

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

/назва факультету/

Кафедра комп'ютерних наук

/назва кафедри/

“ЗАТВЕРДЖУЮ”

Декан факультету

Ігор БАРАН

2024 року

“ 02 ”



РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Технологія створення програмних продуктів

/назва дисципліни/

галузь знань 12 Інформаційні технології

/шифр і назва галузі знань/

рівень вищої освіти Перший (бакалаврський рівень)

/назва/

спеціальність 122 «Комп'ютерні науки»

/шифр і назва/

освітня програма Комп'ютерні науки

/назва/

спеціалізація

/назва/

вид дисципліни обов'язкова дисципліна циклу професійної підготовки

/обов'язкова / вибіркова/

Робоча програма з навчальної дисципліни Технологія створення програмних продуктів

/назва дисципліни/

для студентів факультету комп'ютерно-інформаційних систем і програмної інженерії

/назва факультету(ів)/

Розробник:

асистент кафедри комп'ютерних наук,
доктор філософії

/посада, науковий ступінь та вчене звання/

/підпис/

Олег ПАЛКА

/ім'я та прізвище/

Робоча програма розглянута та схвалена

на засіданні кафедри комп'ютерних наук

/назва/

Протокол від « 26 » серпня 2024 року № 1

Завідувач кафедри

/підпис/

Ігор БОДНАРЧУК

/ім'я та прізвище/

Робоча програма розглянута та схвалена НМК

факультету комп'ютерно-інформаційних систем і програмної інженерії

Протокол від « 02 » вересня 2024 року № 1

Секретар НМК

/підпис/

Богдана МЛИНКО

/ім'я та прізвище/

Робоча програма погоджена:

Спеціальність 122 «Комп'ютерні науки»

/шифр і назва/

освітня програма Комп'ютерні науки

/назва/

Завідувач випускової кафедри

/підпис/

Ігор БОДНАРЧУК

/ім'я та прізвище/

Гарант освітньої програми

/підпис/

Леся ДМИТРОЦА

/ім'я та прізвище/

1. Структура навчальної дисципліни

Показник	Всього годин	
	Денна форма навчання (ДФН)	Заочна (дистанційна) форма навчання (ЗФН)
Кількість кредитів/год	4/120	4/120
Аудиторні заняття, год.	54	12
Самостійна робота, год.	66	108
Аудиторні заняття:		
– лекції, год.	18	6
– лабораторні заняття, год.	36	6
– практичні заняття, год.	–	–
– семінарські заняття, год.	–	–
Самостійна робота:		
– підготовка до лабораторних (практичних – семінарських) занять, год.	24	36
– опрацювання окремих розділів програми, які не виносяться на лекції, год.	30	48
– виконання контрольних завдання, год.	–	–
– виконання індивідуальних завдань, год.	–	–
– виконання курсових проектів (робіт), год.	–	–
– підготовка та складання <u>заліків</u> , екзаменів, контрольних робіт, рефератів, есе, <u>тестування</u> , год.	12	24
Екзамен	–	–
Залік	3	3

Частка годин самостійної роботи студента:

денна форма навчання - 55 %;

заочна (дистанційна) форма навчання - 90 %.

2. Мета та завдання навчальної дисципліни

2.1. Мета вивчення навчальної дисципліни

Основною метою дисципліни “Технологія створення програмних продуктів” є отримання студентами теоретичних знань та практичних навичок у сфері сучасних підходів, методів і технологій проєктування, розробки, тестування та підтримки програмного забезпечення. Особлива увага приділяється розумінню життєвого циклу програмного продукту, методологій розробки, аналізу вимог, архітектурних моделей, управління версіями коду, тестування та документування.

Предметом вивчення дисципліни “Технологія створення програмних продуктів” є сучасні підходи, технології та інструменти розробки програмного забезпечення, включаючи життєвий цикл розробки ПЗ, методології управління проєктами, архітектурні патерни, моделювання систем, управління версіями, клієнт-серверну архітектуру, тестування та забезпечення якості.

Важливим аспектом успішного засвоєння навчального матеріалу дисципліни є самостійна робота студентів з використанням інструментальних засобів проєктування, розробки та тестування програмного забезпечення (інтегрованих середовищ розробки) та відповідної технічної літератури.

2.2. Завдання навчальної дисципліни

Завданням дисципліни є формування у студентів теоретичних знань і практичних навичок проєктування архітектури та розробки програмних продуктів. Особливий акцент робиться на освоєнні сучасних технологій розробки, управлінні проєктами, використанні UML-діаграм, Git/GitHub, реалізації клієнт-серверних застосунків, тестуванні програмного забезпечення та документуванні.

По завершенню вивчення дисципліни студент повинен набути таких знань та вмінь:

Знати:

1. Основні терміни та поняття, пов'язані з технологією створення програмного забезпечення.

2. Життєвий цикл програмного продукту та основні моделі життєвого циклу.

3. Методології розробки програмного забезпечення (Agile, Scrum, Waterfall, Kanban) та принципи організації командної роботи.

4. Основи аналізу вимог до програмного забезпечення та методи їх верифікації.

5. Архітектурні моделі програмних продуктів (монолітна, клієнт-серверна, мікросервісна) та принципи проєктування.

6. Основи моделювання систем за допомогою UML (варіанти використання, класи, послідовності, стани).

7. Основи управління версіями коду, використання Git та GitHub.

8. Принципи побудови клієнт-серверних застосунків та роботи з хмарними сервісами.

9. Методи тестування програмного забезпечення (юніт-тести, інтеграційні тести, UI тести).

10. Види документації програмного забезпечення та основи їх складання.

Вміти:

1. Аналізувати вимоги до програмного забезпечення та формалізувати їх у технічному завданні.

2. Використовувати сучасні методології управління проєктами для організації розробки програмного продукту.

3. Проєктувати архітектуру програмного продукту та застосовувати архітектурні патерни.

4. Виконувати моделювання предметної області з використанням UML-діаграм.

5. Реалізовувати програмні продукти, використовуючи сучасні фреймворки та мови програмування.

6. Управляти версіями коду за допомогою Git/GitHub та організовувати командну роботу.

7. Створювати клієнт-серверні застосунки, використовуючи багаторівневу архітектуру.

8. Виконувати тестування програмного забезпечення та виправляти помилки за допомогою дебагінгу та оптимізації коду.

9. Оформлювати документацію програмного продукту відповідно до стандартів.

Вивчення навчальної дисципліни передбачає формування та розвиток у студентів інтегральної компетентності, загальних компетентностей та спеціальних (фахових) компетентностей згідно освітньої програми «Комп'ютерні науки» першого (бакалаврського) рівня вищої освіти.

Інтегральна компетентність (ІК):

Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі комп'ютерних наук або у процесі навчання, що передбачає застосування теорій та методів інформаційних технологій і характеризується комплексністю та невизначеністю умов.

Загальні компетентності (ЗК):

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК2. Здатність застосовувати знання у практичних ситуаціях.

ЗК3. Знання та розуміння предметної області та розуміння професійної діяльності.

ЗК6. Здатність вчитися й оволодівати сучасними знаннями.

ЗК9. Здатність працювати в команді.

ЗК11. Здатність приймати обґрунтовані рішення.

ЗК12. Здатність оцінювати та забезпечувати якість виконуваних робіт.

Спеціальні (фахові, предметні) компетентності (СК):

СК8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-

орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

СК9. Здатність реалізувати багаторівневу обчислювальну модель на основі архітектури клієнт-сервер, включаючи бази даних, знань і сховища даних, виконувати розподілену обробку великих наборів даних на кластерах стандартних серверів для забезпечення обчислювальних потреб користувачів, у тому числі на хмарних сервісах.

СК10. Здатність застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника.

Вивчення навчальної дисципліни передбачає формування у студента таких програмних результатів навчання (ПР):

ПР1. Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.

ПР9. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.

ПР10. Використовувати інструментальні засоби розробки клієнт-серверних застосувань, проектувати концептуальні, логічні та фізичні моделі баз даних, розробляти та оптимізувати запити до них, створювати розподілені бази даних, сховища та вітрини даних, бази знань, у тому числі на хмарних сервісах, із застосуванням мов веб-програмування.

ПР11. Володіти навичками управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти розробляти проектну документацію (техніко-економічне обґрунтування, технічне завдання, бізнес-план, угоду, договір, контракт).

3. Опис навчальної дисципліни

3.1. Лекційні заняття

№	Тема заняття та короткий зміст	Кількість годин	
		ДФН	ЗФН
1.	Тема 1. Вступ до технології створення програмного забезпечення. Основні поняття та терміни життєвого циклу (ЖЦ) розробки ПЗ. Життєвий цикл програмного продукту. Моделі життєвого циклу.	2	0,5
2.	Тема 2. Методології розробки програмного забезпечення. Agile. Scrum. Waterfall. Kanban. Організація командної роботи в проєкті.	2	0,5
3.	Тема 3. Аналіз вимог до програмного забезпечення. Збір та формалізація вимог замовника. Написання технічного завдання (ТЗ). Верифікація вимог.	2	0,5
4.	Тема 4. Архітектура програмних продуктів. Архітектурні моделі: монолітна, клієнт-серверна та мікросервісна. Принципи проєктування архітектури.	2	0,5
5.	Тема 5. Проєктування програмного забезпечення. Архітектурні патерни: види та призначення. Моделювання предметної області. Застосування UML-діаграм для моделювання (варіантів використання, класів, послідовностей, станів).	2	1
6.	Тема 6. Розробка та конструювання програмного забезпечення. Вибір мови програмування для реалізації програмного продукту. Фреймворки та бібліотеки: переваги від застосування. Поняття систем управління версіями. Управління версіями з Git.	2	0,5
7.	Тема 7. Побудова програмної системи. Повний цикл створення програмного продукту. Етапи конструювання, тестування та валідації вимог.	2	1

8	Тема 8. Розробка клієнт-серверних програмних продуктів. Використання багаторівневої архітектури. Організація серверної частини проєкту. Сховища даних та хмарні сервіси. Розробка мобільних додатків.	2	1
9.	Тема 9. Документація та забезпечення якості ПЗ. Види документації: технічна, користувацька. Методи тестування програмного забезпечення. Етика та правові аспекти щодо програмних продуктів. Авторські права та ліцензії.	2	0,5
Усього годин		18	6

3.2. Лабораторні заняття

№	Тема заняття	Кількість годин	
		ДФН	ЗФН
1.	Аналіз вимог до програмного забезпечення.	6	1
2.	Концепція створення програмного забезпечення.	6	1
3.	Моделювання предметної області з використанням UML.	6	1
4.	Проектування програмного забезпечення з використанням патернів.	6	1
5.	Управління проєктом за допомогою Git/GitHub.	6	1
6.	Реалізація проєкту з клієнт-серверною архітектурою.	6	1
Усього годин		36	6

3.3. Самостійна робота

№	Найменування робіт	Кількість годин	
		ДФН	ЗФН
1.	Тема 1. Вступ до технології створення програмного забезпечення. Переваги та недоліки моделей життєвого циклу ПЗ. Підготовка до лабораторної роботи 1.	6	9
2.	Опрацювання лекційного матеріалу теми 1. Тема 2. Методології розробки програмного забезпечення. Особливості організації командної роботи в проєкті. Підготовка до лабораторної роботи 1, 2.	6	9

3.	Опрацювання лекційного матеріалу теми 2. Тема 3. Аналіз вимог до програмного забезпечення. Використання матриць відстеження вимог. Підготовка до лабораторної роботи 1, 2.	6	10
4.	Опрацювання лекційного матеріалу теми 3. Тема 4. Архітектура програмних продуктів. Масштабованість та гнучкість як принципи проєктування архітектури. Підготовка до лабораторної роботи 2.	6	9
5.	Опрацювання лекційного матеріалу теми 4. Тема 5. Проєктування програмного забезпечення. Порівняльна характеристика UML та BPMN-діаграм. Підготовка до лабораторної роботи 3, 4.	6	9
6.	Підготовка до модуля 1 у вигляді тестування.	6	12
7.	Опрацювання лекційного матеріалу теми 5. Тема 6. Розробка та конструювання програмного забезпечення. Системи управління версіями: варіації, переваги, недоліки, тип інтерфейсу. Підготовка до лабораторної роботи 4, 5.	6	9
8.	Опрацювання лекційного матеріалу теми 6. Тема 7. Побудова програмної системи. Методи тестування та їх значення. Підготовка до лабораторної роботи 5, 6.	6	10
9.	Опрацювання лекційного матеріалу теми 7. Тема 8. Розробка клієнт-серверних програмних продуктів. Використання баз даних у клієнт-серверних системах: SQL vs NoSQL бази даних. Підготовка до лабораторної роботи 6.	6	10
10.	Опрацювання лекційного матеріалу тем 8. Тема 9. Документація та забезпечення якості ПЗ. Авторські права та ліцензії. Підготовка до лабораторної роботи 6.	6	9
11.	Підготовка до модуля 2 у вигляді тестування.	6	12
Усього годин		66	108

4. Критерії оцінювання результатів навчання студентів

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.

Захист звіту з лабораторної роботи оцінюється відповідною кількістю балів поданою в таблиці. Після проходження теоретичного матеріалу проводиться електронне тестування його засвоєння у вигляді проміжного (модульного) контролю.

Контроль здійснюється засобами електронного навчального курсу (ЕНК) на сервері дистанційного навчання <http://dl.tntu.edu.ua>, ідентифікатор дисципліни ID 6705.

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% 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 A, B, C, D, E, F, FX при цьому чотирибальна шкала оцінок (з записом семестрової оцінки «відмінно» - A, «добре» - B,C, «задовільно» - D, E відповідають підсумковому результату «зараховано», «незадовільно» - F, FX відповідає підсумковому результату «незараховано»).

Модуль 1			Модуль 2			Підсумковий контроль	Разом з дисципліни
Аудиторна та самостійна робота			Аудиторна та самостійна робота				
Теоретичний курс (тестування)	Лабораторна робота		Теоретичний курс (тестування)	Лабораторна робота			
17	21		16	21		25	100
№ тем	Вид робіт	бал	№ тем	Вид робіт	бал	за кожних три бали семестрової оцінки студент отримує 1 бал підсумкової семестрової оцінки автоматично	
Тема 1	Лаб. роб. №1	7	Тема 6	Лаб. роб. № 4	7		
Тема 2			Тема 7	Лаб. роб. №5	7		
Тема 3			Тема 8	Лаб. роб. №6	7		
Тема 4	Лаб. роб. №2	Тема 9					
Тема 5	Лаб. роб. №3	7					

5. Навчально-методичне забезпечення

1. Методичні вказівки для самостійної роботи студентів з дисципліни «Технологія створення програмних продуктів» для спеціальності 122 «Комп'ютерні науки».

2. Методичні вказівки для виконання лабораторних робіт з дисципліни «Технологія створення програмних продуктів» для студентів спеціальності 122 «Комп'ютерні науки».

3. Лекції в електронному навчальному курсі «Технологія створення програмних продуктів» для студентів напряму підготовки 122 «Комп'ютерні науки», ідентифікаційний номер якого ID 6705, <https://dl.tntu.edu.ua/>

6. Рекомендована література

Базова

1. Sommerville, I. (2021). Software Engineering (11th Edition). Pearson. ISBN-13: 9780133943030.

2. Sutherland, J. (2014). Scrum: The Art of Doing Twice the Work in Half the Time. Crown Business. ISBN-13: 9780385346450.

3. Pressman, R. S., & Maxim, B. R. (2020). Software Engineering: A Practitioner's Approach. McGraw-Hill Education. ISBN-13: 9781260548006.
4. Bass, L., Clements, P., & Kazman, R. (2021). Software Architecture in Practice (4th Edition). Addison-Wesley. ISBN-13: 9780136886099.
5. Martin, R. C. (2018). Clean Architecture: A Craftsman's Guide to Software Structure and Design. Pearson. ISBN-13: 9780134494166.
6. Трофименко О. Г. Основи програмної інженерії : навч.-метод. посібник / О. Г. Трофименко, С. Ю. Манаков, Д. Г. Ларін. – Одеса : Фенікс, 2022. – 197 с. – Режим доступу : <https://hdl.handle.net/11300/22773>.
7. ДСТУ ISO 9000:2007. Системи управління якістю. Основні положення та словник термінів. – К.: Держспоживстандарт, 2008. – [Чинний від 2008-01-01] – 35 с.– (Державний стандарт).
8. ДСТУ ISO 9001:2009. Системи управління якістю. Вимоги. [Текст]: – К.: Держспоживстандарт, 2009. – [Чинний від 2009-06-22] – 80 с.– (Державний стандарт).
9. Systems and software engineering – Software Life Cycle Processes. ISO 12207:2008. – [Чинний від 2008-02-01] – II, 122 с.– (Міжнародний стандарт).
10. IEEE Standard Glossary of Software Engineering Terminology, Глосарій. IEEE Std 610.12- 1990. – (Галузевий стандарт).

Допоміжна

1. Kruchten P. The 4+1 View of Architecture / Philippe Kruchten, IEEE Software, 12(6) Nov. 1995, pp. 45–50.
2. Leffingwell D. Managing Software Requirements: A Use Case Approach, 2nd Edition / Dean Leffingwell, Don Widrig - Addison_Wesley, 2003, ISBN 032112247X.
3. Meyer B. Object Oriented Software Construction / Bertrand Meyer, Prentice Hall, 1997, ISBN 0136291554.
4. Mellor S. Agile MDA / Stephen J Mellor, MDA Journal», June 2004. [Електронний ресурс] – Режим доступу: www.bptrends.com.

7. Інформаційні ресурси

1. Електронний навчальний курс «Технологія створення програмних продуктів» в системі дистанційного навчання ТНТУ імені І. Пулюя. [Електронний ресурс]. – Режим доступу: <https://dl.tntu.edu.ua/bounce.php?course=6705>
2. Agile Alliance. [Електронний ресурс]. – Режим доступу: <https://www.agilealliance.org/>
3. GitHub Documentation. [Електронний ресурс]. – Режим доступу: <https://docs.github.com/>
4. UML Resource Center. [Електронний ресурс]. – Режим доступу: <https://www.uml.org/>
5. Software Testing & QA Resources. [Електронний ресурс]. – Режим доступу: <https://www.softwaretestinghelp.com/>
6. IEEE Software Engineering Standards. [Електронний ресурс]. – Режим доступу: <https://standards.ieee.org/>
7. Youtube-канал “Programming with Mosh”. [Електронний ресурс]. – Режим доступу: <https://www.youtube.com/c/programmingwithmosh>

8. Зміни та доповнення до робочої програми навчальної дисципліни

№	Зміст внесених змін (доповнень)	Дата і № протоколу засідання кафедри	Примітки