

29.

Návrhové vzory – uveďte příklad a příklad využití minimálně 5 návrhových vzorů.
Uveďte a popište různé oblasti a druhy aplikace jednotlivých vzorů.

Návrhové vzory

- jedná se o popis řešení problému nebo šablonu, která může být použita v různých situacích
- Objektově orientované návrhové vzory typicky ukazují vztahy a interakce mezi třídami a objekty
- Algoritmy nejsou považovány za návrhové vzory, protože řeší konkrétní problémy a nikoliv problémy návrhu

Creational Patterns (vytvářející)

- Řeší problémy související s vytvořením objektů v systému.
- Snaha popsat postup výběru třídy nového objektu a zajištění správného počtu těchto objektů.
- Většinou se jedná o dynamická rozhodnutí učiněná za běhu programu.

Abstract factory (Abstraktní továrna)

- Poskytnutí interface pro vytváření rodin souvisejících objektů, aniž by bylo třeba specifikovat jejich konkrétní třídu
- Jednotlivé továrny určují přímo typy objektů, které vytvářejí. Ve většině jazyků se tak obvykle děje použitím klíčového slova **new**. Nicméně továrna vrací abstraktní ukazatel na vytvořený konkrétní objekt.
- Díky tomu **není klientský kód zatěžován vytvářením objektů**, ale o vytvoření požádá objekt továrny, která vrací objekt požadovaného abstraktního typu pomocí ukazatele na tento objekt.prototypovou instanci, která je klonována pro vytváření nových objektů
-
- **Př:** CREATEBUTTON - nezajímá mě jak to funguje na pozadí

Prototype (prototyp)

- Používá se, když typ objektu, který je třeba vytvořit, je určen prototypovou instancí, která je klonována pro vytváření nových objektů
- Místo toho abychom tvořili objekt od nuly, **vytvoření kopie** původní instance a upravíme je podle potřeby
- **Př:** Dělicí se buňky.

Singleton (jedináček)

- Řeší problém, kdy je třeba, aby v celém programu běžela **pouze jedna instance třídy**
- Zabezpečuje, že třída bude mít jedinou instanci a poskytne k ní **globální přístupový bod**

- **Př:** "V systému by měl být pouze jeden správce oken (souborový systém, zařazovací jednotka tisku). Singleton se zpravidla používá pro centralizované řízení vnitřních nebo vnějších zdrojů a poskytuje jim globální přístup"

Object pool (bazén objektů)

- Pokud potřebujeme větší množství instancí, které jsou časově náročné na vytvoření, ale nepotřebujeme jich mnoho zároveň.
- Namísto **neustálé tvorby nových instancí použijeme** ty, které jsme **vytvořili již dříve** a nyní nejsou používané.
- **Př:** Půjčovna kostýmů
- **Př:** "Když chceme najmout nového zaměstnance tak pro něj připravíme pracovní plochu (PC, stůl, židle). Pokud už není prostor tak ho nemůže najmout. Pokud ještě je místo tak kouknem do skladu jestli máme, ještě vybavení. Pokud ještě máme tak mu ho dáme, jinak ho objednáme v obchodě další. A až ho vyhodíme tak stále budem mít vybavení pro dalšího zaměstnance. "

Structural Patterns (strukturální)

- Zaměřující se na možnosti uspořádání jednotlivých tříd nebo komponent v systému.
- Snahou je zpřehlednit systém a využít možností strukturalizace kódu.

Adapter (adaptér)

- Slouží jako prostředník mezi prostředím, které požaduje nějaké rozhraní, a třídou,
- jejíž rozhraní neodpovídá požadovanému.
- Umožňuje spolupráci třídám, které by spolu jinak spolupracovali.
- **Př:** Nabíječka na telefon, API, ETL (datová pumpa), EAI

Proxy

- Nabízí náhradu nebo zástupný objekt za nějaký jiný pro kontrolu přístupu k danému objektu
- **Př:** Šek jako proxy pro peníze na účtě

Behavioral Patterns (chování)

- Mohou být založeny na třídách nebo objektech.
- U tříd využívají při návrhu řešení především principu dědičnosti.
- V druhém přístupu je řešena spolupráce mezi objekty a skupinami objektů, která zajišťuje dosažení požadovaného výsledku.

Iterator (Iterátor)

- Zajišťuje možnost procházení prvků bez znalosti jejich vnitřní implementace
- **Př:** Tlačítko next, for loop

Mediator (Prostředník)

- Mezi komunikující objekty je vložen prostředník
- Někdy může být mezi komunikující objekty vloženo i několik prostředníků.
- Objekty si **nemusí pamatovat odkazy na všechny ostatní objekty**, ale pouze na prostředníka.
- **Př:** Řídicí věž na letišti (místo toho, aby každé letadlo blízko letiště komunikovalo se všemi okolo tak komunikace se směřuje na řídicí věž. Je důležité si uvědomit, že věž nekontroluje celý let. Existuje pouze pro vynucení omezení v oblasti terminálu.)

Dependency injection, Inversion of control (Obrácení řízení)

- Je technika, při které **jeden objekt** (nebo statická metoda) **dodává závislost** jinému objektu.
- Závislost je objekt, který lze použít (služba).
- Vstřikem je přemístění závislosti na závislý objekt (klienta), který by ho použil.
- Služba je součástí stavu klienta.
- Poskytování služby klientovi, spíše než umožnit klientovi udělat nebo najít službu, je základním požadavkem tohoto vzoru
- **Př:** "Nevolejte, my vám zavoláme"

Concurrent patterns

Vzory řešící problémy vzniklé spouštěním programů ve vláknech a tudíž při souběžném řešení úlohy

Guarded (Chráněný)

- Vzor obstarávající operace, které požadují uzamčení a navíc mají podmínku, která musí být splněna předtím, než může být operace provedena.

Thread pool (Bazén vláken)

- V bazénku vláken je vytvořen **určitý počet vláken pro řešení úloh**, které jsou organizovány ve frontě.
- Zpravidla je výrazně **více úloh než vláken**