ենթարկվի:
- Դ** (C++) Անհրաժեշտ է արստրակցիաի իրականացումը ամբողջովին թաքցնել օգտագործողներից:

Կիրառելիությունը

Այս Ն.Ձ. պետք է օգտագործել երբ.

- Ա** Անհրաժեշտ է արստրակցիաի և իրականացման մեջ մշտական կապ չստեղծել:
- Բ** Ա՛ արստրակցիան, Ա՛ իրականացումը պետք է ընդլայնելի լինեն ժառանգության միջոցով:
- Գ** Արստրակցիաի իրականացման մեջ փոփոխությունները օգտագործողների վրա չպետք է ազդեցություն ունենան, այսինքն նրանց կողմը չպետք է վերակոմպիլիացիաի ենթարկվի:
- Դ** (C++) Անհրաժեշտ է արստրակցիաի իրականացումը ամբողջովին թաքցնել օգտագործողներից:
- Ե** Անհրաժեշտ է միևնույն իրականացումը օգտագործել մի քանի օբյեկտների համար և այդ փաստը պետք է թաքնված լինի օգտագործողից:

Կառուցվածքը



Հետևանքները

Այս Ն.Ձ. ունի հետևյալ առավելություններն ու թերությունները.

Ա Բաժանում է ինտերֆեյսը իրականացումից:

Հետևանքները

Այս Ն.Ձ. ունի հետևյալ առավելություններն ու թերությունները.

Ա Բաժանում է ինտերֆեյսը իրականացումից:

Բ Բարելավում է ընդլայնելիությունը:

Հետևանքները

Այս Ն.Ձ. ունի հետևյալ առավելություններն ու թերությունները.

- Ա** Բաժանում է ինտերֆեյսը իրականացումից:
- Բ** Բարելավում է ընդլայնելիությունը:
- Գ** Իրականացման մանրամասները քողարկում է օգտագործողից:

Իրականացումը

- Ա Միայն մեկ իրականացում:
- Բ Համապատասխան իրականացնող օբյեկտի ստեղծում:
- Գ Ընդհանուր իրականացնող օբյեկտների կիրառում:
- Դ Բազմակի ժառանգության կիրառում:

Օրինակ

```
class Window {  
  
    public:  
        Window(View* contents);  
        virtual void DrawContents();  
        virtual void Open(); // Close();  
        virtual void Iconify(); // Deiconify();  
        virtual void SetOrigin(const Point& at);  
        virtual void SetExtent(const Point& extent);  
        virtual void DrawRect(const Point&, const Point&);  
        // DrawLine, DrawPolygon, DrawText  
  
    protected:  
        WindowImp* GetWindowImp();  
        View* GetView();  
  
    private:  
        WindowImp* imp;  
        View* contents;  
};
```

Օրինակ

```
class WindowImp {  
  
public:  
    virtual void ImpTop() = 0;  
    virtual void ImpBottom() = 0;  
    virtual void ImpSetExtent(const Point&) = 0;  
    virtual void ImpSetOrigin(const Point&) = 0;  
  
    virtual void DeviceRect(Coord, Coord, Coord, Coord) = 0;  
    virtual void DeviceText(const char*, Coord, Coord) = 0;  
    virtual void DeviceBitmap(const char*, Coord, Coord) = 0;  
    // more functions for drawing on windows  
  
protected:  
    WindowImp();  
};
```


Օրինակ

```
WindowImp* Window::GetWindowImp() {  
  
    if (imp == 0) {  
        imp = WindowSystemFactory::Instance()->MakeWindowImp();  
    }  
    return imp;  
}  
  
void Window::DrawRect(const Point& p1, const Point& p2) {  
  
    WindowImp* imp = GetWindowImp();  
  
    imp->DeviceRect(p1.X(), p1.Y(), p2.X(), p2.Y());  
}
```

Օրինակ

```
class ApplicationWindow : public Window {  
  
public:  
    virtual void DrawContents() {  
        GetView()->DrawOn(this);  
    }  
};  
  
class IconWindow : public Window {  
  
public:  
    virtual void DrawContents() {  
        WindowImp* imp = GetWindowImp();  
        imp->DeviceBitmap(bitmapName, 0.0, 0.0);  
    }  
  
private:  
    const char* bitmapName;  
};
```

Առնչվող Նախագծման Ձևանմուշները

- Abstract Factory

- Adapter