



ОСНОВИ ПРОГРАМУВАННЯ. Частина 2. МЕТОДОЛОГІЯ ПРОГРАМУВАННЯ

Силабус освітнього компонента

Реквізити навчальної дисципліни

Рівень вищої освіти	Перший (бакалаврський)
Галузь знань	F Інформаційні технології
Спеціальність	F2 Інженерія програмного забезпечення
Освітня програма	Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці
Статус дисципліни	Обов'язкова (нормативна)
Форма навчання	Очна (денна)
Рік підготовки, семестр	I курс, другий (весінній) семестр
Обсяг дисципліни	180 годин / 6 кредитів ЄКТС
Семестровий контроль / контрольні заходи	Екзамен
Розклад занять	http://rozklad.kpi.ua/
Мова викладання	Українська
Інформація про керівника курсу / викладачів	<u>Лектор:</u> доцент, к.т.н. Шпурик Вадим Вадимович <u>Практичні заняття:</u> доцент, к.т.н. Шпурик Вадим Вадимович старший викладач Оленева Ксенія Миколаївна <u>Лабораторні роботи:</u> доцент, к.т.н. Шпурик Вадим Вадимович старший викладач Оленева Ксенія Миколаївна
Розміщення курсу	Кампус – campus.kpi.ua http://route.ho.ua/kpi/c2/

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Майбутньому фахівцеві варто вивчати дисципліну «Основи програмування» тому, що вона є базовим курсом для всіх тих, хто хоче вивчити мови програмування, зокрема, C і C++, та стати досвідченим розробником програмного забезпечення. Курс містить також спеціальні розділи, присвячені життєвому циклу програмних засобів сучасної ІТ-індустрії, проблеми помилок в програмах і методах верифікації програмного забезпечення, стилю програмування. Після проходження лабораторного практикуму студент буде здатний писати програми середньої складності на мовах C, C++. Процес побудований таким чином, що слухачі навчаються не тільки писати код і розуміти код, написаний іншими програмістами, але і вирішувати проблеми, з якими неминуче доведеться зіткнутися при створенні програмного забезпечення.

Метою опанування дисципліни є освоєння фундаментальних основ програмування; ознайомлення з усіма основними термінами і поняттями; здобуття практичних навичок розробки алгоритмів; отримання реального досвіду розробки; навчитися писати доволі складні

програми; познайомитися з передовими мовами програмування і напрямками програмування; чітко розуміти, який напрямок вам найцікавіше; отримати матеріали та інструкції для подальшого розвитку.

Предметом кредитного модуля є вивчення фундаментальних принципів сучасного процедурного та об'єктно-орієнтованого програмування на базі мови системного програмування C.

Програмні результати навчання

Дисципліна сприяє формуванню у студентів певних компетентностей, тобто здатностей до виконання певних завдань самостійно. Компетентності формуються як фахові, так і загальні.

Загальні компетентності:

- здатність до абстрактного мислення, аналізу та синтезу (ЗК 01);
- здатність до пошуку, оброблення та аналізу інформації з різних джерел (ЗК 06).

Фахові компетентності:

- здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення (ФК 01);
- здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування (ФК 02);
- здатність розробляти архітектури, модулі та компоненти програмних систем (ФК 03);
- володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних (ФК 07);
- здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення (ФК 08);
- здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя (ФК 10);
- здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення (ФК 11);
- здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення (ФК 13);
- здатність до алгоритмічного та логічного мислення (ФК 14);
- володіти скриптовими та декларативними мовами програмування (ФК 16).

Згідно з вимогами програми навчальної дисципліни після засвоєння навчальної дисципліни студенти мають продемонструвати певні **програмні результати навчання**, тобто набуті ними знання й уміння описані далі:

- аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки (ПРН 01);
- знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення (ПРН 03);
- уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення (ПРН 06);
- знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення (ПРН 07);

- знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань (ПРН 13);
- мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення (ПРН 15);
- знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних (ПРН 18).

2. Пререквізити та постреквізити дисципліни

Вивчення дисципліни «Основи програмування» проходить паралельно з вивченням студентами основоположних дисциплін за спеціальністю, тому розміщується дисципліна у структурно-логічній схемі паралельно з нормативними освітніми компонентами циклу загальної підготовки та з дисциплінами циклу професійної підготовки, такими як: «Алгоритми та структури даних», «Основи комп'ютерних систем і мереж», «Компоненти програмної інженерії. Частина 1. Вступ до програмної інженерії» та «Комп'ютерна дискретна математика».

Дисципліна «Основи програмування» забезпечує вивчення таких дисциплін, як: «Бази даних», «Компоненти програмної інженерії», «Об'єктно-орієнтований аналіз та конструювання програмних систем», а також забезпечує написання курсової роботи з циклу професійної підготовки «Основи програмування. Курсова робота».

3. Зміст навчальної дисципліни

Навчальна дисципліна «Основи програмування» складається з двох кредитних модулів, другий з них – «Основи програмування. Частина 2. Методології програмування» має структуру представлену в таблиці далі.

Розділ 1. Додаткові відомості мови C. Структури даних	Тема – 1. Прості абстрактні структури даних
Розділ 2. Введення в мову C++: основні поняття та конструкції	Тема – 2.1. Вступ до C++: основні поняття
	Тема – 2.2. Вступ до C++: конструкції мови
	Тема – 2.3. Масиви об'єктів, використання вказівників і посилання
Розділ 3. Конструкції мови C++	Тема – 3.1. Використання перевантажених конструкцій
	Тема – 3.2. Використання наслідування (успадкування)
Розділ 4. Введення та виведення даних у C++. Робота з файлами	Тема – 4.1. Введення та виведення даних
	Тема – 4.2. Файлове введення та виведення даних
Розділ 5. Віртуальні функції, шаблони та виключення, RTTI	Тема – 5.1. Поліморфізм і віртуальні функції в програмах
	Тема – 5.2. Шаблони та обробка виключень
	Тема – 5.3. Типи даних і дії з ними
Розділ 6. Простори імен і шаблони проектування	Тема – 6.1. Простори імен і перетворення типів
	Тема – 6.2. Шаблони проектування
Розділ 7. Додаткові відомості мови C++. Структури даних, хеш-таблиці	Тема – 7. Алгоритм Марківського ланцюга

4. Навчальні матеріали та ресурси

Для кращого засвоєння дисципліни, виконання лабораторних і самостійної роботи (СРС) студентам надається список рекомендованої літератури та навчальні матеріали, розміщені в Інтернеті та на університетських ресурсах.

Ресурси та література:

1. Конспект лекцій. 2 семестр – URL: <http://route.ho.ua/kpi/c2/>

2. Ivor Horton Peter Van Weert. Beginning C++23: From Beginner to Pro. 7th Edition : Apress. 2023. 948 p.
3. Nesteruk D. Design Patterns in Modern C++20. 2nd Edition : Apress. 2022. 386 p.
4. Bjarne Stroustrup. Programming: Principles and Practice Using C++. 2nd Edition : Addison-Wesley Professional. 2014. 1312 p.
5. Bjarne Stroustrup. A Tour of C++ (2nd Edition) (C++ In-Depth Series) 2nd Edition : Addison-Wesley Professional. 2018. 256 p.

Навчальний контент

1. Методика опанування навчальної дисципліни (освітнього компонента)

Навчальна дисципліна складається з двох кредитних модулів, які охоплюють 11 кредитів або 330 годин. За порядком слідування та розподілом годин кредитні модулі розміщуються на першому курсі в осінньому та весінньому семестрах.

У весняному семестрі першого курсу передбачено другий кредитний модуль дисципліни «Основи програмування» – освітній компонент «Основи програмування. Частина 2. Методології програмування», на який відводиться 6 кредитів або 180 годин.

Другий кредитний модуль дисципліни включає в себе проведення протягом семестру:

- лекційних занять – 30 годин;
- практичних занять (комп'ютерні практикуми) – 14 годин;
- лабораторних робіт – 30 годин.

Також в освітньому компоненті відводиться 106 годин на самостійну роботу студента.

Календарний план запланованих заходів: лекційні заняття

№	Розподіл лекцій за темами	
1	Тема – 1. Прості абстрактні структури даних	Лекція 1. Прості абстрактні структури даних <i>перелік тем: абстрактні структури даних, однозв'язний список, бінарні дерева, хеш-таблиці</i>
2	Тема – 2.1. Вступ до C++: основні поняття	Лекція 2. Вступ до C++ <i>перелік тем: консольний ввід/вивід, конструктори та деструктори, перевантаження функцій, g++</i>
3	Тема – 2.2. Вступ до C++: конструкції мови	Лекція 3. Вступ до C++. Продовження <i>перелік тем: основні структури мови, що для чого і як використовувати</i>
4	Тема – 2.3. Масиви об'єктів, використання вказівників і посилання	Лекція 4. Масиви, вказівники та посилання <i>перелік тем: масиви об'єктів, використання вказівників і посилань, незалежні посилання та обмеження</i>
5	Тема – 3.1. Використання перевантажених конструкцій	Лекція 5. Перевантаження функцій <i>перелік тем: як перевантажують функції, випадки коли це потрібно, конструктор копіювання, проблеми при використанні, ситуація неоднозначності</i>
6		Лекція 6. Перевантаження операторів <i>перелік тем: як перевантажують оператори, перевантаження унарних і бінарних операторів, порядок операндів, перевантаження логічних операторів і оператора індексації, використання дружніх оператор-функцій</i>
7	Тема – 3.2. Використання наслідування (успадкування)	Лекція 7. Наслідування: типи, можливості, обмеження <i>перелік тем: види наслідування, базовий і похідний класи, контроль доступу до даних, маніпуляції з різними модифікаторами доступу, множинне наслідування, віртуальні базові класи</i>

8	Тема – 4.1. Введення та виведення даних	Лекція 8. Введення та виведення даних <i>перелік тем: основи вводу/виводу C++, форматований ввід/вивід, використання маніпуляторів вводу/виводу, створення власних операторів вводу/виводу</i>
9	Тема – 4.2. Файлове введення та виведення даних	Лекція 9. Розширені можливості вводу/виводу. Файлове введення та виведення даних <i>перелік тем: створення власних маніпуляторів, файловий ввід/вивід, неформатований ввід/вивід, перевірки, налаштований ввід/вивід а файли</i>
10	Тема – 5.1. Поліморфізм і віртуальні функції в програмах	Лекція 10. Поліморфізм: поняття та використання. Віртуальні функції <i>перелік тем: основні відомості про поліморфізм, вказівники на похідні класи, віртуальні функції, застосування поліморфізму</i>
11	Тема – 5.2. Шаблони та обробка виключень	Лекція 11. Шаблони та обробка виключень <i>перелік тем: поняття шаблонів, універсальні функції та класи (дженерики), виключення в програмах, обробка винятків</i>
12	Тема – 5.3. Типи даних і дії з ними	Лекція 12. Ідентифікація типів даних (RTTI), оператори приведення типів <i>перелік тем: що таке RTTI, використання операторів typedef, dynamic-, const-, static- і reinterpret_cast</i>
13	Тема – 6.1. Простори імен і перетворення типів	Лекція 13. Простори імен, функції перетворення типів і додаткові питання <i>перелік тем: простори імен: зокрема, глобальний і безіменний; функції перетворення; статичні члени класу; координація доступу до спільного ресурсу</i>
14	Тема – 6.2. Шаблони проектування	Лекція 14. Шаблони проектування <i>перелік тем: основні поняття, випадки та приклади використання</i>
15	Тема – 7. Алгоритм Марківського ланцюга	Лекція 15. Структура та реалізація простого алгоритму Марківського ланцюга <i>перелік тем: алгоритм Марківського ланцюга, побудова структури даних, побудова хеш-таблиці, генерація виходу</i>

Календарний план запланованих заходів: практичні заняття

№	Назва роботи
1	Практичне заняття 1. Основні конструкції
2	Практичне заняття 2. Перевантаження функцій і операторів. Конструктори: перевантаження та копіювання
3	Практичне заняття 3. Введення та виведення даних. Файлове введення та виведення даних
4	Практичне заняття 4. Наслідування класів
5	Практичне заняття 5. Віртуальні функції. Поліморфізм
6	Практичне заняття 6. Шаблони та обробка виключень. Універсальні функції та класи
7	Практичне заняття 7. Простори імен, функції перетворення типів

Календарний план запланованих заходів: лабораторні роботи

№	Назва роботи
1	Лабораторна робота 1. Структури даних
2	Лабораторна робота 2. Введення та виведення даних, приведення типів, конструктори та деструктори класів
3	Лабораторна робота 3. Основні конструкції: класи, об'єкти та функції
4	Лабораторна робота 4. Робота з масивами та їх елементами: виділення пам'яті та контроль границь
5	Лабораторна робота 5. Робота з перевантаженими функціями
6	Лабораторна робота 6. Робота з перевантаженими операторами
7	Лабораторна робота 7. Робота з наслідуванням
8	Лабораторна робота 8. Введення та виведення даних: прапори форматів, маніпулятори вводу/виводу
9	Лабораторна робота 9. Введення та виведення даних: робота з файлами
10	Лабораторна робота 10. Робота з віртуальними функціями. Поліморфізм
11	Лабораторна робота 11. Робота з універсальними функціями та класами
12	Лабораторна робота 12. Робота з операторами приведення типів
13	Лабораторна робота 13. Простори імен, функції перетворення типів
14	Лабораторна робота 14. Генерація тексту з заданими характеристиками за допомогою Марківського алгоритму
15	Лабораторна робота. Проведення модульної контрольної роботи

2. Самостійна робота студента

Самостійна робота студента в кредитному модулі «Основи програмування. Частина 2. Методології програмування» розрахована на 106 годин і поділена на завдання відповідно до розділів і тем.

Самостійна робота складається з завдань, які студенти мають виконати самостійно в поза аудиторні години навчання. Завдання містять як теоретичні, так і практичні задачі та розташовані в календарному плані дисципліни так, щоб доповнювати лекційний матеріал або передувати виконанню практичних занять та лабораторних робіт.

Також до годин самостійної роботи студентів входять підготовка до модульної контрольної роботи та екзамену.

Далі наведений перелік завдань.

Календарний план запланованої самостійної роботи студентів

№	Назви самостійних робіт за розділами та темами		Кіл-сть годин СРС
Розділ 1. Додаткові відомості мови С. Структури даних			
1	Тема. Прості абстрактні структури даних	СРС 1. Випадки «тупикових» ситуацій при роботі з деревами	2
Розділ 2. Введення в мову С++: основні поняття та конструкції			
2	Тема – 1. Вступ до С++: основні поняття	СРС 2. Особливості використання оператора “<<” з різними типами даних	3
3	Тема – 2. Вступ до С++: основні конструкції	СРС 3. Відмінність у використанні struct і class	4
Розділ 3. Конструкції мови С++: перевантаження та наслідування			
6	Тема – 1. Використання перевантажених конструкцій	СРС 6. Обмеження при перевантаженні методів	4
7		СРС 7. Обмеження при перевантаженні операторів	4
8	Тема – 2. Використання наслідування	СРС 8. Наслідування від декількох базових класів. Порядок виклику конструкторів	4
9	(успадкування) в класах	СРС 9. Виникнення помилок у роботі деструкторів при множинному наслідуванні	3
Розділ 4. Введення та виведення даних у С++. Робота з файлами			
10	Тема – 1. Введення та виведення даних	СРС 10. Експерименти з iostream з метою вивчення іо-можливостей	3
11	Тема – 2. Файлове введення та виведення даних	СРС 11. Зберігання даних на зовнішніх носіях	4
12		СРС 12. Особливості роботи з бінарними файлами	2
Розділ 5. Віртуальні функції, шаблони та виключення, RTTI			
13	Тема – 1. Поліморфізм і віртуальні функції в програмах	СРС 13. Віртуальний базовий клас, його необхідність	2
14	Тема – 2. Шаблони та обробка виключень	СРС 14. Особливості використання шаблонів (templates)	5
15	Тема – 3. Типи даних і дії з ними	СРС 15. Шаблон RTTI. Приклади застосування	4
Розділ 6. Простори імен і шаблони проектування			
16	Тема – 1. Простори імен і перетворення типів	СРС 16. Багатомодульна програма. Особливості використання простору імен (namespace)	4
17	Тема – 2. Шаблони проектування	СРС 17. Бібліотека STL. Базові можливості	4
Розділ 7. Додаткові відомості мови С++. Структури даних, хеш-таблиці			
18	Тема. Алгоритм Марківського ланцюга	СРС 18. Інші можливості генерації тексту	5
Контрольні заходи			
19	Підготовка до модульної контрольної роботи		9
20	Підготовка до екзамену		40

1. Політика навчальної дисципліни (освітнього компонента)

В освітньому компоненті «Основи програмування. Частина 2. Методології програмування» протягом семестру передбачено проведення: лекційних і практичних занять, лабораторних робіт і модульної контрольної роботи (МКР).

Відвідування занять

Відвідування усіх видів аудиторних занять, а саме: лекційних, практичних і лабораторних є обов'язковим.

Виняток становить лише відсутність на заняттях з поважних причин (тобто хвороби, форс-мажорні обставини, участь в олімпіадах, тощо), про що студент має повідомити викладача завчасно та надати підтвердження причини. У разі відсутності студента на аудиторному занятті з поважних причин нарахування штрафних балів не здійснюється.

Правила захисту лабораторних робіт

Протягом семестру студенти мають виконувати і захищати лабораторні роботи. Для кожної з робіт встановлені граничні терміни захисту (дедлайни).

Для отримання максимальних балів за кожну з передбачених програмою робіт студент повинен виконати роботу самостійно, в повній мірі, без помилок, вчасно здати звіт з роботи та вчасно роботу захистити (захист очний).

Самостійне виконання роботи має на увазі виконання роботи студентом на основі лекційних і практичних матеріалів, за необхідності використання додаткових рекомендованих джерел. Самостійне виконання забороняє використання для розв'язку завдань матеріалів інших студентів (чи випускників кафедри), матеріали (програмні коди) згенеровані штучним інтелектом чи матеріали скопійовані з будь-яких доступних ресурсів через Інтернет. Також лабораторні роботи мають виконуватись з урахуванням вивченого матеріалу в рамках дисципліни, а не чогось іншого, вивченого з інших джерел, не затверджених для дисципліни.

Якщо студент пропустив термін захисту роботи з поважної причини, він має можливість виконати та здати роботу без втрати балів.

Якщо студент пропустив термін захисту роботи без поважних причин – йому надається можливість відпрацювання пропущеної роботи шляхом написання самостійної роботи за відповідною темою. Але в такому випадку максимальний бал за роботу становитиме 60% від найвищої оцінки за завдання.

Причиною, що унеможливорює захист пропущених лабораторних робіт є дедлайн на відповідну роботу та закінчення навчального процесу (семестру).

Політика дедлайнів та перескладань

На кожну лабораторну роботу за календарним планом кредитного модуля відводиться по два тижні для виконання та захисту роботи. Всі терміни оголошуються студентам на початку семестру.

Дедлайни можуть коригуватись протягом семестру у випадку форс-мажорних ситуацій.

Перескладання лабораторних робіт не передбачено.

Відпрацювання пропущеної роботи передбачено у вигляді самостійної роботи протягом двох тижнів після завершення основного дедлайну на захист відповідної роботи.

Перескладання модульної контрольної роботи не передбачено. Написання модульної контрольної роботи поза визначеною датою можливе лише за поважних причин, що мають підтвердження.

2. Види контролю результатів навчання

Для кредитного модуля «Основи програмування. Частина 2. Методології програмування» передбачені наступні види контролю:

- проведення поточного контролю;
- проведення календарного контролю;

– проведення семестрового контролю.

Поточний контроль

Поточний контроль протягом семестру проводиться у вигляді двох самостійних робіт і модульної контрольної роботи. Поточний контроль має на меті перевірку знань та вмінь студентів, засвоєних і здобутих ними на усіх видах аудиторних занять.

Модульна контрольна робота проводиться за матеріалами усіх аудиторних видів занять, тобто: лабораторних робіт, лекційних і практичних занять. Завдання складаються з теоретичних і практичних запитань. Контрольна робота проводиться в кінці семестру. Питання та опорні теми для повторення оголошуються студентам заздалегідь.

Для проведення модульної контрольної роботи виділяються одна учбова година за рахунок лабораторного заняття.

Самостійні роботи проводяться за матеріалами лекційних і практичних занять, лабораторних робіт, складаються з теоретичних і практичних завдань. Самостійні роботи передбачені для перевірки поточних знань студентів, для проведення календарного контролю та не входять у загальний рейтинг студента. Опорні теми до повторення оголошуються студентам заздалегідь.

Календарний контроль

Календарний контроль проводиться двічі на семестр як моніторинг поточного стану знань та вмінь студентів і виконання ними умов навчальної програми освітнього компонента.

Умовою отримання атестації (позитивної оцінки) за проходження календарного контролю є виконання вимог за кількістю та якістю виконаних лабораторних робіт, а також отримання позитивної оцінки (≥60% від максимального балу) за написання відповідної самостійної роботи.

Причинами, що призводять до отримання незадовільного результату атестації під час календарного контролю є отримання незадовільної оцінки за самостійну роботу та захист передбачених календарним планом лабораторних робіт на незадовільні бали або ж у кількості менше 60% запланованих.

Семестровий контроль

Семестровий контроль з кредитного модуля здійснюється у вигляді екзамену.

Умовами допуску до здачі екзамену є виконання завдань, передбачених лабораторними роботами у кількості ≥ 75% від загальної кількості лабораторних. А також, написання модульної контрольної роботи на оцінку не нижче оцінки «достатньо» (60% від максимального балу).

Загальний рейтинг студента за семестр для допуску до екзамену має становити не менше 36 балів.

Екзамен проводиться в письмовому вигляді, містить теоретичні питання та практичні завдання. Перелік питань та опорні теми для підготовки до екзамену заздалегідь надаються студентам.

3. Рейтингова система оцінювання результатів навчання

Система оцінювання результатів навчання

Оцінювання результатів навчання здійснюється згідно рейтингової системи оцінювання за 100-бальною шкалою.

Рейтинг студента складається з балів за захист лабораторних робіт, написання модульної контрольної роботи та складання іспиту. Протягом семестру студент може набрати 49 балів за лабораторні роботи та 11 балів за МКР, що в сумі дає 60 балів.

Також під час роботи на лекційних і практичних заняттях студенти можуть отримати заохочувальні (додаткові) бали. За невиконання умов навчання передбачені штрафні бали. Кількість заохочувальних або штрафних балів рівна 5 балам на семестр.

Оцінювання лабораторних робіт

Для освітнього компонента «Основи програмування. Частина 2. Методології програмування» передбачено 14 лабораторних робіт на оцінку.

Загалом за лабораторні роботи за семестр студент може набрати 49 балів. Для отримання оцінки за лабораторну роботу потрібно виконати роботу, здати звіт і захистити роботу. Без попередньо зданого звіту до захисту студенти не допускаються.

Оцінюються роботи по 3,5 бали кожна за наступними критеріями:

- бездоганна робота оцінюється в 3,5 бали – завдання виконано в повній мірі, вирішує всі поставлені задачі, виконано без помилок або є лише незначні недоліки;
- є певні недоліки у виконанні роботи – 3-2,5 бали – завдання виконано в повній мірі, але допущені незначні помилки, вчасно здано звіт у належній якості та виконано очний захист на дуже добре/добре;
- є певні недоліки в виконанні роботи – 2-1 бали – завдання виконано неповністю, допущені суттєві помилки, вчасно здано звіт в неналежній якості чи допущені незначні помилки, очний захист виконано на задовільно/достатньо;
- робота не виконана, не захищена або не зданий звіт – 0 балів – робота здана лише у вигляді звіту, але не захищена особисто; робота здана особисто, але немає звіту; робота виконана чи захищена менш, ніж на 60% або здана пізніше дедлайну.

Оцінювання самостійних робіт

Самостійні роботи призначені для проведення календарного контролю оцінюються в 10 балів. Критерії оцінювання самостійних робіт аналогічні критеріям, що встановлені до оцінювання модульної контрольної роботи (описані далі).

Самостійні роботи, що виконуються студентами з умовою відпрацювання пропущених робіт без поважної причини оцінюються в 60% від максимального балу (тобто в 2,1 бали) за критеріями відповідними лабораторним роботам, без здачі звіту.

Оцінювання модульної контрольної роботи

Контрольна робота складається з чотирьох завдань: двох теоретичних питань і двох практичних завдань. Оцінювання завдань контрольної роботи виконується за таким розподілом: теоретичні питання оцінюються в 1,5 бали, а практичні завдання по 4 бали кожне.

Оцінюються питання модульної контрольної роботи за такими критеріями:

- «відмінно», повна відповідь, не менше 90% потрібної інформації, програмний код написаний згідно стандартів кодування, вирішує основне завдання, включає всі необхідні програмні модулі, конструкції та хоча б частину перевірок; відповіді написані точними формулюваннями та розгорнуто;
- «добре», достатньо повна відповідь, не менше 75% потрібної інформації, що виконана згідно з вимогами, є незначні неточності, відсутні перевірки та код неструктурований; відповіді короткі, але точні;
- «задовільно», неповна відповідь, не менше 60% потрібної інформації, завдання виконане з певними недоліками, в коді відсутні перевірки, наявні помилки; відповіді неповні чи містять суттєві помилки;
- «незадовільно», відповідь не відповідає умовам до «достатньо», наявні суттєві помилки, відповідь відсутня або не вирішує поставлені задачі; наявне копіювання, списування роботи, використання сторонньої допомоги (людини, штучного інтелекту, тощо).

Оцінювання екзаменаційної роботи

Умовою допуску до екзамену є виконання завдань, передбачених лабораторними роботами у кількості $\geq 75\%$ від загальної кількості лабораторних, написання модульної контрольної роботи на $\geq 7,2$ бали і стартовий рейтинг не менше 36 балів.

Екзамен студенти виконують письмово. Білети складаються з теоретичних питань і практичних завдань.

Теоретичні запитання оцінюються по 7 балів кожне, практичні завдання по 13 балів кожне, за наступними критеріями:

- «відмінно», повна відповідь, не менше 90% потрібної інформації, що виконана згідно з вимогами до рівня вмінь;
- «добре», достатньо повна відповідь, не менше 75% потрібної інформації, що виконана згідно з вимогами до рівня вмінь або є незначні неточності;
- «задовільно» і «достатньо», неповна відповідь, не менше 60% потрібної інформації, завдання виконане з суттєвими недоліками;
- «незадовільно», відповідь не відповідає умовам до «достатньо» – 0 балів.

Сума стартових балів та балів за екзаменаційну роботу переводиться до екзаменаційної оцінки згідно з таблицею:

Бали	Оцінка
100...95	Відмінно
94...85	Дуже добре
84...75	Добре
74...65	Задовільно
64...60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

За умови проведення навчання у дистанційному режимі екзамен не проводиться. В такому випадку стартові бали, отримані за виконання лабораторних робіт і написання МКР, переводяться у 100-бальну шкалу шляхом приведення через пропорцію.

Робочу програму навчальної дисципліни (силабус):

Складено

доцент, к.т.н., Шпурик Вадим Вадимович,
старший викладач Оленєва Ксенія Миколаївна

Ухвалено

кафедрою ІПЗЕ (протокол № 43 від 24.06.2026)

Погоджено

методичною комісією ННІАТЕ (протокол № 9 від 26.06.2026)