VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

WAP – Internetové aplikace Projekt 1: Memento

Návrhový vzor – Memento

Memento je návrhový vzor, který slouží k uložení vnitřního stavu objektu při zachování principu zapouzdření. Nejedná se o kompletní záznam změn, nýbrž jen o jeden konkrétní stav, rozšířit vzor by však nebylo těžké, stačilo by ukládat změny příklad do zásobníku. Takový způsob ukládání by však mohl vést na nadužívání paměti, proto může být u takového rozšíření vhodné ukládat pouze měněné údaje, kdežto u základního vzoru memento standardně není problém ukládat celý vnitřní stav objektů.

Motivací pro tento návrhový vzor je možnost návratu k předchozímu stavu objektů. Toho můžeme chtít docílit například, když uživatel začne vyplňovat formulář ke změně údajů a v posledním kroku zruší celou operaci, aby následně nebylo nutné žádat server duplicitně o původní data.

Vzor využívá následující třídy:

- Originator Označení pro třídy, jejichž stav ukládáme. Tyto třídy musí implementovat metody pro uložení stavu do nového mementa a obnovení stavu z daného mementa.
- *Memento* Třída, která uchovává reprezentaci vnitřního stavu objektů tříd *Originator*. Neobsahuje žádnou další vnitřní logiku.
- Caretaker Tato třída obstarává ukládání a načítání mementa do/z originatoru.

Implementace a API

Originator

Jako základ tříd *Originator* jsem implementoval jednoduchou třídu *OriginatorBase*, kterou následně třídy rozšiřují. *OriginatorBase* poskytuje privátní proměnnou *id*, se kterou pracuje třída *Caretaker* a uživateli je úplně skryta. Tento přístup jsem zvolil, aby šlo snadno přidávat mementa nových objektů do pole v rámci třídy *Caretaker* a šlo snadno a jednoznačně odvodit, ke kterému objektu dané memento patřilo, a to, protože v jazyce JavaScript nemá každý objekt jednoznačné oid, které by jinak šlo uplatnit např. v rámci slovníku (struktury dict). Jako třídy *Originator* pak lze označit třídy *Element, Spell, Wizard* a *Grimoar*, které jsem implementoval jako odlehčenou variantu ze zadání z předmětu IUS "Svět Magie" pro testování a předvedení návrhového vzoru. Každá třída implementuje vlastní variantu metod *saveState* a *loadState*, které jsou využívány skrze třídu *Caretaker*.

Uživatelská část poskytuje konstruktory tříd a gettery a settery pro následnou práci s jednotlivými atributy, ty jsou vhodně okomentované pro poskytnutí nápovědy správných typů pro používání.

Caretaker

Třída *Caretaker* se interně stará o přidělení unikátního ID objektům, které má zachovávat a následně je využívá pro indexaci mement ve svém poli.

Dále poskytuje uživateli API pro uložení aktuálních a načítání původních stavů objektů za pomoci funkcí *loadState* a *saveState*, které přijímají jako parametr konkrétní objekt tříd *Originator*. Jako drobné rozšíření od základního vzoru pak implementuje pro uživatelskou pohodlnost také metody *loadStates* a *saveStates*, kterým se zadává pole objektů, a které následně volají výše zmíněné metody nad každým objektem v poli.

Memento

Poslední třída Memento implementuje pouze vytvoření nového mementa s uložením nějakých údajů do jedné proměnné a vrácení obsahu této proměnné, uživatel s touto třídou přímo nepracuje.

Zdroje

Z následujících zdrojů jsem vycházel při snaze pochopit jakým způsobem má návrhový vzor fungovat, všechny obsahují různé ukázky kódu, které se principiálně liší hlavně v implementaci třídy *Caretaker*, ve které se zároveň nejvíce liší i mé řešení, které využívá navíc id přidělené do *Originator* tříd, pro které jsem také vytvořil navíc základní třídu *OriginatorBase*.

- https://www.itnetwork.cz/navrh/navrhove-vzory/gof/memento
- http://voho.eu/wiki/memento/
- https://designpatternsgame.com/patterns/memento
- https://dev.to/shikaan/design-patterns-in-web-development---2-memento-253j