

Objektové modelování, B6B36OMO

Návrhové vzory Strukturální

Tomáš Mayer

Katedra počítačů
Fakulta elektrotechnická
České vysoké učení technické v Praze

<https://cw.fel.cvut.cz/wiki/courses/b6b36omo>

CO JSOU STRUKTURÁLNÍ NÁVRHOVÉ VZORY

- STRUKTURÁLNÍ VZORY SE SNAŽÍ POPSAT MOŽNOSTI ORGANIZACE OBJEKTŮ DO STRUKTUR
- VZORY VZNIKLY NA ZÁKLADĚ OPAKOVANÝCH TYPICKÝCH PROBLÉMU, SE KTERÝMI SE VÝVOJÁŘ SETKÁVÁ
- KATALOG VZORŮ ZAZNAMENÁVÁ OSVĚDČENÉ STRUKTURY A POPISUJE JEJICH ZPŮSOB POUŽITÍ

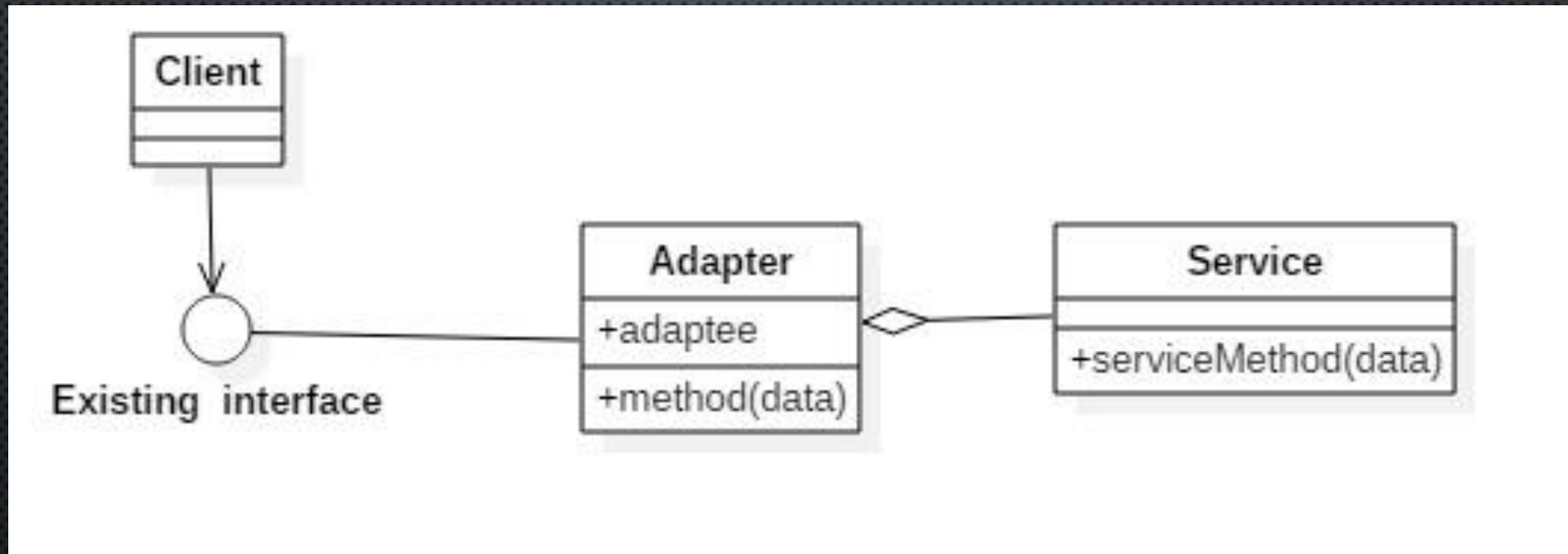
GOF STRUKTURÁLNÍ NÁVRHOVÉ VZORY

- **ADAPTER** – UMOŽNÍ SPOLUPRÁCI DVOU OBJEKTŮ S ROZDÍLNÝM ROZHRAŇÍM
- **BRIDGE** – ROZDĚLENÍ IMPLEMENTACE NA DVĚ HIERARICHICKÉ ČÁSTI – ABSTRAKCI A IMPLEMENTACI, KTERÉ MOHOU BÝT ROZVÍJENY ODDĚLENĚ
- **COMPOSITE** – ORGANIZACE OBJEKTŮ DO STRUKTUR, KTERÁ UMOŽNÍ PRACOVAT SE STRUKTROU JAKO KDYBY SE JEDNALO O INDIVIDUÁLNÍ OBJEKT
- **DECORATOR** – UMOŽNÍ VLOŽIT NOVÉ CHOVÁNÍ FORMOU WRAPPERU, DO KTERÉHO JE VLOŽEN PŮVODNÍ OBJEKT
- **FACADE** – VYSTAVENÍ JEDNODUŠŠÍHO ROZHRAŇÍ, KTERÝ SCHOVÁ SLOŽITOU STRUKTURU OBJEKTŮ NEBO ROZHRAŇÍ
- **PROXY** – UMOŽNÍ POSKYTNOUT SUBSTITUCI ROZHRAŇÍ OBJEKTU TAK, ŽE JE MOŽNÉ VLOŽIT PŘED NEBO ZA VOLÁNÍ PŮVODNÍHO OBJEKTU NĚJAKOU VLASTNÍ LOGIKU
- **FLYWEIGHT** – UMOŽNÍ SDÍLET SPOLEČNÝ SDÍLENÝ STAV VÍCE OBJEKTŮ NA JEDNOM MÍSTĚ, MÍSTO TOHO, ABY SI KAŽDÝ OBJEKT DRŽEL SVOU KOPII

ADAPTER

- APLIKACE, KTERÁ POUŽÍVÁ VOLÁNÍ PŘES JIŽ DEFINOVANÉ ROZHRAŇÍ
- POTŘEBUJEME ZAPOJIT NOVOU KNIHOVNU, KTERÁ MÁ ODLIŠNÉ ROZHRAŇÍ
- VYTVOŘÍME TŘÍDU (ADAPTÉR), KTERÁ VYSTAVUJE PŮVODNÍ ROZHRAŇÍ A PROVÁDÍ PŘEVOLÁNÍ DO NOVÉ KNIHOVNY PŘES JEJÍ ROZHRAŇÍ

ADAPTER



ADAPTER

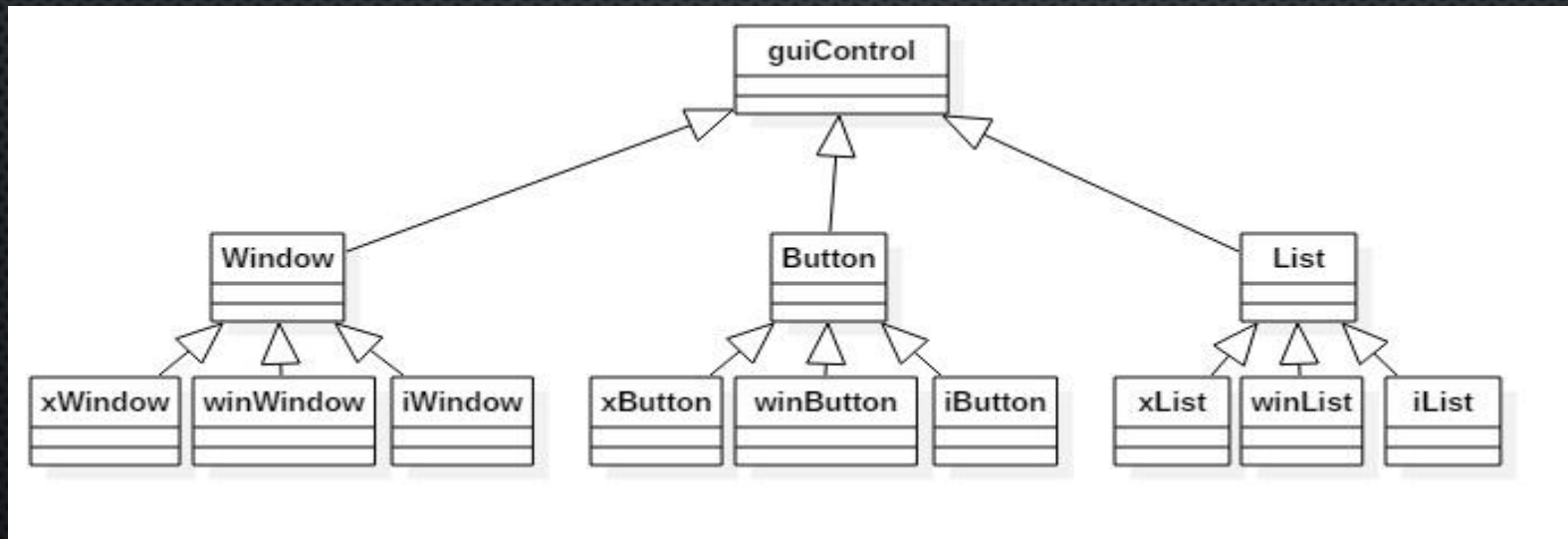
- KDY POUŽÍT
 - CHCEME POUŽÍT ROZHRANÍ, KTERÉ ALE NENÍ KOMPATIBILNÍ SE ZBYTKEM SYSTÉMU
- VÝHODY
 - ODSTIŇUJE KLIENTA OD IMPLEMENTAČNÍCH DETAILŮ KONVERZE ROZHRANÍ A DAT

BRIDGE

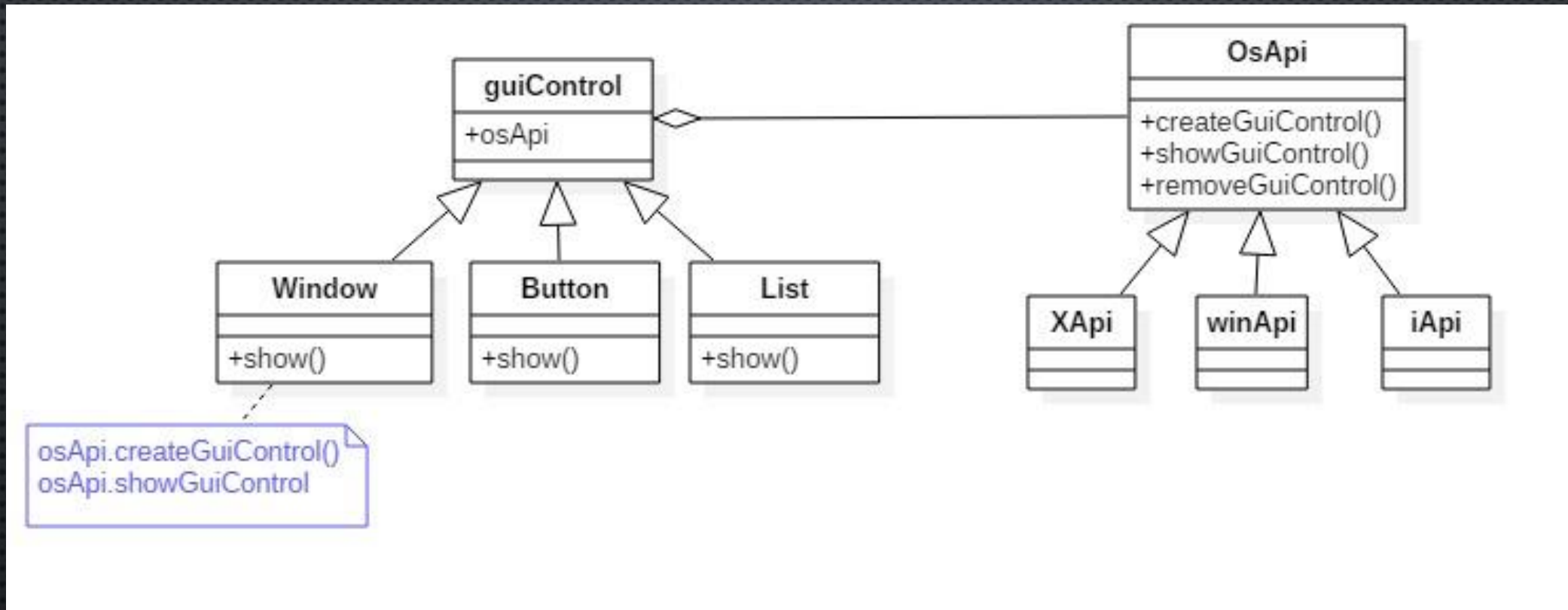
- ROZDĚLÍ ŘEŠENÍ NA DVĚ ČÁSTI – ABSTRAKCI OD IMPLEMENTACE A UNMOŽNÍ TAK NEZÁVISLOU HIERARCHII JAK ABSTRAKCE, TAK IMPLEMENTACE
- VIZ NÁSLEDUJÍCÍ PŘÍKLAD

BRIDGE

- PŘÍKLAD GUI, NA JEDNÉ STRANĚ JE SPOLEČNÁ ABSTRAKCE GUI – OKNA, TALČÍTKA, LIŠTY, NA STRANĚ DRUHÉ IMPLENENTACE PRVKŮ GUI PROSTŘEDNICTVÍM API OPERAČNÍHO SYSTÉMU.
- BEZ BRIGDE MŮŽE ŘEŠENÍ VYPADAT:
- xWINDOW, winWINDOW, iWINDOW, xBUTTON, winBUTTON, IBUTTON.....

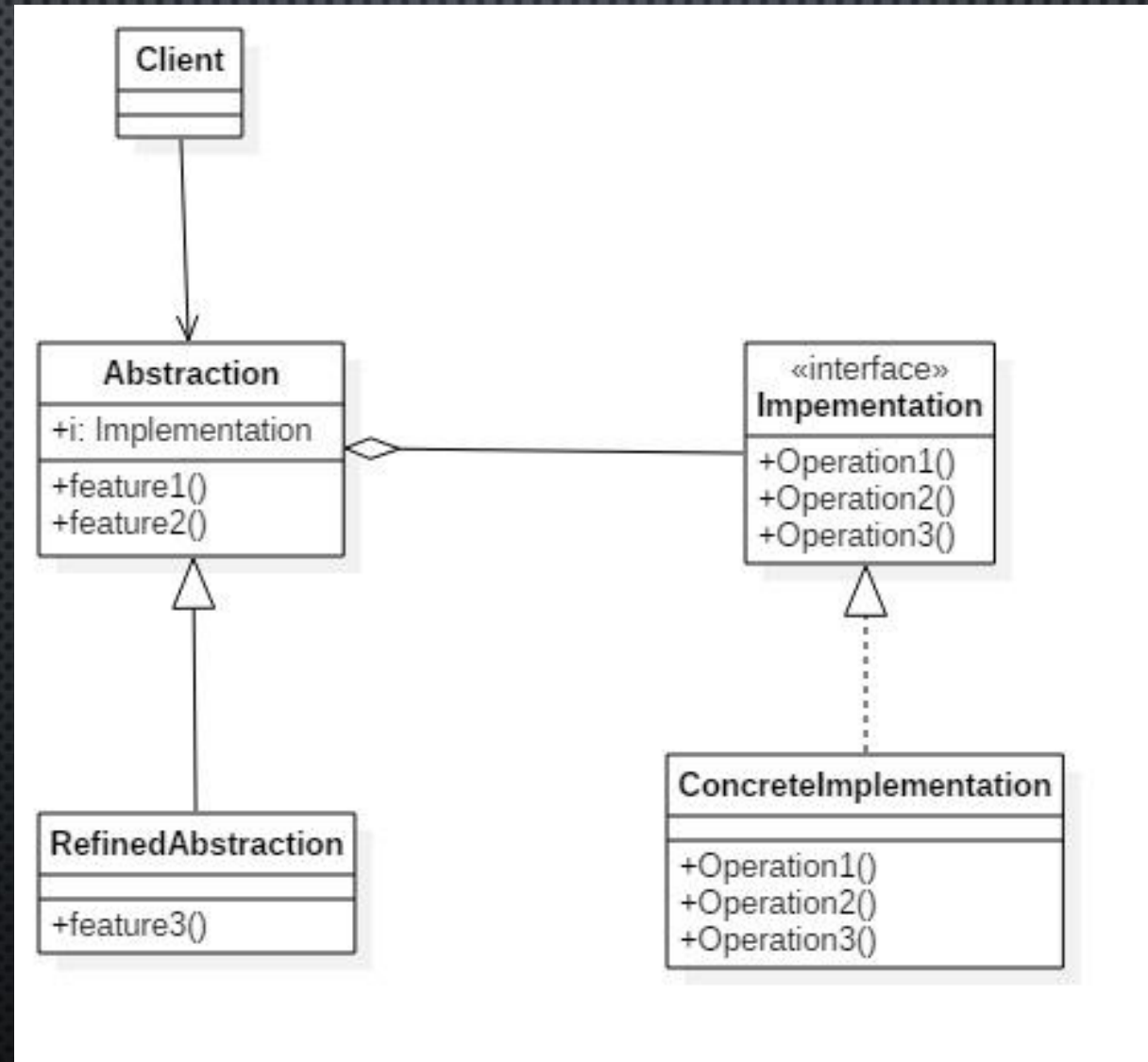


BRIDGE



- Odělením abstrakce od implementace zároveň vznikne možnost k abstrakci **přřadit implementaci dynamicky v runtime**

BRIDGE

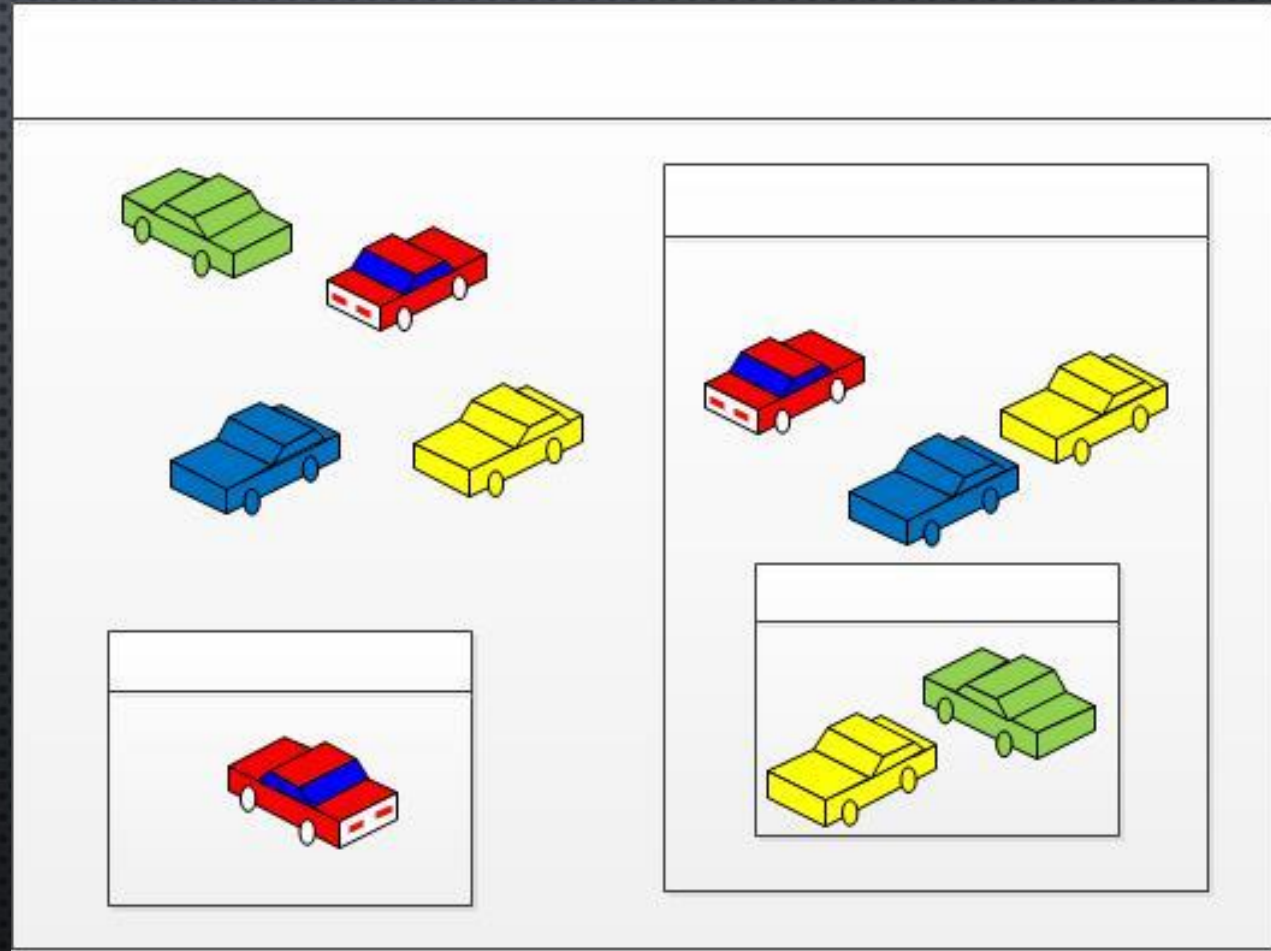


BRIDGE

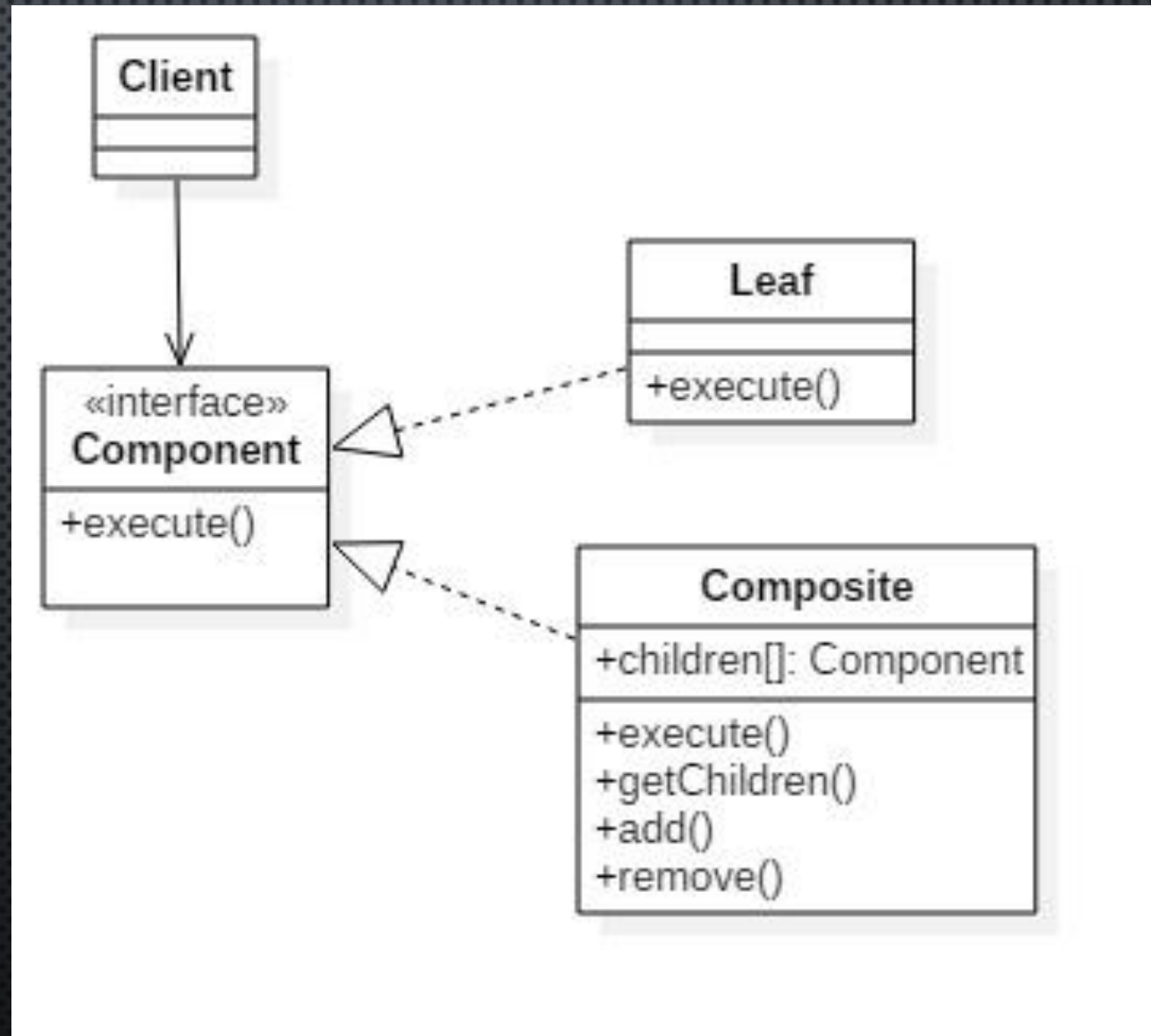
- KDY POUŽÍT
 - POKUD MÁME HIEARACHIE TŘÍD, KTERÉ MAJÍ VÍCE VARIANT (GUI CONTROL X OPERAČNÍ SYSTÉMY)
 - POKUD POTŘEBUJEME ROZŠÍŘIT SADU TŘÍD DO NĚKOLIKA NEZÁVISLÝCH DIMENZÍ
 - POKUD CHCEM MĚNIT IMPLEMENTACI ZA BĚHU V RUNTIME
- VÝHODY
 - UMOŽNÍ PROGRAMOVAT ŘEŠENÍ NEZÁVISLÉ NA PLATFORMĚ
 - NEZATĚŽUJE KLIENTA IMPLEMENTAČNÍMI DETAILS

COMPOSITE

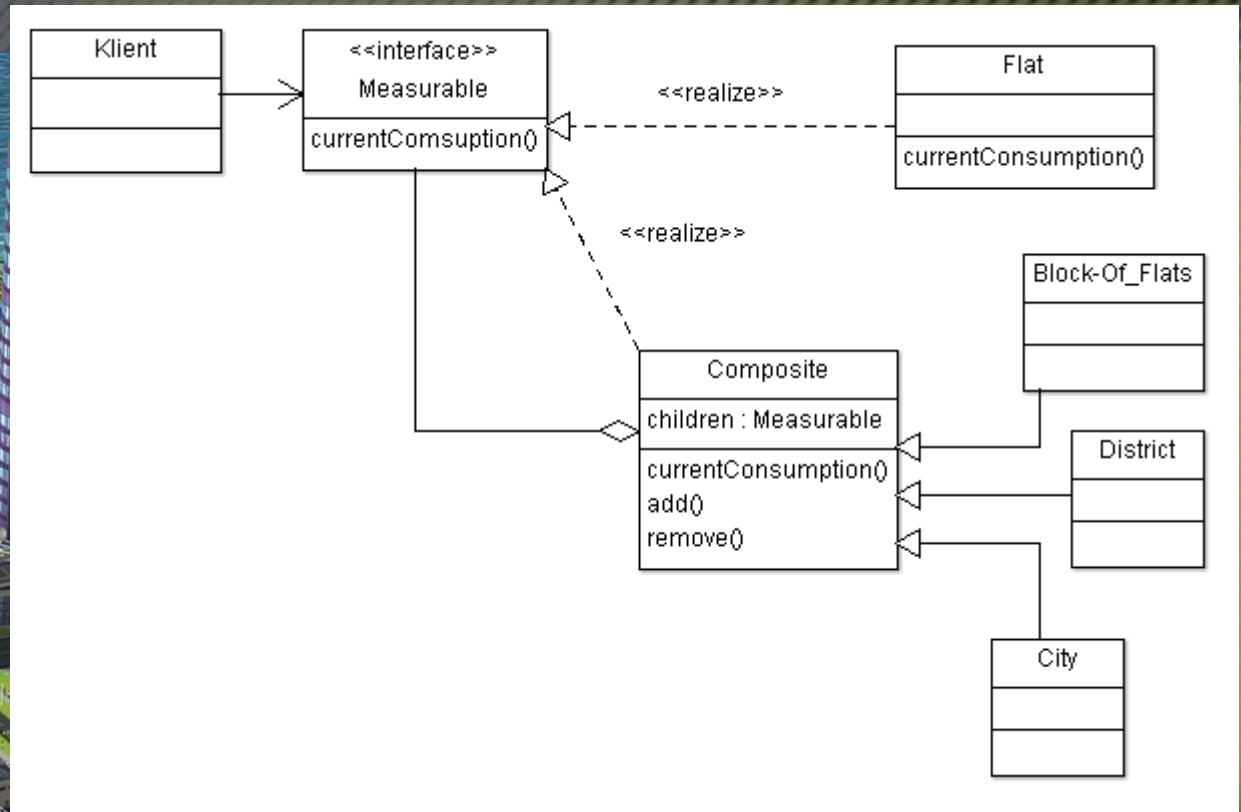
- HIEARARCHICKÁ STROMOVÁ STRUKTURA
- VNOŘENÉ OBJEKTY
- KLIENT PRACUJE S VLASTNÍMI OBJEKTY SHODNĚ JAKO S KONTEJNERY



COMPOSITE



COMPOSITE – PŘÍKLAD – MĚŘENÍ SPOTŘEBY



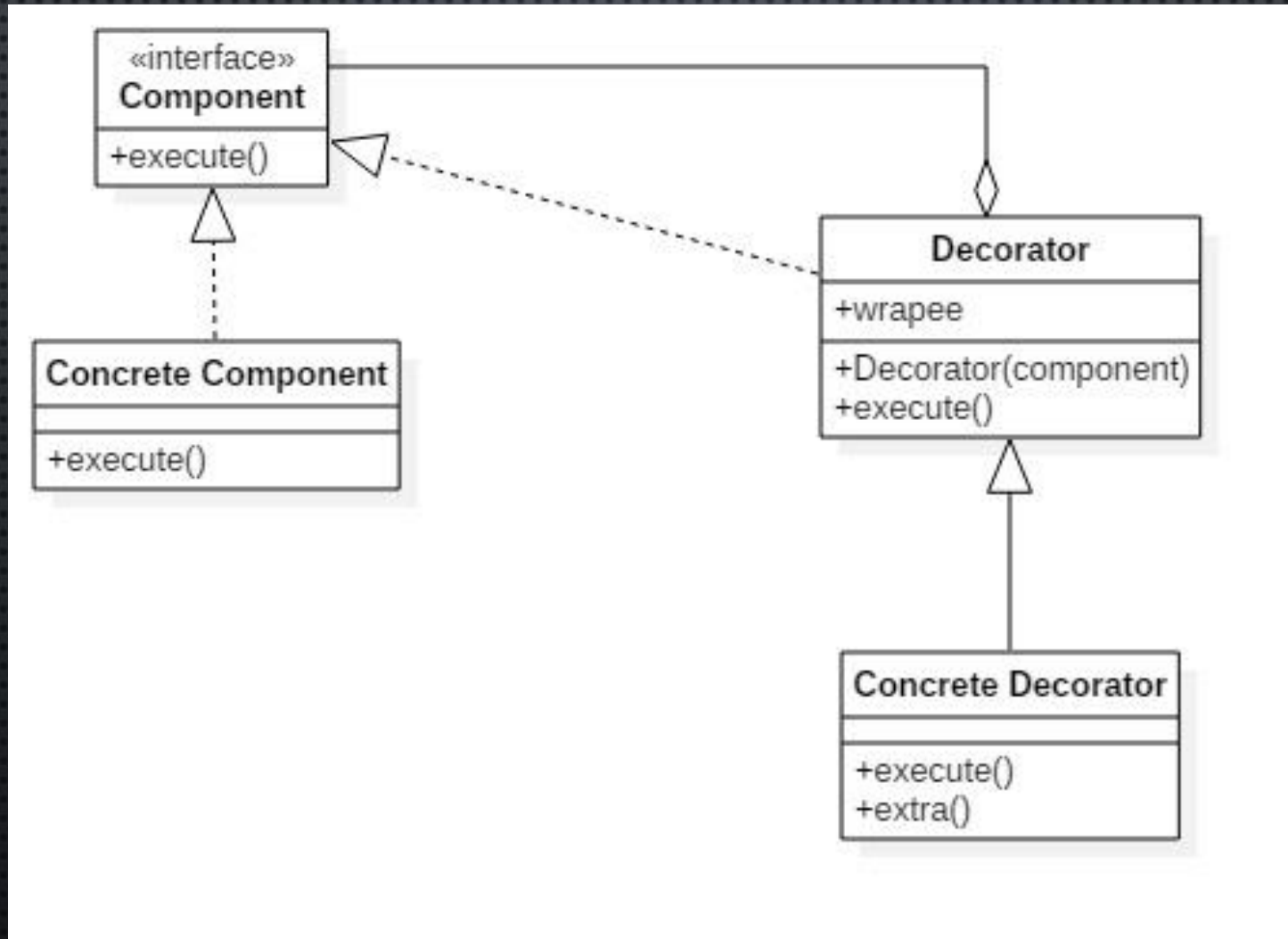
COMPOSITE

- KDY POUŽÍT
 - POKUD MÁME STROMOVOU HIERARICHU, KTERÁ MÁ JEDNODUCHÉ ELEMENTY A KONTEJNERY
 - POKUD MÁ KLIENT JEDNOTNĚ PRACOVAT JAK S JEDNODUCHÝMI TAK SLOŽENÝMI ELEMENTY
- VÝHODY
 - ZJEDNODUŠUJE KÓD KLIENTA, KTERÝ TAK NEMUSÍ PRACOVAT SE SLOŽITOU STROMOVOU STRUKTUROU
 - USNADŇUJE PŘIDÁVÁNÍ NOVÝCH KOMPONENT DO HIERARCHIE

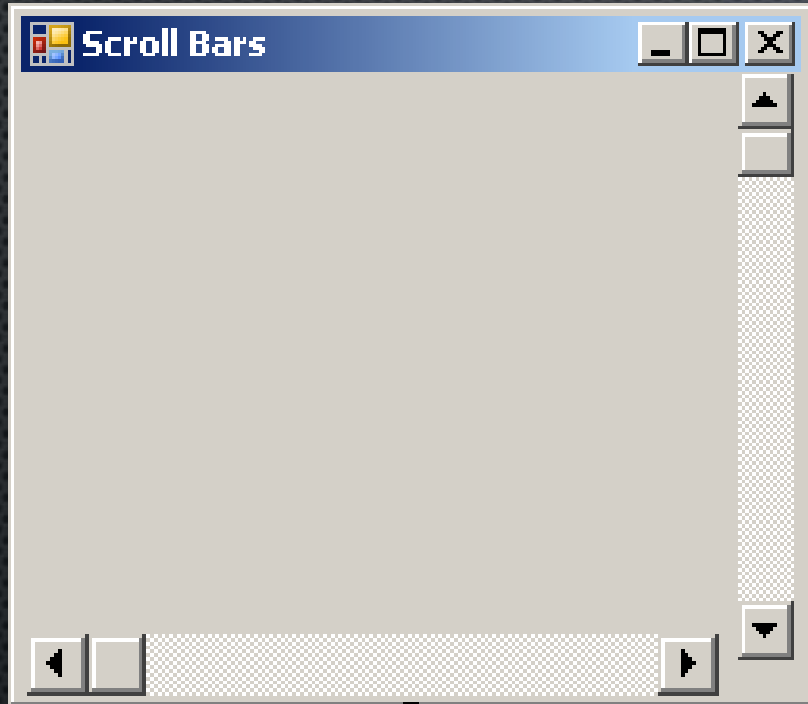
DECORATOR

- POTŘEBUJEME MÍT MOŽNOST DYNAMMICKY PŘIDÁVAT A ODEBÍRAT FUNKCIONALITU K JIŽ EXISTUJÍCÍM OBJEKTŮM TAK, ABY TO NEMĚLO VLIV NA OSTATNÍ KÓD V PROGRAMU
- PRVNÍ MOŽNOST – VYUŽITÍ POLYMORFISMU, Z PŮVODNÍ TŘÍDY ZDĚDÍM A PŘIDÁM DO PŘEPSANÉ METODY POTŘEBNOU FUNKCIONALITU
 - NEVÝHODA – STATICKÉ, NELZE PŘIDÁVAT A ODEBÍTAT BEZ KOMPILACE KÓDU
- JINÁ MOŽNOST - POUŽITÍ OBALOVÉHO OBJEKTU (DECORATOR)
 - DECORATOR MÁ STEJNÝ INTERFACE A OBALUJE PŮVODNÍ OBJEKT
 - KAŽDÝ DEKORÁTOR MÁ ATRIBUT, DO KTERÉHO SI ULOŽÍ REFERENCI NA PŮVODNÍ OBJEKT
 - LZE TAKTO REKURZIVNĚ OBALOVAT DALŠÍMI DECORÁTORY

DECORATOR



DECORATOR - PŘÍKLAD



```
Widget* aWidget = new BorderDecorator(  
    new HorizontalScrollBarDecorator(  
        new VerticalScrollBarDecorator(  
            new Window( 80, 24 )))); aWidget->draw();
```

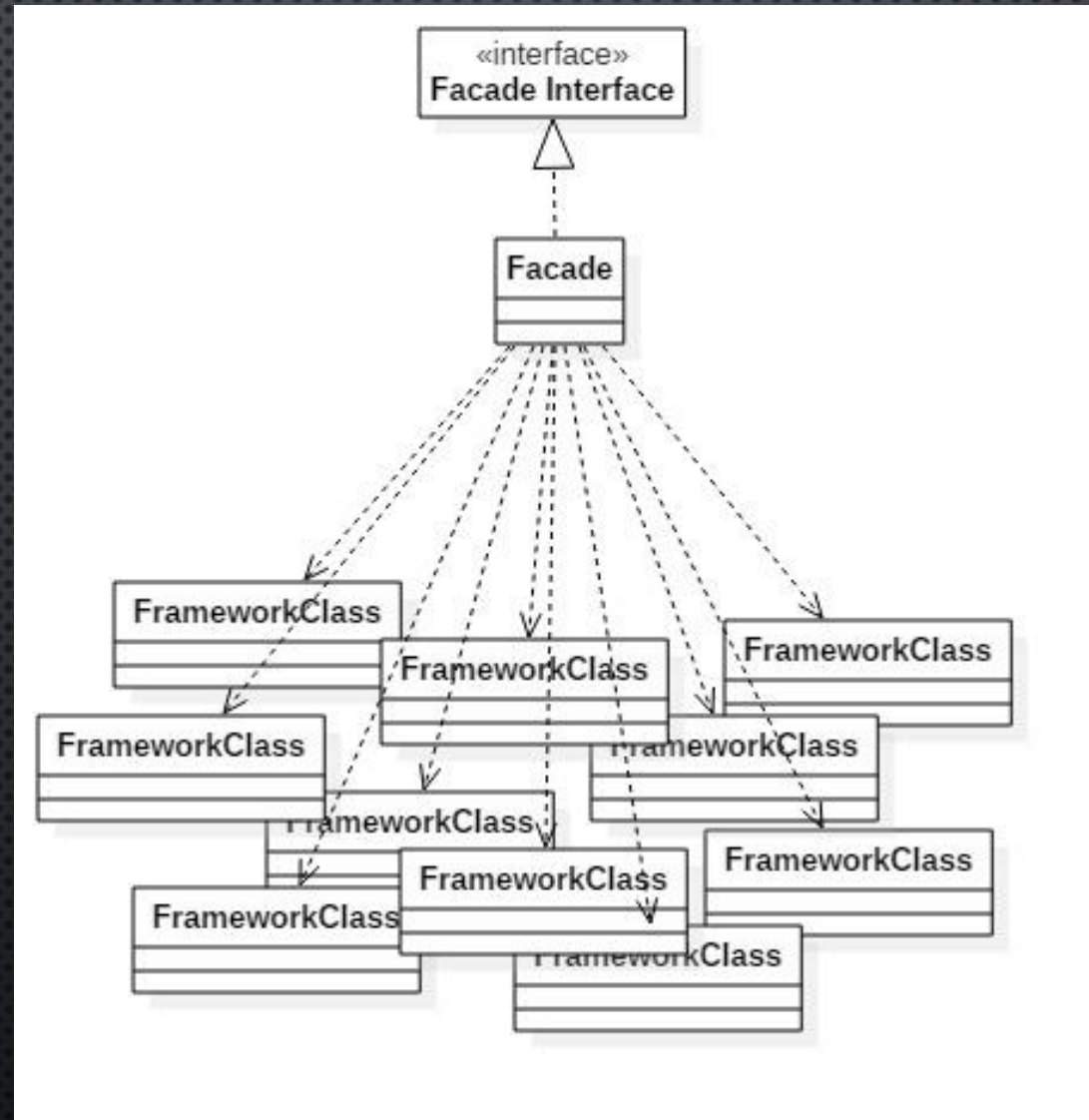
DECORATOR

- KDY POUŽÍT
 - POKUD POTŘEBUJI DYNAMICKY PŘIDÁVAT NEBO UBÍRAT FUNKCIONALITU
 - POKUD NEMOHU NEBO NENÍ JEDNODUCHÉ POUŽÍT DĚDIČNOST
- VÝHODY
 - FLEXIBILNĚJŠÍ NEŽ POUŽITÍ DEDIČNOSTI
 - LZE PŘIDÁVAT DYNAMICKY I V RUNTIME
 - LZE PŘIDÁVAT DALŠÍ FUNKCIONALITU FORMOU DALŠÍCH DEKORÁTORŮ
 - UMOŽŇUJE SKLÁDÁNÍ KOMPLEXNEJŠÍCH OBJEKTŮ Z JEDNODUŠŠÍCH
- NEVÝHODY
 - TEŽŠÍ KONFIGUROVAT KASKÁDNĚ DEKOROVANÝ OBJEKT
 - VĚTŠÍ MNOŽSTVÍ MALÝCH TŘÍD

FACADE

- FACADE – INTERFACE, KTERÝ SCHOVÁ KOMPLEXITU FRAMEWORKU. NEMUSÍ VYSTAVOVAT CELOU FUNKCIONALITU FRAMEWORKU, POUZE TO CO KLIENT OPRAVDU VYUŽIJE
- PŘÍKLAD – MÁME K DISPOZICI FRAMEWORK OBSAHUJÍCÍ VĚTŠÍ SADU OBJEKTŮ, KTERÉ V PROGRAMU BUDU MUSET VOLAT, KONFIGUROVAT, ZACHOVAT NAPŘ. URČITÉ POŘADÍ PRÁCE S OBJEKTY NEBO POŘADÍ VOLÁNÍ OPERACÍ. PROGRAM SE TAK ÚZCE SVÁŽE S IMPLEMENTACÍ FRAMEWORKU.

FACADE

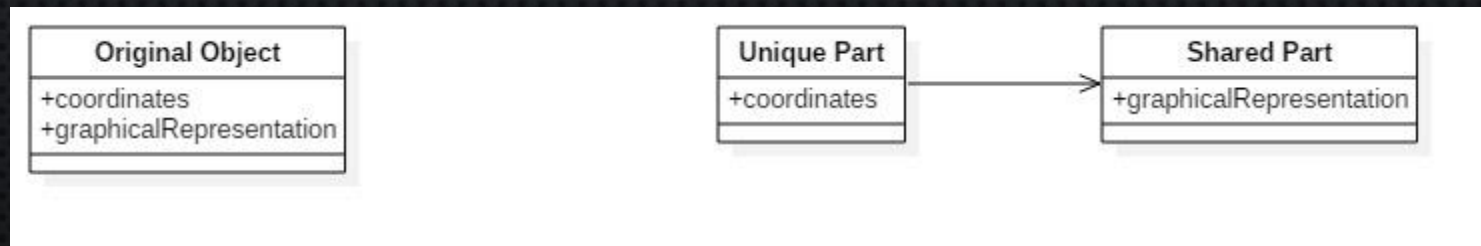


FACADE

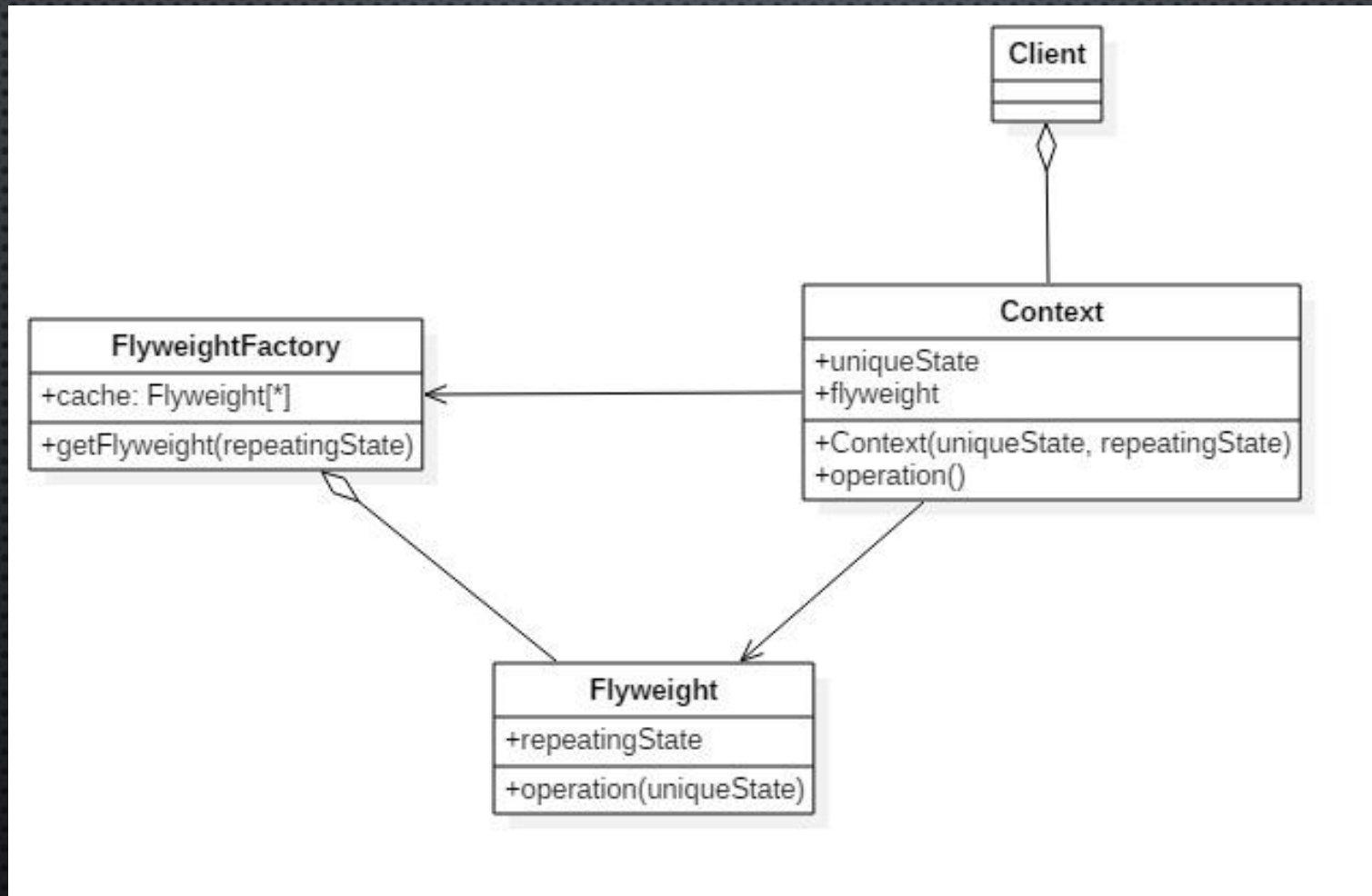
- KDY POUŽÍT
 - KDYŽ POTŘEBUJEME VYSTAVIT JEDNODUCHÝ INTERFACE, ZA KTERÝ SCHOVÁME KOMPLEXITU IMPLEMENTACE NEBO FRAMEWORKU
 - POKUD CHCEME STRUKTUROVAT ŘEŠENÍ DO VRSTEV
- VÝHODY
 - IZOLUJE PROGRAM OD KOMPONENT SUBSYSTÉMU NEBO FRAMEWORKU
 - IZOLUJE OD PŘÍLIŠNÉ TĚSNÉ VAZBY MEZI PROGRAMEM A SUBSYSTÉMEM NEBO FRAMEWORKEM
- NEVÝHODY
 - NEBEZPEČÍ, ŽE SE FACADE MŮŽE STÁT TŘÍDOU, KTERÁ ŘEŠÍ VŠECHNO A JE SVÁZÁNA SE VŠEMI TŘÍDAMI PROGRAMU

FLYWEIGHT

- UMOŽŇUJE EFEKTIVNĚJŠÍ VYUŽITÍ PAMĚTI SDÍLENÍM SPOLEČNÝCH PRVKŮ
- PŘEDSTAVME SI SYSTÉM, KTERÝ BUDE PRACOVAT S 1.000.000 OBJEKTŮ, OBJEKTY BUDOU JEDNOHO Z 20 TYPŮ. KAŽDÝ TYP OBJEKTU BUDE MÍT GRAFICKOU REPREZENTACI V PRŮMĚRU O VELIKOSTI 100 KB.
 - POKUD BUDE MÍT KAŽDÝ OBJEKT SVOU GRAFICKOU REPREZENTACI U SEBE:
 - 1.000.000 x 100 KB – CELKEM 100 GB
 - POKUD BUDE GRAFICKOU REPREZENTACI SDÍLET PŘES TYP (ODKAZ NA SÍLEDNOU INFROMACI POČÍTÁM 4 B)
 - 1.000.000 x 4 + 20 x 100 KB – CELKEM 6 MB



FLYWEIGHT



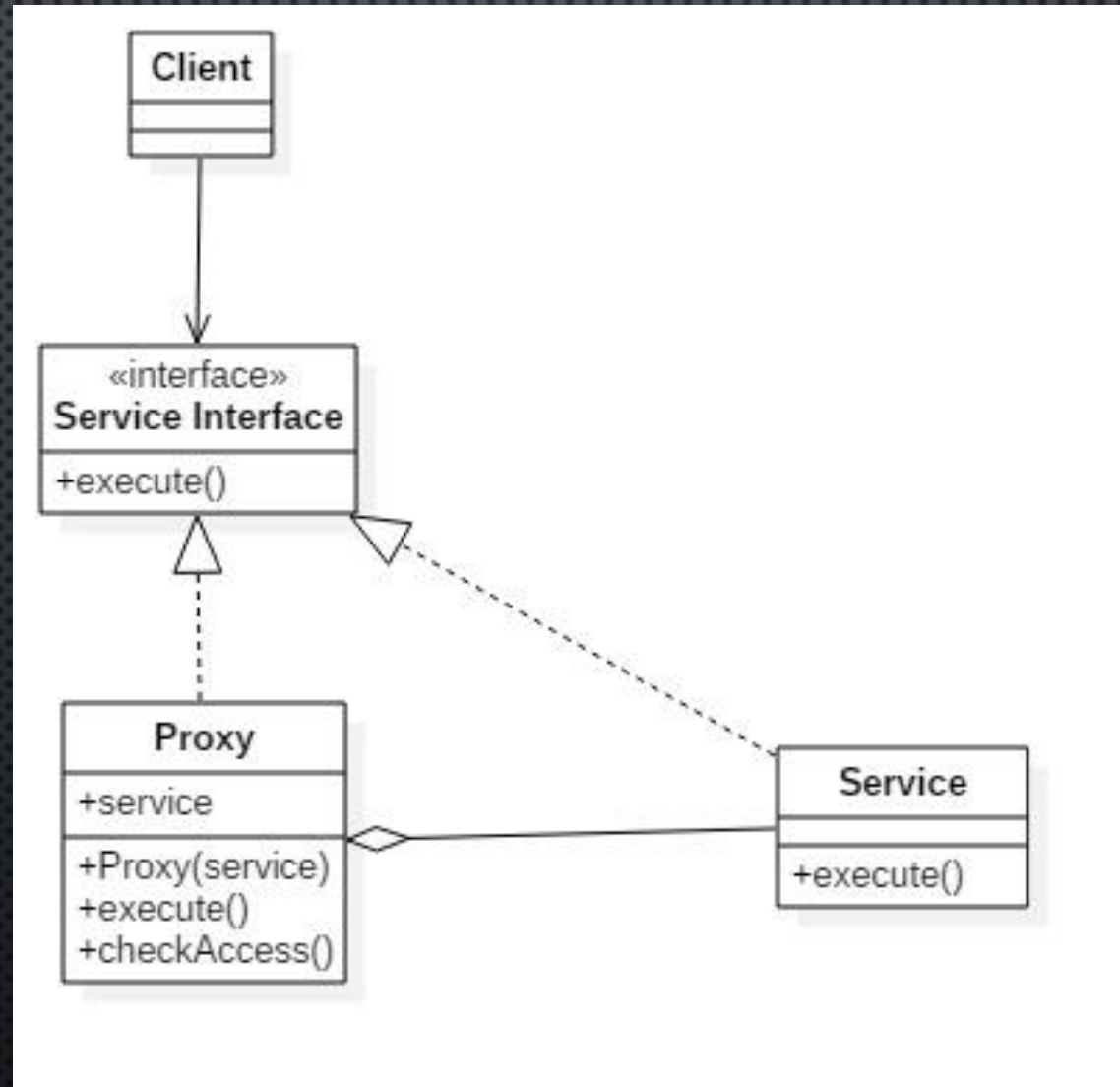
FLYWEIGHT

- KDY POUŽÍT
 - KDYŽ POTŘEBUJI PRACOVAT S VELKÝM MNOŽSTVÍM OBJEKTŮ, KTERÉ SE OBTÍŽNĚ VEJDOU DO OPERAČNÍ PAMĚTI
- VÝHODY
 - ŠETŘÍ PAMĚŤOVÝ PROSTOR
- NEVÝHODY
 - VĚTŠÍ SLOŽITOST ŘEŠENÍ
 - VYŠŠÍ ZÁTĚŽ CPU PŘI VYHLEDÁVÁNÍ FLYWEIGHT

PROXY

- UMOŽŇUJE ZASTUPOVAT OBJEKT, ABY BYLO MOŽNÉ NAPŘÍKLAD ŘÍDIT PŘÍSTUP K PŮVODNÍMU OBJEKTU
- PROXY OBJEKT MÁ STEJNÝ INTERFACE JAKO PŮVODNÍ OBJEKT
- PROXY DODÁVÁ DALŠÍ LOGIKU DO VOLÁNÍ PŮVODNÍ SLUŽBY

PROXY



PROXY

- KDY POUŽÍT
 - ODDÁLENÁ INICIALIZACE. VHODNÁ PRO TĚŽKÉ OBJEKTY, KTERÉ NENÍ POTŘEBA INICIALIZOVAT A DRŽET OD STARTU PROGRAMU, ALE AŽ VE CHVÍLI, KDY JSOU POTŘEBA
 - ŘÍZENÍ PŘÍSTUPU
 - MOŽNOST LOKÁLNÍHO PROVOZOVÁNÍ SLUŽBY, KTERÁ JE VOLÁNA VZDÁLENĚ
 - CACHOVÁNÍ OBJEKTŮ
 - LOGOVÁNÍ PŘÍSTUPU K OBJEKTU
- VÝHODY
 - MOŽNOST ŘÍZENÍ PŘÍSTUPU K OBJEKTU
 - FUNKČNÍ I V PŘÍPADĚ, ŽE PŮVODNÍ SLUŽBA NEBĚŽÍ
 - MOŽNOST ŘÍZENÍ ŽIVOTNÍHO CYKLU OBJEKTU

VZTAH MEZI ADAPTER, FACADE, DECORATOR, PROXY

- ADAPTER POSKYTUJE ROZDÍLNÝ INTERFACE OPROTI OBJEKTU, KTERÝ ZPŘÍSTUPŇUJE, PROXY POSKYTUJE SHODNÝ INTERFACE, DECORATOR SHODNÝ NEBO ROZŠÍŘENÝ
- FACADE A PROXY
 - PODOBNÉ V TOM, ŽE POSKYTUJÍ ODSTÍNĚNÍ KLIENTA OD VYUŽÍVANÝCH OBJEKTŮ
 - ROZDÍL V TOM, ŽE **PROXY** POSKYTUJE SHODNÝ INTERFACE JAKO SERVISNÍ OBJEKT
- DECORATOR A PROXY
 - PODOBNÁ STRUKTURA, ALE ROZDÍLNÝ ÚČEL
 - PROXY SPRAVUJE ŽIVOTNÍ CYKLUS SERVISNÍHO OBJEKTU
 - STRUKTURA DECORATORU JE ŘÍZENA KLIENTEM
- FACADE DEFINUJE NOVÉ ROZHRAŇÍ, ADAPTER SPOJUJE EXISTUJÍCÍ ROZHRAŇÍ

VZTAH BRIDGE, ADAPTER, BUILDER

- BRIDGE – NAVRHOVANÝ OD POČÁTKU PRO MOŽNOST NEZÁVISLÉHO ROZVÍJENÍ ABSTRAKCE A IMPLEMENTACE - ŘEŠÍ SE V PRŮBĚHU NÁVRHU
- ADAPTER – ÚPRAVA UMOŽŇUJÍCÍ SPOLUPRÁCI JIŽ EXISTUJÍCÍCH NEODPOVÍDAJÍCÍCH SI ROZHRAŇÍ – ŘEŠENÍ PRO OBJEKTY, KTERÉ JIŽ BYLY NAVRŽENY
- BUILDER MŮŽE BÝT STRUKTUROVANÝ JAKO BRIDGE – DIRECTOR PŘEDSTAVUJE ABSTRAKCI A BUILDERS PŘEDSTAVUJÍ IMPLEMENTACE
- ABSTRACT FACTORY – MŮŽE BÝT POUŽITA SPOLU S BRIDGE, UŽITEČNÁ V PŘÍPADĚ, ŽE ABSTRAKCE MŮŽE SPOLUPRACOVAT POUZE S URČITOU IMPLEMENTACÍ

VZTAH MEZI COMPOSITE A DECORATOR

- PODOBNÁ REKURZIVNÍ STRUKTURA
- DECORATOR MŮŽE BÝT VNÍMÁN JAKO ZVLÁŠTNÍ PŘÍPAD COMPOSITE POUZE S JEDNOU KOMPONENTOU.
- NICMÉNĚ DECORATOR SLOUŽÍ PRO PŘIDÁNÍ DALŠÍ ZODPOVĚDNOSTI, KDEŽTO COMPOSITE UMOŽNÍ TRANSPARENTNÍ VOLÁNÍ METODY BEZ OHLEDU NAT TO, V JAKÉ ČÁSTI HIERARCHIE SE NACHÁZÍM
- NÁVRH, KTERÝ VÝRAZNĚ POUŽÍVÁ COMPOSITE A DECORATOR VELMI ČASTO VYUŽÍVÁ TAKÉ PROTOTYPE – KLONOVÁNÍ SLOŽITÉ STRUKTURY NAMÍSTO JEJÍHO VYTVÁŘENÍ OD NULY

ZDROJE

- ERICH GAMMA, RICHARD HELM, RALPH JOHNSON, JOHN VLISSIDES: DESIGN PATTERNS: ELEMENTS OF REUSABLE OBJECT-ORIENTED SOFTWARE; ADDISON-WESLEY; 1994
- [HTTPS://EN.WIKIPEDIA.ORG/WIKI/DESIGN_PATTERNS](https://en.wikipedia.org/wiki/Design_Patterns)
- [HTTPS://SOURCEMAKING.COM/DESIGN_PATTERNS/STRUCTURAL_PATTERNS](https://sourcemaking.com/design_patterns/structural_patterns)