

Peter Markuš - Finančné spravovanie vo veľkoobchode

Cieľom práce je vytvoriť systém pre spravovanie financií veľkoobchodu. Spravovanie spočíva v investíciách do produktov alebo rôznych typov zamestnancov, plniac každý svoju určitú úlohu v obchode. **Používateľské rozhranie** je sprostredkované formou CLI ktorá je zahrnutá v triede Konzola. Používa tri hlavné príkazy: nakupovanie produktov alebo zamestnancov, výpis informácií ohľadom obchodu a štart dňa.

(Splnenie hlavných kritérií)

Na typoch zamestnancov bolo použité **dedenie** triedy zamestnanca, ktorý zároveň dedil od abstraktnej triedy Človek. Každý zamestnanec môže vyhodiť človeka s výnimkou zamestnanca. Pri tejto príležitosti som použil **prekonávanie** vlastnej metódy u administrátora, ktorý mohol vyhodiť aj zamestnanca. Pri vyhodení zákazníkov administrátorom bol použitý **polymorfizmus**. V triede Velkoobchod som zakomponoval **agregáciu** objektov ako je sklad, predajňa, pokladňa či košík. Najviditeľnejšie **zapúzdrenie** je možné vidieť pri košíku, ktorý má produkty ako atribúty. Triedy som zorganizoval do troch **balíkov**: interagujuceObjekty, objektyPreManipuláciu a veľkoobchod.

(Splnenie ďalších kritérií)

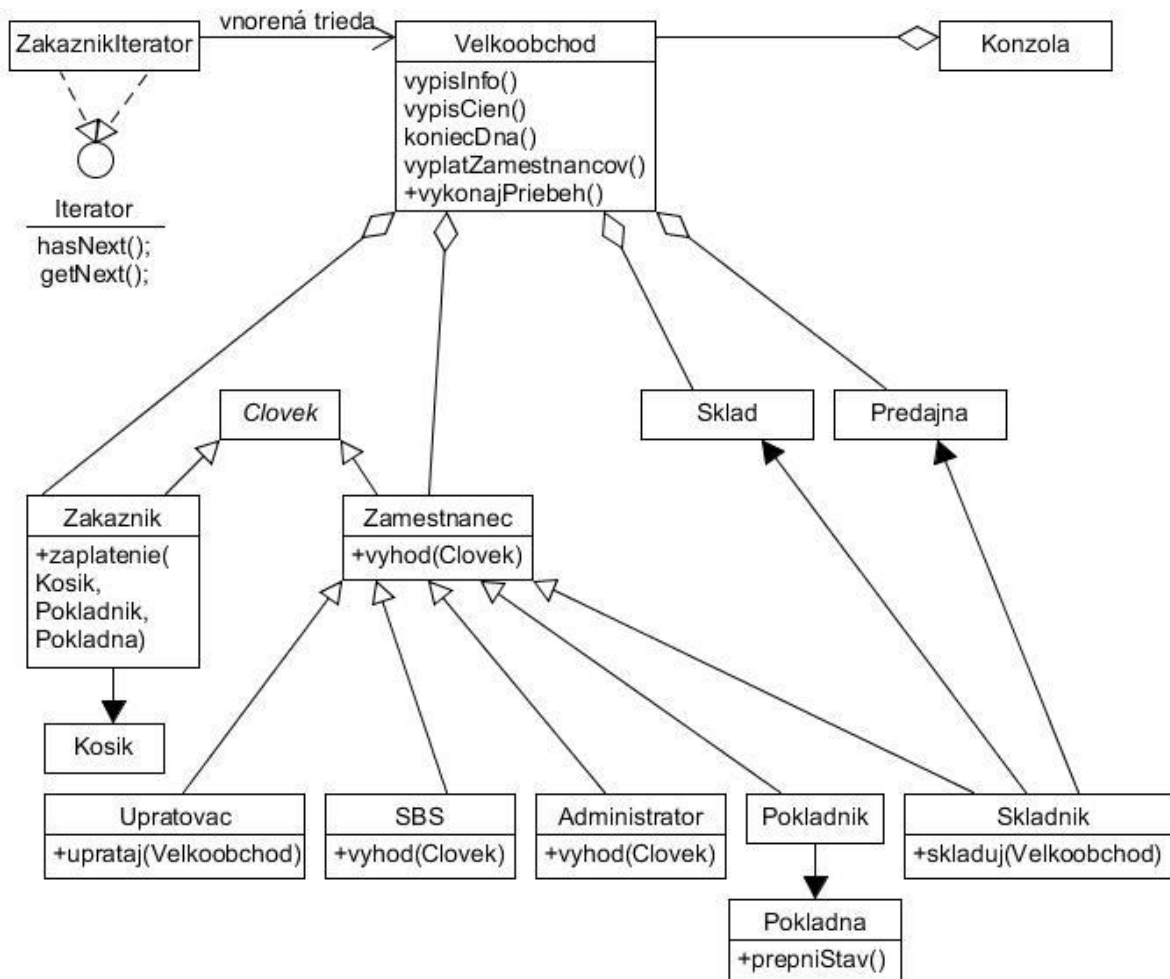
Ošetrovanie mimoriadnych stavov (**výnimky**) som použil pri počte kupovaných produktov, pričom keď nebolo zadané číslo, tak rozhranie informuje používateľa o jeho nesprávnom zadaní čísla.

Pri štarte dňa je využitý **návrhový vzor** iterátor, ktorý som použil na spracovanie každého zákazníka zvlášť. Na jeho zrealizovanie som vytvoril **vnorenú triedu** ZakaznikIterator ktorá implementovala **rozhranie** Iterator.

Na zjednodušenie zamestnávania nových pracovníkov obchodu bola použitá **generickosť** v triede Velkoobchod ako zoznam zamestnancov.

Identifikáciu typov v čase vykonávania (**RTTI**) som použil pri metóde vyhod v triede Zamestnanec, ktorý zaručoval to, aby nebolo možné vyhodiť ktoréhokoľvek zamestnanca. To mohol jedine už spomínaný administrátor pri ktorom sa táto metóda prekonala.

UML Diagram:



Pri štarte dňa sa postupne spracúva každý zákazník zvlášť metódou `vykonajPribeh()` u **Velkoobchodu**. V tejto metóde každý zamestnanec koná svoju úlohu. Pokladníci používajú pokladňu na to, aby mohol zákazník zaplatiť metódou `zaplatenie(...)` a čím je väčší počet pokladníkov v obchode, tým bude väčšia pravdepodobnosť, že spracujú všetkých zákazníkov cez deň. Podobne to takto funguje aj u ostatných typoch zamestnancov. Čím je teda väčší počet, tým viac budú stíhať svoje úlohy. Skladníci skladujú keď sa míňa tovar a teda manipulujú triedami **Sklad** a **Predajna**. SBS dávajú pozor na zlodejov, ak však tam žiadny nieje prítomný, tak človeka vyhodia iný zamestnanci ak ho prichytia metódou `vyhod(Clovek)`. Zákazník pri nákupe používa košík do ktorého pridáva produkty za ktoré zaplatí pri pokladni.

Zoznam odovzdaných pracovných verzií programu:

Práca na projekte – 6. Cvičenie – “static atributy a vela opakovaného kodu este opravim” (zlý popis, keďže to bola prvá verzia programu, neboli v nej správne využité objektovo-orientované princípy)

Práca na projekte – 8. Cvičenie – “” (prerobenie celého projektu s účelom využitia objektovo-orientovaných princípov)

Práca na projekte – 9. Cvičenie – “Oprava polymorfizmu a konvencii”

Práca na projekte – 11. Cvičenie – “” (vytvorenie triedy Rozhranie a aplikovanie návrhového vzoru Iterátor, pridanie nových metód do Velkoobchod)

Finálna verzia – úprava konvencií, premenovanie balíkov a triedy, zmena statických metód na nestatické u triedy Konzola, úprava metódy vyhod())