Основи ООП.  
Лабораторна 1.

Звіт.

Виконав студент ІПС-21  
Ольховатий Ігор

Завдання 1а

**Design/Implementation Modeling**

1. У якості існуючого проєкту було обрано роботу AlgorithmsGroupProject з минулого семестру: <https://github.com/RyanGoslenko/AlgorithmsGroupProject>
2. Реалізовано unit tests використовуючи бібліотеку doctest.
3. Побудовано UML діаграми: **Use Case, Class, Activity**. Інші з існуючих типів діаграм було вирішено не реалізовувати через недоцільність. За умовою завдання зроблено версію діаграм після рефакторингу коду, але оскільки рефакторинг вплинув лише на діаграму **Class**, то її нова версія позначена як **Class(2.0),** а інші діаграми стосуються обох версій програми: до та після рефакторингу. Також реалізовано **глоссарій** проєктуу вигляді однойменного розділу у README.md файлі проєкту.
4. Було запропоновано наступні зміни:

* Complex<T> getRandomNumber

Тепер реалізація має лише одне визначення з bool параметрами real (для типу розподілу) і complex (можливість створити випадкове число як з реальною, так і з уявною частиною або лише з дійсною частиною).

* ComplexMatrix.h

Позбулися приватного методу nextPowerOf2, оскільки він не має нічого спільного з самим класом ComplexMatrix.

Видалили визначення getRandom для квадратних матриць, щоб зберегти простоту.

Створено загальний метод getInverse, який має параметр "algorithm", який визначає алгоритм, який використовуватиметься для пошуку оберненої матриці. getInverseGaussJordan і getInverseLU тепер є приватними членами ComplexMatrix. Тому всі використання цих методів було змінено.

* LinearRegression.h

Створено клас LinearRegression, який реалізує попередньо реалізовану функцію BuildLinearRegression як конструктор.

1. Усі перелічені зміни реалізовано.
2. Нова версія програми повністю зберігає попередню функціональність і коректну поведінку, у цьому можна впевнитися за допомогою реалізованих раніше unit tests.
3. Раніше було реалізовано benchmarks для порівняння часу виконання програм. Шляхом тестування виявлено, що запропоновані зміни фактично не змінили час виконання програми.
4. На мою думку, у даній програмі строго дотримано принципів ООП та принципу DRY. Використання патернів, про які варто згадати, не було реалізовано.

\* Було реалізовано створення документації коду виконаних завдань з використанням Doxygen. Згенеровану документацію викладено в репозиторій та налаштовано GitHub Pages ( github.io ) для можливості перегляду документації без клонування репозиторію.

<https://ryangoslenko.github.io/AlgorithmsGroupProject/>

Завдання 1d

У процесі виконання лабораторної роботи №1 було використано інструмент UML-моделювання draw.io

1. Чому було обрано саме цей інструмент, а не аналоги?

Цей інструмент було обрано оскільки він є досить зрозумілим у використанні та безкоштовним. До того ж, є експорт у форматі .svg

1. Наскільки просто та зрозуміло було отримати, встановити, налаштувати та почати використовувати цей інструмент?

Цей інструмент було просто отримати, можна було встановити desktop версію, але мною було використано лише браузерну версію.

1. Наскільки зрозумілою та корисною була документація інструменту?

Документація інструменту була дуже зрозумілою і корисною. При відносно доброму знанні UML та принципів ООП можна зрозуміти документацію без складнощів, а корисною вона виявилась при побудові діаграм (наприклад, який об’єкт використати для ілюстрації класу, зв’язку тощо)

1. Наскільки було зрозуміло, як саме використовувати інструмент, які функції/засоби/вікна/елементи керування використовувати для вирішення поставлених задач?

Використання досить інтуїтивне, тому складнощів не виникло. Використовувалися елементи з бокової панелі, де можна обирати компоненти UML діаграм.

1. Чи всі 14 типів діаграм з UML 2.5 підтримує інструмент? Якщо ні – вкажіть, які саме не підтримуються

Інструмент підтримує усі типи, але шаблони є лише для activity, Sequence, State і Class діаграм.

1. Чи всі можливості, доступні на кожному типі діаграм, підтримує інструмент? Якщо ні – вкажіть, що саме не підтримується, та для якого типу діаграм.

Для використаних діаграм у лабораторній роботі – так.

1. Чи використовували якісь додаткові можливості інструменту, наприклад генерацію коду з діаграм чи відновлення діаграм з коду? Наскільки гарно та правильно працюють ці можливості?

Додаткові можливості не були використані, адже у цьому не було потреби.

1. Наскільки зручно було використовувати інструмент, чи не треба було виконувати багато надлишкових дій?

Єдиний мінус зручності інструмента це складнощі під час додавання стрілок на діаграмі. Довелося витратити багато час, що вони виглядали гарно.

1. Наскільки зрозумілою була поведінка інструменту в різних ситуаціях? Чи не виникали ситуації, коли незрозуміло, чому були виконані якісь дії чи як досягли певного стану?

Одного разу виникла ситуація з незрозумілим форматуванням тексту під час вставки з буферу обміну.

1. Чи виникали якісь проблеми з використанням інструменту? Чи вдалось їх вирішити, як саме?

Загалом інструмент працює непогано, проблем які б вплинули на якість результату не виникло.

1. Що хорошого можна сказати про цей інструмент, які були позитивні аспекти використання інструменту?

Гарні шаблони для діаграм, зрозумілий інтерфейс та купа форматів експорту. А ще зручна веб-версія інструменту.

1. Що поганого можна сказати про цей інструмент, які були негативні аспекти використання інструменту?

Як було зазначено раніше, трохи незручно розставляти стрілки на діаграмі.

1. Якби довелось вирішувати аналогічну задачу, але вже враховуючи досвід використання в цій лабораторній роботі, що варто було б робити так само, а що змінити? Можливо, використати інший інструмент, чи використати інші можливості цього інструменту, чи інакше організувати процес розробки діаграм, чи ще щось?

Інструмент я б використовував той самий, але вже використовував більше можливостей і спробував використати більш «екзотичні» типи діаграм.