



**ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ
ФАКУЛЬТЕТ КОМП'ЮТЕРНО-ІНФОРМАЦІЙНИХ СИСТЕМ І
ПРОГРАМНОЇ ІНЖЕНЕРІЇ (ФІС)
КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ**

1. Загальна інформація про дисципліну:

Назва дисципліни	Архітектура та проектування програмного забезпечення
Викладач	Петрик Михайло Романович
Профайл викладача	https://tntu.edu.ua
Контактний тел.	Комутатор (0352) 51-97-00, внутрішній 1104 (із зовнішніх телефонів через комутатор, тоді в режимі тонального набору набрати внутрішній номер), 380973405705
E-mail:	mykhaylo_petryk@tu.edu.ua
Сторінка дисципліни в A-Tutor	https://dl.tntu.edu.ua , ID:1616
Консультації	Згідно графіку консультацій у п'ятому семестрі

2. Коротка анотація до дисципліни

Навчальна дисципліна «**Архітектура та проектування програмного забезпечення**» належить до обов'язкових дисциплін циклу професійної підготовки освітньої програми «Інформаційні системи та технології» першого (бакалаврського) рівня вищої освіти. Галузь знань 12 «Інформаційні технології», спеціальності 126 «Інформаційні системи та технології». Викладається у 5 семестрі (третій курс) обсягом 4,5 кредитів ECTS. Формою підсумковою контролю є залік.

3. Мета та завдання дисципліни

Метою дисципліни «Архітектура та проектування програмного забезпечення»: сформувати у студентів уявлення та надання майбутнім фахівцям у галузі комп'ютерних наук знань про сучасні вимоги щодо принципів, технологій, підходів, методів та засобів проектування архітектури програмних систем, а також практичних навичок щодо виконання відповідних фаз створення програмних проектів та продуктів.

Завданням дисципліни є:

є надання студентам базових знань та компетенцій щодо сучасних архітектур програмних систем, моделей життєвого циклу програмних систем, методів, засобів та технологій розробки архітектури складних програмних систем; вміння розв'язувати задачі, що виникають на окремих фазах життєвого циклу програмних систем та пов'язані із проектування архітектури програмних систем; володіти сучасними засобами проектування інформаційних систем.

4. Формат дисципліни:

Дисципліна передбачає проведення лекційних, лабораторних занять та консультацій. Для кращого розуміння та засвоєння викладеного матеріалу дисципліна має супровід у вигляді електронного навчального курсу в системі A-Tutor (<https://dl.tntu.edu.ua>). Електронний навчальний курс має лекційний матеріал, лабораторні роботи, питання підсумкового контролю та систему оцінювання.

5. Результати навчання:

За результатами вивчення дисципліни "Архітектура та проектування програмного забезпечення" студент повинен:

знати:

- проблеми, що виникають під час розробки програмних систем;
- поняття архітектури програмних засобів, підходи до її аналізу та побудови;

- основні існуючі архітектури програмних засобів, їх характеристики та особливості, а також застосування архітектур на практиці;
- мету та задачі проектування програмного забезпечення;
- поняття про життєвий цикл програмної системи;
- основні та допоміжні процеси життєвого циклу
- моделі життєвого циклу програмних систем;
- визначення вимог до програмних систем;
- відмінності функціональних та нефункціональних вимог
- методології проектування програмних систем;
- структурне проектування;
- методику розробки модульної архітектури програмних систем;
- об'єктний підхід до проектування архітектури програмних систем; мову моделювання UML.
- патерни проектування;
- рефакторинг архітектури програмних систем

вміти:

- вірно обирати необхідну модель архітектури програмної системи;
- розробляти вимоги до програмного забезпечення;
- будувати необхідні моделі для етапів життєвого циклу програмного забезпечення;
- користуватись методами об'єктно-орієнтованого проектування програмного забезпечення та застосовувати ці методи на практиці;
- використовувати мову моделювання UML для проектування програмного забезпечення;
- застосовувати множину різних шаблонів проектування, середовищ розробки і архітектур в проектуванні різноманітного програмного забезпечення;
- проектувати і реалізовувати програмне забезпечення, використовуючи декілька різних технологій проміжного програмного забезпечення.

здобути навички:

- роботи в CASE-системах, які дозволяють проектування архітектури програмного забезпечення.

Вивчення навчальної дисципліни передбачає формування та розвиток у студентів **загальних компетентностей (КЗ) та спеціальних (фахових) компетентностей (КС)** згідно освітньої програми.

Загальні:

КЗ1. Здатність до абстрактного мислення, аналізу та синтезу.

Спеціальні (фахові):

КС4. Здатність проектувати, розробляти та використовувати засоби реалізації інформаційних систем, технологій та інфокомунікацій (методичні, інформаційні, алгоритмічні, технічні, програмні та інші).

КС13. Здатність проводити обчислювальні експерименти, порівнювати результати експериментальних даних і отриманих рішень..

Вивчення навчальної дисципліни передбачає формування у студента таких **прикладних результатів навчання (ПР)** згідно освітньої програми:

ПР1. Знати лінійну та векторну алгебру, диференціальне та інтегральне числення, теорію функцій багатьох змінних, теорію рядів, диференціальні рівняння для функції однієї та багатьох змінних, операційне числення, теорію ймовірностей та математичну статистику в обсязі, необхідному для розробки та використання інформаційних систем, технологій та інфокомунікацій, сервісів та інфраструктури організації.

6. Обсяг дисципліни

Вид заняття	Загальна кількість годин
лекції	32
лабораторні заняття	32
самостійна робота	71
Всього за дисципліну	135

7. Ознаки дисципліни:

Рік викладання	Семестр	Курс	Спеціальність	Нормативна/вибіркова
2019-2020	5	3	126 «Інформаційні системи та технології»	Обов'язкова

8. Пререквізити

Студенти повинні володіти базовими знаннями з основ програмування, об'єктно-орієнтованого програмування, основ програмної інженерії, аналізу вимог до програмного забезпечення, моделювання та аналізу програмного забезпечення.

9. Технічне й програмне забезпечення /обладнання:

Студент повинен мати рівень впевненого користувача прикладних програм пакету Microsoft visual studio та IBM Rational Architect.

10. Політика дисципліни

Усі процедури навчального процесу під час викладання дисципліни відповідають положенню про академічну доброчесність учасників освітнього процесу та недопущення академічного плагіату в Тернопільському національному технічному університеті імені Івана Пулюя.

11. Схема дисципліни

Тиж./дата/год.	Тема, підтеми	Форма заняття, формат	Матеріали	Література, ресурси в інтернеті	Завдання, год	Засоби діагностики
1	2	3	4	5	6	7
Тиж. 1/ 4 акад. год.	Тема 1. Процес і персонал розробки ПЗ. Задачі проектування ПЗ. Техніка безпеки. Дослідження моделей архітектур програмного забезпечення та базових стилів реалізації програмних систем.	Лекція, F2F Лаб. роб. № 1	Лекційний матеріал Лаб. роб. № 1	Дистанційний курс	Опрацювання лекцій, Лаб. роб. № 1 (3)	Дистанційний курс
Тиж. 2/ 4 акад. год.	Тема 2. Моделі та інструментальні засоби процесу розробки ПЗ. Дослідження моделей архітектур програмного забезпечення та базових стилів реалізації програмних систем.	Лекція, F2F Лаб. роб. № 1	Лекційний матеріал Лаб. роб. № 1	Дистанційний курс	Опрацювання лекцій, Лаб. роб. № 1 (2)	Дистанційний курс
Тиж. 3/ 4 акад. год.	Тема 3. Життєвий цикл уніфікованого процесу. Реалізація ООП проектів із застосування породжуючих шаблонів проектування Фабричний метод (Factory Method) та Абстрактна фабрика (Abstract Factory)	Лекція, F2F Лаб. роб. № 2	Лекційний матеріал Лаб. роб. № 2	Дистанційний курс	Опрацювання лекцій, Лаб. роб. № 2 (2)	Дистанційний курс

1	2	3	4	5	6	7
Тиж. 4/ 4 акад. год.	Тема 4. Архітектурно-центрований процес розробки ПЗ. Реалізація ООП проектів із застосування породжуючих шаблонів проектування Фабричний метод (Factory Method) та Абстрактна фабрика (Abstract Factory)	Лекція, F2F Лаб. роб. № 2	Лекційний матеріал Лаб. роб. № 2	Дистанційний курс	Опрацювання лекцій, Лаб. роб. № 2 (2)	Дистанційний курс
Тиж. 5/ 4 акад. год.	Тема 5. Процес проектування ПЗ, керований варіантами використання. Реалізація ООП проектів із застосування структурного шаблону проектування Пристосуванець (Flyweight)	Лекція, F2F Лаб. роб. № 3	Лекційний матеріал Лаб. роб. № 3	Дистанційний курс	Опрацювання лекцій, Лаб. роб. № 3 (2)	Дистанційний курс
Тиж. 6/ 4 акад. год.	Тема 6. Концептуальні засади, завдання і роль архітектури ПЗ. Реалізація ООП проектів із застосування структурного шаблону проектування Пристосуванець (Flyweight)	Лекція, F2F Лаб. роб. № 3	Лекційний матеріал Лаб. роб. № 3	Дистанційний курс	Опрацювання лекцій, Лаб. роб. № 3 (2)	Дистанційний курс
Тиж. 7/ 4 акад. год.	Тема 7. Описи моделей та проектування варіантів використання на основі аналізу вимог. Реалізація ООП проектів із застосування поведінкового шаблону проектування Спостерігач (Observer).	Лекція, F2F Лаб. роб. № 4	Лекційний матеріал Лаб. роб. № 4	Дистанційний курс	Опрацювання лекцій, Лаб. роб. № 4 (2)	Дистанційний курс
Тиж. 8/ 4 акад. год.	Тема 8. Архітектурні вигляди ПЗ. Реалізація ООП проектів із застосування поведінкового шаблону проектування Міст (Bridge)	Лекція, F2F Лаб. роб. №5	Лекційний матеріал Лаб. роб. №5	Дистанційний курс	Опрацювання лекцій, Лаб. роб. № 5 (2)	Дистанційний курс
Тиж. 9/ 4 акад. год.	Тема 8. Архітектурні вигляди ПЗ. Реалізація ООП проектів із застосування поведінкового шаблону проектування Міст (Bridge)	Лекція, F2F Лаб. роб. №5	Лекційний матеріал Лаб. роб. №5	Дистанційний курс	Опрацювання лекцій, Лаб. роб. № 5 (2)	Дистанційний курс

1	2	3	4	5	6	7
Тиж. 10/ 4 акад. год.	Тема 9. Архітектурні уявлення моделей розгортання і реалізації. Реалізація ООП проектів із застосування структурного шаблону проектування Декоратор (Decorator)	Лекція, F2F Лаб. роб. №6	Лекційний матеріал Лаб. роб. №6	Дистанційний курс	Опрацювання лекцій, Лаб. роб. № 6 (2)	Дистанційний курс
Тиж. 11/ 4 акад. год.	Тема 10. Архітектурні зразки та патерни проектування. Реалізація ООП проектів із застосування структурного шаблону проектування Декоратор (Decorator)	Лекція, F2F Лаб. роб. №6	Лекційний матеріал Лаб. роб. №6	Дистанційний курс	Опрацювання лекцій, Лаб. роб. № 6 (2)	Дистанційний курс
Тиж. 12/ 4 акад. год.	Тема 10. Архітектурні зразки та патерни проектування. Реалізація ООП проектів із застосування поведінкового шаблону проектування Стан (State) або Відвідувач (Visitor)	Лекція, F2F Лаб. роб. №7	Лекційний матеріал Лаб. роб. №7	Дистанційний курс	Опрацювання лекцій, Лаб. роб. № 7 (2)	Дистанційний курс
Тиж. 13/ 4 акад. год.	Тема 11. Патерни проектування структурного типу. Реалізація ООП проектів із застосування поведінкового шаблону проектування Стан (State) або Відвідувач (Visitor)	Лекція, F2F Лаб. роб. №7	Лекційний матеріал Лаб. роб. №7	Дистанційний курс	Опрацювання лекцій, Лаб. роб. № 7 (2)	Дистанційний курс
Тиж. 14/ 4 акад. год.	Тема 11. Патерни проектування структурного типу. Реалізація ООП проектів із застосування структурного шаблону проектування Стратегія (Strategy)	Лекція, F2F Лаб. роб. №8	Лекційний матеріал Лаб. роб. №8	Дистанційний курс	Опрацювання лекцій, Лаб. роб. № 8 (2)	Дистанційний курс
Тиж. 15/ 4 акад. год.	Тема 12. Поведінкові патерни проектування. Реалізація ООП проектів із застосування структурного шаблону проектування Стратегія (Strategy)	Лекція, F2F Лаб. роб. №8	Лекційний матеріал Лаб. роб. №8	Дистанційний курс	Опрацювання лекцій, Лаб. роб. № 8 (2)	Дистанційний курс

Продовження таблиці

1	2	3	4	5	6	7
Тиж. 16/ 4 акад. год.	Тема 12. Поведінкові патерни проектування. Реалізація ООП проектів із застосування структурного шаблону проектування Стратегія (Strategy) Моделі реалізації ПЗ.	Лекція, F2F Лаб. роб. №8	Лекційний матеріал Лаб. роб. №8	Дистанційний курс	Опрацювання лекцій, Лаб. роб. № 8 (2)	Дистанційний курс

12. Система оцінювання та вимоги

Форма підсумкового семестрового контролю – залік.

Підсумкова семестрова оцінка заліку складається з суми балів, отриманих студентом при проведенні проміжних (модульних) контролів рівня засвоєння теоретичних знань (за перший та другий модулі) та отриманих балів за лабораторні роботи.

Модуль 1			Модуль 2			Підсумковий контроль	Разом з дисципліни
Аудиторна та самостійна робота			Аудиторна та самостійна робота				
Теоретичний курс (тестування)	Практична робота		Теоретичний курс (тестування)	Практична робота			
15	20		20	20		25	100
№ лекцій	Вид робіт	Бал	№ лекцій	Вид робіт	Бал	За кожних 3 бали семестрової оцінки студент отримує 1 бал підсумкової семестрової оцінки автоматично	
Вступна	Техніка безпеки		Лекція № 8	Лаб. роб. № 5			
Лекція № 1	Лаб. роб. № 1		Лекція № 9				
Лекція № 2			Лекція № 10	Лаб. роб. № 6			
Лекція № 3	Лаб. роб. № 2		Лекція № 11				
Лекція № 4			Лекція № 12	Лаб. роб. № 7			
Лекція № 5	Лаб. роб. № 3		Лекція № 13				
Лекція № 6			Лекція № 14	Лаб. роб. № 8			
Лекція № 7	Лаб. роб. № 4		Лекція № 15-16				
	Проміжний контроль			Проміжний контроль			

До підсумкового семестрового контролю (складання семестрового заліку) допускаються студенти, які протягом семестру виконали всі види навчальної роботи, успішно пройшли проміжні (модульні) контролі і набрали не менше 45 балів семестрової бальної оцінки та за умови отримання не менше 60% (15) балів за результатами кожного проміжного (модульного) контролю рівня знань.

Підсумкова оцінка записується за 100-бальною шкалою із подальшим переведенням її у шкалу Європейської кредитно-трансферної системи (ECTS) відповідно А, В, С, D, E, F, FX при цьому чотирибальна шкала оцінок (з записом семестрової оцінки «відмінно» - А, «добре» - В,С, «задовільно» - D, E відповідають підсумковому результату «зараховано», «незадовільно» - F, FX відповідає підсумковому результату «незараховано»).

13. Навчально-методичне забезпечення

1. Методичні вказівки для виконання лабораторних робіт з дисципліни «Архітектура та проектування програмного забезпечення» для студентів спеціальності. (ID:1616) «Архітектура та проектування програмного забезпечення» Електронні навчальні курси ТНТУ імені І. Пулюя.

2. Методичні вказівки для самостійної роботи студентів з дисципліни «Архітектура та проектування програмного забезпечення». (ID:1616) «Архітектура та проектування програмного забезпечення» Електронні навчальні курси ТНТУ імені І. Пулюя.

14. Рекомендована література

Базова

1. Arlow J. UML 2 & the Unified Process: Practical ObjectOriented Analysis and Design, 2nd Ed. / Jim Arlow, Ila Neustadt - Addison_Wesley, 2005.
2. Rumbaugh J. The Unified Modeling Language Reference Manual», 2nd Ed. / James Rumbaugh, Ivar Jacobson, Grady Booch - Addison_Wesley, 2005.
3. Jacobson I. Unified Software Development Process /Ivar Jacobson, Grady Booch, James Rumbaugh - Addison_Wesley, 1999.
4. Booch G. Object-Oriented Analysis and Design with Applications, 3 Ed. / Grady Booch - Addison-Wesley, 2007.
5. Booch G. Object Solutions / Grady Booch - Addison_Wesley, 1995.
6. Gamma E. Design Patterns / Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides - Addison_Wesley, 1995.
7. UML 2.0 Object Constraint Language (OCL), www.omg.org. (Об'єктна мова обмежень – текстова мова обмежень і запитів), OMG (Object Management Group). www.uml.org (2004)
8. Cours de Jean-Marie Favre. Paris, IMAG, 2014.
9. Петрик М. Моделювання та аналіз програмного забезпечення / Петрик М., Петрик О. - Тернопіль: Видавництво ТНТУ, 2015. – 210 с.
10. UML 2.0 et UML 2.0 Infrastructure /Benoît Charoux, Aomar Osmani, Yann Thierry-Mieg. Editions Pearson, Education France, 2008.

Допоміжна

1. Alexander I. Writing Better Requirements/ Ian F. Alexander, Richard Stevens, Ian Alexander, Addison_Wesley - 2002, ISBN 0321131630.
2. Ambler S.W. The Unified Process Inception Phase /Scott W. Ambler, Larry L. Constantine - CMP Books, 2000, ISBN 1929629109.
3. Ambler S.W. The Unified Process Elaboration Phase / Scott W. Ambler, Larry L. Constantine, Roger Smith - CMP Books, 2000, ISBN 1929629052.
4. Ambler S.W. The Unified Process Construction Phase / Scott W. Ambler, Larry L. Constantine - CMP Books, 2000, ISBN 192962901X.
5. Arlow J. Enterprise Patterns and MDA: Building Better Software with Archetype Patterns and UML», Jim Arlow, Ila Neustadt, Addison Wesley - 2004, ISBN 032111230X.
6. Booch G. Object Solutions/ Grady Booch, Addison_Wesley - 1995, ISBN 0805305947.
7. Booch G. The Unified Modeling Language User Guide/ Grady Booch, Ivar Jacobson, James Rumbaugh, Addison_Wesley - 1998, ISBN 0201571684.
8. Chomsky N. Syntactic Structures / Noam Chomsky, Peter Lang Publishing, 1975, ISBN 3110154129.
9. Frankel D.S. Model Driven Architecture: Applying MDA to Enterprise Computing/ David S. Frankel, John Wiley & Sons, 2003, ISBN 0471319201.
10. Gamma E. Design Patterns / Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Addison_Wesley, 1995, ISBN 0201633612.
11. Harel D. Modeling Reactive Systems with Statecharts: The Statemate Approach/ David Harel, Michal Politi, McGraw Hill, 1998, ISBN 0070262055.
12. Jacobson I. Unified Software Development Process / Ivar Jacobson, Grady Booch, James Rumbaugh - Addison_Wesley, 1999, ISBN 0201571692.
13. Kay M. XSLT Programmer's Reference. 2nd Edition / Michael Kay - Wrox Press Inc., 2001, ISBN 1861005067.
14. Kleppe A. MDA Explained: The Model Driven Architecture – Practice and Promise /Anneke Kleppe, Jos Warmer, Wim Bast - Addison_Wesley, 2001, ISBN 032119442X.
15. Kroll P. The Rational Unified Process Made Easy: A Practitioner's Guide to Rational Unified Process /Per Kroll, Philippe Kruchten, Grady Booch - Addison_Wesley, 2003, ISBN 0321166094.
16. Kruchten P. The Rational Unified Process, An Introduction/ Philippe Kruchten - Addison_Wesley, 2000, ISBN 0201707101.

17. Kruchten P. The 4+1 View of Architecture / Philippe Kruchten, IEEE Software, 12(6) Nov. 1995, pp. 45–50.
18. Leffingwell D. Managing Software Requirements: A Use Case Approach, 2nd Edition / Dean Leffingwell, Don Widrig - Addison_Wesley, 2003, ISBN 032112247X.
19. Meyer B. Object Oriented Software Construction / Bertrand Meyer, Prentice Hall, 1997, ISBN 0136291554.
20. Mellor S. Agile MDA / Stephen J Mellor, MDA Journal», June 2004. [Електронний ресурс] – Режим доступу: www.bptrends.com.
21. Unified Modeling Language: OCL, version 2.0 . [Електронний ресурс] – Режим доступу: www.omg.org.
22. Pitts N. XML Black Book, 2nd Edition // Natalie Pitts, Coriolis Group, 2000, ISBN 1576107833.
23. Rumbaugh J. The Unified Modeling Language Reference Manual, 2nd Edition/ James Rumbaugh, Ivar Jacobson, Grady Booch - Addison Wesley, 2004, ISBN 0321245628.
24. The CHAOS Report (1994), The Standish Group. . [Електронний ресурс] – Режим доступу: www.standishgroup.com/sample_research/chaos_1994_1.php.
25. Unified Modeling Language: Superstructure, version 2.0. . [Електронний ресурс] – Режим доступу: www.omg.org.
26. Warmer J. The Object Constraint Language: Getting Your Models Ready for MDA / Jos Warmer, Anneke Kleppe - Addison_Wesley, 2003, ISBN 0321179366.
27. Гамма Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования/ Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес – Пер. с англ. – СПб.: Питер, 2007.
28. Майкл Кэй «XSLT. Справочник программиста», 2_е издание. – Пер. с англ. – СПб.: Символ_Плюс, 2002.
29. Фаулер М. Рефакторинг: улучшение существующего кода. - СПб: Символ-Плюс, 2003. - 432 с
30. Фаулер М. Шаблоны корпоративных приложений / М.Фаулер, Д.Райс, М.Фоммел и др. - М.: Вильямс, 2010.
31. Фримен Э. Паттерны проектирования/ Э. Фримен, Э. Фримен, К. Сьерра, Б. Бейтс – СПб.: Питер, 2011.
32. Руководство Microsoft по проектированию АРХИТЕКТУРЫ ПРИЛОЖЕНИЙ Microsoft 2009.

15. Інформаційні ресурси

1. Best Practices for Software Development Teams [Електронний ресурс]. – Режим доступу: URL : https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_T_P026B.pdf
2. Введение в RUP [Електронний ресурс]. – Режим доступу: URL : http://dit.isuct.ru/Publish_RUP/core.base_rup/guidances/supportingmaterials/introduction_to_rup_36B6_3436.html
3. Unified Modeling Language™ (UML®) Resource Page [Електронний ресурс]. Режим доступу: www.uml.org
4. Руководство Microsoft по проектированию архитектуры приложений [Електронний ресурс]. Режим доступу: http://www.dut.edu.ua/uploads/1_1507_99407341.pdf
5. Як навчати архітектурі ПЗ [Електронний ресурс]. Режим доступу: <https://dou.ua/forums/topic/3398/>
6. School bionic [Електронний ресурс]. Режим доступу: http://school.bionic-university.com/programs/java_ee
7. ПРОФЕСІЙНИЙ СТАНДАРТ [Електронний ресурс]. Режим доступу: <http://mon.gov.ua/content/%D0%9D%D0%BE%D0%B2%D0%B8%D0%BD%D0%B8/2016/03/15/6-ps-rozrobnik-pz-13.12.2014.pdf> Як навчати архітектурі ПЗ [Електронний ресурс]. Режим доступу: <https://dou.ua/forums/topic/3398/>
8. Патерни Програмування: Що таке патерни та їхні типи? [Електронний ресурс]. Режим доступу: <http://www.vitaliyupodoba.com/2014/06/programming-patterns-intro/>
9. Що таке архітектура програмного забезпечення?, [Електронний ресурс]. Режим доступу: <http://easy-code.com.ua/2012/07/shho-take-arxitektura-programnogo-zabezpechennya-rizne-programuvannya-statti/>
10. SO AND SO [Електронний ресурс]. Режим доступу: <https://soandso.biz/blog/software-engineering/arhitektura-programnogo-zabezpechennya.html>