**Додаткове завдання 4**

**Тищенко Михайло. ІПС-22**

В цьому звіті буде розглянуто переважно інструмент для створення UML-діаграм Visual Paradigm Community Edition. Також буде оглядово вказано про такі інструменти як Lucid Chart, Draw.io та Star UML. Коментарі про відповідні інструменти буде залишено у місцях, де вони мають місце.

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

**Visual Paradigm:** Відповідь: даний інструмент розрахований безпосередньо для створення UML-діаграм. До того ж, Community Edition дозволяє користуватися ним безкоштовно та без жодних обмежень на кількість діаграм чи елементів діаграм.

**Lucid Chart:** цей інструмент є привабливим не лише за рахунок зручного інтерфейсу, а й вбудованих уроків по використанню та самим діаграмам. Але строгі обмеження на кількість діаграм та деякі неточності в загальнодоступних туторіалах відштовхують від його використання. Серед помилок, зокрема, занадто детальні Use Case diagrams.

**Star UML:** рекомендація зі списку найбільш популярних інструментів для UML-діаграм.

**Draw.io:** рекомендація знайомих, які вже працювали з UML-діаграмами.

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

**Visual Paradigm:** Програма мала простий та зручний інсталятор. Перед встановленням потрібно було детально вивчити сайт, перш ніж знайти посилання на Community Edition: <https://www.visual-paradigm.com/download/community.jsp>

**Star UML:** так само може бути встановлена з власного веб-сайту. Установка нескладна, але багато параметрів неможливо змінити (зокрема, опцію про створення ярлика на робочому столі): <https://staruml.io/>

**Draw.io:** загальнодоступний за посиланням, ідентичним назві, інструмент. Окрім браузерної, наявна десктопна версія.

https://app.diagrams.net/

**Lucid Chart:** браузерний інструмент, необхідна авторизація. https://www.lucidchart.com/pages/ru

1. Наскільки зрозумілою та корисною була документація інструмент?

**Visual Paradigm:** Документація програми, опис UML-діаграм є доступними на сайті продукту. Окрім того, наявні форуми, де користувачі мають можливість ставити та відповідати на запитання. Працювати з документацією було зручно та зрозуміло.

**Draw.io:** наявна вкладка Tutorials. В основному інтерфейс є інтуїтивно зрозумілим, тож документацією користувався нечасто.

**Star UML:** досить зрозуміла документація, переважно лише по специфіці даної програми. Описано коротко, лаконічно

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

**Visual Paradigm:** В основному труднощів з використанням програми не виникало. Основні елементи діаграм згруповані у панелі зліва. До того ж, був вбудований провідник, за допомогою якого можна було взаємодіяти з файлами діаграм.

**Draw.io:** простий інтерфейс: робоча область розташовується в центрі, зліва – контейнер елементів діаграм, які можна отримати drag-and-drop, справа – панель для редагування вибраних елементів.

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

**Visual Paradigm:** у наявній версії відсутня Profile діаграма.

**Star UML:** підтримуються всі типи діаграм.

**Draw.io:** наявні елементи багатьох діаграм; легко доповнити відсутні елементи власноруч створеними з базових інструментів. Але через відсутність посилань не буде доцільним створювати таку діаграму як Interaction Overview (чи інші, де можна використати посилання).

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

**Visual Paradigm:** в деяких джерелах вказувався інший тип Timing діаграм, якого немає в даному застосунку. В якості проміжків часу описаної діаграми використовуються многокутники. Загалом більшість можливостей діаграм можливо тим або іншим чином реалізувати.

**Draw.io:** можуть бути труднощі зі створенням Interaction Overview

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

**Visual Paradigm:** наявна генерація з Java та C++ коду та навпаки: генерація коду з діаграм, але доступна лише в Standart версії програми. Ця можливість не була перевірена мною; всі діаграми були побудовані вручну.

**Draw.io**: додаткові можливості відсутні.

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

**Star UML:** при кожному запуску треба було згортати вікно з пропозицією придбати розширену версію. Будь-які зміни (на прикладі class diagram) вносяться складно. Щоб додати атрибут до класу необхідно було зробити багато додаткових «кліків»; важко дублювати елементи.

**Visual Paradigm:** компоненти в редакторі добре упорядковані, з ними легко взаємодіяти. Деякі властивості необхідно було реалізовувати через Open Specification, що трохи ускладнює роботу.

**Draw.io:** застосунок, в якому найлегше робляться зміни на діаграмах, оскільки кожен об’єкт можна вважати просто графічним об’єктом. При цьому незручністю є відсутність поділу на різні діаграми, а всі компоненти скупчені в одному меню.

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?

**Visual Paradigm:** незрозумілі ситуації виникли у таких випадках:

1. Під час копіювання об’єктів по суті створювалося посилання на попередній, а не дублікат. Відповідно будь-які зміни одного з об’єктів стосувалися й іншого.
2. Під час роботи з Interaction Overview не відразу було зрозуміло, як мають додаватися / відображатися вбудовані Sequence діаграми. В якийсь момент здалося, що до одної компоненти прив’язано відразу декілька посилань. Потім було перероблено за документацією.
3. Зазвичай у дроп-лістах наявна опція, яка зараз використовується. Наприклад, якщо маємо зараз увімкнено режим «1», то коли ми бажаємо змінити його на інший, то нашими опціями є «1, 2, 3». У Visual Paradigm же опціями будуть «2, 3», що дещо викликає плутанину. Зокрема, було важко відрізнити (і знайти) такі елементи як Merge Node та Decision Node.
4. Досить важко було знайти можливість додавання вкладених станів.

**Draw.io:** через відносну вільність у діаграмі нерідко виникали моменти, коли лінії, що з’єднують компоненти, вели себе непередбачено або неправильно.

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

Проблем, пов’язаних з некоректною чи дуже непередбачуваною поведінкою інструментів не виникало.

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

**Draw.io:** застосунок не має строгих обмежень на вміст діаграми. Зокрема, дозволяється фарбувати елементи, змінювати шрифти, додавати власні позначення. Це дозволяє дещо розширити можливості UML, наприклад, вказуючи на діаграмах зміни «до» та «після». Інструмент легкий у використанні, розрахований на швидку роботу.

**Visual Paradigm:** наявність багатьох діаграм та чудової документації. Дозволяє зберігати всі діаграми в одному проекті, легко експортує зображення. Впорядкованість та наявність елементів, що пришвидшують роботу. Наприклад, Control Flow можна розбити на два, помістивши всередину Action всього у два кліки.

**Star UML:** наявність всіх видів діаграм, наявність документації.

**Lucid chart:** наявність багатьох діаграм та поділу та інструкцій

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

**Draw.io:** окрім того, що компоненти діаграм розташовані хаотично в одному меню, при чому відразу всіх діаграм UML 2.5, відсутній пошук потрібних компонент. Також відсутній експорт відразу всіх діаграм у графічні файли, потрібно обходити по черзі. Draw.io за замовчуванням генерує багато back-up файлів, які зберігаються в тому ж каталозі, що й діаграма; такий підхід спричиняє засмічення у папці.

**Visual Paradigm:** часом надто важко зробити саме те, що потрібно користувачу. Іноді важко відрізнити елементи, що потрібні на постійній основі від тих, що програма пропонує додатково. Community edition добре захований на сайті.

**Star UML:** відносно важкий інтерфейс, нав’язливі pop-up вікна.

**Lucid chart:** велика кількість обмежень у безкоштовній версії.

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?

Я дуже задоволений, що використав декілька інструментів і мав можливість їх порівняти. Найдоцільнішим вважаю використання Visual Paradigm. Цей інструмент є найбільш адаптованим до роботи з UML-діаграмами, на мою думку. Draw IO найбільше підходить для створення відносно невеликих та швидких діаграм. Star UML та Lucid Chart були особисто для мене незручними у використанні, я не став би їх рекомендувати іншим.