

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Волинський національний університет імені Лесі Українки
Факультет інформаційних технологій і математики
Кафедра комп'ютерних наук та кібербезпеки

СИЛАБУС

вибіркового освітнього компонента

ЧИСТИЙ КОД ТА КРАЩІ ПРАКТИКИ ПРОГРАМУВАННЯ

підготовки бакалавра

Луцьк – 2026

Силабус освітнього компонента «Чистий код та кращі практики програмування» підготовки бакалавра, всіх галузей знань, всіх спеціальностей, за всіма освітніми програмами.

Розробник: Борозенець Д.В., асистент кафедри комп'ютерних наук та кібербезпеки.

Силабус освітнього компонента затверджено та погоджено на засіданні кафедри комп'ютерних наук та кібербезпеки

протокол № 6 від 15.01.2026 р.

Завідувач кафедри:



Гришанович Т. О.

I. Опис освітнього компонента

Найменування показників	Галузь знань, спеціальність, освітньо-професійна /освітньо-наукова/освітньо-творча програма, освітній рівень	Характеристика освітнього компонента
Денна форма здобуття освіти	Всі галузі знань Всі спеціальності Всі освітньо-професійні програми Бакалавр	Вибірковий
Кількість годин/кредитів 150/5		Рік навчання 2
		Семестр 4-ий
ІНДЗ: <u>немає</u>		Лекції 10 год.
		Лабораторні 20 год.
		Самостійна робота 110 год.
Мова навчання		Консультації 10 год.
	Форма контролю: залік	
		Українська

II. Інформація про викладача

ППІ Борозенець Дмитро Володимирович
 Науковий ступінь -
 Вчене звання -
 Посада асистент кафедри комп'ютерних наук та кібербезпеки
 Контактна інформація ел. скринька: Borozenets.Dmytro@vnu.edu.ua
 Дні занять <https://ps.vnu.edu.ua/cgi-bin/timetable.cgi?n=700>

III. Опис освітнього компонента

1. Анотація освітнього компонента

Силабус освітнього компонента «**Чистий код та кращі практики програмування**» спрямований на формування професійної культури розробника програмного забезпечення та підготовку здобувачів освіти до створення читабельного, підтримуваного та масштабованого коду.

Дисципліна охоплює принципи написання чистого коду, сучасні підходи до проєктування програмного забезпечення (SOLID, DRY, KISS, YAGNI), методи рефакторингу, основи тестування та кращі практики організації командної розробки. Особлива увага приділяється формуванню відповідального ставлення до якості програмного продукту, зменшенню технічного боргу та забезпеченню довготривалої підтримки програмних систем.

Курс орієнтований на студентів 2 курсу та формує фундамент для подальшого вивчення програмної інженерії, архітектури програмного забезпечення та професійної розробки ПЗ.

2. Пререквізити та постреквізити

Пререквізити.

Знання та вміння, отримані в результаті вивчення базових дисциплін з програмування (Python, Java, C++, C# або інших мов), розуміння алгоритмів і структур даних, базові знання об'єктно-орієнтованого програмування, навички роботи з системами контролю версій (на початковому рівні).

Постреквізити.

Знання та навички, здобуті під час вивчення дисципліни, є основою для опанування курсів з програмної інженерії, архітектури програмного забезпечення, DevOps, тестування ПЗ, розробки великих інформаційних систем, а також для участі у реальних командних проєктах та проходження виробничої практики.

3. Мета і завдання освітнього компонента

Мета дисципліни: сформувати у здобувачів освіти системне розуміння принципів створення якісного, читабельного, структурованого та підтримуваного програмного коду відповідно до сучасних стандартів програмної інженерії.

Завдання:

- сформувати навички написання чистого, структурованого та зрозумілого коду;
- навчити застосовувати принципи SOLID, DRY, KISS, YAGNI у практичній розробці;
- опанувати методи виявлення та усунення «code smells»;
- навчитися виконувати рефакторинг без зміни функціональності програмного продукту;
- сформувати розуміння ролі тестування у забезпеченні якості коду;
- розвинути навички організації структури проєкту та дотримання стандартів програмування;
- сформувати відповідальне ставлення до якості програмного продукту та мінімізації технічного боргу.

4. Soft skills

- **Критичне та аналітичне мислення** – уміння оцінювати якість коду, знаходити приховані проблеми та прогнозувати наслідки архітектурних рішень.
- **Увага до деталей** – здатність помічати неточності у структурі, логіці та стилі програмного коду.
- **Системне мислення** – розуміння взаємозв'язків між модулями та компонентами програмної системи.
- **Problem Solving** – здатність знаходити оптимальні способи покращення коду та зменшення його складності.
- **Командна взаємодія** – навички роботи в команді, участі у code review, аргументованого обговорення технічних рішень.
- **Відповідальність за результат** – формування професійної культури розробника, орієнтованої на якість, а не лише на швидкість реалізації.

- **Безперервне навчання (Lifelong Learning)** – готовність постійно вдосконалювати підходи до розробки відповідно до розвитку технологій та індустріальних стандартів.

5. Структура освітнього компонента.

Назви змістових модулів і тем	Усього	Лек.	Лабор.	Сам. роб.	Конс.	Форма контролю / Бали
Змістовий модуль 1. Сучасні практики розробки та забезпечення якості програмного коду						
Тема 1. Концепція чистого коду: принципи читабельності, зрозумілості та структурованості програмного коду. Розгляд характеристик якісного коду, правил іменування, форматування, структурування файлів і модулів, мінімізації складності, уникнення дублювання та типових помилок (code smells).	23	2	4	15	2	Звіт/6/6
Тема 2. Принципи проєктування програмного забезпечення: SOLID, DRY, KISS, YAGNI та їх практичне застосування. Формування навичок побудови гнучкої, масштабованої та розширюваної архітектури програм із використанням сучасних підходів до модульності та відповідальності компонентів.	23	2	4	15	2	Звіт /6/6
Тема 3. Рефакторинг як інструмент покращення якості коду без зміни функціональності. Методи виявлення проблем у коді, техніки рефакторингу, робота з технічним боргом, поступове покращення існуючих програмних рішень.	23	2	4	15	2	Звіт /6/6
Тема 4. Тестування програмного забезпечення як складова якісної розробки. Основи модульного тестування, принципи TDD, забезпечення покриття коду тестами, роль автоматизації перевірок у підтримці	23	2	4	15	2	Звіт /6/6

чистоти та стабільності програмного продукту.						
Тема 5. Кращі практики організації коду та командної розробки програмного забезпечення. Стандарти структурування проєктів, контроль версій, code review, стилістичні гіді, статичний аналіз коду та інструменти підтримки якості в командному середовищі.	23	2	4	15	2	Звіт/6/6
Разом за модулем 1	115	10	20	75	10	60
Контрольна робота				20		20
Тестування				15		20
Всього годин/Балів	150	10	20	110	10	100

1. Завдання для самостійного опрацювання.

№ з/п	Тема (опрацювати)	Кількість год.
1	Опрацювання та аналіз лекційного матеріалу	20
2	Опрацювання додаткових джерел та відео-роликів мережі Інтернет	15
3	Підготовка до лабораторних робіт	20
4	Підготовка до контрольної роботи	20
5	Підготовка до тестування	15
6	Підготовка відповідей на питання до заліку	20

IV. Політика оцінювання

Політика викладача щодо здобувача освіти

Усі учасники освітнього процесу повинні дотримуватись вимог чинного законодавства України, Статуту і Правил внутрішнього розпорядку ВНУ імені Лесі Українки, загально-прийнятих моральних принципів, правил поведінки та корпоративної культури; підтримувати атмосферу доброзичливості, відповідальності, порядності й толерантності. Атмосфера на заняттях повинна бути творчою, відкритою до конструктивної критики. Недопустимі запізнення на заняття; користування мобільним телефоном, планшетом чи іншими мобільними пристроями під час заняття; списування. Очікується, що всі студенти відвідають усі лекції і практичні заняття курсу.

Політика щодо академічної доброчесності

Дотримання академічної доброчесності здобувачами передбачає: самостійне виконання завдань поточного та підсумкового контролю (для осіб з особливим освітніми потребами ця вимога застосовується з урахуванням їх індивідуальних потреб і можливостей); посилення на джерела інформації у разі використання ідей, тверджень,

відомостей; дотримання норм законодавства про авторське право; надання достовірної інформації про результати власної навчальної (наукової, творчої) діяльності.

Порушенням академічної доброчесності вважається: академічний плагіат, самоплагіат, фабрикація, фальсифікація, списування. За порушення академічної доброчесності здобувачі освіти можуть бути притягнені до такої академічної відповідальності: повторне проходження оцінювання; повторне проходження відповідного освітнього компонента освітньої програми.

Під час модульного та підсумкового контролю (заліку) студентам заборонено користуватися такими засобами як мобільний телефон, планшет, конспект, навчальна література, інші джерела інформації, в тому числі Інтернет-ресурси.

Політика щодо дедлайнів та перескладання

Усі передбачені завдання мають бути виконані у встановлений термін. Несвоєчасно виконані завдання оцінюються на нижчу оцінку. Виключенням можуть бути завдання, які не вдалося зробити з поважних причин, в такому випадку студент може доробити вказані завдання у вказаний термін.

Якщо здобувач вищої освіти був відсутній на заняттях з будь-якої причини, то він (вона) вивчає матеріал самостійно, використовуючи навчальні посібники, конспекти лекцій, матеріали дистанційного курсу, у випадку розміщення його на платформі дистанційного навчання Moodle, виконує всі домашні завдання (завдання подані на самостійну роботу). Прозвітуватися про виконання завдань можна, використовуючи дистанційний курс, прикріпивши виконанні завдання у відповідні комірки та попередити викладача про здане завдання, або під час консультацій або надіслати виконане завдання на корпоративну пошту викладача. Зворотній зв'язок з викладачем для з'ясування всіх питань: використання форуму, чату дистанційного курсу, корпоративної пошти університету або відповідної бесіди у певному месенджері.

Можливість визнання результатів навчання, отриманих у формальній, неформальній та інформальній освіті

Під час вивчення освітнього компонента можливе визнання інших результатів навчання, отриманих у формальній, неформальній та/або інформальній освіті. Порядок визнання результатів навчання для здобувачів вищої освіти, набутих у: формальній освіті (академічна мобільність студентів на території України чи поза її межами, для студентів, які переводяться, поновлюються з інших ЗВО (вітчизняних чи іноземних); неформальній та/або інформальній освіті здійснюється згідно «ПОЛОЖЕННЯ про визнання результатів навчання, отриманих у формальній, неформальній та/або інформальній освіті у Волинському національному університеті імені Лесі Українки».

Можливість отримати додаткові (бонусні) бали

Відповідно до пункту 4.5 Положення про поточне та підсумкове оцінювання знань здобувачів вищої освіти Волинського національного університету імені Лесі Українки здобувачам освіти, які брали участь у роботі конференцій, підготовці наукових публікацій, в олімпіадах, конкурсах студентських наукових робіт, спортивних змаганнях, мистецьких конкурсах тощо й досягли значних результатів, може бути присуджено додаткові (бонусні)

бали, які зараховуються як результати поточного контролю з відповідного ОК. Систему бонусних балів погоджує науково-методична комісія факультету (інституту).

І так, здобувачі освіти мають можливість отримати додаткові бали за вказаний вид робіт з ОК «Програмування» відповідно до таблиці витягу з протоколу № 1 засідання НМТ ФІТІМ ВНУ ім. Лесі Українки від 3.09.2025 р.

Система бонусних балів для здобувачів освіти

Вид діяльності	Рівень / результат	Кількість бонусних балів
Студентські олімпіади	I місце	7
	II місце	5
	III місце	3
	Участь в олімпіаді	2
Конкурси студентських наукових робіт	Диплом I ступеня	7
	Диплом II ступеня	5
	Диплом III ступеня	3
Підготовка наукових публікацій	Публікація в WoS / Scopus	10
	Фахова стаття	7
	Нефахова стаття	5
	Публікація тез	2
Участь у конференціях	Виступ на конференції	2
Першість України з командного програмування	I місце	10
	II місце	8
	III місце	6
	Участь	4

V. Підсумковий контроль

Підсумковий контроль з даної дисципліни передбачено у вигляді заліку.

Оцінювання здійснюється за 100-бальною шкалою. Оцінка включає в себе оцінювання всіх видів запланованої навчальної роботи протягом семестру: нараховується за якісне виконання лабораторних, індивідуальних робіт та виконання самостійної роботи. Максимальна кількість балів, яку може отримати студент під час поточного оцінювання, у випадку заліку, за семестр – 100 балів.

Залік викладач виставляє за результатами поточної роботи за умови, що здобувач освіти виконав ті види навчальної роботи, які визначено силабусом ОК. У випадку, якщо здобувач освіти не відвідував окремі аудиторні заняття (з поважних причин), на консультаціях він має право відпрацювати пропущені заняття та добрати ту кількість балів, яку було визначено на пропущені теми. У дату складання заліку викладач записує у відомість суму поточних балів, які здобувач освіти набрав під час поточної роботи (шкала від 0 до 100 балів).

У випадку, якщо здобувач освіти протягом поточної роботи набрав менше як 60 балів, він складає залік під час ліквідації академічної заборгованості. У цьому випадку бали, набрані під час поточного оцінювання анулюються. Максимальна кількість балів на залік під час ліквідації академічної заборгованості, як правило, 100. У день складання заліку за основною сесією заборонено проводити додаткові опитування здобувача освіти, а також здобувач освіти не має права доздавати будь-який вид робіт, передбачений силабусом освітнього компоненту. На заліку, під час ліквідації академічної заборгованості, здобувач отримує комплексне завдання, яке охоплює всі теми і всі форми контролю, які пропонувалися при вивченні освітнього компонента. Порядок проведення заліку-ліквідації – залік відбувається у вигляді виконання комплексного завдання.

Питання до заліку-ліквідації та приклади практичних завдань

1. Дайте визначення поняттю «чистий код» та охарактеризуйте його основні ознаки.
2. Які критерії визначають читабельність і зрозумілість програмного коду?
3. Поясніть принципи правильного іменування змінних, функцій, класів і модулів.
4. Яку роль відіграє форматування коду у забезпеченні його якості?
5. Що таке цикломатична складність та як її можна зменшити?
6. Поясніть поняття «code smells» та наведіть приклади типових проблем коду.
7. У чому полягає принцип DRY та які наслідки його порушення?
8. Розкрийте сутність принципів KISS та YAGNI у проектуванні ПЗ.
9. Охарактеризуйте кожен із принципів SOLID та наведіть приклади їх застосування.
10. Що таке модульність та як вона впливає на масштабованість програмного забезпечення?
11. Дайте визначення рефакторингу та поясніть його відмінність від переписування коду.
12. Які основні техніки рефакторингу ви знаєте?
13. Що таке технічний борг та як він впливає на розвиток програмного продукту?
14. Яким чином виявляються проблемні ділянки коду перед рефакторингом?
15. Поясніть роль модульного тестування у забезпеченні якості програмного коду.
16. У чому полягає концепція TDD (Test-Driven Development)?
17. Що таке покриття коду тестами та як його оцінюють?
18. Які переваги автоматизації тестування у процесі розробки програмного забезпечення?
19. Назвіть основні принципи організації структури програмного проекту.
20. Яку роль відіграють стилістичні гіді та стандарти кодування у командній розробці?
21. Поясніть сутність процесу code review та його значення для якості ПЗ.
22. Які можливості надають системи контролю версій у командній роботі?
23. Що таке статичний аналіз коду та які його переваги?
24. Як поєднання принципів чистого коду, рефакторингу та тестування впливає на підтримуваність програмного продукту?

Приклади практичних завдань

1. Покращення якості коду

Надається готовий фрагмент програми з помилками стилю та структури.

Потрібно:

- знайти недоліки (погані назви змінних, дублювання коду, занадто складні функції);
- переписати код так, щоб він став більш зрозумілим і структурованим;
- коротко пояснити внесені зміни.

2. Простий рефакторинг програми

Надано працюючу програму з довгими або перевантаженими функціями.

Потрібно:

- розділити код на менші функції або класи;

- усунути дублювання;
- не змінювати роботу програми;
- показати, що програма після змін працює так само, як і раніше.

3. Невеликий проєкт з тестами

Розробити просту програму (наприклад, калькулятор, систему обліку оцінок або список справ).

Вимоги:

- зрозуміла структура проєкту;
- дотримання правил іменування та форматування;
- написання декількох модульних тестів для перевірки роботи програми;
- збереження проєкту у системі контролю версій.

Шкала оцінювання знань здобувачів освіти з освітніх компонентів, де формою контролю є залік

Оцінка в балах	Лінгвістична оцінка
90–100	Зараховано
82–89	
75–81	
67–74	
60–66	
1–59	Незараховано (необхідне перескладання)

VII. Рекомендована література та інтернет-ресурси

1. Чистий кодер: Кодекс поведінки для професійних розробників / пер. з англ. Г. Якубовська. — Харків : ВД Фабула, 2023. — 256 с. — Режим доступу: https://ridmi.com.ua/contents/chystyi-koder-chystyi-koder-ebook.pdf?srsId=AfmBOoolxzA_LDfws7auXWg5Wa1Gr8Tc91g_fQvL9U9F6TYO4168sDyV
2. Посібник із принципів чистого коду [Електронний ресурс] // *Clean Code Guy*. — 2025. — Режим доступу: <https://cleancodeguy.com/uk/blog/clean-coding-principles>
3. Найкращі книги з чистого коду для розробників [Електронний ресурс] // *Clean Code Guy*. — 2025. — Режим доступу: <https://cleancodeguy.com/uk/blog/clean-code-books>
4. Що таке чистий код? Методи рефакторингу [Електронний ресурс] // *IT Ukraine Association*. — 2024. — Режим доступу: <https://itukraine.org.ua/it-events/shho-take-chistij-kod-metodi-refaktoringu/>
5. Онлайн-курс «Чистий код та патерни проєктування» [Електронний ресурс] // *DOU.ua Calendar*. — 2024. — Режим доступу: <https://dou.ua/calendar/51098/>
6. Онлайн-вебінар «Що таке чистий код? Методи рефакторингу» [Електронний ресурс] // *DOU.ua Calendar*. — 2024. — Режим доступу: <https://dou.ua/calendar/49457/>

7. Шевченко А. Огляд книжки «Чистий код» Роберта Мартіна [Електронний ресурс] // Форум *DOU.ua*. — 18 грудня 2023. — Режим доступу: <https://dou.ua/forums/topic/46697/>
8. qa семпай | автоматизація тестування. Чи потрібен "Clean Code" у 2026? Книги, AI та реалії IT, 2026. *YouTube*. URL: <https://www.youtube.com/watch?v=NyebjaMiY70>
9. CodeUA. Що таке чистий код? Методи рефакторингу, 2024. *YouTube*. URL: <https://www.youtube.com/watch?v=0C-bPxAkHcg>
10. Що таке best practice у програмуванні? [Електронний ресурс] // *Tseivo.com*. — 09 грудня 2025. — Режим доступу: <https://tseivo.com/b/memecode/t/qg2k5kq7lr>
11. Good practice, bad practice, хард код та сучасні практики програмування [Електронний ресурс] // Форум *DOU.ua*. — опубліковано ~1,5 року тому. — Режим доступу: <https://dou.ua/forums/topic/49945/>