

Նախագծման Ձևանմուշներ: Adapter

Հրաչյա Թանդիլյան

2020

Adapter

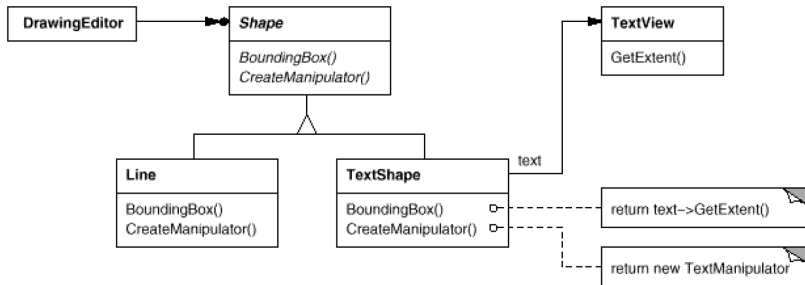
Նպատակը

Դասի ինտերֆեյսը փոխակերպում է օգտագործողի կողմից նախատեսված ինտերֆեյսի:

Նաև հայտնի է որպես

- Wrapper

Մոտիվացիան



Կիրառելիությունը

Այս Ն.Ձ. պետք է օգտագործել երբ.

- Ա** Ցանկանում եք օգտագործել գոյություն ունեցող դաս, բայց նրա ինտերֆեյսը չի համընկնում ձեզ անհրաժեշտ ինտերֆեյսի հետ:

Կիրառելիությունը

Այս Ն.Ձ. պետք է օգտագործել երբ.

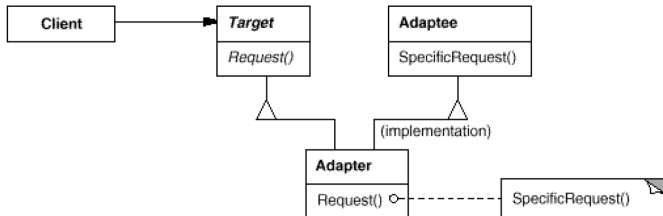
- Ա** Ցանկանում եք օգտագործել գոյություն ունեցող դաս, բայց նրա ինտերֆեյսը չի համընկնում ձեզ անհրաժեշտ ինտերֆեյսի հետ:
- Բ** Ցանկանում եք ստեղծել վերօգտագործելի դաս, որ կաշխատի չնախատեսված դասերի հետ, այսինքն այնպիսի դասերի հետ, որոնք պարտադիր չէ, որ ունենան պահանջվող ինտերֆեյսը

Կիրառելիությունը

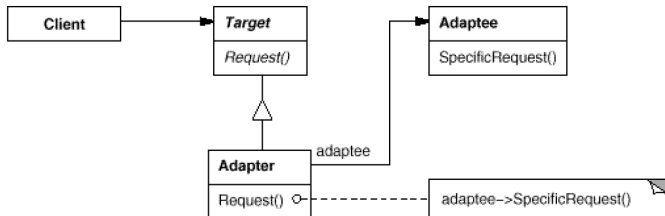
Այս Ն.Ձ. պետք է օգտագործել երբ.

- Ա** Ցանկանում եք օգտագործել գոյություն ունեցող դաս, բայց նրա ինտերֆեյսը չի համընկնում ձեզ անհրաժեշտ ինտերֆեյսի հետ:
- Բ** Ցանկանում եք ստեղծել վերօգտագործելի դաս, որ կաշխատի չնախատեսված դասերի հետ, այսինքն այնպիսի դասերի հետ, որոնք պարտադիր չէ, որ ունենան պահանջվող ինտերֆեյսը
- Գ** Միայն օբյեկտային Adapter-ի համար.
ցանկանում եք օգտագործել մի շարք գոյություն ունեցող ենթադասեր, բայց նրանց ինտերֆեյսի փոխակերպում յուրաքանչյուրից ժառանգելու միջոցով արակտիկ չէ:

Կառուցվածքը: Class Adapter



Կառուցվածքը: Object Adapter



Հետևանքները: Class Adapter

Այս Ն.Ձ. ունի հետևյալ առավելություններն ու թերությունները.

- Ա Փոխակերպում է կոնկրետ դասը և որպես հետևանք կիրառելի չէ երբ անհրաժեշտ է փոխակերպել դասը և նրա բոլոր ժառանգները:
- Բ Թույլ է տալիս վերասահմանել ադապտացվող դասի որոշ վարվելակերպ, քանի որ հանդիսանում է նրա ժառանգը:
- Գ Օգտագործում է միայն մեկ օբյեկտ, բացառելով հավելյալ վերահղման անհրաժեշտությունը:

Հետևանքները: Object Adapter

Այս Ն.Ձ. ունի հետևյալ առավելություններն ու թերությունները.

- Թույլ է տալիս միակ Adapter-ին աշխատել շատ ադապտացվող դասերի հետ, այսինքն ադապտացվող դասի և նրա բոլոր ժառանգերի հետ:
- Դժվարեցնում է ադապտացվող դասի վարվելակերպի վերասահմանումը:

Հետևանքները

Այս Ն.Ձ. ունի նաև հետևյալ դիտարկման արժանի հարցերը.

- Ա Որքան փոխակերպում է Adapter-ը իրականացնումը:
- Բ Pluggable adapter-ներ:
- Գ Թափանցիկության համար երկակի Adapter-ների կիրառում:

Իրականացումը

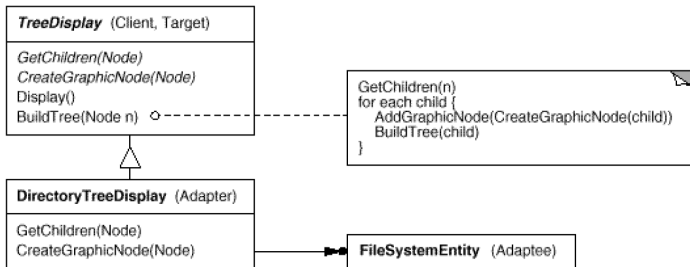
Ա Class adapter-ի իրականացումը C++

Բ Pluggable adapter-ներ:

- Աբստրակտ մեթոդների կիրառմամբ:
- Ներկայացուցիչ (delegate) օբյեկտների կիրառմամբ:

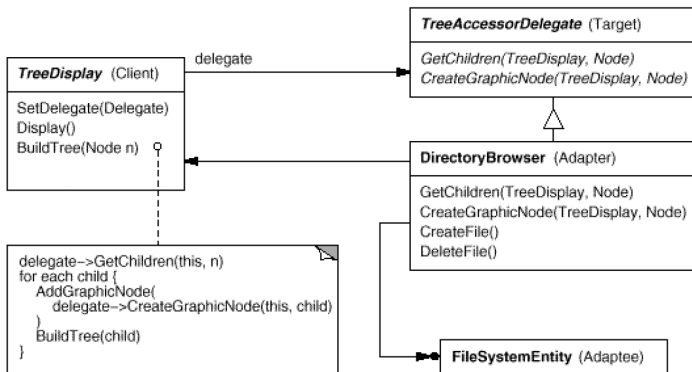
Pluggable Adapter-ների Իրականացումը

Աբստրակտ մեթոդների կիրառմամբ:



Pluggable Adapter-ների Իրականացումը

Ներկայացուցիչ օբյեկտների կիրառմամբ:



Օրինակ

```
class Shape {  
  
public:  
    Shape();  
    virtual void BoundingBox(Point& bottomLeft, Point& topRight) const;  
    virtual Manipulator* CreateManipulator() const;  
};  
  
class TextView {  
  
public:  
    TextView();  
    void GetOrigin(Coord& x, Coord& y) const;  
    void GetExtent(Coord& width, Coord& height) const;  
    virtual bool IsEmpty() const;  
};
```

Օրինակ

```
class TextShape : public Shape, private TextView {  
  
public:  
    TextShape();  
    virtual void BoundingBox(Point& bottomLeft, Point& topRight) const;  
    virtual bool IsEmpty() const;  
    virtual Manipulator* CreateManipulator() const;  
};  
  
void TextShape::BoundingBox(Point& bottomLeft, Point& topRight) const {  
  
    Coord bottom, left, width, height;  
    GetOrigin(bottom, left); GetExtent(width, height);  
  
    bottomLeft = Point(bottom, left);  
    topRight = Point(bottom + height, left + width);  
}
```


Օրինակ

```
class TextShape : public Shape {  
  
public:  
    TextShape(TextView* t) : text(t) {}  
    virtual void BoundingBox(Point& bottomLeft, Point& topRight) const;  
    virtual bool IsEmpty() const;  
    virtual Manipulator* CreateManipulator() const;  
  
private:  
    TextView* text;  
};  
  
void TextShape::BoundingBox(Point& bottomLeft, Point& topRight) const {  
  
    Coord bottom, left, width, height;  
    text->GetOrigin(bottom, left); text->GetExtent(width, height);  
  
    bottomLeft = Point(bottom, left);  
    topRight = Point(bottom + height, left + width);  
}
```

Առնչվող Նախագծման Ձևանմուշները

- Bridge

- Decorator

- Proxy