Fakulta riadenia a informatiky

Informatika

SEMESTRÁLNA PRÁCA I. – ALGORITMY A ÚDAJOVÉ ŠTRUKTÚRY I.

# AoE

Ing. Michal Varga, PhD. – PIA 12:00

2018/2019 Dominik Bullo 5ZY029

## Návrh aplikácie

Aplikáciu som rozvrhol do nasledujúcich tried: CentrralnySklad, Datum, Dron, Firma, Konstatny, Objednavka, Prekladisko, UI, Vozidlo, Zasielka.

Trieda CentralnySklad zastrešuje všetky veci, dejúce sa v centralnom sklade, ako napríklad prijatie zásielok z vozidiel, preloženie do vozidiel, ktoré budú na danej trase opäť ráno rozvážať zásielky a pod.

Trieda Datum rieši prácu s dátumom, poskytuje nám len jednu inštanciu, obsahuje statické metódy na prekonvertovanie dátumu na string a naopak a taktiež posúvanie dní.

Trieda Dron obsahuje všetky potrebné informácie ohľadom drona, ako je napríkla stav nabitia, počet nalietaných kilometrov, dokedy bude dron vyťažený a pod. Taktiež v nej môžem vytvárať rôzne typy dronov, s rozličnými priemernými rýchlosťami, časom nabíjania, či maximálnou dobou letu (podľa zadania). Trieda má front zásielok, ktorý má v daný deň spracovať.

Trieda Firma zastrešuje jednotlivé funkcie a spájanie ostatných tried, uchováva zoznamy na všetky prekladiská, všetky autá ktoré má k dispozicií a taktiež všetky objednávky, ktoré dostala. Taktiež rieši celý proces prijatia objednávky, aj s možnosťou zamietnutia, zo strany firmy, alebo usera.

Trieda Konstanty obsahuje ako vyplýva z názvu konštanty a enumy pre jednoduchšie sa orientovanie v kóde, prípadnú zmenu.

Trieda Objednavka obsahuje ešte subclassy Odosielatel a Adresat, kedy si pri každom uchováva región a vzdialenosť od lokálneho prekladiska v danom regióne. Objednávka ďalej vie svoj stav, dátum a čas spracovania, či dôvod priípadného zamietnutia zo strany firmy.

Trieda Prekladisko, má na starosť správu prekladísk v regiónoch (podľa mapy). Obsahuje zoznam dronov, ktorí sú k dispozícií v danom prekladisku, či zoznam, ktoré cez deň prijalo a večer (pod 21:00) sa odovzdajú vozidlu, ktoré ich prevezie do centrálneho skladu.

Trieda Vozidlo obsahuje všetky potrebné informácie o vozidle, hlavne zoznam prekladísk, ktorými bude prechádzať a vykladať/nakladať zásielky a taktiež zoznam zásielok, ktoré už „vyzdvihol“ z lokálnych prekladsísk.

Trieda Zasielka obsahuje objednávku zákazníka, ale funguje ako pomocná trieda pre zasielku, aby som ju mohol jednoduchšie prepravovať či už vozidlom do centrálneho sklad, alebo k adresátovi pomocou drona.

Trieda UI má „nekonečnú slučku“ a rieši jednotlivú obsluhu triedy firma.

Trieda main spúšťa UI.

## Popis implemetácie údajových štruktúr

V Dronovi používam fron, pretože potrebujem zásielky spracovávať postupne podľa času pridania objednávky, to znamená. Vyberám z prvého miesta, ukladám na posledné.

V triede Firma si ukladám do ArrayListu zoznam všetkých vozidiel, ktoré má firma k dispozícií. Vozidlá mám zoradené podľa dátumu pridania. Na zoznam prekladíska danej firmy používam Array, kvôli presnému počtu prekladísk, ktoré mám je toto najvyhovujúcejší spôsob ukladania. Na objednávky, ktoré firma má som použil LinkedList, z dôvodu veľkého množstva pamäte, ktoré by mohla pri veľkom množstve objednávok táto štruktúra zabrať. V triede si totižto značím každú objednávku aj ak bola zamietnutá, zrušená, či zrealizovaná.

V triede Prekladisko používam dvakrát štruktúru ArrayList to konkrétne na zoznam dronov, ktoré má lokálne prekladisko k dispozicíí a zoznam zásielok, ktoré sa nachádzajú v lokálnom prekladisku a pre ktoré večer príde vozidlo.

Trieda Vozidlo obsahuje zoznam lokálnych prekladísk, ktorými bude ráno a večer prechádzaž. Je tvorený pomocou štruktúry LinkedList. Trieda ďalej obsahuje zoznam zásielok, ktoré sú vo vozidle „naložené“ a smerujú do lokálneho prekladiska alebo centrálneho skladu.

## Zložitosti jednotlivých operácii

1. Pridanie Biofarmára do linkListu, rovno ich triedim čiže hľadám správnu pozíciu => N operácií a volám insert, čo je ďalších N operácii
2. Vypísanie zoznamu Biofarmárov foreach => O(N)
3. Vloženie vozidla do LinkListu O(1).
4. Vypísanie vozidiel podľa dátumu evidencie, pri vložení sa vkladajú s aktuálnym dátumom čiže už sú zoradené. Zložitosť O(N).
5. Pridanie nového zákazníka. Utriedenie podľa abecedy ako biofarmárov, zložitosť O(N).
6. Kontrola či existuje Zákazník ktorý si objednal tovar N operácií, pridanie objednávky ak spĺňa požadované vlastnosti 1 operácia.
7. Prebehne všetky objednávky a spočíta počet tovarov O(N), prepočíta koľko potrebujeme polotovarov 1 operácia pre každý polotovar, dokúpenie surovín => napĺňam haldu všetkými biofarmarmi čiźe pridanie do haldy zložitosť O(logN) a prejdenie foreachom cez zoznam Biofarmarov O(N) , prebehnutie cez haldu kým nenakúpim alebo nedôjdu Biofarmári N- krát pop, ktorý má zložitosť logN, pridanie nakúpeného tovaru do zoznamu nakúpenyTovar O(1). Naskladnenie má zložitosť O(N).
8. Výpis objednávok na zajtra zložitosť O(N).
9. a  10. vyrobenie produktov => odčítanie surovín zo skladov jednotková zložitosť, naplnenie prioritného frontu implemetovaného haldou pripravenými objednávkami O(N), vytvorenie podzoznamov pre vozidlá, podľa typu vozidla O(N), Rozloženie objednávok podľa regiónu do áut zmena nákladov áut O(N) v halde mám objednávky už zoradené podľa regiónu.
10. Pri každom zákazníkovi pozerám zoznam objednávok teda O(N^2).
11. Vloženie cez cyklus N operácií do Haldy O(logN) potom opäť N operácií pri výpise.
12. Objednávky sú v zozname podľa dátumu evidencie takže sa len vypíšu podľa podmienky ktoré potrebujem O(N).
13. Vkladanie do haldy O(logN) a cez všetkých biofarmárov foreach O(N) a v každom sa spočítava počet dní kedy kúpil (priorita) O(N). Výpis O(1).
14. Neimplementované

## Diagram Tried



## Používateľská príručka

Program obsahuje jednoduché konzolové užívateľské rozhranie. Jednotlivé pokyny sú číslované stačí sa riadiť upresňujúcim textom ktorý vyzýva užívateľa k akcií.