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 – 015.39 Професійна освіта (Цифрові технології)
освітня програма – Професійна освіта (комп`ютерні технології)

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 «Професійна освіта (комп`ютерні технології)», протокол № 1а від «28» серпня 2026 р.

Гарант ОП «Професійна освіта (комп`ютерні технології)» к.пед.н., доцент Віталій КАБАК

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

Затверджений силабус онлайн:
https://e.lntu.edu.ua/syllabus/4124

Луцьк – 2026

СИЛАБУС ОСВІТНЬОГО КОМПОНЕНТА «Технології створення програмних продуктів»

Освітня програма	Професійна освіта (комп`ютерні технології)
Галузь знань / Спеціальність	01 Освіта/Педагогіка / 015.39 Професійна освіта (Цифрові технології)
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 здобувачів вищої освіти системи теоретичних знань і практичних умінь щодо технологій, методів та інструментальних засобів створення програмних продуктів, набуття здатності здійснювати аналіз вимог, проєктування, розроблення, тестування, впровадження та супровід програмного забезпечення з урахуванням сучасних підходів до організації процесів розроблення, вимог якості та потреб користувачів, зокрема під час створення компонентів електронних освітніх ресурсів.

Завдання: Основними завданнями вивчення дисципліни є:
- ознайомлення з основними поняттями, принципами та технологіями створення програмних продуктів;
- вивчення моделей життєвого циклу програмного забезпечення, методологій та підходів до організації процесів його розроблення;
- формування вмінь аналізувати предметну область, визначати зацікавлені сторони, виявляти, формулювати та документувати вимоги до програмних продуктів;
- набуття практичних навичок розроблення технічного завдання та проєктної документації;
- опанування методів моделювання бізнес-процесів, функціональної поведінки та структури програмних систем із використанням сучасних нотацій і засобів моделювання;
- формування вмінь обґрунтовувати архітектурні та технологічні рішення, застосовувати принципи модульності й повторного використання програмних компонентів;
- набуття навичок використання середовищ розроблення, систем контролю версій та інструментальних засобів створення програмного забезпечення;
- опанування основних методів тестування, налагодження, забезпечення якості, впровадження та супроводу програмних продуктів;
- розвиток здатності до самостійного та командного виконання завдань зі створення програмних продуктів, оцінювання результатів розроблення та підготовки відповідної документації.

Пререквізити: Алгоритмізація і програмування з методикою навчання, Web Technologies and Web Design, Програмування на Python, Організація баз даних і знань, Об'єктно-орієнтоване програмування.

2. Компетентності

СК17 — Здатність розробляти програмне забезпечення із застосуванням сучасних технологій об’єктно-орієнтованого, прикладного та web-програмування з відповідними моделями, методами та алгоритмами обчислень, структурами даних і механізмами управління.

СК18 — Здатність до алгоритмічного та логічного мислення, приймати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.

СК20 — Здатність розробляти архітектури, модулі та компоненти програмних систем,

застосовувати методи, засоби і принципи проектування та створення електронних освітніх ресурсів.

СК16 — Здатність до системного аналізу технічних і педагогічних систем, процесів та ситуацій, вивчення передового виробничого та педагогічного досвіду, впровадження досягнень вітчизняної й зарубіжної науки і техніки.

СК14 — Здатність збирати, аналізувати та інтерпретувати інформацію (дані) відповідно до спеціалізації.

СК08 — Здатність використовувати відповідне програмне забезпечення для вирішення професійних завдань, відповідно до спеціалізації.

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

№	Назва теми	**Лекції** денна/ заочна	**Практ./ Семін./ лаб** денна/ заочна	**Самост. робота** денна/ заочна	**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	2/0	2/0	6/10	10/10	МН1, МН2, МН3, МН4, МН6	МО2, МО4, МО7	СК16
2	Тема 2. Життєвий цикл програмного забезпечення та моделі його реалізації	2/2	2/0	6/8	10/10	МН1, МН2, МН3, МН4, МН6	МО2, МО4, МО7	СК18, СК16
3	Тема 3. Аналіз предметної області та зацікавлених сторін програмного продукту	2/0	2/0	6/10	10/10	МН1, МН2, МН3, МН4, МН6	МО2, МО4, МО7	СК16, СК14
4	Тема 4. Інженерія вимог до програмного забезпечення	2/2	2/0	6/8	10/10	МН1, МН2, МН3, МН4, МН6	МО2, МО4, МО7	СК18, СК14
5	Тема 5. Технічне завдання та документування вимог до програмного продукту	2/0	2/0	6/10	10/10	МН1, МН2, МН3, МН4, МН6	МО2, МО4, МО7	СК18, СК14
6	Тема 6. Моделювання бізнес-процесів і потоків даних програмних систем	2/0	2/0	6/10	10/10	МН1, МН2, МН3, МН4, МН5, МН6	МО2, МО4, МО7	СК18, СК16, СК08

№	Назва теми	Лекції денна/ заочна	Практ./ Семін./ лаб денна/ заочна	Самост. робота денна/ заочна	Разом	Методи навчання	Методи оцінювання	СК
7	Тема 7. Функціональне моделювання програмних систем засобами UML	2/0	2/2	6/8	10/10	МН1, МН2, МН3, МН4, МН5, МН6	МО2, МО4, МО7, Модульна контрольна робота	СК18, СК08
Змістовий модуль 2. Проєктування, реалізація та забезпечення якості програмних продуктів								
8	Тема 8. Моделювання поведінки програмних систем: діаграми діяльності та станів	2/0	2/0	6/10	10/10	МН1, МН2, МН3, МН4, МН5, МН6	МО2, МО4, МО7	СК18, СК20, СК08
9	Тема 9. Моделювання взаємодії та структури компонентів програмних систем	2/0	2/0	6/10	10/10	МН1, МН2, МН3, МН4, МН5, МН6	МО2, МО4, МО7	СК18, СК20, СК08
10	Тема 10. Архітектура програмного забезпечення та принципи її проєктування	2/2	2/0	6/8	10/10	МН1, МН2, МН3, МН4, МН6	МО2, МО4, МО7	СК20, СК16
11	Тема 11. Технології та інструментальні засоби розроблення програмних продуктів	2/0	2/2	6/8	10/10	МН1, МН2, МН3, МН4, МН5, МН6	МО2, МО4, МО7	СК17, СК08
12	Тема 12. Організація командної розробки та управління версіями програмного забезпечення	2/0	2/0	6/10	10/10	МН1, МН2, МН3, МН4, МН5, МН6	МО2, МО4, МО5, МО7	СК17, СК08
13	Тема 13. Тестування, налагодження та забезпечення якості програмних продуктів	2/0	2/2	6/8	10/10	МН1, МН2, МН3, МН4, МН5, МН6	МО2, МО4, МО7	СК17, СК14, СК08

№	Назва теми	**Лекції** денна/ заочна	**Практ./ Семін./ лаб** денна/ заочна	**Самост. робота** денна/ заочна	**Разом**	**Методи навчання**	**Методи оцінювання**	**СК**
14	Тема 14. Інтеграція, розгортання та впровадження програмного забезпечення	2/0	2/0	6/10	10/10	МН1, МН2, МН3, МН4, МН5, МН6	МО2, МО4, МО7	СК17, СК20, СК08
15	Тема 15. Супровід, модернізація та розвиток програмних продуктів	2/0	2/0	6/10	10/10	МН1, МН2, МН3, МН4, МН5, МН6	МО2, МО4, МО7, Модульна контрольна робота	СК17, СК20, СК16, СК08
	Разом	**30/6**	**30/6**	**90/138**	**150/150**			

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:

– поняття, класифікацію та основні характеристики програмних продуктів і програмних систем;

– процеси життєвого циклу програмного забезпечення, особливості каскадної, ітераційної, інкрементної, спіральної та V-подібної моделей, підходів Agile, Scrum, Kanban, RUP і DevOps;

– методи аналізу предметної області, визначення зацікавлених сторін, виявлення, класифікації, пріоритизації та перевірки вимог;

– призначення й структуру технічного завдання та специфікації вимог, принципи простежуваності й управління змінами вимог;

– основні елементи BPMN і DFD та правила моделювання процесів і потоків даних;

– призначення та основні елементи UML-діаграм варіантів використання, діяльності, станів, послідовності, класів і компонентів;

– архітектурні стилі, принципи модульності, інкапсуляції, розділення відповідальності, SOLID та повторного використання компонентів;

– призначення середовищ розроблення, фреймворків, бібліотек, API, засобів збирання програм і керування залежностями;

– принципи організації командної розробки, контролю версій у Git, перевірки коду та безперервної інтеграції;

– характеристики якості програмного забезпечення, рівні й види тестування, принципи розроблення тест-кейсів та документування дефектів;

– етапи інтеграції, контейнеризації, розгортання й упровадження програмних продуктів, призначення конвеєра CI/CD та користувацької документації;

– види супроводу, підходи до рефакторингу, модернізації й розвитку програмного забезпечення

на основі зворотного зв’язку користувачів;

– особливості застосування методів проєктування програмних систем під час створення компонентів електронних освітніх ресурсів.

Вміти:

– аналізувати предметну область, визначати зацікавлені сторони, потреби користувачів і цілі створення програмного продукту;

– формулювати функціональні й нефункціональні вимоги, користувацькі історії та перевірювані критерії прийнятності;

– розробляти технічне завдання, фрагмент специфікації вимог і матрицю простежуваності вимог до проєктних рішень і тестів;

– будувати контекстні діаграми й моделі процесів і потоків даних у нотаціях BPMN та DFD;

– створювати узгоджені UML-діаграми варіантів використання, діяльності, станів, послідовності, класів і компонентів відповідно до вимог навчального програмного проєкту;

– обґрунтовувати вибір архітектурного стилю, декомпонувати програмну систему на модулі й компоненти та визначати їхні інтерфейси;

– налаштовувати середовище розроблення й реалізовувати прототип програмного продукту або компонента електронного освітнього ресурсу з дотриманням принципів модульності;

– використовувати бібліотеки й API, керувати залежностями та документувати програмний код;

– організовувати репозиторій Git, працювати з гілками й комітами, виконувати злиття змін і базову перевірку коду під час командної роботи;

– складати тест-кейси, виконувати перевірки відповідності реалізації вимогам, документувати дефекти й налагоджувати програмний код;

– створювати та запускати прості автоматизовані тести навчального програмного продукту, аналізувати результати й відтворювати виявлені помилки;

– інтегрувати компоненти, готувати навчальний застосунок до розгортання, перевіряти його працездатність і складати інструкцію користувача;

– аналізувати запити на зміни та зворотний зв’язок, виконувати обґрунтований рефакторинг і планувати подальший супровід програмного продукту.

5. Цілі сталого розвитку

ЦСР 4 – Якісна освіта

ЦСР 8 – Гідна праця та економічне зростання

ЦСР 9 – Промисловість, інновації та інфраструктура

6. Неформальна освіта

Рекомендовано навчальну вправу GitHub Skills «Introduction to GitHub»: https://github.com/skills/introduction-to-github. Вправа охоплює репозиторії, гілки, коміти та pull requests і може доповнювати опрацювання теми 12 «Організація командної розробки та управління версіями програмного забезпечення». Результат виконання можна підтвердити посиланням на навчальний репозиторій та історію внесених змін.

Результати тематичних онлайн-курсів, тренінгів та інших заходів неформальної або інформальної освіти можуть бути визнані відповідно до процедур ЛНТУ за умови відповідності результатам навчання дисципліни. Здобувач подає опис опанованих тем і досягнутих результатів та наявні підтвердження: сертифікат, портфоліо, програмний код, моделі або звіт. За потреби проводиться співбесіда чи практична перевірка. Обсяг і спосіб зарахування визначаються за встановленою в університеті процедурою.

7. 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.

8. 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.

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.

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.

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.

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 та структурного 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 окремого компонента електронного освітнього ресурсу як прикладу навчального програмного продукту.

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).

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.

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.

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.

9. Опис ІЗ/КПІЗ

Не передбачено навчальним планом.

10. 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.

11. 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.

ІІ. 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.