FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

WAP – Projekt č. 1
Objekty v jazyku JavaScript – Reťazec zodpovednosti

Ján Folenta (xfolen00) 13.3.2021

1 Reťazec zodpovednosti

Reťazec zodpovednosti je návrhový vzor, ktorý umožňuje posielať požiadavku sériou objektov nazývaných handlers, pričom každý objekt sa môže rozhodnúť, či požiadavku spracuje alebo ju pošle ďalšiemu objektu v reťazci pre spracovanie (prípadne môže požiadavku aj spracovať aj ju poslať následníkovi). Jednotlivé objekty sú navzájom pospájané pomocou referencií a tvoria tak reťazec zodpovednosti. Tento návrhový vzor je možné použiť napríklad v stromových hierarchiách, kde je požiadavka postupne predávaná smerom ku koreňu stromu, pri procese, pri ktorom dochádza k postupnému spracovaniu požiadavky alebo v prípade, že si nechceme udržiavať referencie na každý handler.

2 Implementácia

Na demonštráciu vybraného návrhového vzoru som sa rozhodol použiť príklad kontroly správneho formátu údajov (meno, emailová adresa a telefónne číslo), ktorý je možné použiť pri spracovaní formulára. Implementované boli 2 súbory a to **library.js** a **model.js**.

2.1 Library.js

V tomto súbore sú implementované univerzálne objekty, ktoré implementujú daný návrhový vzor. Súbor obsahuje 5 tried, a to triedy *Person, Handler, NameHandler, EmailHandler* a *PhoneNumberHandler*. Trieda *Person* slúži na uloženie a prístup k údajom, ktoré budú spracovávané. Pre demonštračné účely sú ukladané len 3 typy údajov a to meno, emailová adresa a telefónne číslo. Trieda *Handler* poskytuje rozhranie pre zvyšné triedy a obsahuje metódu ***nextObjectInChain(nextObj),*** ktorá nastaví následníka v reťazci zodpovednosti a metódu ***processData(person),*** ktorá bude kontrolovať správnosť formátu predaných dát. Triedy *NameHandler, EmailHandler* a *PhoneNumberHandler* implementujú metódy triedy *Handler*. *NameHandler* vykoná kontrolu správnosti formátu mena objektu *person*. V prípade, že meno je v nesprávnom formáte, tak je vyvolaná výnimka a program končí s chybou. V prípade, že formát je správny, tak sú dáta poslané následníkovi v reťazci zodpovednosti, ktorý vykoná kontrolu ďalšieho údaju objektu *person* a takým spôsobom je prejdená celá reťaz. *EmailHandler* vykonáva iba základnú kontrolu správnosti formátu emailovej adresy a *PhoneNumberHandler* vykonáva kontrolu tel. čísla vo formáte +421xxxxxxxxx. Na konci sú vytvorené objekty a na základe metódy ***nextObjectInChain(nextObj)*** je vytvorená reťaz zodpovednosti.

2.2 Model.js

Súbor **model.js** demonštruje využitie návrhového vzoru implementovaného v súbore **library.js**. Sú v ňom vytvorené 2 objekty (*person1* a *person2*), ktoré sú predané reťazcu zodpovednosti, ktorá vykoná kontrolu správnosti formátu dát. Objekt *person1* obsahuje správny formát dát, tým pádom reťazec zodpovednosti nevyvolá žiadnu výnimku a po vykonaní kontroly je na štandardný výstup vypísaná správa „Success“. Objekt *person2* obsahuje telefónne číslo s chybným formátom, preto reťazec zodpovednosti vyvolá výnimku a program skončí s chybou.

3 Použitie

Program sa spúšťa príkazom:

```
./node model.js
```

Pri implementácii a testovaní bol použitý node vo verzii 14.16.0. Pre vlastné otestovanie návrhového vzoru je potrebné v súbore **model.js** vytvoriť objekt *person* s potrebnými údajmi a následne zavolať metódu pre kontrolu dát:

```
const person = new library.person("name", "email", "number")

library.nameHandler.processData(person)
```

(Hlavný informačný zdroj, z ktorého bolo čerpané pri riešení je:
https://refactoring.guru/design-patterns/chain-of-responsibility)