

УДК 004.94+006(043.2)

Ю. Безкоровайна, А. Телька

(Національний авіаційний університет)

ВІЗУАЛЬНЕ ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ РЕДАКТОРОМ ECLIPSE PAPYRUS

UDC 004.94+006(043.2)

Y. Bezkorovaina, A. Telka

(National Aviation University, Ukraine)

VISUAL DESIGN OF SOFTWARE BY ECLIPSE PAPYRUS EDITOR

Одним із етапів розробки програмне забезпечення (ПЗ) є проектування. Проектування – це один із етапів життєвого циклу ПЗ, під час якого вимоги до ПЗ аналізуються та створюється опис внутрішньої структури ПЗ. Опис внутрішньої структури ПЗ включає : декомпозицію ПЗ, організацію його компонентів та інтерфейсів між цими компонентами [1]. Тому виникла потреба створення «мови» проектування і була стандартизована мова UML (англ. Unified Modeling Language).

Мова UML – це графічна інтерпретація компонентів та їх зв'язків ПЗ. За допомогою UML створюють діаграми, які описують структуру та стани ПЗ. Наприклад, діаграма класів – структуру та залежності між класами ПЗ; діаграма компонентів – з яких компонентів складається ПЗ та зв'язки між ними; діаграма розгортання – фізичні вузли (пристрої) та компоненти, які на них необхідно встановити для роботи з ПЗ; тощо.

Існують спеціалізовані середовища розробки для проектування, які орієнтовані на створення та опис проектів ПЗ, проте стає популярним вбудовувати плагіни в середовища розробки коду, наприклад, графічний редактор Eclipse Papyrus [2].

Графічний редактор Papyrus має наступні переваги [2]:

- Відкритий код;
- Повністю реалізовує специфікацію OMG (англ. Object Management Group);
- Підтримує всі діаграми UML та генерацію коду;
- Повністю підтримує SysML для розробки систем основаних на моделях;
- Реалізує моделі, які виконуються;
- Гнучкий до внесення змін. Усі функції в Eclipse Papyrus розроблені так, щоб вони були налаштовані та максимально повторно використані. При необхідності, можна конфігурувати для конкретного домену, нотації, або моделі.

Для розробки ПЗ важливим є – яке середовище використовують для розробки. Як видно з переліченого вище в редакторі Papyrus можна розробляти діаграми, які будуть однозначно транслюватися в виконуваний код, наприклад, на діаграмі класів відношення будуть транслюватися так як визначено в специфікаціях OMG та відповідно синтаксису мови програмування. В достатній мірі допомагати швидко та якісно реалізувати ПЗ для різних потреб кінцевих користувачів. Наприклад, діаграми варіантів використання на етапі аналізу вимог, а діаграми класів, компонентів, пакетів, станів на етапах проектування, кодування та супроводження.

Література

1. Software Engineering Book of Knowledge. Software Design. URL: <http://swbookwiki.org/> (дата звернення: 28.11.2019).
2. Eclipse Papyrus. Documentation. URL: <https://www.eclipse.org/papyrus/> (дата звернення: 28.11.2019).