

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



INTERNETOVÉ APLIKACE
2020/2021

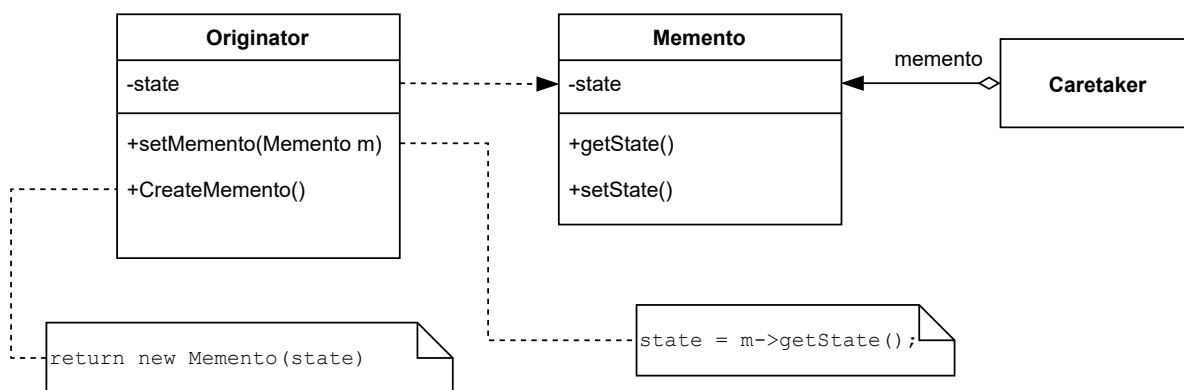
Projekt 1: Objekty v jazyce JavaScript Memento

Tomáš Odehnal (xodehn08)

Brno, 7. března 2021

1 Struktura

Cílem tohoto návrhového vzoru je uložení a zpřístupnění vnitřního stavu objektu, tak že objekt může být obnoven do toho stavu později. Zároveň je zachováno pravidlo zapouzdření. Typickým případem použití návrhového vzoru memento, když potřebujeme objekt vrátit do předchozího stavu (angl. "undo" nebo "rollback" operace).



Obrázek 1: Třídní diagram návrhového vzoru Memento.

Části návrhového vzoru:

- **Memento** - Uchovává stav objektu Originátor. Obsah uložené proměnné závisí pouze na Originátorovi.
- **Originátor** - Vytváří memento obsahující snímek jeho aktuálního vnitřního stavu. A používá memento k obnovení svého předchozího vnitřního stavu.
- **Caretaker** - Je zodpovědný za bezpečné uložení mementa. Nikdy nepřistupuje nebo neprovádí operace nad obsahem mementa.

2 Můj příklad použití

Návrhový vzor memento jsem použil v aplikaci, která má za úkol seřadit vstupní pole čísel. Toto řazení se provádí po jednotlivých krocích, kde přepínání mezi nimi je řízeno uživatelem aplikace. Zároveň je možno v procesu řazení provádět i kroky zpět.

2.1 Návrh

Aplikace bude rozdělena na dvě části. První bude obsahovat memento a možnost jeho uložení do datové struktury (např. pole). Druhá část bude používat prvky z první části. Zde bude implementace řadicího algoritmu a možnost interakce s uživatelem. Memento se bude vytvářet v Originátoru, což je v případě této aplikace řadicí algoritmus, a následně v hlavní části programu se uloží do datové struktury z první části. Díky uložení do této struktury bude možné provádět i kroky zpět, protože se budou moci uložená mementa použít k navrácení algoritmu k předchozím stavům. Po každém kroku se uživatel, podle vypsaných informací, rozhodne o následující akci, kterou aplikaci předává v podobě příkazů. Uživateli je k dispozici nápověda. Pro ilustraci jsem zvolil implementaci jednoduchého řadicího algoritmu Bubblesort.

2.2 Implementace

Podle návrhu je aplikace rozdělena do dvou souborů:

`library.mjs` obsahuje definici objektů **Memento** a **Caretaker**.

Memento obsahuje atribut `state`, ve kterém bude uložen stav Originátoru (řadicího algoritmu). Tato proměnná je privátní, tedy k ní mimo tento objekt nemá nikdo přístup. Přístup k ní je možný pouze pomocí definovaného rozhraní v podobě funkce `setState(st)`, která uloží obsah proměnné `st` do proměnné `state` a funkce `getState()`, která vrátí obsah proměnné `state`.

Caretaker obsahuje proměnnou `mementoList`, což je seznam chronologicky seřazených objektů `memento`. Objekt umožňuje přidat na konec tohoto seznamu další `memento` pomocí metody `add(memento)`. Také umožňuje smazat a vrátit poslední přidáný prvek to seznamu pomocí metody `pop()`. Seznam `mementoList` je tedy implementovaný jako zásobník. Tento soubor je implementovaný jako modul a exportu je tyto dva definované objekty.

`module.js` obsahuje implementaci řadicího algoritmu a interakci s uživatelem aplikace. Objekty **Memento** a **Caretaker** jsou do souboru `naimportovány`. Kvůli interaktivitě s uživatelem je hlavní funkce `run()` definována jako asynchronní. V ní je s operátorem `await` volána funkce `repl(bubblesort, caretaker)` (s objekty algoritmu a `caretaker`), která vrací `Promise`, ve kterém je implementována tato interakce s uživatelem. Před provedením dalšího kroku řazení, vytvoří algoritmus novou instanci `Memento`. Následně uloží stav algoritmu do této instance, provede krok řazení a vrátí tuto instanci (již s uloženým stavem). Ve funkci `repl` je toto `memento` uloženo pomocí objektu `Caretaker`. Při vrácení o krok zpět je z objektu `Caretaker` pomocí funkce `pop()` získám objekt `Memento`. Stav uložený v tomto `Memento` je předá funkci `prevStep()` řadicího algoritmu a je obnoven stav, před provedením poslední kroku řazení.

3 Spuštění a ovládání

Aplikace byla testována pomocí Node.js verze 14.15.5. Pro spuštění aplikace je potřeba vytvořit soubor s názvem `package.json`, do kterého je potřeba vložit následující:

```
{
  "type": "module"
}
```

Aplikace by pak měla jít spustit následujícím příkazem¹:

```
$ node module.js <numbers>
```

Kde `<numbers>` je seznam čísel a jednotlivá čísla jsou oddělená čárkou (bez mezery) např. `5, 6, 8, 3, 1`.

Aplikaci je možné ovládat pomocí nabízených příkazů:

- `h` - vypíše nápovědu pro ovládání aplikace.
- `e` - ukončí aplikaci.
- `n` - provede aktuální krok řadicího algoritmu.
- `p` - vrátí se k předchozímu kroku řadicího algoritmu.

¹Předpokládá se, že cesta, ke spuštění `node` je v `PATH` proměnné.