МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЦИФРОВИХ, ОСВІТНІХ ТА СОЦІАЛЬНИХ ТЕХНОЛОГІЙ

ЗАТВЕРДЖУЮ
Декан факультету
_____ Галина ГЕРАСИМЧУК
«28» серпня 2026 р.

ЗАТВЕРДЖУЮ
Перший проректор
_____ Надія КОВАЛЬЧУК
«29» серпня 2026 р.

СИЛАБУС

з освітнього компонента «Технології створення програмних продуктів»
ступінь вищої освіти – бакалавр
галузь знань – 01 Освіта/Педагогіка
спеціальність – 014.09 Середня освіта (Інформатика)
освітня програма – Середня освіта. Інформатика

Кафедра цифрових освітніх технологій

Силабус склав(ла) доцент кафедри цифрових освітніх технологій, к.пед.н., доцент Юлія МЕЛЬНИЧУК

Силабус затверджений на засіданні кафедри цифрових освітніх технологій, протокол № 1а від «28» серпня 2026 р.

Завідувач кафедри к.пед.н., доцент Віталій КАБАК

Розглянуто та схвалено групою забезпечення освітньої програми «Середня освіта. Інформатика», протокол № 1а від «28» серпня 2026 р.

Гарант ОП «Середня освіта. Інформатика» к.пед.н., доцент Юлія МЕЛЬНИЧУК

Схвалено науково-методичною радою університету, протокол № 1/а від «29» серпня 2026 р.

Затверджений силабус онлайн:
https://e.lntu.edu.ua/syllabus/4128

Луцьк – 2026

СИЛАБУС ОСВІТНЬОГО КОМПОНЕНТА «Технології створення програмних продуктів»

Освітня програма	Середня освіта. Інформатика
Галузь знань / Спеціальність	01 Освіта/Педагогіка / 014.09 Середня освіта (Інформатика)
Рівень вищої освіти	перший (бакалаврський)
Статус дисципліни	вибіркова
Мова викладання	українська
Обсяг	5 кредитів ECTS / 150 годин
Форма контролю	залік
Викладач(і)	к.пед.н., доцент Юлія МЕЛЬНИЧУК

1. Мета і завдання дисципліни

Мета: Формування у майбутніх учителів інформатики теоретичних знань і практичних умінь зі створення програмних продуктів: аналізу вимог, моделювання, проєктування, розроблення, тестування, впровадження та супроводу програмного забезпечення. Розвиток здатності обирати й застосовувати сучасні технології та інструментальні засоби, працювати в команді, створювати програмні компоненти й освітні вебзастосунки з урахуванням потреб користувачів, вимог якості та завдань навчання інформатики.

Завдання: Основними завданнями вивчення дисципліни є:
Опанування основних понять програмної інженерії, моделей життєвого циклу та підходів до організації розроблення програмних продуктів.
Формування вмінь досліджувати предметну область, визначати зацікавлені сторони, збирати, аналізувати й документувати вимоги, зокрема до програмних засобів освітнього призначення.
Набуття навичок укладання технічного завдання, специфікації вимог і критеріїв приймання програмного продукту.
Опанування моделювання процесів, алгоритмів, поведінки та структури програмних систем засобами BPMN, DFD і UML.
Формування вмінь обґрунтовувати архітектурні рішення, застосовувати модульність, принципи SOLID і повторне використання компонентів.
Набуття навичок роботи із середовищами розроблення, бібліотеками, API та інструментами створення прототипів і вебзастосунків.
Опанування командної роботи із застосуванням Git, перевірки коду та основ безперервної інтеграції.
Формування вмінь розробляти тестові сценарії, виявляти й документувати дефекти, налагоджувати та оцінювати якість програмних продуктів.
Набуття навичок інтеграції, розгортання, документування, супроводу й удосконалення програмного забезпечення.
Розвиток здатності застосовувати технології створення програмних продуктів у професійній діяльності вчителя інформатики та добирати посильні завдання для навчальних програмних проєктів.

Пререквізити: Для опанування дисципліни необхідні базові знання з алгоритмізації та програмування, структур даних, об’єктно-орієнтованого програмування, основ баз даних і вебтехнологій; уміння розробляти та налагоджувати нескладні програми, працювати з файлами й використовувати середовище розроблення.

2. Компетентності

СК11 — Володіння методами інформаційного моделювання; здатність реалізовувати інформаційну модель засобами інформаційно-комунікаційних технологій; проводити комп’ютерний експеримент, інтерпретувати, аналізувати та узагальнювати його результати.

СК12 — Здатність до використання сучасних методів розробки та дослідження алгоритмів розв'язування задач у моделюванні об'єктів і процесів та реалізації цих алгоритмів сучасними мовами програмування.

СК13 — Здатність використовувати програмні засоби загального та спеціального призначення для розв’язання прикладних задач з інформатики.

СК21 — Здатність розробляти програмне забезпечення із застосуванням сучасних технологій об’єктно-орієнтованого, прикладного та web-програмування, організації баз даних та знань з відповідними моделями, методами та алгоритмами обчислень, структурами даних і механізмами управління.

СК22 — Здатність застосовувати в професійній діяльності сучасні веб-технології, основні підходи, методи і засоби веб-дизайну.

3. Структура дисципліни, розподіл годин, зв'язок тем з компетентностями

№	Назва теми	Лекції	Практ./ Семін./ лаб	Самост. робота	Разом	Методи навчання	Методи оцінювання	СК
Змістовий модуль 1. Теоретичні та методологічні основи створення програмних продуктів								
1	Тема 1. Програмні продукти та програмні системи: поняття, класифікація, характеристики	2	2	6	10	МН1, МН2, МН3, МН4, МН6	МО2, МО4, МО7	СК21
2	Тема 2. Життєвий цикл програмного забезпечення та моделі його реалізації	2	2	6	10	МН1, МН2, МН3, МН4, МН6	МО2, МО4, МО7	СК21
3	Тема 3. Аналіз предметної області та зацікавлених сторін програмного продукту	2	2	6	10	МН1, МН2, МН3, МН4, МН6	МО2, МО4, МО7	СК11, СК21
4	Тема 4. Інженерія вимог до програмного забезпечення	2	2	6	10	МН1, МН2, МН3, МН4, МН6	МО2, МО4, МО7	СК11, СК21
5	Тема 5. Технічне завдання та документування вимог до програмного продукту	2	2	6	10	МН1, МН2, МН3, МН4, МН6	МО2, МО4, МО7	СК11, СК13, СК21
6	Тема 6. Моделювання бізнес-процесів і потоків даних програмних систем	2	2	6	10	МН1, МН2, МН3, МН4, МН5, МН6	МО2, МО4, МО7	СК11, СК13

№	Назва теми	Лекції	Практ./ Семін./ лаб	Самост. робота	Разом	Методи навчання	Методи оцінювання	СК
7	Тема 7. Функціональне моделювання програмних систем засобами UML	2	2	6	10	МН1, МН2, МН3, МН4, МН5, МН6	МО2, МО4, МО7, Модульна контрольна робота	СК11, СК13
Змістовий модуль 2. Проєктування, реалізація та забезпечення якості програмних продуктів								
8	Тема 8. Моделювання поведінки програмних систем: діаграми діяльності та станів	2	2	6	10	МН1, МН2, МН3, МН4, МН5, МН6	МО2, МО4, МО7	СК11, СК12, СК13
9	Тема 9. Моделювання взаємодії та структури компонентів програмних систем	2	2	6	10	МН1, МН2, МН3, МН4, МН5, МН6	МО2, МО4, МО7	СК11, СК21
10	Тема 10. Архітектура програмного забезпечення та принципи її проєктування	2	2	6	10	МН1, МН2, МН3, МН4, МН6	МО2, МО4, МО7	СК11, СК21
11	Тема 11. Технології та інструментальні засоби розроблення програмних продуктів	2	2	6	10	МН1, МН2, МН3, МН4, МН5, МН6	МО2, МО4, МО7	СК12, СК13, СК21, СК22
12	Тема 12. Організація командної розробки та управління версіями програмного забезпечення	2	2	6	10	МН1, МН2, МН3, МН4, МН5, МН6	МО2, МО4, МО5, МО7	СК13, СК21
13	Тема 13. Тестування, налагодження та забезпечення якості програмних продуктів	2	2	6	10	МН1, МН2, МН3, МН4, МН5, МН6	МО2, МО4, МО7	СК12, СК13, СК21

№	Назва теми	Лекції	Практ./ Семін./ лаб	Самост. робота	Разом	Методи навчання	Методи оцінювання	СК
14	Тема 14. Інтеграція, розгортання та впровадження програмного забезпечення	2	2	6	10	МН1, МН2, МН3, МН4, МН5, МН6	МО2, МО4, МО7	СК13, СК21, СК22
15	Тема 15. Супровід, модернізація та розвиток програмних продуктів	2	2	6	10	МН1, МН2, МН3, МН4, МН5, МН6	МО2, МО4, МО7, Модульна контрольна робота	СК13, СК21
	Разом	**30**	**30**	**90**	**150**			

Легенда: СК — спеціальна (фахова) компетентність (номер за п. 2). Повне формулювання кожного коду наводиться в ОП / стандарті спеціальності.

Методи навчання: МН1 — словесний метод (лекція, дискусія, співбесіда тощо); МН2 — практичний метод (лабораторні та практичні заняття); МН3 — наочний метод (метод ілюстрацій і метод демонстрацій); МН4 — робота з навчально-методичною літературою (конспектування, тезування, анотування, рецензування, складання реферату); МН5 — відеометод у сполученні з новітніми інформаційними технологіями та комп'ютерними засобами навчання (дистанційні, мультимедійні, веб-орієнтовані тощо); МН6 — самостійна робота (розв'язання завдань).

Методи оцінювання: МО2 — усне або письмове опитування; МО4 — тестування; МО5 — командні проєкти; МО7 — презентації результатів виконаних завдань та досліджень.

4. Результати навчання

Знати:

– Основні поняття, характеристики та класифікацію програмних продуктів, зокрема програмних засобів освітнього призначення.

– Моделі життєвого циклу програмного забезпечення, підходи Agile, Scrum, Kanban, RUP і засади DevOps.

– Методи дослідження предметної області, визначення зацікавлених сторін та виявлення потреб користувачів, зокрема вчителів і учнів.

– Види вимог, способи їх специфікації, пріоритизації, перевірки й управління змінами; структуру технічного завдання та SRS.

– Принципи моделювання процесів і потоків даних у нотаціях BPMN та DFD.

– Призначення та основні елементи діаграм UML: варіантів використання, діяльності, станів, послідовності, класів і компонентів.

– Архітектурні стилі, принципи модульності, SOLID, повторного використання компонентів і вибору технологічних рішень.

– Можливості середовищ розроблення, бібліотек, фреймворків, API, засобів збирання й керування залежностями.

– Основи створення користувацьких інтерфейсів та освітніх вебзастосунків з урахуванням зручності, доступності й навчальних завдань.

– Принципи командної розробки, контролю версій Git, перевірки коду й безперервної інтеграції.

– Рівні та методи тестування, правила підготовки тест-кейсів, документування дефектів і засоби налагодження та автоматизації тестування.

– Основні підходи до інтеграції, контейнеризації, розгортання, CI/CD та підготовки користувацької документації.

– Види супроводу програмного забезпечення, принципи рефакторингу, аналізу зворотного зв’язку та планування розвитку продукту.

Вміти:

– Аналізувати предметну область, визначати користувачів, межі системи та цілі створення програмного продукту.

– Формулювати функціональні й нефункціональні вимоги, користувацькі історії та критерії прийнятності з урахуванням освітніх потреб.

– Укладати технічне завдання й специфікацію вимог, забезпечувати їх узгодженість, перевірюваність і простежуваність.

– Будувати та аналізувати моделі процесів і потоків даних у нотаціях BPMN і DFD.

– Створювати узгоджені UML-діаграми для опису функцій, алгоритмів, поведінки, взаємодії та структури програмної системи.

– Обґрунтовувати архітектуру, декомпозицію, інтерфейси компонентів і вибір технологій розроблення.

– Використовувати середовище розроблення для реалізації й налагодження програмних компонентів та алгоритмів.

– Створювати прототип інтерфейсу й нескладний освітній вебзастосунок з урахуванням зручності та доступності для користувачів.

– Підключати бібліотеки й API, керувати залежностями та документувати програмний код.

– Організовувати командну роботу в Git: створювати коміти й гілки, зливати зміни, усувати конфлікти та брати участь у перевірці коду.

– Розробляти й виконувати тест-кейси та нескладні автоматизовані тести, документувати дефекти й перевіряти відповідність продукту вимогам.

– Інтегрувати компоненти, розгортати навчальний програмний продукт і готувати інструкцію користувача.

– Аналізувати відгуки користувачів, пропонувати зміни, виконувати рефакторинг та планувати супровід програмного продукту.

– Добирати приклади й завдання зі створення програмних продуктів для навчальних проєктів з інформатики відповідно до рівня підготовки учнів.

5. Неформальна освіта

Рекомендовано практичний курс GitHub Skills «Introduction to GitHub» (https://github.com/skills/introduction-to-github) для опанування роботи з репозиторієм, гілками, комітами та запитами на злиття за темою 12. Також можуть бути розглянуті результати тематичних курсів із вимог, UML, розроблення та тестування програмного забезпечення. Визнання результатів неформальної та інформальної освіти здійснюється відповідно до процедур ЛНТУ після зіставлення досягнутих результатів зі змістом дисципліни. Підтвердженнями можуть бути сертифікат, репозиторій, портфоліо або інші матеріали виконаних завдань; за потреби проводиться перевірка набутих умінь.

6. Критерії та форми контролю

Оцінювання — за 100-бальною шкалою (національна шкала / ЄКТС). Підсумковий бал = поточний контроль (50 %) + модульний контроль (50 %).

Академічна доброчесність: самостійне виконання завдань; коректне посилання на джерела; позначення частин тексту, згенерованих засобами ШІ. Порушення академічної доброчесності не толеруються.

7. Зміст тем

Тема 1. Програмні продукти та програмні системи: поняття, класифікація, характеристики

Поняття програмного продукту, програмного забезпечення та програмної системи. Основні

характеристики, властивості та класифікація програмних продуктів. Структура програмних систем, їхні функціональні та нефункціональні складові. Програмний продукт як результат інженерної діяльності. Системний підхід до розроблення програмного забезпечення. Межі програмної системи та її взаємодія із зовнішнім середовищем. Основні положення програмної інженерії та структура галузі знань SWEBOK. Сучасні тенденції розвитку технологій створення програмних продуктів. Програмні засоби освітнього призначення: навчальні тренажери, тестові системи та освітні вебзастосунки. Їх використання у професійній діяльності вчителя інформатики.

Тема 2. Життєвий цикл програмного забезпечення та моделі його реалізації

Поняття життєвого циклу програмного забезпечення. Основні процеси та етапи створення програмних продуктів. Моделі життєвого циклу: каскадна, V-подібна, ітераційна, інкрементна, спіральна. Гнучкі методології розроблення програмного забезпечення: Agile, Scrum, Kanban. Методологія Rational Unified Process (RUP). Концепція DevOps та її роль у процесах розроблення й експлуатації програмного забезпечення. Критерії вибору моделі життєвого циклу залежно від особливостей програмного продукту, вимог замовника та умов реалізації проєкту.

Тема 3. Аналіз предметної області та зацікавлених сторін програмного продукту

Поняття предметної області програмної системи та методи її дослідження. Ідентифікація й класифікація зацікавлених сторін програмного продукту. Аналіз потреб, очікувань та інтересів користувачів і замовників. Джерела інформації для визначення вимог до програмних систем. Методи збирання інформації: інтерв'ювання, анкетування, спостереження, аналіз документації, мозковий штурм. Моделювання поточного та бажаного стану процесів за підходами AS-IS і TO-BE. Визначення проблем предметної області та формування цілей розроблення програмного продукту. Аналіз освітньої предметної області: потреби вчителя й учнів, навчальні цілі та сценарії використання програмного засобу.

Тема 4. Інженерія вимог до програмного забезпечення

Сутність та основні процеси інженерії вимог. Класифікація вимог до програмного забезпечення. Бізнес-вимоги, користувацькі та системні вимоги. Функціональні й нефункціональні вимоги до програмного продукту. Методи формулювання та специфікації вимог. Користувацькі історії (User Stories), сценарії використання та критерії прийнятності. Аналіз, узгодження, перевірка й валідація вимог. Методи пріоритизації вимог. Управління змінами вимог протягом життєвого циклу програмного забезпечення. Формулювання вимог до освітнього застосунку: відповідність навчальній меті, зручність інтерфейсу, доступність, коректність результатів і захист даних користувачів.

Тема 5. Технічне завдання та документування вимог до програмного продукту

Призначення та роль технічного завдання у процесі створення програмного продукту. Основні складові технічного завдання та вимоги до його оформлення. Документування функціональних, нефункціональних і технічних вимог. Специфікація вимог до програмного забезпечення (SRS). Формування критеріїв приймання програмного продукту. Забезпечення повноти, однозначності, узгодженості та перевірюваності вимог. Простежуваність вимог і матриця їх відповідності результатам проєктування та тестування. Організація погодження й актуалізації проєктної документації.

Тема 6. Моделювання бізнес-процесів і потоків даних програмних систем

Призначення моделювання бізнес-процесів під час аналізу та проєктування програмних систем. Поняття процесу, функції, події, виконавця та інформаційного потоку. Основні підходи до опису й моделювання процесів предметної області. Нотація BPMN та її базові елементи. Діаграми потоків даних (DFD): процеси, зовнішні сутності, сховища даних і потоки інформації. Контекстні діаграми та декомпозиція процесів. Побудова моделей інформаційної взаємодії компонентів системи. Аналіз і вдосконалення процесів на основі побудованих моделей. Моделювання процесів виконання навчального завдання та інформаційних потоків освітнього застосунку.

Тема 7. Функціональне моделювання програмних систем засобами UML

Призначення уніфікованої мови моделювання UML у проєктуванні програмного забезпечення.

Основні поняття, принципи та класифікація UML-діаграм. Функціональне представлення програмної системи. Діаграми варіантів використання (Use Case Diagram). Актори, варіанти використання та зв'язки між ними. Визначення меж системи та її функціональних можливостей. Текстовий опис сценаріїв використання, основні та альтернативні потоки подій. Узгодження функціональних моделей із вимогами до програмного продукту.

Тема 8. Моделювання поведінки програмних систем: діаграми діяльності та станів

Призначення поведінкового моделювання програмних систем. Діаграми діяльності UML: дії, переходи, розгалуження, паралельні потоки виконання та розподіл відповідальності. Моделювання алгоритмів і робочих процесів програмного продукту. Діаграми станів: стани об'єктів, події, переходи та умови їх виконання. Визначення життєвого циклу об'єктів програмної системи. Узгодження моделей діяльності та станів із функціональними вимогами. Використання поведінкових моделей для аналізу логіки функціонування програмного забезпечення.

Тема 9. Моделювання взаємодії та структури компонентів програмних систем

Призначення моделей взаємодії та структури програмних систем. Діаграми послідовності UML: об'єкти, лінії життя, повідомлення та порядок їх виконання. Моделювання сценаріїв взаємодії між компонентами програмного продукту. Діаграми класів: класи, атрибути, операції та зв'язки. Асоціація, агрегація, композиція, наслідування та залежності. Діаграми компонентів, інтерфейси та залежності між програмними модулями. Узгодження моделей взаємодії зі структурними моделями програмної системи.

Тема 10. Архітектура програмного забезпечення та принципи її проєктування

Поняття архітектури програмного забезпечення та її роль у забезпеченні якості програмних продуктів. Основні архітектурні принципи: модульність, інкапсуляція, розділення відповідальності, слабка зв'язаність і висока згуртованість компонентів. Архітектурні стилі та підходи: монолітна, багаторівнева, клієнт-серверна, мікросервісна та подієво-орієнтована архітектури. Принципи SOLID і повторного використання програмних компонентів. Основні шаблони проєктування програмного забезпечення. Вибір архітектурних рішень відповідно до функціональних і нефункціональних вимог.

Тема 11. Технології та інструментальні засоби розроблення програмних продуктів

Класифікація сучасних технологій та інструментальних засобів розроблення програмного забезпечення. Мови програмування, фреймворки та бібліотеки як складові технологічного середовища. Інтегровані середовища розроблення (IDE), редактори коду та засоби автоматизації. Організація структури програмного проєкту. Принципи модульного програмування, повторного використання коду та керування залежностями. Засоби збирання й виконання програмного продукту. Використання програмних інтерфейсів API та інтеграція зовнішніх бібліотек. Документування програмного коду та застосування стандартів його оформлення. Прототипування користувацького інтерфейсу та реалізація нескладного освітнього вебзастосунку або його компонента. Добір елементів вебдизайну з урахуванням зручності й доступності навчального ресурсу.

Тема 12. Організація командної розробки та управління версіями програмного забезпечення

Принципи організації командної розробки програмних продуктів. Ролі та відповідальність учасників команди розроблення. Планування завдань, розподіл робіт і контроль виконання проєкту. Системи контролю версій та їх призначення. Основні поняття Git: репозиторій, коміт, гілка, злиття та конфлікти змін. Організація спільної роботи з використанням платформ GitHub або GitLab. Стратегії розгалуження та інтеграції програмного коду. Перевірка коду (Code Review), управління задачами та документування змін. Основи безперервної інтеграції (CI). Використання репозиторію для спільного навчального програмного проєкту; добір посильних завдань і розподіл ролей учасників.

Тема 13. Тестування, налагодження та забезпечення якості програмних продуктів

Поняття якості програмного забезпечення та основні характеристики якісного програмного продукту. Мета, завдання та принципи тестування програмного забезпечення. Рівні тестування:

модульне, інтеграційне, системне та приймальне. Функціональне й нефункціональне тестування. Методи тестування «чорної» та «білої» скриньки. Розроблення тестових сценаріїв і тест-кейсів. Виявлення, документування та класифікація дефектів. Засоби налагодження програмного коду. Автоматизація тестування. Перевірка відповідності програмного продукту встановленим вимогам і критеріям якості. Перевірка правильності роботи освітнього застосунку, оброблення помилкового введення, зручності й доступності інтерфейсу відповідно до критеріїв приймання.

Тема 14. Інтеграція, розгортання та впровадження програмного забезпечення

Призначення та основні етапи інтеграції програмних компонентів. Організація взаємодії модулів і підсистем програмного продукту. Використання API та засобів обміну даними між програмними компонентами. Підготовка програмного забезпечення до розгортання. Середовища розроблення, тестування та експлуатації. Основи контейнеризації програмних застосунків. Автоматизація процесів збирання, тестування та розгортання програмного забезпечення (CI/CD). Організація впровадження програмного продукту. Перевірка працездатності, налаштування середовища та підготовка користувацької документації. Розгортання навчального вебзастосунку, перевірка його доступності та підготовка інструкції для вчителя й учнів.

Тема 15. Супровід, модернізація та розвиток програмних продуктів

Поняття супроводу програмного забезпечення та його місце в життєвому циклі програмного продукту. Основні види супроводу: коригувальний, адаптивний, удосконалювальний і профілактичний. Виявлення й усунення помилок у процесі експлуатації програмних систем. Управління змінами, оновленнями та версіями програмного забезпечення. Аналіз зворотного зв'язку користувачів і формування вимог до вдосконалення програмного продукту. Рефакторинг, модернізація та оптимізація програмного коду. Забезпечення сумісності, надійності й підтримуваності програмного забезпечення. Планування подальшого розвитку програмного продукту. Аналіз відгуків учителів і учнів для вдосконалення освітнього програмного продукту.

8. Опис ІЗ/КПІЗ

Не передбачено навчальним планом.

9. Загальні вимоги

Освітній процес регулюється академічними правилами ЛНТУ. Здобувачі освіти зобов’язані ознайомитися з вимогами, визначеними Кодексом академічної доброчесності ЛНТУ та Політикою етичного використання штучного інтелекту в ЛНТУ, і дотримуватися принципів академічної доброчесності та етичного використання штучного інтелекту в освітній діяльності.

Академічна доброчесність. Учасники освітнього процесу зобов’язані дотримуватися Закону України «Про академічну доброчесність» від 18.12.2025 № 4742-IX та внутрішніх нормативних документів ЛНТУ. Не допускаються академічний плагіат, фабрикація, фальсифікація, привласнення авторства, несамостійне виконання завдань, а також недоброчесне використання результатів, згенерованих штучним інтелектом.

Використання штучного інтелекту. Інструменти штучного інтелекту можуть використовуватися для пошуку ідей, пояснення методів, попереднього структурування, мовного редагування та допоміжного аналізу неконфіденційних даних, якщо конкретне завдання не передбачає повністю самостійного виконання. Здобувач освіти має зазначати використаний інструмент, мету та етап його застосування, перевіряти факти й розрахунки та самостійно обґрунтовувати висновки.

10. Література

І. Навчально-методичні матеріали та розробки

Праці автора силабуса:

1. Емпіричні методи програмної інженерії : методичні вказівки до лабораторних робіт / уклад. Ю. А. Лук’янчук, Ю. Є. Мельничук. Луцьк : Луцький НТУ, 2021. 44 с.

2. Алгоритмізація та програмування з методикою навчання : методичні вказівки до виконання лабораторних робіт / уклад. Ю. Є. Мельничук. Луцьк : ЛНТУ, 2025. 52 с.

3. Веб-технології та веб-дизайн : методичні вказівки до виконання лабораторних робіт / уклад. Ю. Є. Мельничук. Луцьк : Луцький НТУ, 2022. 44 с.

Праці інших авторів:

4. Khononov V. Learning Domain-Driven Design: Aligning Software Architecture and Business Strategy. Sebastopol : O'Reilly Media, 2021. 342 p.

5. Ford N., Richards M., Sadalage P., Dehghani Z. Software Architecture: The Hard Parts: Modern Trade-Off Analyses for Distributed Architectures. Sebastopol : O'Reilly Media, 2021. 462 p.

6. Guide to the Software Engineering Body of Knowledge (SWEBOK Guide). Version 4.0 / ed. H. Washizaki. IEEE Computer Society, 2024.

II. Наукові праці

Праці автора силабуса:

7. Мельничук Ю. Є. Розвиток алгоритмів створення веб-орієнтованих інформаційних систем. Наука і техніка сьогодні. 2023. № 2(16). С. 392–400. DOI: https://doi.org/10.52058/2786-6025-2023-2(16)-392-400.

8. Мельничук Ю. Є. Алгоритми функціонування інформаційних систем освітнього призначення. Наука і техніка сьогодні. 2023. № 3(17). С. 576–585. DOI: https://doi.org/10.52058/2786-6025-2023-3(17)-576-585.

9. Мельничук Ю. Є. Принципи побудови інформаційних систем освітнього призначення. Комп'ютерно-інтегровані технології: освіта, наука, виробництво. 2023. № 50. С. 77–84. DOI: https://doi.org/10.36910/6775-2524-0560-2023-50-11.

10. Мельничук Ю. Є., Редько О. І., Сушик О. Г. Інтерфейси інформаційних систем у професійній підготовці майбутніх фахівців з інформатики. Наука і техніка сьогодні. 2025. № 8(49). С. 829–840. DOI: https://doi.org/10.52058/2786-6025-2025-8(49)-829-840.

11. Карпишин Р., Мельничук Ю. Використання технологій штучного інтелекту для автоматизації процесів тестування програмного забезпечення. Волинська весна: перші паростки науки : збірник матеріалів X Всеукраїнської студентської науково-практичної онлайн-конференції (Луцьк, 11 березня 2026 р.). Луцьк, 2026. С. 680–681.

12. Квасний О., Мельничук Ю. Практичні аспекти використання Selenium WebDriver для автоматизації тестування вебзастосунків. Волинська весна: перші паростки науки : матеріали X Всеукраїнської студентської науково-практичної онлайн-конференції (Луцьк, 11 березня 2026 р.). Луцьк, 2026. С. 682–683.

13. Левковець О., Мельничук Ю. Автоматизація процесів тестування реєстрації та генерації API-ключів засобами Python. Волинська весна: перші паростки науки : матеріали X Всеукраїнської студентської науково-практичної онлайн-конференції (Луцьк, 11 березня 2026 р.). Луцьк, 2026. С. 696–697.

14. Шолох Н., Мельничук Ю. Проблеми та підходи до тестування хмарних сервісів. Волинська весна: перші паростки науки : матеріали X Всеукраїнської студентської науково-практичної онлайн-конференції (Луцьк, 11 березня 2026 р.). Луцьк, 2026. С. 770–772.

Праці інших авторів:

15. Forsgren N., Storey M.-A., Maddila C., Zimmermann T., Houck B., Butler J. The SPACE of Developer Productivity: There's More to It Than You Think. ACM Queue. 2021. Vol. 19, No. 1. P. 20–48. DOI: https://doi.org/10.1145/3454122.3454124.

16. Umar M. A., Lano K. Advances in automated support for requirements engineering: a systematic literature review. Requirements Engineering. 2024. Vol. 29. P. 177–207. DOI: https://doi.org/10.1007/s00766-023-00411-0.

17. Necula S.-C., Dumitriu F., Greavu-Şerban V. A Systematic Literature Review on Using Natural Language Processing in Software Requirements Engineering. Electronics. 2024. Vol. 13, No. 11. Art. 2055. DOI: https://doi.org/10.3390/electronics13112055.

18. Hou X. et al. Large Language Models for Software Engineering: A Systematic Literature Review. ACM Transactions on Software Engineering and Methodology. 2024. Vol. 33, No. 8. Art. 220. DOI:

https://doi.org/10.1145/3695988.