

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Objektovo-orientované programovanie

Semestrálna práca OOP
Simulácia hutníckej firmy

Jakub Hrnčár
AIŠ ID: 110 800, ak. rok 2020/2021

1. Zámer práce

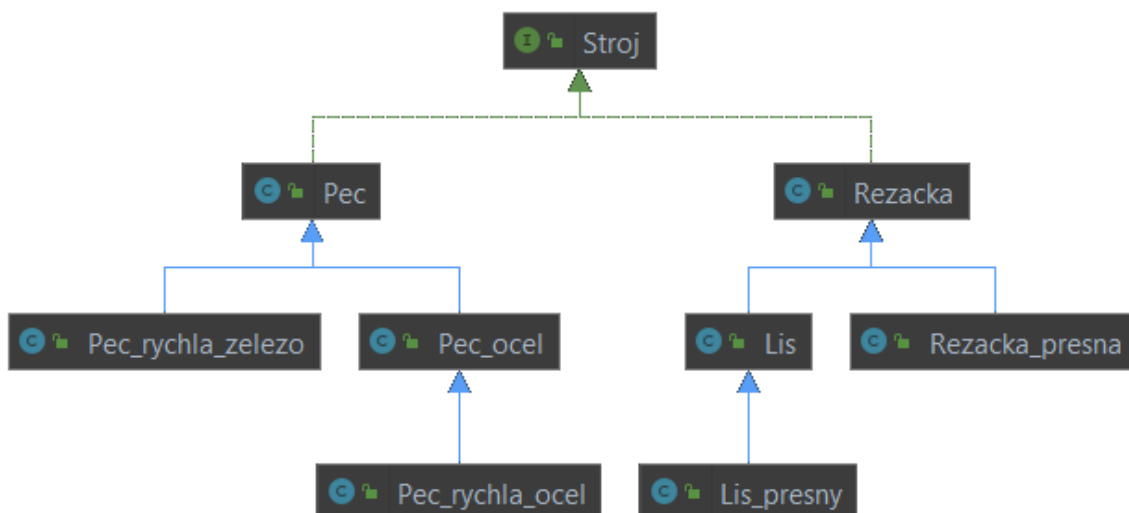
Hutnícky priemysel je celosvetovo stále veľmi potrebný, preto je nutné, aby sa čo najviac zefektívnili jeho výrobné a plánovacie procesy. Úlohou hutníckeho priemyslu je efektívne spracovávať surové rudy a vytvárať z nich použiteľné kovy alebo už priamo hotové výrobky. celý proces je ale veľmi komplexný, napríklad: železná ruda sa musí kúpiť, priviesť, očistiť, roztaviť a následne sa z nej vytvára buď hotový produkt, alebo sa používa ďalej na výrobu ocele. V tomto procese je zapojených množstvo zamestnancov, ktorých je nutné menežovať. V tomto odvetví priemyslu je extrémne dôležité, aby výroba sa nezastavila, pretože je veľmi náročné a veľmi drahé ju následne obnoviť (rozžhaviť pece a pod.). Preto je mojou úlohou navrhnúť taký systém, ktorý naplánuje všetky procesy od zadania objednávky až po vývoz hotového produktu.

2. Hlavné kritériá

Dve 3-vrstvové hierarchie dedenia s implementovaným rozhraním:

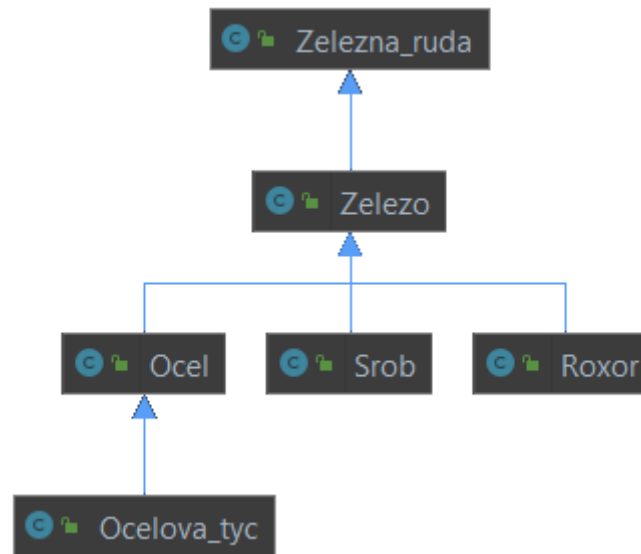
Umiestnenie: model.pristroj

Diagram:



1. hierarchia: Rozhranie Stroj->Trieda Pec->Trieda Pec_Ocel->Trieda Pec_rychla_ocel
->Trieda Pec_rychla_zelezo
2. hierarchia: Rozhranie Stroj->Trieda Rezacka->Trieda Lis->Trieda Lis_presny
->Trieda Rezacka_presna

Poprípade tretia hierarchia, v ktorej ale nie je využité rozhranie:



Polymorfizmus:

Umiestnenie: 1. `model.pristroj.pec`
`model.pristroj.pec.rychla.zelezo`

2. `model.pristroj.lis`
`model.pristroj.lis_presny`

1. V triede `Pec` používam metódu `spracuj()`, ktorú prepisujem v zdedenej triede `Pec_rychla_zelezo`. Rozdiel medzi nimi je, že v triede `Pec` je táto metóda implementovaná s možným pozastavením programu, avšak v triede `Pec_rychla_zelezo` takéto obmedzenie nie je.
2. V triede `Lis` je používam metódu `spracuj_zelezo()`, ktorú prepisujem v zdedenej triede `Lis_presny`. Rozdiel medzi nimi je, že v triede `Pec` môže táto metóda s 15% šancou zmeniť atribút `power` na `false`, avšak v triede `Lis_presny` sa toto nedeje.

V programe je prepísaný viac, všetky sú označené anotáciou

Zapuzdrenie:

Umiestnenie: `model.pristroj.Pec`

Atribút `teplo` má obmedzenie `protected`, kde následne sú metódy getter - `getTeplo()`, a setter - `zohrej()`.

Metódy, ktoré fungujú ako settery a gettery je v programe viac.

Agregácia:

Umiestnenie: `mode.zamestnanec.Obsluha`

Triede obsluha je agregáciou priradený objekt Pec, s ktorým následne pracuje.

MVC Model: Program je riadne členený do MVC modelu a dodržiava ho.

Balíky: Triedy programu sú podľa vnútornej logiky programu jasne rozdelené do balíkov.

3. Vedľajšie kritériá:

Vlastná výnimka:

Umiestnenie: `control.Invalid_input`

Zavolá sa v prípade, že do objednávky vstúpili iba nuly. Je v try-catch bloku vo `view.GUI` a vyhadzuje ju `control.control.spracuj()`

GUI:

Umiestnenie: `view.GUI`

GUI má štyroch spracovateľov udalostí, pre každé tlačidlo jeden. Je riadne oddelené od aplikačnej logiky.

Multithreading:

Umiestnenie: `model.pristroj.Pec.spracuj()`

Multithreading je využitý len zľahka, kedy je volaná metóda `Thread.sleep()`, slúžiaca na pozastavenie programu.

RTTI:

Umiestnenie: `model.zamestnanec.Manazer.uzavri_objednavku()`

Používam operátor `instanceof`, aby som zistil správne spracovanie objednávky a následne ju uzavrel.

Lambda:

Umiestnenie: `model.pristroj.Pec.spracuj()`

Lambda výraz tu je použitý na rýchly výpočet času, ktorý bude potrebný na zastavenie programu, kvôli nízkej hodnote tepla v peci.

Serializácia:

Umiestnenie: `model.zamestnanec.Manazer.uzavri_objednavku()`

Po správnom uzavretí objednávky manažér serializuje výstupný zoznam "taska", v ktorom sú uložené všetky vytvorené produkty, ako si ich užívateľ objednal. Táto trieda sa následne môže v dodatku ku tomuto programu deserializovať a je možné s touto zásielkou ďalej pracovať.

4. Verzie v GitHube

30.3.2021 - Začiatok práce

12.4.2021 - Odovzdanie pracovnej verzie

22.4.2021 - Pridanie triedy Skladník

24.4.2021 - Serializácia, RTTI

25.4.2021 - GUI

26.4.2021 - MVC, lambda, zmena polymorfizmu, rozdelenie hierarchií

27.4.2021 - Vlastný exception, posun a výpis objednávky

14.5.2021 - JavaDoc, Dokumentácia