

Projektowanie obiektowe oprogramowania

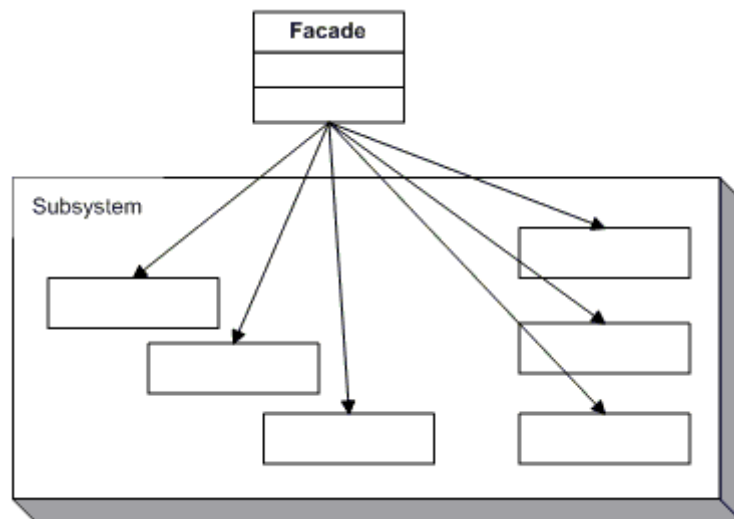
Wykład 5 – wzorce strukturalne

Wiktor Zychła 2014

1 Wzorce strukturalne

1.1 Facade

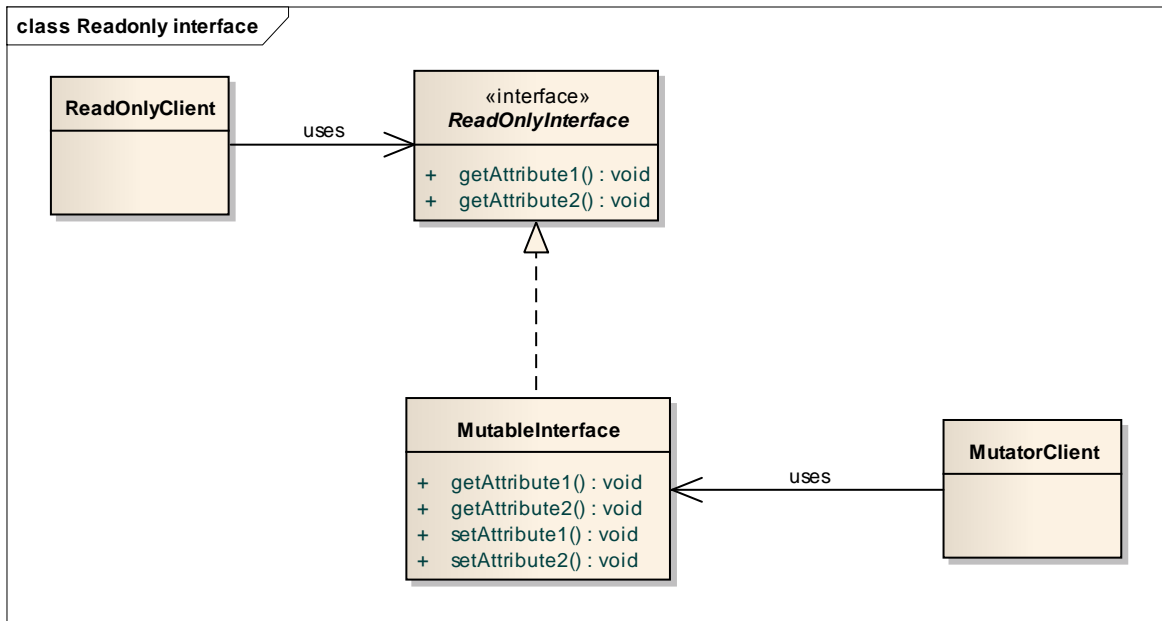
Motto: uproszczony interfejs dla podsystemu z wieloma interfejsami



```
class SntpFacade {
    public void Send( string From, string To,
                    string Subject, string Body,
                    Stream Attachment, string AttachmentMimeType );
}
```

1.2 Read-only interface

Motto: interfejs do odczytu dla wszystkich, a do zapisu tylko dla wybranych

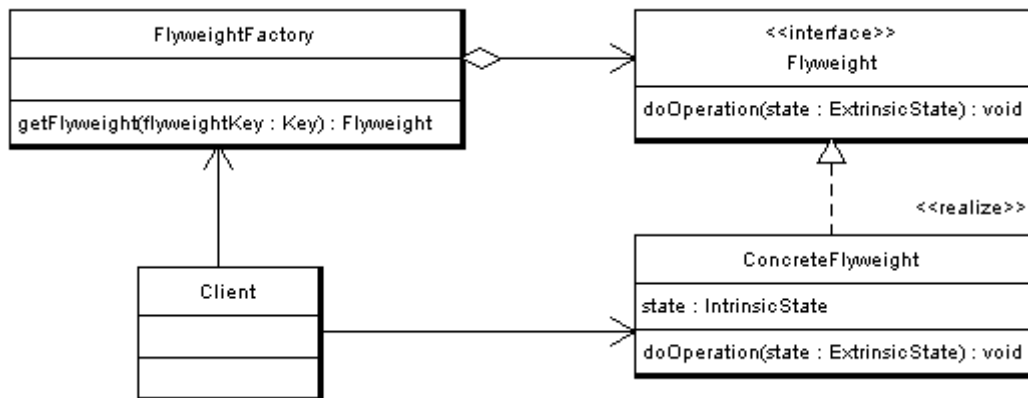


Przykład: ReadonlyCollection, AsReadOnly()

1.3 Flyweight

Motto: Efektywne zarządzanie wieloma drobnymi obiektami

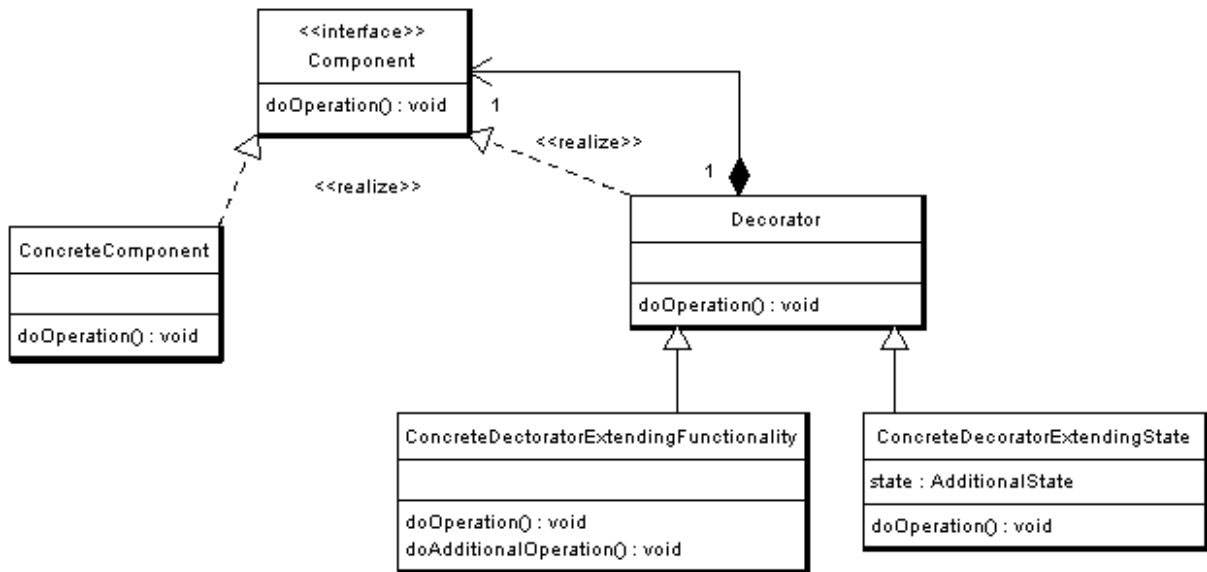
Kojarzyć: Object Pool + immutable + bardzo dużo danych – zapamiętać przykład z wykładu Board vs Checker



1.4 Decorator

Motto: Dynamicznie rozszerzanie odpowiedzialności obiektów (alternatywa dla podklas)

Kojarzyć: System.IO.Stream.Stream



1.5 Proxy

Motto: substytut (zamiennik) obiektu w celu sterowania dostępem do niego

- Proxy zdalne – reprezentant lokalny obiektu zdalnego
- Proxy wirtualne – tworzy kosztowny obiekt na żądanie
- Proxy ochraniające – kontroluje dostęp do obiektu
- Proxy logujące – loguje dostęp do obiektu

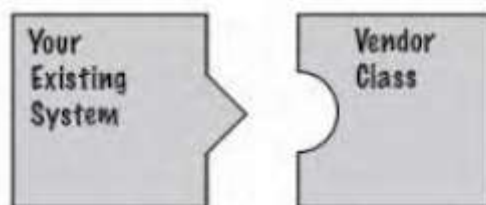
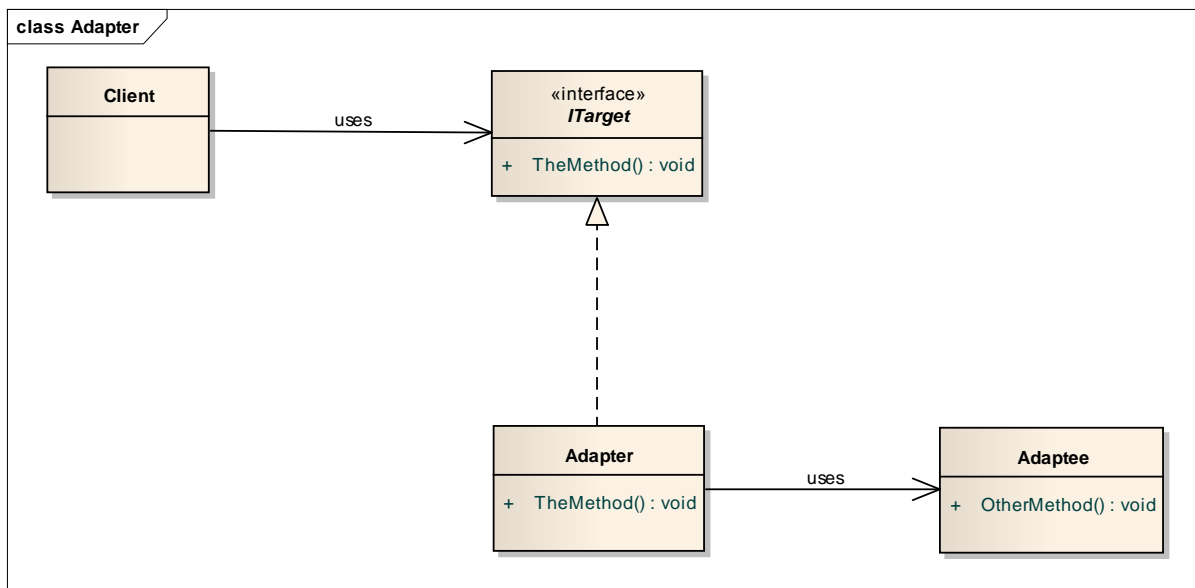
Przykład virtual proxy z biblioteki standardowej .NET - Lazy<T>.

Uwaga! Struktura Proxy i Decoratora może wydawać się podobna. Różnice są następujące:

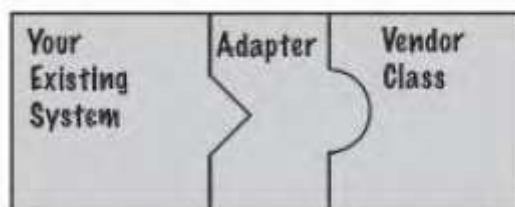
- Każdy z Decoratorów może dodawać nowe, specyficzne dla siebie operacje/informacje; Proxy nigdy nie zmienia (nie rozszerza) interfejsu obiektu
- Proxy z zasady nie jest przeznaczone do stosowania rekursywnego; Dekorator przeciwnie

1.6 Adapter

Motto: uzgadnianie niezgodnych interfejsów



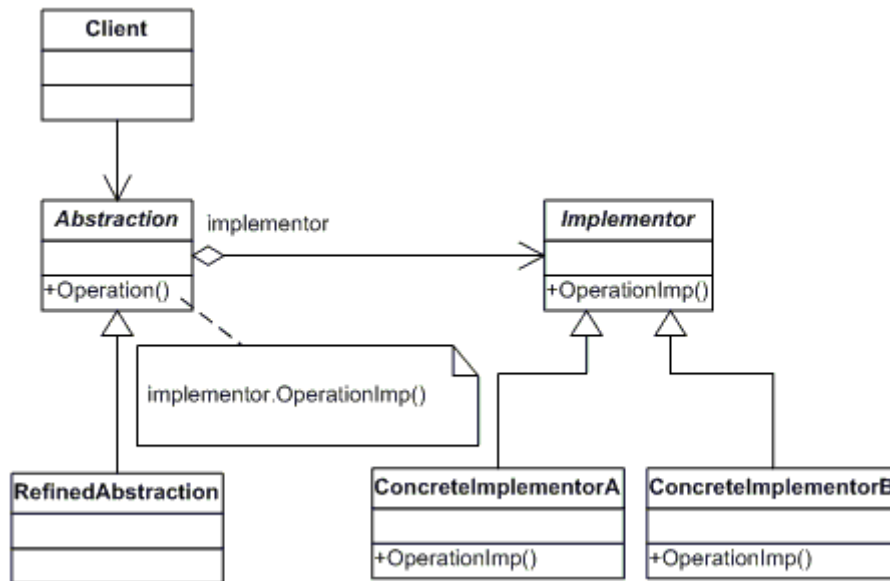
Not Compatible? We are in trouble



God Adapter Save our Life

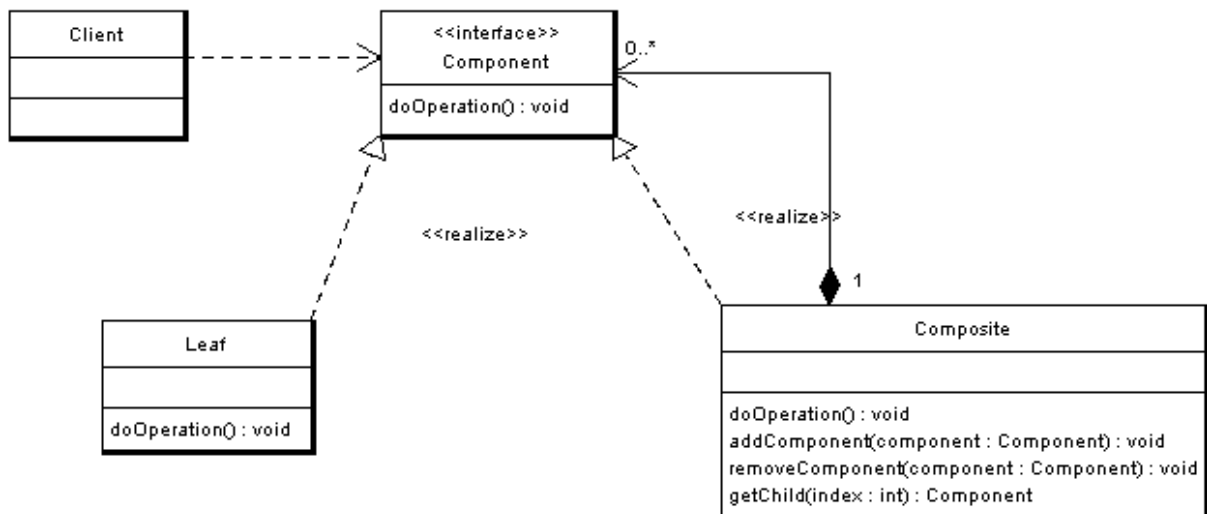
1.7 Bridge

Motto: SRP + ISP + DIP dla hierarchii obiektowej o dwóch stopniach swobody; oddzielenie abstrakcji od implementacji



1.8 Composite

Motto: składanie obiektów w struktury „drzewiaste”
 Kожarzyć: Tree, Expression



2 Literatura

- [1] Gamma, Helm, Johnson, Vlissides – Wzorce projektowe
- [2] Martin, Martin – Zasady, wzorce i praktyki zwinnego wytwarzania oprogramowania w C#
- [3] Grand, Merrill – Wzorce projektowe
- [4] Freeman, Freeman, Sierra, Bates – Head First Design Patterns
- [5] <http://www.oodeesign.com>