



СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ТЕХНОЛОГІЯ СТВОРЕННЯ ПРОГРАМНИХ ПРОДУКТІВ

ID 6705

Шифр, назва спеціальності та освітній рівень	122 Комп'ютерні науки (бакалавр)	Назва освітньої програми	Комп'ютерні науки (2024) Computer science (2024)
Тип програми	Освітньо-професійна	Мова викладання	Українська
Факультет	Факультет комп'ютерно-інформаційних систем і програмної інженерії (ФІС)	Кафедра	Каф. комп'ютерних наук (КН)

Викладач/викладачі

Палка Олег Вікторович, доктор філософії, Асистент кафедри КН, [профіль на порталі "Науковці ТНТУ"](#)

Боднарчук Ігор Орестович, канд. техн. наук, доцент, Завідувач кафедри КН, [профіль на порталі "Науковці ТНТУ"](#)

Бревус Галина Богданівна, Асистент кафедри КН

Загальна інформація про дисципліну

Мета курсу	Отримання студентами теоретичних знань та практичних навичок у сфері сучасних підходів, методів і технологій проєктування, розробки, тестування та підтримки програмного забезпечення; розуміння життєвого циклу програмного продукту, методологій розробки, аналізу вимог, архітектурних моделей, управління версіями коду, тестування та документування.
Формат курсу	Змішаний – курс, що передбачає проведення лекцій, лабораторних та консультацій для кращого розуміння викладеного матеріалу і має супровід в електронному навчальному курсі системи A-Tutor, що має структуру, контент, завдання і систему оцінювання.
Компетентності ОП	<p>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.</p> <p>Інтегральна компетентність (ІК):</p> <p>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.</p> <p>Загальні компетентності (ЗК):</p> <p>ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.</p> <p>ЗК2. Здатність застосовувати знання у практичних ситуаціях.</p> <p>ЗК3. Знання та розуміння предметної області та розуміння професійної діяльності.</p> <p>ЗК6. Здатність вчитися й оволодівати сучасними знаннями.</p> <p>ЗК9. Здатність працювати в команді.</p> <p>ЗК11. Здатність приймати обґрунтовані рішення.</p> <p>ЗК12. Здатність оцінювати та забезпечувати якість виконуваних робіт.</p> <p>Спеціальні (фахові, предметні) компетентності (СК):</p> <p>СК8. Здатність проєктувати 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.</p> <p>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.</p>

	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.
Програмні результати навчання з ОП	<p>Вивчення навчальної дисципліни передбачає формування у студента таких програмних результатів навчання (ПР):</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>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).</p>
Обсяг курсу	<p>Очна (денна) форма здобуття освіти:</p> <p>Кількість кредитів ECTS — 4; лекції — 18 год.; лабораторні заняття — 36 год.; самостійна робота — 66 год.;</p> <p>Заочна форма здобуття освіти:</p> <p>Кількість кредитів ECTS — 4; лекції — 6 год.; лабораторні заняття — 6 год.; самостійна робота — 108 год.;</p>
Ознаки курсу	Рік навчання — 2; семестр — 4; Обов'язкова (для здобувачів інших ОП може бути вибірковою) дисципліна; кількість модулів — 2;
Форма контролю	<p>Поточний контроль: захист лабораторних робіт і два модульних тестування</p> <p>Підсумковий контроль: залік</p>
Компетентності та дисципліни, що є передумовою для вивчення	Дисципліни: Вступ до спеціальності, Програмування, Об'єктно-орієнтоване програмування, Архітектура комп'ютерних систем

Матеріально-технічне
та/або інформаційне
забезпечення

Комп'ютеризоване робоче місце з встановленим програмним забезпеченням для моделювання діаграм (Visual Paradigm) та проєктування ПЗ для налагодження практичних навичок. Середовище розробки та програмування (MS Visual Studio).
Проектор для презентування матеріалів з предмету.

СТРУКТУРА КУРСУ

Лекційний курс	Годин	
	<u>ОФЗО</u>	<u>ЗФЗО</u>
Лекція 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
Лекція 2. Методології розробки програмного забезпечення. Agile. Scrum. Waterfall. Kanban. Організація командної роботи в проєкті.	2	0,5
Лекція 3. Аналіз вимог до програмного забезпечення. Збір та формалізація вимог замовника. Написання технічного завдання (ТЗ). Верифікація вимог.	2	0,5
Лекція 4. Архітектура програмних продуктів. Архітектурні моделі: монолітна, клієнт-серверна та мікросервісна. Принципи проектування архітектури.	2	0,5
Лекція 5. Проектування програмного забезпечення. Архітектурні патерни: види та призначення. Моделювання предметної області. Застосування UML-діаграм для моделювання (варіантів використання, класів, послідовностей, станів).	2	1
Лекція 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
Лекція 7. Побудова програмної системи. Повний цикл створення програмного продукту. Етапи конструювання, тестування та валідації вимог.	2	1
Лекція 8. Розробка клієнт-серверних програмних продуктів. Використання багаторівневої архітектури. Організація серверної частини проєкту. Сховища даних та хмарні сервіси. Розробка мобільних додатків.	2	1
Лекція 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
РАЗОМ:	18	6

Годин

Теми занять, короткий зміст

Лабораторний практикум (теми)**ОФЗО** **ЗФЗО**

Лабораторна робота 1. Аналіз вимог до програмного забезпечення.

6

1

Лабораторна робота 2. Концепція створення програмного забезпечення.

6

1

Лабораторна робота 3. Моделювання предметної області з використанням UML.

6

1

Лабораторна робота 4. Проектування програмного забезпечення з використанням патернів.

6

1

Лабораторна робота 5. Управління проектом за допомогою Git/GitHub.

6

1

Лабораторна робота 6. Реалізація проекту з клієнт-серверною архітектурою.

6

1

РАЗОМ: 36 6

ІНШІ ВИДИ РОБІТ

Теми, короткий зміст

Самостійна робота студента включає опрацювання теоретичного матеріалу за допомогою рекомендованих літературних та Інтернет джерел по наступних темах:

Тема 1. Вступ до технології створення програмного забезпечення. Переваги та недоліки моделей життєвого циклу ПЗ.

Тема 2. Методології розробки програмного забезпечення. Особливості організації командної роботи в проєкті.

Тема 3. Аналіз вимог до програмного забезпечення. Використання матриць відстеження вимог.

Тема 4. Архітектура програмних продуктів. Масштабованість та гнучкість як принципи проєктування архітектури.

Тема 5. Проєктування програмного забезпечення. Порівняльна характеристика UML та BPMN-діаграм.

Тема 6. Розробка та конструювання програмного забезпечення. Системи управління версіями: варіації, переваги, недоліки, тип інтерфейсу.

Тема 7. Побудова програмної системи. Методи тестування та їх значення.

Тема 8. Розробка клієнт-серверних програмних продуктів. Використання баз даних у клієнт-серверних системах: SQL vs NoSQL бази даних.

Тема 9. Документація та забезпечення якості ПЗ. Авторські права та ліцензії.

Важливою частиною самостійної роботи є практичне застосування засобів моделювання при побудові UML-діаграм та середовища розробки (наприклад, Visual Studio) для виконання як пропонованих завдань лабораторних робіт так і зразків програмного коду, які містяться в літературних джерелах.

Крім того, до самостійної роботи також належать підготовка до захисту лабораторних робіт і підготовка до змістових модулів.

Інформаційні джерела для вивчення курсу

Рекомендована література:

Основна:

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).
11. Палка О., Станько А., Шимчук Г. та Герасимчук О. Запобігання поширення коронавірусної інфекції у «розумних містах». Комп'ютерно-інтегровані технології: освіта, наука, виробництво. Луцьк. 2021. № 42, ст. 79-88.
12. Палка О., Мацюк О., Дуда О., Пасічник В. та Кунанець Н. Комп'ютерна програма «Визначення розумності міста». А.с. про реєстрацію авторського права на твір № 101119. 2020.

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.

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>

Політики курсу

Політика контролю	Використовуються такі засоби оцінювання та методи демонстрування результатів навчання: поточне опитування; тестування; виконання індивідуальних завдань та презентацій; оцінювання результатів виконаних самостійних робіт; бесіди та обговорення проблемних питань; дискусії; індивідуальні консультації; екзамен. Можливий ректорський контроль.
Політика щодо консультування	Консультації при вивченні дисципліни проводяться згідно затвердженого на кафедрі . Консультування передбачено як очно , так і з використанням ресурсів електронного навчального курсу у середовищі електронного навчання університету.
Політика щодо перескладання	Студент має право на повторне складання модульного контролю з метою підвищення рейтингу протягом тижня після складання модульного контролю за графіком. Перескладання екзамену відбувається в терміни, визначені графіком освітнього процесу. Здобувач ВО має право на зарахування результатів навчання здобутих у неформальній чи інформальній освіті.
Політика щодо академічної доброчесності	При складанні усіх видів контролю у середовищі електронного навчання завжди активується система розпізнавання особи, що складає контроль. Усі практичні роботи у ЕНК перевіряються вбудованою системою Антиплагіат. При складанні усіх форм контролю забороняється списування, у тому числі з використанням сучасних інформаційних технологій.
Політика щодо відвідування	Відвідування занять є обов’язковим компонентом освітнього процесу. За наявності поважних причин (наприклад, хвороба, особливі потреби, відрадження, сімейні обставини, участь у програмах академічної мобільності тощо) навчання може здійснюватися за індивідуальним графіком, погодженим з деканом факультету.

СИСТЕМА ОЦІНЮВАННЯ

Розподіл балів, які отримують студенти за курс

Модуль 1			Модуль 2			Підсумковий контроль	Разом з дисципліни
Аудиторна та самостійна робота			Аудиторна та самостійна робота			Одна третя від суми балів, набраних здобувачем впродовж семестру	100
Теоретичний курс (тестування)	Лабораторна робота		Теоретичний курс (тестування)	Лабораторна робота			
17	21		16	21			
№ лекції	Види робіт	К-ть балів	№ лекції	Види робіт	К-ть балів		
Лекція №1			Лекція №6	Лабораторна робота №4	7		
Лекція №2	Лабораторна робота №1	7	Лекція №7	Лабораторна робота №5	7		
Лекція №3			Лекція №8	Лабораторна робота №6	7		
Лекція №4	Лабораторна робота №2	7	Лекція №9				
Лекція №5	Лабораторна робота №3	7					

Розподіл оцінок

Сума балів за навчальну діяльність	Шкала ECTS	Оцінка за національною шкалою
90-100	A	Зараховано
82-89	B	Зараховано
75-81	C	Зараховано
67-74	D	Зараховано
60-66	E	Зараховано
35-59	FX	Не зараховано
1-34	F	Не зараховано

Затверджено рішенням кафедри КН, протокол №1 від «26» серпня 2024 року.

ПОГОДЖЕНО

Гарант освітньої програми канд. техн. наук, доцент кафедри КН

Леся ДМИТРОЦА