

# СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ «МЕТОДОЛОГІЇ ТА ТЕХНОЛОГІЇ РОЗРОБКИ ПРОГРАМНИХ СИСТЕМ»



Ступінь освіти	магістр
Спеціальність	121 Інженерія програмного забезпечення
Освітня програма	Інженерія програмного забезпечення
Тривалість викладання	осінній семестр (1, 2 чверті)
Кількість кредитів	5 кредитів ЄКТС (150 годин)
Заняття:	
лекції:	2 год/тиждень
лабораторні:	2 год/тиждень
Мова викладання	українська

Сторінка курсу в СДО НТУ «ДП»: <https://do.nmu.org.ua/course/view.php?id=5504>

**Кафедра, що викладає** Програмного забезпечення комп'ютерних систем



**Викладач:**

Удовик Ірина Михайлівна  
професор, к.т.н

**Персональна сторінка**

[https://pzks.nmu.org.ua/ua/teachers/head\\_of\\_the\\_department.php](https://pzks.nmu.org.ua/ua/teachers/head_of_the_department.php)

**E-mail:** [udovyk.i.m@nmu.one](mailto:udovyk.i.m@nmu.one)

## 1. Анотація до курсу

Методологія розробки програмних систем - сукупність методів, що застосовуються на різних стадіях життєвого циклу програмного забезпечення. Кожна методологія характеризується своїми принципами, від яких залежить ефективність всієї методології, узгодженої множиною моделей методів, що реалізують цю методологію та концепціями (поняттями), що дозволяють точніше визначити методи розробки. У окремому випадку, коли методологія застосовується на стадії програмування (конструювання), її зазвичай називають парадигмою програмування.

У результаті опанування курсу здобувачі вищої освіти зможуть аналізувати, оцінювати і застосовувати на системному рівні сучасні програмні та апаратні платформи для розв'язання складних задач інженерії програмного забезпечення; обґрунтовано обирати парадигми і мови програмування для розроблення програмного забезпечення; застосовувати на практиці сучасні засоби розроблення програмного забезпечення; прогнозувати розвиток програмних систем та інформаційних технологій; вміти використовувати методи фундаментальних і прикладних дисциплін інженерії програмного забезпечення при проектуванні архітектури та розробці програмних систем.

## **2. Мета та завдання курсу**

**Мета дисципліни** – формування компетентностей щодо використання методів та парадигм, що застосовуються на різних стадіях життєвого циклу програмного забезпечення.

### **Завдання курсу:**

Навчити здобувачів вищої освіти:

- знанням і практичним навичкам з проектування програмних систем;
- обґрунтовано обирати парадигми і мови програмування для розроблення програмного забезпечення;
- застосовувати на практиці сучасні методології та технології розроблення програмного забезпечення;
- підготувати студентів до провадження дослідницької та/або інноваційної діяльності в галузі проектування програмних систем, сформувані компетентності, необхідні при проектуванні програмних систем у складі колективу.

## **3. Результати навчання**

Дисциплінарні результати навчання:

- оцінювати і обирати ефективні методи і моделі розроблення програмного забезпечення;
- обирати технології з використанням мультипарадигмового підходу;
- розробляти і модифікувати архітектуру програмного забезпечення для ідентифікації, класифікації та опису роботи програмно-технічних засобів;
- розробляти і модифікувати архітектуру програмного забезпечення для класифікації та опису в контексті мультипарадигмного підходу до створення програмних продуктів;
- застосовувати на практиці сучасні засоби розроблення програмного забезпечення з урахуванням мультипарадигмного підходу.

Дисциплінарні результати навчання сформовано на основі ПРН освітньо-професійної програми (РН02, РН03, РН07, РН08, РН09).

#### 4. Структура курсу

Вид заняття	Внесок в загальну оцінку, %
<b>ЛЕКЦІЇ</b>	<b>40</b>
<p><b>1.</b> Сучасний стан розробки програмних засобів. Порівняльний аналіз та вибір життєвого циклу розроблення програмного забезпечення. Розповсюджені процеси та етапи розроблення програмних систем.</p> <ol style="list-style-type: none"> <li>1. Сучасний стан розробки програмних засобів.</li> <li>2. Огляд та порівняльний аналіз моделей життєвого циклу програмного забезпечення.</li> <li>3. Вибір прийнятної моделі життєвого циклу розроблення програмного забезпечення.</li> <li>4. Розповсюджені процеси та етапи розроблення програмних систем.</li> </ol>	
<p><b>2.</b> Основні фази, стандарти та засоби розроблення програмного забезпечення. Сучасні технології проектування програмного забезпечення.</p> <ol style="list-style-type: none"> <li>1. Основні фази розроблення ПЗ: формулювання вимог, формулювання цілей проекту, аналіз прикладної галузі, створення функційної специфікації, проектування, реалізація.</li> <li>2. Стандарти в галузі розроблення ПЗ.</li> <li>3. Програмні засоби підтримки життєвого циклу.</li> <li>4. Визначення технології проектування програмного забезпечення (ТППЗ). Загальні вимоги, пропоновані до ТППЗ. Приклади ТППЗ.</li> <li>5. Моделі систем. Прототипування програмних систем</li> </ol>	
<p><b>3.</b> Етапи визначення вимог до програмної системи. Проектування та модифікація архітектури програмних систем. Управління ризиками при розробці програмного забезпечення.</p> <ol style="list-style-type: none"> <li>1. Формалізація вимог: виділення вимог за допомогою прецедентів.</li> <li>2. Формалізація вимог: псевдокод, кінцеві автомати, графічні дерева рішень.</li> <li>3. Завдання та результати етапу аналізу вимог.</li> <li>4. Планування архітектури.</li> <li>5. Проектування архітектури.</li> <li>6. Документування архітектури.</li> <li>7. Аналіз архітектури.</li> <li>8. Поняття ризику.</li> <li>9. Управління ризиками. Метод PERT-аналізу.</li> </ol>	
<p><b>4.</b> Сучасні паттерни програмування та їх застосування при розробці програмних систем.</p> <ol style="list-style-type: none"> <li>1. Circuit Breaker.</li> <li>2. Command and Query Responsibility Segregation (CQRS).</li> </ol>	

3. Event Sourcing. 4. Sidecar. 5. Backend-for-Frontend. 6. Strangler.	
<i>Тестова контрольна робота №1</i>	<b>20</b>
<p><b>5.</b> Мультипарадигмове програмування: поняття, порівняння, програмні та апаратні платформи для розв'язування задач з інженерії програмного забезпечення.</p> <p>1. Мультипарадигмове програмування. Інженерія предметної області в контексті парадигм програмування.</p> <p>2. Необхідність кількох парадигм програмування.</p> <p>3. Визначення мультипарадигмового підходу. Аналіз предметних областей, та застосування декількох парадигм.</p> <p>4. Мультипарадигмове проектування та реалізація.</p> <p>5. Складні рішення та підключення паттернів.</p>	
<p><b>6.</b> Використання мультипарадигмового підходу та його засобів в проектуванні програмного забезпечення.</p> <p>1. Огляд методів мультипарадигмового проектування. Міри складності в проектуванні програмних систем.</p> <p>2. Особливість застосування декількох парадигм в кожній підобласті.</p> <p>3. Динамічне зв'язування мультипарадигмових елементів.</p> <p>5. Дедуктивна та індуктивна спільність мультипарадигмових елементів.</p> <p>6. Сімейства програм. Словник предметної області. Аналіз мінливості.</p> <p>7. Аналіз прикладної області і області рішень мультипарадигмової системи.</p>	
<p><b>7.</b> Формальна специфікація та дизайн програмного забезпечення.</p> <p>1. Методи моделювання на основі подій і РАТ.</p> <p>2. Теоретичні основи перевірки моделей.</p> <p>3. Мови властивостей та їх використання.</p> <p>4. Виконання автоматизованої перевірки за допомогою інструменту перевірки моделі програмного забезпечення.</p>	
<i>Тестова контрольна робота №2</i>	<b>20</b>
<b>ЛАБОРАТОРНІ ЗАНЯТТЯ</b>	<b>60</b>
Лабораторна робота №1 Розроблення технічного завдання. Методологія об'єктоорієнтованого аналізу й проектування.	
Звіт та захист роботи №1	<b>10</b>
Лабораторна робота №2 Розроблення діаграм варіантів використання і діаграм класів. Розроблення діаграм станів і діаграм діяльності.	
Звіт та захист роботи №2	<b>10</b>
Лабораторна робота №3 Розроблення діаграм послідовності і діаграм кооперації. Розроблення діаграм компонентів і діаграм розгортання.	
Звіт та захист роботи №3	<b>10</b>

Лабораторна робота №4 Управління розробкою програмних проєктів	
Звіт та захист роботи №4	<b>10</b>
Лабораторна робота №5 Командна розробка програмних проєктів у Microsoft Team Foundation Server	
Звіт та захист роботи №5	<b>10</b>
Лабораторна робота №6 Командний рефакторинг попереднього проєкту за допомогою мультипарадигмового підходу.	
Звіт та захист роботи №6	<b>10</b>
<b>Загальна кількість балів</b>	<b>100</b>

## 5. Технічне обладнання та/або програмне забезпечення\*

Активованій акаунт університетської пошти (student.i.p.@nmu.one) на Офіс365, використання дистанційної платформи (<https://do.nmu.org.ua/>). Microsoft Team Foundation Server. StarUML. Пакет Офіс365.

## 6. Система оцінювання та вимоги

**6.1. Навчальні досягнення здобувачів вищої освіти** за результатами вивчення курсу оцінюватимуться за шкалою, що наведена нижче:

Рейтингова шкала	Інституційна шкала
90 – 100	відмінно
74-89	добре
60-73	задовільно
0-59	незадовільно

Загальні критерії досягнення результатів навчання відповідають описам 6-го кваліфікаційного рівня НРК.

**6.2.** Здобувачі вищої освіти можуть отримати **підсумкову оцінку** з навчальної дисципліни **на підставі поточного оцінювання знань** за умови, якщо набрана кількість балів з поточного тестування та виконання і захисту практичних робіт складатиме не менше 60 балів.

**Теоретична частина** оцінюється за результатами здачі двох контрольних тестових робіт, кожна з яких містить тестові закриті запитання з однією вірною відповіддю, максимальна кількість – 20 питань та вираховується відсоток кожної (розподіл % за окремими контрольними роботами див. в таблиці розділу 4). Загалом за дві контрольні тестові роботи отримується **максимум 40 балів**, тобто 40% від оцінки за дисципліну.

**Лабораторні роботи** (шість лабораторних робіт – у вигляді індивідуального завдання з кожної, розподіл % див. в таблиці розділу 4) виконуються у письмовому вигляді (звіт з кожної лабораторної роботи оцінюється в межах 10 балів (5 балів

максимальна оцінка за звіт та 5 балів за захист), загалом шість лабораторних враховуються, як 60% (максимум 60 балів). У сумі за практичну частину курсу при поточному оцінюванні отримується **максимум 60 балів**.

Отримані бали за теоретичну частину та лабораторні роботи додаються і є підсумковою оцінкою за вивчення навчальної дисципліни. Максимально за поточною успішністю здобувач вищої освіти може набрати 100 балів.

Максимальне оцінювання поточного контролю в балах:

Теоретична частина	Практична частина	Разом
40	60	<b>100</b>

**6.3. Критерії оцінювання підсумкової роботи.** У випадку якщо здобувач вищої освіти за поточною успішністю отримав менше 60 балів та/або прагне поліпшити оцінку проводиться **підсумкове оцінювання (іспит)** під час сесії.

**Іспит** проводиться у вигляді комплексної контрольної роботи, яка включає запитання з теоретичної та практичної частини курсу. Білет складається з **30 тестових завдань** з чотирма варіантами відповідей, одна правильна відповідь оцінюється в 2 бали (**разом 60 балів**) та **2 практичні завдання**, кожне з запитань оцінюється максимум у 20 балів (**разом 40 балів**), причому:

- 20 балів – відповідність еталону;
- 15 балів – відповідність еталону з незначними помилками;
- 10 балів – часткова відповідність еталону, питання повністю не розкриті;
- 5 балів – невідповідність еталону, але відповідність темі запитання;
- 0 балів – відповідь не наведена або не відноситься до теми запитання.

Отримані бали за запитання з теоретичної та практичної частини курсу додаються і є підсумковою оцінкою за вивчення навчальної дисципліни. Максимально за підсумковою роботою здобувач вищої освіти може набрати 100 балів.

## 7. Політика курсу

**7.1. Політика щодо академічної доброчесності.** Академічна доброчесність здобувачів вищої освіти є важливою умовою для опанування результатами навчання за дисципліною і отримання задовільної оцінки з поточного та підсумкового контролів. Академічна доброчесність базується на засудженні практик списування (виконання письмових робіт із залученням зовнішніх джерел інформації, крім дозволених для використання), плагіату (відтворення опублікованих текстів інших авторів без зазначення авторства), фабрикації (вигадування даних чи фактів, що використовуються в освітньому процесі). Політика щодо академічної доброчесності регламентується положенням "Положення про систему запобігання та виявлення плагіату у Національному технічному університеті "Дніпровська політехніка" <https://cutt.ly/IBesJEc>

У разі порушення здобувачем вищої освіти академічної доброчесності (списування, плагіат, фабрикація), робота оцінюється незадовільно та має бути виконана повторно. При цьому викладач залишає за собою право змінити тему завдання.

**7.2. Комунікаційна політика.** Здобувачі вищої освіти повинні мати активовану корпоративну університетську пошту.

Усі письмові запитання до викладачів стосовно курсу мають надсилатися на університетську електронну пошту.

**7.3. Політика щодо перескладання.** Роботи, які здаються із порушенням термінів без поважних причин оцінюються на нижчу оцінку. Перескладання підсумкового оцінювання відбувається із дозволу деканату за наявності поважних причин (наприклад, лікарняний).

**7.4 Політика щодо оскарження оцінювання.** Якщо здобувач вищої освіти не згоден з оцінюванням його знань він може опротестувати виставлену викладачем оцінку у встановленому порядку.

**7.5. Відвідування занять.** Для здобувачів вищої освіти денної форми відвідування занять є обов'язковим. Поважними причинами для неявки на заняття є хвороба, участь в університетських заходах, академічна мобільність, які необхідно підтверджувати документами. Про відсутність на занятті та причини відсутності здобувач вищої освіти має повідомити викладача або особисто, або через старосту.

За об'єктивних причин (наприклад, міжнародна мобільність) навчання може відбуватись в он-лайн формі за погодженням з керівником курсу.

**7.6. Опитування.** Наприкінці вивчення курсу та перед початком сесії здобувача вищої освіти буде запропоновано анонімно заповнити електронні анкети (Microsoft Forms Office 365), які буде розіслано на ваші університетські поштові скриньки. Заповнення анкет є важливою складовою вашої навчальної активності, що дозволить оцінити дієвість застосованих методів викладання та врахувати ваші пропозиції стосовно покращення змісту навчальної дисципліни.

## **8 Рекомендовані джерела інформації**

1. Авраменко В.С. Технологія програмування та створення програмних продуктів. Електронний конспект лекцій. ЧНУ. 2017. – 192 с.
2. Elad Shalom A Review of Programming Paradigms Throughout the History: With a Suggestion Toward a Future Approach / USA: Independently published, 2015. – 345 p.
3. Гнатовська Г.А. Конспект лекцій з дисципліни «Технологія створення програмних продуктів» Одеса, 2015 – 97 с.
4. Мартін Р. Чистий код: створення і рефакторинг за допомогою Agile / пер. з англ. І. БондарТерещенко. – Харків: Вид-во «Ранок»: Фабула, 2019. – 448 с.
5. Systems and software engineering – Software Life Cycle Processes. ISO 12207:2008. – [Чинний від 2008-02-01] – II, 122 с.– (Міжнародний стандарт).