

Fakulta riadenia a informatiky - Žilinská univerzita

Semestrálna práca 1

Algoritmy a údajové štruktúry 1

2017/2018

Tomáš Blázy

5ZY02A

1. Zadanie

Cieľom tejto semestrálnej práce bolo vytvorenie aplikácie, ktorá predstavuje informačný systém spoločnosti C&C, prostredníctvom ktorého bude možné evidovať požiadavky zákazníkov, objednávať polotovary, sledovať časový vývoj nákupných a predajných cien, ako aj náklady na údržbu vozového parku, slúžiaceho pre rozvoz produktov k zákazníkom.

Aplikácia je vytvorená v jazyku C++ s manuálnou správou pamäte a s využitím vlastných údajových štruktúr, ktoré som sa snažil implementovať efektívne z hľadiska výpočtovej a pamäťovej zložitosti.

2. Návrh aplikácie

2.1. Rozbor použitých údajových štruktúr

Do aplikácie som sa rozhodol implementovať nasledujúce štruktúry:

- **Array** - pole
- **ArrayList** - zoznam implementovaný poľom
- **ArrayList2** - zoznam implementovaný poľom s premenlivou počiatočnou veľkosťou poľa
- **LinkedList** - jednostranne zreťazený zoznam
- **ImplicitQueue** - implicitný front

Využitia jednotlivých štruktúr:

Array - pre uchovávanie typov polotovarov, ktoré dodávajú jednotliví biofarmári pretože vždy viem aký maximálny počet polotovarov bude môcť každý farmár dodávať.

ArrayList/ArrayList2 - evidovanie denného zisku spoločnosti, kde ako index slúži konkrétny deň. Výhodou takejto implementácie je prakticky okamžitý prístup so zložitosťou $O(1)$. Pri načítavaní zo súboru je zase výhodné použitie parametrického konštruktora štruktúry ArrayList2, v ktorom sa určí počiatočná veľkosť poľa a nie je potrebná viacnásobná realokácia pamäte.

LinkedList – použitý všade tam, kde je potrebné rýchlo pridávať alebo odoberať prvky a nie je potrebný rýchly prístup ku konkrétnemu prvku. Napr. Pri vytváraní utriedených zoznamov biofarmárov, vozidiel, zákazníkov, zásob atď.

ImplicitQueue – výpočet priemernej nákupnej ceny za posledných 30 dní v triede UdajePolotovarov. Použitím implicitného frontu sa táto operácia veľmi zjednodušuje a zaisťuje zložitost' operácie rádovo $O(1)$ aj pri veľkom počte cien.

2.2. Zložitosti jednotlivých operácií

1. Operácia vloženie biofarmára má zložitost' $O(n)$, kde „n“ je počet biofarmárov v zozname, pretože musíme nájsť miesto, na ktoré sa má v zozname vložiť. Pridanie každého jedného polotovaru, ktorý farmár dodáva má zložitost' $O(n + m)$, kde „n“ je počet farmárov a „m“ je počet všetkých možných polotovarov, ktoré farmár môže dodávať. Polotovarov farmára sa vkladá aj do zoznamu usporiadaného podľa priemerných nákupných cien, ale keďže nový polotovarov má nulovú priemernú nákupnú cenu, vloží sa na začiatok, t.j. $O(1)$.

Výsledná zložitost': $O(2n + m)$

2. Výpis biofarmárov, ktorí dodávajú zvolený polotovarov podľa abecedy, musí prejsť celý zoznam všetkých farmárov (n) a pre každého farmára všetky jeho polotovary (m). Keďže sú zoradení podľa abecedy, nemusí sa nič triediť.

Výsledná zložitost': $O(n + m)$

3. Operácia vloženie vozidla má zložitosť $O(n)$, kde „n“ je počet vozidiel v zozname, pretože musíme nájsť miesto, na ktoré sa má v zozname vložiť.

Výsledná zložitosť: $O(n)$

4. Výpis vozidiel uložených v usporiadanom zozname podľa dátumu zaradenia do evidencie.

Výsledná zložitosť: $O(n)$

5. Operácia registrácia nového zákazníka má zložitosť $O(n)$, kde „n“ je počet zákazníkov v zozname, pretože musíme nájsť miesto, na ktoré sa má v zozname vložiť.

Výsledná zložitosť: $O(n)$

6. Operácia zaevidovanie novej objednávky od zákazníka. Najprv sa musí nájsť zákazník podľa zadaného názvu – zložitosť $O(n)$, kde „n“ je počet zákazníkov. Zistenie celkovej kapacity vozidiel na daný deň doručenia – zložitosť $O(m)$, kde „m“ je počet vozidiel. Nájdienie správneho miesta na vloženie objednávky – $O(o)$, kde „o“ je počet objednávok.

Výsledná zložitosť: $O(n + m + o)$

7. Kontrola požiadaviek na najbližších 7 dní. Pre každý možný vyrábaný polotovár (n) a tovar (m) sa zistí množstvo tohto tovaru, ktoré treba vyrobiť v nadchádzajúcom období, prejdením všetkých objednávok (o). Ak množstvo polotovaru na sklade je menšie ako požadované množstvo potom prejde všetkých farmárov (p) podľa priemerných cien a pokúša sa objednať zásoby. Pri každom objednaní polotovaru od biofarmára zaeviduje novú objednávku polotovaru do zoznamu objednávok polotovarov (q). Nakoniec pre každého farmára, od ktorého sa nakúpilo aktualizuje zoznam priemerných nákupných cien polotovarov biofarmárov. Aktualizácia prebieha tak, že najprv sa nájde starý záznam o cene v zozname priemerných cien (r) a potom sa doňho vloží nový záznam na správne miesto.

Výsledná zložitosť: $O((n * m * o) + (n * p * q) + (n * p * 2 * r))$

8. Výpis objednávok, ktoré majú dátum zrealizovania nasledujúci deň zo zoznamu objednávok.

Výsledná zložitosť: $O(n)$

9. Naplnenie vozidiel. Vytvorenie zoznamu objednávok, ktoré majú byť zrealizované v daný deň (n) zo zoznamu všetkých objednávok (m) a jeho usporiadanie podľa regiónov, v ktorých sa budú doručovať, od najväčšieho čísla regiónu. Pre všetky možné typy vyrábaných tovarov (o), vytvorí zoznam napĺňaných vozidiel (p) rozvážajúcich daný typ tovaru. Pre všetky zostávajúce objednávky prechádza napĺňané vozidlá a postupne ich plní. Keď sú všetky objednávky vo vozidlách, tak prejde všetky vozidlá a aktualizuje prevádzkové náklady.

Výsledná zložitosť: $\sim O((n * m) + (o * p) + (o * (p * 2 * n)))$

10. Zrealizovanie dnešných objednávok. Prejdenie zoznamu všetkých objednávok (n) a vloženie všetkých zrealizovaných objednávok z dnešného dňa (m) do zoznamu zrealizovaných objednávok (o) na správne miesto.

Výsledná zložitosť: $O(n + (m * o))$

11. Výpis zákazníkov z daného regiónu. Vytvorenie zoznamu zákazníkov z daného regiónu (n) zo zoznamu všetkých zákazníkov (m). Pre každého zákazníka, prechádzanie zoznamu všetkých objednávok (o).

Výsledná zložitosť: $O((n * m) + (n * o))$

12. Výpis zrealizovaných objednávok zo zoznamu zrealizovaných objednávok (n)

Výsledná zložitosť: $O(n)$

13. Výpis nezrealizovaných objednávok zo zoznamu všetkých objednávok (n)

Výsledná zložitosť: $O(n)$

14. Výpis farmára od ktorého spoločnosť nakúpila najväčšie množstvo daného polotovaru za posledných 30 dní. Pre každého farmára (n) prechádzanie všetkých objednávok (m).

Výsledná zložitosť: $O(n * m)$

15. Výpis celkových ziskov firmy za dané obdobie. Pre každý deň z daného obdobia (n) prechádzanie ArrayListu ziskov.

Výsledná zložitosť: $O(n)$

2.3. Diagramy tried

Diagram použitých štruktúr:

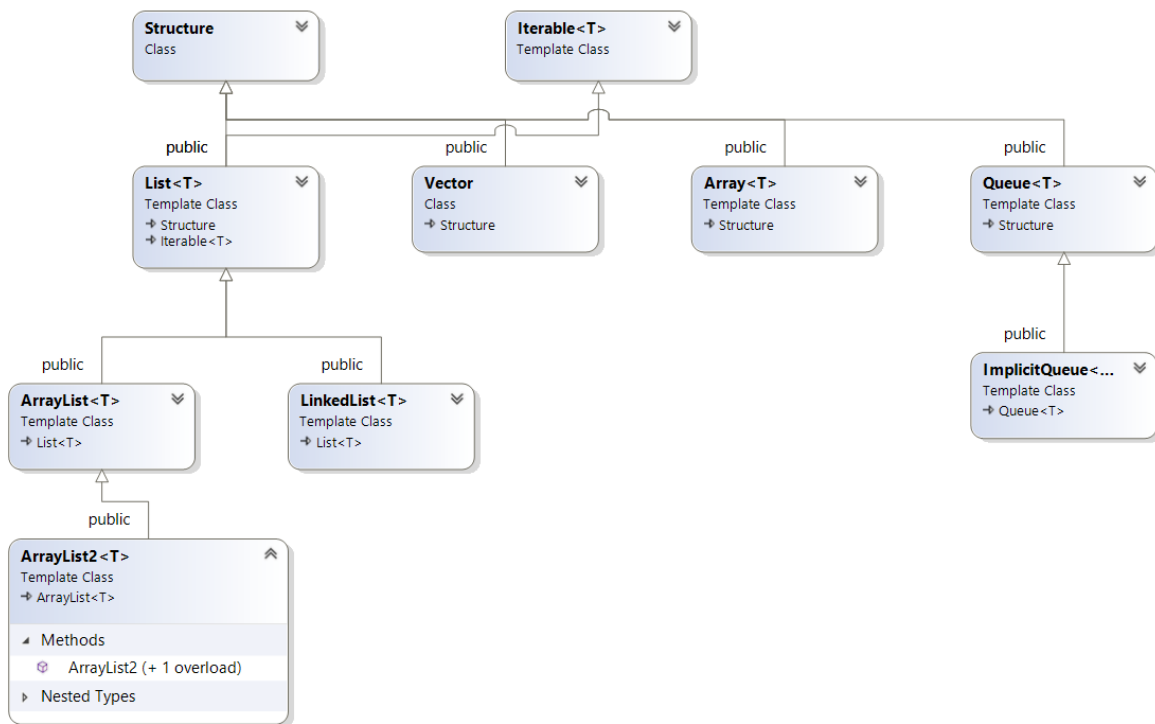
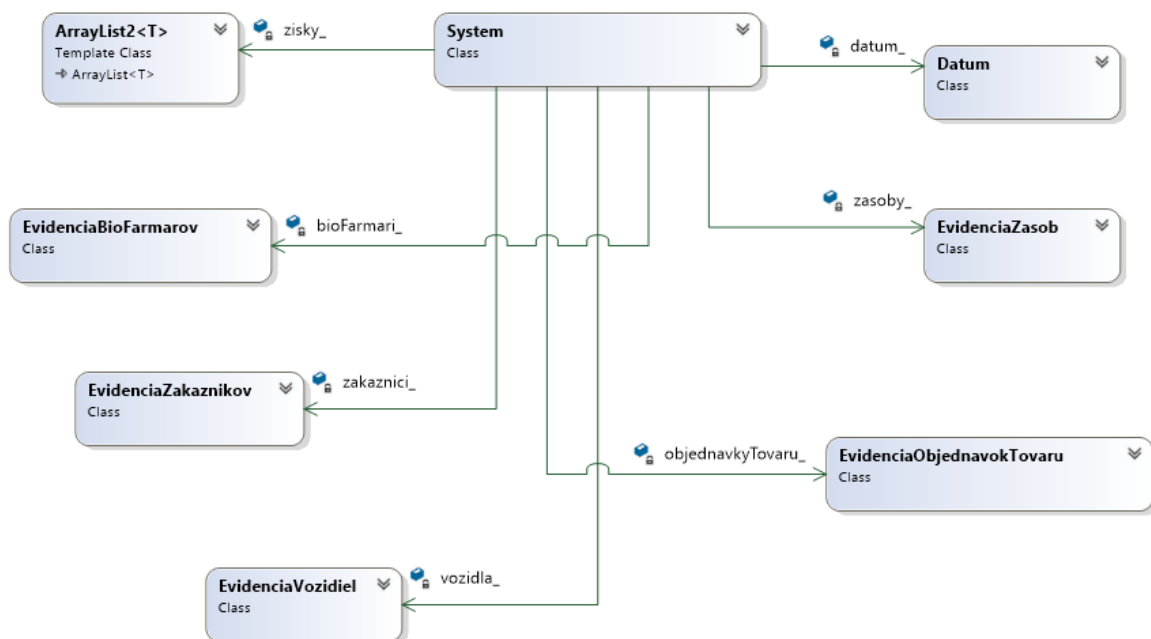


Diagram hlavnej triedy System a všetkých jej súčastí



3. Používateľská príručka

Ovládanie je zabezpečené pomocou konzoly. Po spustení aplikácie si môže užívateľ vybrať či chce vytvoriť nový informačný systém alebo či chce nejaký načítať zo súboru.

Potom už môže začať systém používať pomocou jednoduchého menu.

Obsah

1.	Zadanie	2
2.	Návrh aplikácie.....	3
2.1.	Rozbor použitých údajových štruktúr	3
2.2.	Zložitosť jednotlivých operácií	3
2.3.	Diagramy tried.....	6
3.	Používateľská príručka	7