

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

HARDWARE A APLIKAČNÍ SOFTWARE

- Design pattern
- Představuje obecné řešení problému.
- Šablona, popis řešení, interakce mezi třídami.
- Nejedná se o algoritmus (konkrétní způsob řešení).

Vytvářecí vzory

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

Abstraktní továrna

- Definuje rozhraní pro vytváření rodin objektů.
- Např. Create button (vytvořit tlačítko, nezajímá mě, jak ho vytvořím)
- Aplikace nemá tušení, jaký druh GUI továrna vytvoří.

Object pool

- Bazén objektů, které použijeme při potřebě.
- Umožňuje sdílení zdrojů pomocí recyklování objektů.
- Při nedostatku prostředků pořízení nových nebo časová prodleva.
- Např. půjčovna kostýmů (půjčím si kostým (objekt), upravím ho a použiji, vrátím do poolu)

Prototyp

- Nové objekty se vytváří kopírováním celého objektu (klon) nebo jeho části.
- Např. klonovací razítko

Strukturální vzory (Structural patterns)

- Představují skupinu návrhových vzorů zaměřující se na možnosti uspořádání jednotlivých tříd nebo komponent v systému.
- Snahou je zpřehlednit systém a využít možnosti strukturalizace kódu.

Adaptér

- Převádí rozhraní jedné třídy na rozhraní druhé třídy.
- Umožňuje přístup k elementům skupinového objektu postupně bez toho, abyste vystavovali vnitřní reprezentaci tohoto objektu.
- Typy: API, ETL, EAI
- Např. adaptér do zásuvky (eu/usa koncovka)

Proxy

- Nabízí zástupný objekt za nějaký jiný pro kontrolu přístupu k danému objektu.

- Využití v internetu, všechny lokální (neveřejné) IP adresy přistupují k internetu přes Proxy server pod jedinou IP adresou.

Vzory chování (Behavioral patterns)

- Zajímají se o chování systému.
- 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)

- Umožňuje přístup k elementům skupinového objektu postupně bez toho, abyste vystavovali vnitřní reprezentaci tohoto objektu.

Mediator (Prostředník)

- Umožňuje zajistit komunikaci mezi dvěma komponentami programu, aniž by byly v přímé interakci a tím musely přesně znát poskytované metody.

Souběžné vzory (Convurrent patterns)

- Vzory řešící problémy vzniklé spouštěním programů ve vláknech, a tudíž při souběžném řešení úlohy.
- Např. 2 aktualizací SQL dotazy.

Guarded

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

Thread pool

- V bazénku vláken je vytvořen nějaký počet vláken, pro řešení nějakého množství úloh, které jsou organizovány ve formě (FIFO).
- Zpravidla je výrazně více úloh než vláken.