

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА
ШЕВЧЕНКА**

ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ

ЗВІТ

Додатковому завданню №4

Студентки групи ІПС-23

Матвійчук Анастасії Миколаївни

Київ 2023

Мета: Збір, аналіз та прогнозування інформації щодо часу виконання окремих завдань з програмування.

Було реалізовано: У рамках даної лабораторної роботи було обрано проаналізувати час виконання певних завдань навчального проєкту та 2 лабораторної «Visualization of algorithms» (<https://github.com/yuliapetryk/VisualAlg.git>). Це було здійснено в декілька визначених етапів:

На першому етапі, плануванні та прогнозуванні, було визначено кроки, необхідні для вирішення задачі, та оцінено час, необхідний для виконання кожного з цих кроків. Це допомогло отримати загальне уявлення про обсяг роботи та розподілити час ефективно. На цьому етапі було приблизно оцінено, скільки часу піде на виконання кожного з кроків:

1. Розуміння вимог: Детальне ознайомлення із завданням проєкту, вибір теми, загальних моментів реалізації, обговорення деталей розробки з іншими учасниками команди.
– 1 тиждень
2. Проектування архітектури: Визначення архітектури проєкту, включаючи класи, їх взаємозв'язки та інтерфейси. У контексті обраного мною підваріанту, було розглянуто різноманітні алгоритми пошуку та визначення найбільш актуальних та цікавих для реалізації.
- 4 дні
3. Ознайомлення з патернами: Вивчення принципів роботи патернів, вибір необхідних для реалізації в проєкті:
- 3 дні
4. Реалізація алгоритмів пошуку: Реалізація алгоритмів з використанням відповідних класів та інтерфейсів. Визначення вхідних та вихідних параметрів, а також логіку пошуку патернів.
- 4 дні
5. Візуалізація роботи створених алгоритмів з використанням засобів QML: Створено графічний інтерфейс, який демонструє кроки алгоритму пошуку і результати його роботи.
- 2 тижні
6. Вимірювання часу виконання алгоритмів: Обрахунок часової складності для кожного алгоритму.
- 2 години
7. Створення Unit-tests: Тести для перевірки функціональності програми, зокрема усіх алгоритмів.
- 5 годин

8. Документація: Реалізація документації програмної реалізації з використанням Doxygen
 - 1 година

Під час другого етапу, фіксації часу виконання, було відслідковано реальний час, витрачений на кожне завдання та підзавдання. Це було зроблено більшою мірою вручну, за допомогою текстових файлів та таблиць, а також з орієнтуванням на коміти в репозиторії. Фіксація часу виконання допомогла отримати об'єктивні дані про продуктивність та ефективність роботи. Отже, підрахунок витраченого часу на кожен етап:

1. Розуміння вимог: детальне ознайомлення із завданням проєкту, вибір теми, загальних моментів реалізації, обговорення деталей розробки з іншими учасниками команди.
 - 3 тижні
2. Проектування архітектури: визначення архітектури проєкту, включаючи класи, їх взаємозв'язки та інтерфейси. У контексті обраного мною підваріанту, було розглянуто різноманітні алгоритми пошуку та визначення найбільш актуальних та цікавих для реалізації.
 - 3 дні
3. Реалізація першого алгоритму пошуку: Реалізовано алгоритм Рабіна-Карпа, використовуючи відповідні класи та інтерфейси. Було визначено вхідні та вихідні параметри, а також логіку пошуку патернів.
 - 2 години
4. Візуалізація роботи алгоритму з використанням засобів QML: Створено графічний інтерфейс, який демонструє кроки алгоритму пошуку і результати його роботи.
 - 5 днів
5. Реалізація інших алгоритмів пошуку із застосуванням патерну Стратегія: Створено програмну реалізацію решти алгоритмів: алгоритм скінченних автоматів, Бойера-Мура, Бойера-Мура-Хорспула, Коллінза, Галлахера-Сайда-Галлахера, Кнута-Морріса-Пратта, Сміта. Перед цим було додано базовий клас PatternSearchStrategy, який визначає загальний інтерфейс для всіх алгоритмів, а також клас PatternSearchContext, що виконував алгоритм відповідно до обраної реалізації.
 - 5 днів
6. Візуалізація створених алгоритмів: Додано демонстрацію роботи решти алгоритмів.
 - 3,5 тижнів
7. Застосування патерну Стан: В контексті алгоритмів PatternSearchStrategy, було створено клас StateBoyerMoore, що визначає який з підвидів

алгоритмів Боєра-Мура (BoyerMoore чи BoyerMooreHorspool) доцільно застосувати і забезпечує виконання визначеного алгоритму.

- 1 година

8. Вимірювання часу виконання алгоритмів: Було реалізовано клас `TimedPatternSearchStrategy` для обрахунку часової складності для кожного алгоритму. Клас був створений відповідно до патерну Декоратор.

- 2,5 години

9. Створення Unit-tests: Тести для перевірки функціональності програми, зокрема усіх алгоритмів.

- 4 години

10. Документація: Реалізація документації програмної реалізації з використанням `Doxugen`

- 1,5 години

На третьому етапі, аналізу, ми порівнюємо зібрані дані про час виконання завдань з прогнозованим часом. Це дозволяє нам оцінити, наскільки точні були наші прогнози, і виявити потенційні проблеми або відхилення від плану. Аналіз даних допомагає нам зрозуміти, як можна покращити процес планування та виконання завдань у майбутньому.

Загалом, витрачений час на кожен етап проекту зазвичай перевищував оцінку, яку було заплановано. Особливо значний відхил від плану був помітний на етапі розуміння вимог, де замість запланованого 1 тижня було витрачено 3 тижні. Це зумовлено тим, що всі учасники команди виділили багато часу для детального ознайомлення із запропонованими варіантами завдання – ми детально розібрали кожний із них, щоб зрозуміти, який був би найбільш цікавий та зрозумілий у реалізації конкретно для нас. Візуалізація роботи алгоритмів та реалізація інших алгоритмів пошуку зайняли більше часу, ніж було спочатку передбачено. Це часто було пов'язано з нюансами, що виникали у роботі з QML. Створення unit-тестів і документації зайняло приблизно передбачений час. Це може вказувати на те, що ці етапи розробки є достатньо планомірними і можуть бути більш передбачуваними.

Висновок: В цілому, виконання цього завдання дає нам можливість отримати цінні уроки щодо оцінки часу, управління ресурсами та планування проєктів. Воно допомагає нам покращити наші навички в програмуванні, забезпечує більш реалістичні прогнози та допомагає забезпечити успішне завершення проєктів вчасно та ефективно.