

СП1 – Adapter

Дефиниција

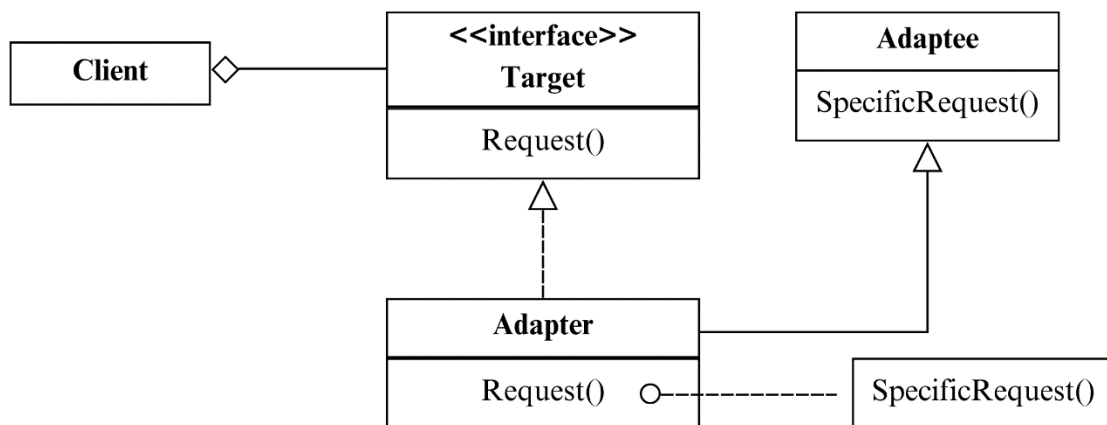
Конвертује интерфејс неке класе у други интерфејс који клијент очекује. Адаптер патерн омогућава заједнички рад класа које имају некомпатибилне интерфејсе.

Појашњење дефиниције

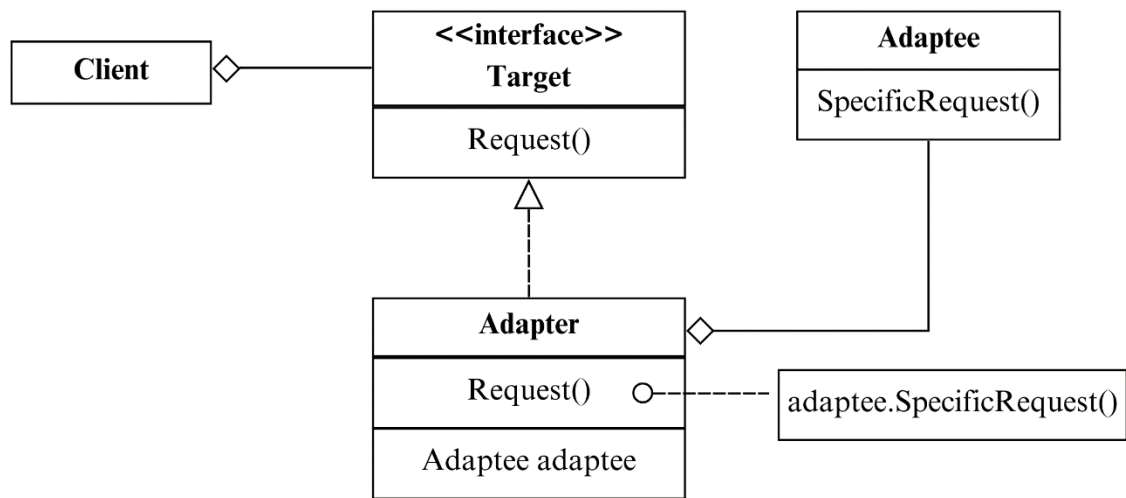
Конвертује интерфејс неке класе (Adaptee) у други интерфејс (Target) који клијент (Client) очекује. Адаптер патерн омогућава заједнички рад класа (Adaptee, Target) које имају некомпатибилне интерфејсе.

Структура Adapter патерна се може јавити у два облика

- а) Класа Adapter користи вишеструко наслеђивање код прилагођавања некомпатибилних интерфејса.



- б) Класа Adapter користи композицију код прилагођавања некомпатибилних интерфејса.



Учесници

- **Client** - Сарађује са интерфејсом Adaptee преко интерфејса Target.
- **Target** - Дефинише доменски - специфичан интерфејс који класа Client користи.
- **Adapter** - Адаптира (прилагођава) интерфејс Adaptee интерфејсу Target.
- **Adaptee** - Дефинише постојећи интерфејс који треба адаптирати.

Кориснички захтев ADP: Шеф развоја софтверског система (за вођење распореда летова) је послао захтев пројектантима софтверског система да креирају софтверски систем и елементе софтверског система:

- a) Екранску форму (view) са пољима за прихват и приказ података.
- b) Контролер за размену података између екранске форме и базе података.
- c) Брокер базе података помоћу кога се омогућава памћење, промена, брисање и приказ података.

Шеф ће да надзире процес израде софтверског система.

Пројектанти имају дефинисану апстрактну класу `Projektant` која има апстрактне методе:

```
void kreirajEkranskuFormu()
void kreirajBrokerBazePodataka()
void kreirajKontroler()
void kreirajSoftverskiSistem()
void prikaziEkranskuFormu()
```

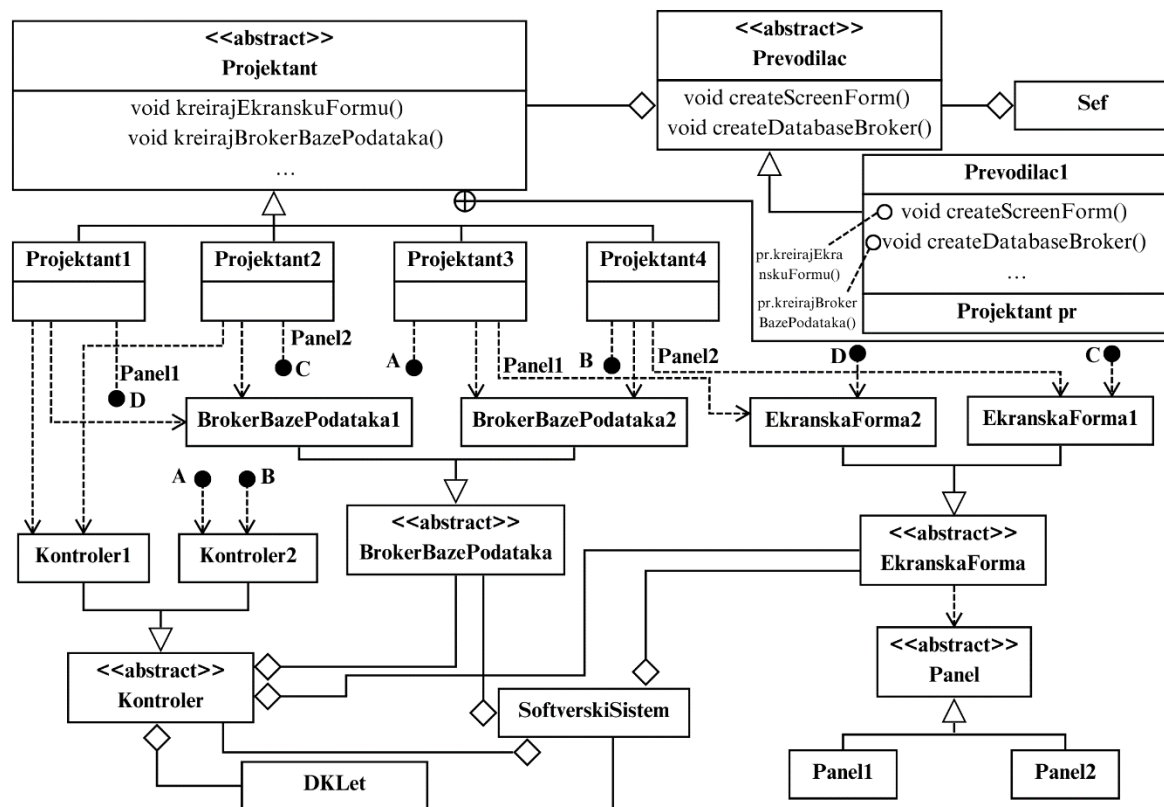
Шеф жели да комуницира са пројектантима преко преводиоца. Преводилац је представљен преко апстрактне класе `Prevodilac` која има следеће апстрактне методе:

```
void createScreenForm()
void createDatabaseBroker()
void createController()
void createSoftwareSystem()
void showScreenForm()
```

Беза између апстрактних метода апстрактне класе `Prevodilac` и апстрактних метода апстрактне класе `Projektant`:

<code>void kreirajEkranskuFormu()</code>	<code>void createScreenForm()</code>
<code>void kreirajBrokerBazePodataka()</code>	<code>void createDatabaseBroker()</code>
<code>void kreirajKontroler()</code>	<code>void createController()</code>
<code>void kreirajSoftverskiSistem()</code>	<code>void createSoftwareSystem()</code>
<code>void prikaziEkranskuFormu()</code>	<code>void showScreenForm()</code>

Дијаграм класа примера ADP



Веза између елемената структуре решења Adapter патерна и структуре решења примера ADP

Елементи Adapter патерна	Елементи структуре примера ADP
Client	Sef
Adaptee	Projektant
ConcreteAdaptee	Projektant1, Projektant2, Projektant3, Projektant4
Adapter	Previdilac1
Target	Previdilac
AbstractProductA	EkranskaForma
AbstractProductB	BrokerBazePodataka
AbstractProductC	Kontroler
ProductA	EkranskaForma1, EkranskaForma2
ProductB	BrokerBazePodataka1, BrokerBazePodataka2
ProductC	Kontroler1, Kontroler2
Product	SoftverskiSistem

Напомене за пример ADP

Овај пример је направљен на основу примера BLP из Builder патерна.

Примењен је облик структуре из примера под b).