**Міністерство Освіти І НАУКИ України**

**Національний університет "Львівська політехніка"**

**КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

**КУРСОВА РОБОТА**

**з дисципліни:** *“Об’єктно-орієнтоване програмування ”*

**На тему: *“ Військкомат ”***

Студента групи ПЗ-25

спеціальності 6.121

“Інженерія програмного забезпечення”

Воін В.О.

Керівник: доцент кафедри ПЗ

к.ф.-м.н., доцент Коротєєва Т. О

Національна шкала\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількіть балів\_\_\_Оцінка ECTS\_\_\_

Члени комісії \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Львів 2023

**Завдання**

на курсову роботу з дисципліни “Об’єктно-орієнтоване програмування”

студента групи ПЗ-25 *Воіна В.О.*

***Тема: “******Військкомат******”***

**Завдання:** (варіант 4)

Створити клас “Військкомат”, який міститиме такі поля як:

* №;
* Військкомат;
* Адреса;
* Витрати на працівника ;
* Кількість працівників;
* Кількість приписаних;

Для класу створити:

1) конструктор за замовчуванням;

2) конструктор з параметрами;

3) конструктор копій;

Реалізувати:

1. Алгоритмом злиття відсортувати записи за кількістю працівників в військкоматі;
2. Визначити середні витрати по всіх військкоматах;
3. Вивести військкомати, в яких працівників менше 50% від кількості приписаних
4. Визначити військкомати, в яких витрати на працівника більші за середні по всіх військкоматах;
5. Вивести 4 військкомати, в яких кількість приписаних більша середньої, проте витрати на працівника менші за середні

**Зміст завдання та календарний план його виконання**

|  |  |  |
| --- | --- | --- |
| **№ з/п** | **Зміст завдання** | **Дата** |
| 1 | Здійснити аналiтичний огляд лiтератури за заданою темою та обґрунтувати вибір інструментальних засобів реалізації. | 02.11 |
| 2 | Побудова UML діаграм | 03.11 |
| 3 | Розробка алгоритмів реалізації | 05.11 |
| 4 | Реалізація завдання (кодування) | 10.11 |
| 5 | Формування інструкції користувача | 13.11 |
| 6 | Оформлення звіту до курсової роботи згідно з вимогами Міжнародних стандартів, дотримуючись такої структури:   * зміст; * алгоритм розв‘язку задачі у покроковому представленні; * діаграми UML [клас](http://vns.lpnu.ua/mod/glossary/showentry.php?eid=54317&displayformat=dictionary)ів, прецедентів, послідовності виконання; * код розробленої програми з коментарями; * протокол роботи програми для кожного пункту завдання * інструкція користувача та системні вимоги; * опис виняткових ситуацій; * структура файлу вхідних даних; * висновки; * список використаних джерел. | 15.11 |

Завдання прийнято до виконання: Воін В.О.

Підпис: 

Керівник роботи: Коротєєва Т. О.

Дата видачі завдання: 07.09.2023

**Зміст**

[1. Алгоритм розв’язку задачі у покроковому представленні 4](file:///C:\Users\voinv\Downloads\Telegram%20Desktop\Yaroshovych.PZ25.docx#_Toc118244368)

[2. Діаграми UML класів, прецедентів, послідовності виконання 8](file:///C:\Users\voinv\Downloads\Telegram%20Desktop\Yaroshovych.PZ25.docx#_Toc118244369)

[3. Код розробленої програми з коментарями 12](file:///C:\Users\voinv\Downloads\Telegram%20Desktop\Yaroshovych.PZ25.docx#_Toc118244370)

[4. Протокол роботи програми для кожного пункту завдання 37](file:///C:\Users\voinv\Downloads\Telegram%20Desktop\Yaroshovych.PZ25.docx#_Toc118244371)

[5. Інструкція користувача та системні вимоги 55](file:///C:\Users\voinv\Downloads\Telegram%20Desktop\Yaroshovych.PZ25.docx#_Toc118244372)

[6. Опис виняткових ситуацій 64](file:///C:\Users\voinv\Downloads\Telegram%20Desktop\Yaroshovych.PZ25.docx#_Toc118244373)

[7. Структура файлів вхідних даних 67](file:///C:\Users\voinv\Downloads\Telegram%20Desktop\Yaroshovych.PZ25.docx#_Toc118244374)

[Висновки 68](file:///C:\Users\voinv\Downloads\Telegram%20Desktop\Yaroshovych.PZ25.docx#_Toc118244375)

[Список використаних джерел 69](file:///C:\Users\voinv\Downloads\Telegram%20Desktop\Yaroshovych.PZ25.docx#_Toc118244376)

# **Алгоритми розв’язку задачі у покроковому представленні**

Програма реалізовує наступні алгоритми:

**Алгоритм зчитування з файлу (Read from File, RF)**

**RF1:** Отримання посилання на HTML-елемент вибору файлу.

**RF2:** Додавання слухача подій.

**RF3:** Отримання вибраних файлів.

**RF4:** Перевірка, чи вибрано хоча б один файл.

**RF5:** Перевірка типу вибраного файлу (перевірка на текстовий файл).

**RF6:** Створення об'єкта *FileReader***.**

**RF7:** Додавання слухача події завантаження файлу.

**RF8:** Отримання вмісту файлу як тексту.

**RF9:** Розбивання текстового вмісту на рядки (з урахуванням різних символів нового рядка).

**RF10:** Розбивання кожного рядка на поля за допомогою коми (CSV формат).

**RF11:** Перевірка типу вибраного файлу (перевірка на текстовий файл).

**RF12:** Зчитування тексту файлу як текст із вказанням файлу для *FileReader*.

**RF13:** Обробка випадку, коли не вибрано жодного файлу.

**RF14:** Обробка помилки (якщо виникає).

**RF15:** Завершення алгоритму.

**Алгоритм додавання військкомату (Adding Commissariat, AC)**

**AC1:** Отримання кнопок сортування.

**AC2:** Додавання обробника подій для кнопки закриття модального вікна.

**AC3:** Отримання елементів форми та додавання обробника події для її відправки.

**AC4:** Отримання даних форми за допомогою *FormData*.

**AC5:** Створення порожнього масиву для нового військкомату.

**AC6:** Ітерація по полях форми та перевірка значень.

**AC7:** Перевірка коректності введених даних.

**AC8:** Обробка винятків та виведення повідомлення про помилку.

**AC9:** Припинення виконання методу у випадку помилки.

**AC10:** Додавання обробника подій для закриття модального вікна при кліку за його межами.

**AC11:** Очищення форми після успішного додавання.

**AC12**: Завершення алгоритму додавання військкомату**.**

**Алгоритм видалення військкомату (Deleting Commissariat, DC)**

**DC1:** Отримання посилання на елементи DOM**.**

**DC2:** Фільтруємо дані, залишаючи тільки ті елементи, номер яких не дорівнює переданому номеру**.**

**DC3:** Оновлення даних**.**

**DC4:** Збереження в локальне сховище**.**

**DC5:** Оновлення відображення.

**DC6:** Завершення алгоритму видалення.

**Алгоритм сортування військкоматів за кількістю працівників у військкоматі (Commissariat Sort, CS)**

**CS1:** Обробка кліку по кнопці.

**CS2:** Алгоритм сортування злиттям (Merge Sort).

**CS2.1**: Визначені дві функції: *merge* та *mergeSort*, які використовують алгоритм сортування злиттям для розділення та злиття масиву.

**CS2.2**: Функція *merge* приймає два відсортованих масиви *left* і *right* та повертає новий відсортований масив, результат їхнього злиття.

**CS2.3**: Функція *mergeSort* рекурсивно розбиває масив на підмасиви, сортує їх за допомогою *merge* і повертає відсортований масив.

**CS3:** Застосування сортування до даних.

**CS4:** Оновлення даних та відображення.

**CS5:** Завершення алгоритму сортування військкоматів**.**

**Алгоритм визначення середніх витрат по військкоматах (Average Expenses, AE)**

**AE1:** Обробка кліку по кнопці.

**AE2:** Обчислення суми та кількості військкоматів.

**AE3:** Прохід по рядках таблиці та сумування витрат.

**AE4:** Обчислення середніх витрат та виведення результату.

**AE5:** Виведення результату**.**

**AE5:** Завершення алгоритму визначення середніх витрат.

**Алгоритм виводу військкоматів в яких працівників менше 50% кількості приписаних (Employee Less Than Half Registered, ELTHR)**

**ELTHR1:** Обробка кліку по кнопці.

**ELTHR2:** Фільтрація рядків за умовою "Працівників менше 50% кількості приписаних"

**ELTHR3:** Оновлення даних.

**ELTHR4:** Виклик методу *renderList* для відображення оновленого списку.

**ELTHR4:** Завершення алгоритму виводу військкоматів в яких менше 50% кількості приписаних.

**Алгоритм виводу військкоматів в яких витрати на працівників більші за середні по військкоматах (Commissariats With Higher Expenses, CWHE)**

**CWHE1:** Обробка кліку по кнопці.

**CWHE2:** Обчислення середніх витрат по всіх військкоматах.

**CWHE2.1:** Цикл проходиться по всіх рядках таблиці і додає витрати до загальної суми.

**CWHE3:** Фільтрація військкоматів з витратами більше за середні.

**CWHE4:** Оновлення даних.

**CWHE5:** Виклик методу *renderList* для відображення оновленого списку.

**CWHE6:** Завершення алгоритму виводу військкоматів в яких витрати на працівників більші за середні по військкоматах.

**Алгоритм виводу 4 військкоматів з вищою кількістю приписаних, але нижчими витратами на працівника порівняно з середніми (High Personnel Low Cost, HPLC)**

**HPLC1:** Обробка кліку по кнопці.

**HPLC2:** Обчислення середніх значень кількості приписаних та витрат на військкомат**.**

**HPLC2.1:** Цикл проходиться по всіх рядках таблиці і додає значення кількості приписаних та витрат до відповідних сум.

**HPLC3:** Обробка середніх значень та фільтрація військкоматів.

**HPLC3.1:** Фільтрація військкоматів за умовами: кількість приписаних більше середньої, а витрати менше середньої.

**HPLC3.2:** Використання методу*slice(0, 4),*щоб взяти перші чотири пасуючі військкомати.

**HPLC4:** Оновлення даних.

**HPLC5:** Виклик методу *renderList* для відображення оновленого списку.

**HPLC6:** Завершення алгоритму виводу 4 військкоматів з вищою кількістю приписаних, але нижчими витратами на працівника порівняно з середніми.

**Алгоритм виведення інформації у файл (Info to File, IF)**

**IF1:** Обробка кліку по кнопці.

**IF2:** Формування тексту для збереження**.**

**IF2.1:** Створюємо текст для збереження, перетворюючи кожен рядок даних у рядок, де значення розділені комою, а рядки - переносами рядків.

**IF3:** Створення *Blob*об'єкта.

**IF4:** Створення посилання для завантаження.

**IF5:** Додавання посилання до DOM та імітація кліку.

**IF6:** Видалення посилання після завантаження**.**

**IF7:** Завершення алгоритму виведення інформації у файл.

**Алгоритм відображення даних в HTML – таблиці (Render List, RL)**

**RL1:** Отримання елементів DOM.

**RL2:** Визначення функції для видалення елемента.

**RL2.1:** Функція *deleteItem* призначена для видалення рядка даних за номером. Після видалення оновлюється внутрішній стан, дані зберігаються в локальному сховищі, і знову викликається *renderList* для оновлення відображення.

**RL3:** Очищення тіла таблиці.

**RL3.1:** Очищення вмісту тіла таблиці перед відображенням нових даних.

**RL4:** Створення рядка таблиці та заповнення комірок даними.

**RL5:** Додавання рядка таблиці до тіла таблиці.

**RL6:** Завершення алгоритму відображення даних.

# **Діаграми UML класів, прецедентів, послідовності виконання**

UML-діаграма класів:

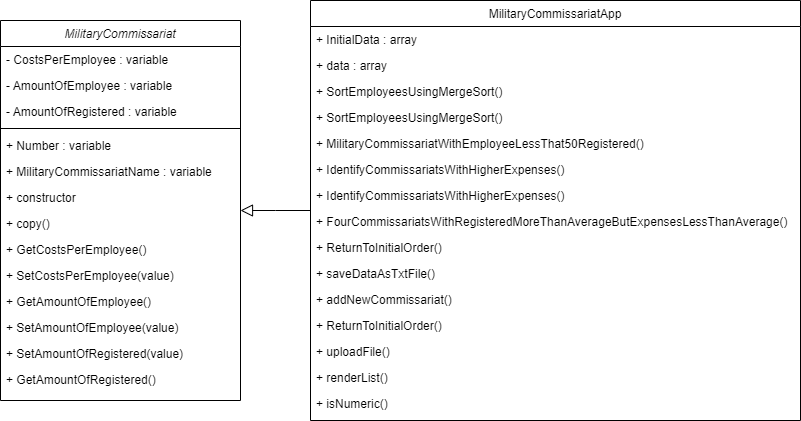


Рис.1 Діаграма класів

UML-діаграма прецедентів: