

Міністерство освіти і науки України
Національний авіаційний університет
Факультет кібербезпеки, комп'ютерної та програмної інженерії
Кафедра комп'ютеризованих систем управління

Модульна контрольна робота № 3
з дисципліни «Технології проектування комп'ютерних систем»
Варіант № 3

Виконав:
студент ФККПІ
групи СП-425
Клокун В. Д.
Перевірів:
Масловський Б. Г.

Київ 2020

1. ХІД РОБОТИ

1.1. Принципи управління розробкою ПЗ

Існує декілька різних методологій управління розробкою програмного забезпечення, кожна з яких має власні принципи управління. Розглянемо деякі з цих методологій.

Принципи каскадної методології:

1. Завершувати проектування програми до початку аналізу і написання коду.
2. Вести документацію своєчасно і у повному обсязі.
3. Якщо це можливо, необхідно виконувати роботу двічі.
4. Планувати, контролювати та наглядати за тестуванням.
5. Залучати замовника до роботи.

Каскадна методологія була популярною у минулому, а зараз більш популярною стала методологія Agile (англ. *гнучка*). Вона побудована на таких принципах:

1. Найвищий пріоритет — задоволення потреб замовника за допомогою частих та неперервних поставок цінного програмного забезпечення.
2. Зміни у вимогах характерні навіть на пізніх етапах реалізації проекту.
3. Працюючий продукт варто випускати кожні декілька тижнів, у крайньому випадку — кожні декілька місяців.
4. Найбільш ефективний спосіб передачі інформації — зустріч членів команди розробки програмного забезпечення.
5. Представники бізнесу і команда розробки повинні працювати над проектом разом.
6. Головна міра прогресу проекту — працююче програмне забезпечення.
7. Гнучкі процеси сприяють неперервному розвитку.
8. Постійна увага до технічної досконалості та якісної архітектури сприяють гнучкості.
9. Найкраща архітектура, вимоги та дизайн створюються у самоорганізовуючихся командах.
10. Команди повинні постійно шукати способи стати більш ефективними, налаштовуючи та коригуючи свої дії.

1.2. Визначення та графічне зображення зв'язків та кратності в асоціаціях

1.2.1. Асоціація

Асоціація (рис. 1) показує відношення між двома класами об'єктів, яке дозволяє одному екземпляру об'єкта викликати інший, щоб виконати певну дію від його

імені. В загальному сенсі, причина використання відношення асоціації називається, на кшталт «відправка повідомлення», «виклик методу» або «виклик функції-члена» у контрольованого об'єкта.

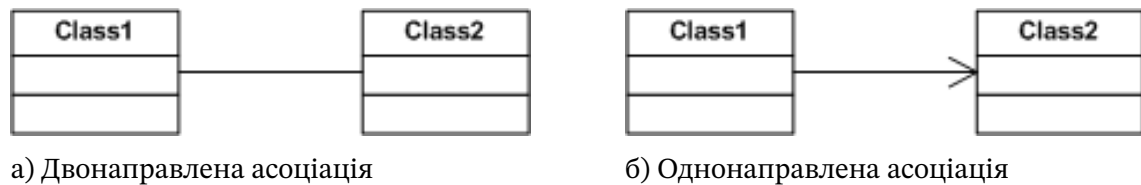


Рис. 1: Графічне зображення відношення асоціації

Відношення асоціації також характеризуються кратністю (рис. 2), тобто кількістю зв'язків між кожним екземпляром класу на початку лінії з екземпляром класу на її кінці. Кратність може бути будь-якою, навіть нескінченною, підмножиною додатних цілих чисел: «1» — рівно 1, «1..*» — 1 або більше, «3..5» — від 3 до 5. Спеціальний символ «*» означає «багато», тобто 0 або більше.

Нотація	Значення	Приклад
0 .. 1	Нуль або один	До робочої станції може бути підключений консольний термінал або ні
1	Рівно 1	У країни є одна столиця
0 .. *	Нуль або більше екземплярів	На одному екрані може бути відкрито багато вікон або жодного
1 .. *	Один або більше	На працюючому комп'ютері може бути встановлена одна операційна система або більше

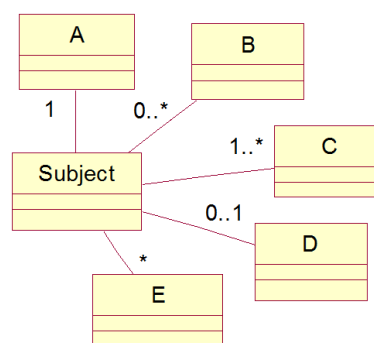


Рис. 2: Графічне позначення кратності асоціацій

1.3. Охарактеризувати СМО, що в нотації Кендалла позначено як M/D/2/3 з дисципліною обслуговування LIFO

Система масового обслуговування, яка в нотації Кендалла позначена як M/D/2/3 з дисципліною обслуговування LIFO має такі характеристики:

1. Перший символ описує процес прибуття до черги. Літера M означає, що система є Марковською (або без пам'яті), тобто час прибуття нових заявок у систему описується розподілом Пуассона:

$$f(k; \lambda) = P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}.$$

2. Другий символ описує розподіл часу на кожну заявку. Літера D показує, що час розподілений вирождено або фіксовано. Наприклад, на обробку кожної заявки йде t часу.
3. Третій символ вказує кількість каналів у системі. В заданій системі 2 канали.
4. Четвертий символ показує кількість місць у черзі або максимальний обсяг заявок, які можуть знаходитись у системі. Дана система може одночасно вмістити в черзі 3 заявки.
5. Дисципліна обслуговування LIFO (last in, first out, останній прибув — перший вибув) означає, що заявки оброблюються у зворотньому порядку: у першу чергу обслуговують ті заявки, які прибули останніми.