Національний технічний університет України
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»



Кафедра обчислювальної техніки

Алгоритми та структури даних 2.

Структури даних

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни	
Рівень вищої освіти	***Перший (бакалаврський)***
Галузь знань	*F Інформаційні технології*
Спеціальність	*F2 Інженерія програмного забезпечення*
Освітня програма	*Інженерія програмного забезпечення*
Статус дисципліни	*Нормативна*
Форма навчання	*очна*
Рік підготовки, семестр	*1 курс, весняний семестр*
Обсяг дисципліни	*4 кредитів/120 годин, денна форма: лекцій 30 годин, лабораторних робіт 14 годин та СРС 76 годин.*
Семестровий контроль/ контрольні заходи	*Залік/ лабораторні роботи*
Розклад занять	*Згідно розкладу на осінній семестр поточного навчального року за адресою http://rozklad.kpi.ua*
Мова викладання	*Українська*
Інформація про керівника курсу / викладачів	**Лектор** д.т.н, професор Анатолій Михайлович Сергієнко , *email Sergiyenko.Anatoliy@lll.kpi.ua* **Викладач** з лабораторних занять д. філ. : *Череватенко Олексій Володимирович, email chereva@ukr.net*
Розміщення курсу	*На платформі дистанційного навчання «Сікорський»: https://classroom.google.com/c/NDIyNTg2NDY3NTY4?cjc=whbtqcx*

Програма навчальної дисципліни

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

Викладання алгоритмів та структур даних в університеті має такі аспекти:

Академічний аспект. Алгоритми та структури даних - це наука, яка дає фундаментальні знання. Мета - ознайомити студентів з основними поняттями та методами представлення даних та виконання з ними алгоритмів, вимірюючи складність алгоритму.

Професійний аспект. Знання структур даних та алгоритмів є обов'язковими для інженерів та програмістів, які розробляють програмне забезпечення для комп'ютера. Мета цього курсу – сформувати навички використання мови С для програмування базових алгоритмів, які використовуються у більшості програмного забезпечення універсальних та спеціалізованих систем.

Інтелектуальні та освітні аспекти. Вивчення структур даних та алгоритмів сприяє розвитку когнітивних навичок, що є суттєвим інтелектуальним фактором у процесі створення програмного забезпечення.

В результаті вивчення цієї дисципліни досягаються такі результати навчання.

Компетенції:

ЗК 01 здатність до абстрактного мислення, аналізу та синтезу.

ЗК 02 здатність до навчання та набуття сучасних знань.

ЗК 05 Здатність вчитися і оволодівати сучасними знаннями.

ФК 01 Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.

ФК 02 Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.

ФК 05 Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу

ФК 14 Здатність до алгоритмічного та логічного мислення.

У відповідності до вищезазначеного, підсилені загальні та фахові компетенції дадуть наступні **знання:**

ПРН 02 Знати кодекс професійної етики, розуміти соціальну значимість та культурні аспекти інженерії програмного забезпечення і дотримуватись їх в професійній діяльності.

ПРН 03 Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.

ПРН 04 Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення.

ПРН 07 Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.

ПРН 13 Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.

ПРН 23 Вміти документувати та презентувати результати розробки програмного забезпечення.

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Передумови, тобто дисципліни, вивчення яких має передувати вивченню цієї дисципліни:

- базовий рівень володіння англійською мовою не нижче А1.
- алгоритми і структури даних, основи алгоритмізації.

Знання та навички, отримані під час вивчення дисципліни «Алгоритми та структури даних», можуть бути застосовані під час вирішення навчальних та лабораторних завдань, що виникають під час засвоєння дисциплін:

- архітектура комп’ютера;
- бази даних;
- розробка програм;
- системне програмне забезпечення;

• чисельні методи;
• комп’ютерні мережі;
• штучний інтелект.

3. Зміст навчальної дисципліни

Тема 1: Рекурсивні функції.

Рекурсивна підпрограма та її реалізація. Рекурсивна функція в математичних обчисленнях. Переваги та недоліки рекурсії.

Тема 2: Пов’язані списки.

Динамічне виділення пам'яті. Функції для роботи з динамічною пам'яттю. Купа. Приклад програмування циклічного буфера. Забезпечення послідовності розподілу динамічної пам'яті. Ініціалізація динамічних змінних.

Послідовності. Мультимножина. Вектор Дерево. Граф. Операції з множинами. Бітова шкала. Послідовне зберігання векторів. Зв’язане зберігання векторів. Лінійний список та його програмування. Зв'язані структури даних. Пошук у зв’язаному списку. Операції зі списком. Різновиди зв’язаних списків.

Стек і черга. Реалізація стека в архітектурі рівня системи команд. Стек як зв'язаний список. Сортування даних у стеках. Мови програмування, орієнтовані на стек. Циклічний буфер як черга. Список як черга. Черга з пріоритетом. Дек.

Тема 3: Алгоритми на графах.

Визначення графів. Диграфи. Типи графів та їх параметри. Маршрути у графах. Граф Ейлера. Гамільтоновий цикл. Частини графа. Подільність діграфа. Дерева і ліс. Кістяк. Плоскі графи.

Графічні структури даних. Графічне прикладне програмування.

Відношення та графи. Перехідне та рефлексивне замикання. Матриця досяжності. Компонент сильного зв'язку. Граф конденсації. Алгоритм Кьоніга.

Обхід графу. Алгоритм пошуку вшир. Алгоритм обчислення відстаней. Алгоритм пошуку вглиб. Топологічне сортування. Пошук вершини з’єднання.

Зважений граф. Найкоротший шлях. Алгоритм Дейкстри. Жадібні алгоритми оптимізації. Мінімальний пошук по кістяку. Алгоритм Краскала. Алгоритм Пріма. Пошук дерева Штейнера.

Орієнтовані дерева. Упорядковані дерева. Визначення виразів, програм за впорядкованими деревами. Прямий обхід бінарних дерев. Представлення дерев у програмах. Представлення вільних дерев. Алгоритми коду Прюфера. Представлення двійкового дерева. Структура списку, упакований масив, польська нотація. Алгоритми обходу двійкових дерев. Польські прямаі та зворотна нотації. Дерево сортування.

Тема 4: Асоціативна пам’ять

Основи асоціативної пам’яті. Операції з асоціативною пам'яттю. Асоціативні масиви та списки та їх програмування. Алгоритм операцій з деревом сортування. Вирівняні та повні дерева. Збалансовані дерева. ABL-дерево. Алгоритми балансування дерев.

Хеш -таблиця та її структура. Створення асоціативної хеш -таблиці. Пошук об’єкта у таблиці за допомогою хеш -функції. Тип даних абстрактної хеш -таблиці. Модульне хешування. Ідеальна хеш -функція. Методи вирішення конфліктів.

4. Навчальні матеріали

1. Базова література

1. Керниган Б., Рітчі Д. Мова програмування Сі. 229 с.
2. Кормен Т. Г., Лейзерсон Ч. Е., Рівест Р. Л., Стайн К. Вступ до алгоритмів. Київ: К. І. С., 2019. 1288 с.
3. Макконелл. Дж. Основи сучасних алгоритмів. 2004. 368 с.

4.2. Додаткова література

4. Кормен Т. Г. Алгоритми доступно. Київ: К.І.С, 2021. 206 с.
5. Ковалюк Т. В. Основи програмування. Київ: BHV. 2005. 384 с.
6. Непийвода Н. Н., Скопін І. Н.. Основи програмування. 2002. 919 с.
7. Вірт, Алгоритми та структури даних Н.. 358 с. ДМК. 2022. 272 с.
8. Sergiyenko A., Serhienko A., Romankevich V. "Genetic Programming of Pipelined Datapaths for FPGA," *2020 IEEE 40th International Conference on Electronics and Nanotechnology (ELNANO)*, Kyiv, Ukraine, 2020, pp. 802-806, doi: 10.1109/ELNANO50318.2020.9088773.

4.3. Інформаційні ресурси

Курс відеолекцій – на платформі дистанційного навчання Big Blue Button: https://bbb.comsys.kpi.ua/b/ana-fxo-y2d-ykw

Слайди лекцій та керівництва до лабораторних робіт https://kanyevsky.kpi.ua/en/for-students/

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Навчальний зміст дисципліни складається з лекцій та лабораторних робіт, які виконуються у лабораторному практикумі.

Лекції.

Тема 1: Рекурсивні функції.

Лекція 1. Рекурсивні алгоритми.

Математична індукція та рекурсія. Рекурсивна підпрограма та її реалізація. Рекурсивна функція. Сигналізуюча функція. Шлях рекурсії. 3 форми рекурсивних дій. Функція з рекурсивним підйомом або спуском. Функція, з частковим спуском і підйомом. Обчислення чисел Фібоначчі. Рекурсивна функція в математичних обчисленнях. Переваги та недоліки рекурсії.

Тема 2: Зв’язані списки.

Лекція 2. Динамічна адресація.

Розподіл статичної пам’яті. Динамічне виділення пам'яті. Тип size_t. Функції для роботи з динамічною пам'яттю. Купа. Приклад програмування циклічного буфера. Забезпечення послідовності розподілу динамічної пам'яті. Ініціалізація динамічних змінних.

Лекція 3. Зв’язані структури даних.

Послідовності. Множини. Вектор Дерево. Граф. Гіперграф. Операції з множинами. Бітова шкала. Послідовне зберігання векторів. Зв’язане зберігання векторів. Зв’язаний список. Вузол списку. Лінійний список та його програмування.

Лекція 4. Зв’язані структури даних (продовження)

Пошук у зв’язаному списку. Видалення елементу зі списку. Додавання елементу до відсортованого списку. З’єднання списків. Переваги та недоліки подання списку. Різновиди зв’язаних списків. Порожній список. Список із заголовком. Циклічний список. Двохзв'язний список. Дввохзв'язний циклічний список. Список з пропусками. Список з самоорганізацією.

Лекція 5. Стек і черга.

Визначення стека. Операції зі стеком. Реалізація стека в архітектурі системи команд. Карта пам'яті звичайного застосунку. Перевірка переповнення стека. Недоліки організації стека в масиві. Стек як зв'язаний список. Використання стека. Сортування даних у стеках. Визначення черги. Циклічний буфер як черга. Список як черга.

Тема 3: Алгоритми на графах.

Лекція 6. Графи.

Визначення графу. Призначення графів. Напрямлений граф. Параметри графів. Види графів. Зважені графи. Ізомрофні графи. Маршрути у графах. Граф Ейлера. Гамільтоновий цикл. Орієнтовані маршрути. Частини графа. З'єднання з графіком. Подільність діграфа. Дерева і ліс. Кістяк графа. Плоскі графи. Графи Понтрягіна - Куратовського. Гомеоморфні графи.

Лекція 7. Структури графічних даних.

Графічний інтерфейс комп’ютера і його основні елементи. Графічний інтерфейс у PC-AT. Багатозадачна операційна система. Інтерфейс графічного пристрою (GDI). Бібліотека windows.h. Дескриптор HANDLE. Графічне прикладне програмування. Основний модуль WinMain (). Приклад програмування фреймового застосунку. Деталі виконання третьої лабораторної роботи.

Лекція 8. Графи та відношення.

Рефлексивні, антирефлексивні, симетричні, антисиметричні, зворотні, універсальні, взаємодоповнюючі відношення. Об'єднання графів. Композиція відношень. Степінь відношення. Транзитивне відношення. Транзитивне та рефлексивне замикання. Матриця досяжності. Компоненти сильної зв'язності. Матриця сильної зв'язності. Граф конденсації. Алгоритм Кеніга.

Лекція 9. Алгоритми обходу графів.

Обхід графів. Алгоритм пошуку вшир. Процедура BFS. Протокол обходу графа. Властивості BFS. BFS-дерево та розрахунок відстані. Геодезичне дерево. Алгоритм обчислення відстаней.

Лекція 10. Алгоритми обходу графів (продовження).

Алгоритм пошуку вглиб. Процедура DFS. Роль стека. Приклад реалізації алгоритму DFS. Властивості алгоритму DFS. DFS-дерево. Топологічна сортування. Спільний точковий пошук.

Лекція 11. Алгоритми найкоротшого шляху та пошуку мінімального кістяка..

Зважений граф. Найкоротший шлях. Алгоритм Дейкстри. Жадібні алгоритми оптимізації. Приклад виконання алгоритму Дейкстри. Мінімальний кістяк. Задачі з мінімальним кістяком. Алгоритм Краскала. Приклад виконання алгоритму Краскала. Алгоритм Пріма. Приклад виконання алгоритму Пріма. Порівняння алгоритмів. Використання алгоритмів мінімального кістяка. Пошук дерева Штейнера.

Лекція 12. Дерева.

Визначення орієнтованого дерева. Частини дерев. Впорядковані дерева. Визначення виразів, програм за впорядкованими деревами. Техніка відступів. Довідники як впорядковані дерева. Обхід дерева. Двійкові дерева. Представлення дерев у програмах. Представлення вільних дерев. Код Прюфера. Алгоритми кодування та декодування коду Прюфера.

Лекція 13. Дерева (продовження).

Двійкові дерева. Представлення двійкового дерева. Представлення двійкового дерева в програмах: структура списку, упакований масив, польська нотація. Алгоритми обходу двійкових дерев: прямі, внутрішні, обернені. Польські прямі та зворотні нотації. Дерево сортування.

Тема 4: Асоціативна пам’ять

Лекція 14. Основи асоціативної пам’яті. Пошук у двійковому дереві

Асоціативна пам’ять. Операції з асоціативною пам'яттю. Асоціативні типи пам’яті. Асоціативні масиви і списки та їх програмування. Порівняння представлень асоціативної пам’яті. Алгоритм пошуку в дереві сортування. Алгоритм вставки в дерево сортування. Алгоритм видалення з дерева сортування. Порівняння представлень асоціативної пам’яті. Вирівняні та повні дерева. Збалансовані дерева. ABL-дерево. Алгоритми балансування дерев.

Лекція 15. Хеш -таблиці

Створення асоціативної таблиці. Пошук в таблиці за допомогою хеш-функції. Хеш-таблиця та її структура. Тип даних абстрактної хеш-таблиці. Хешування ключів. Модульне хешування. Хешування ключа з плаваючою крапкою. Хешування рядків. Ідеальна хеш-функція. Властивості ефективної хеш-функції. Методи вирішення конфліктів. Хешування з відкритою адресацією. Хешування методом виділення та створення ланцюжка. Порівняння представлень асоціативної пам’яті.

Деталізація тем лекційних занять, конспект лекцій та відеоматеріали розміщено на платформі дистанційного навчання «Сікорській»
https://classroom.google.com/c/NDIyNTg2NDY3NTY4?cjc=whbtqcx

Лабораторні роботи

Мета лабораторних робіт – експериментальне дослідження та випробування алгоритмів та структур даних, що вивчаються, при програмуванні завдань лабораторних робіт.

Тематика лабораторних робіт:

Лабораторна робота 1.

Рекурсивні алгоритми.

Лабораторна робота 2.

Програмування зв'язаного списку.

Лабораторна робота 3.

Характеристики графа та його відношень.

Лабораторна робота 4.

Алгоритми обходу графів.

Лабораторна робота 5.

Алгоритми мінімального кістяка.

Загальні методичні рекомендації для дистанційного виконання:

1. **Зосередитись на програмуванні.** Основний навчальний чинник курсу — експериментальне випробування алгоритмів та структур даних, що вивчаються, при програмуванні завдань лабораторних робіт на власному комп’ютері студента. При цьому програмування можна виконувати у будь-яких сприятливих умовах у довільний час. При відсутності центрального електропостачання, програмування можна виконувати використовуючи заряд акумулятора.
2. **Використання матеріалів, які доступні в мережі Інтернет.** При неможливості бути присутнім на лекції кожну лекцію можна продивитись у запису, який зберігається на сервері

кафедри, у довільний час, який є зручним для студента, навіть з повтором, якщо щось є незрозумілим з першого разу. Також в Інтернеті є багато довідкового матеріалу, що стосується курсу, який можна знайти за пошукачем чи у Вікіпедії або користуючись генеративним штучним інтелектом.

3. **Комунікація та підтримка.** Використовувати онлайн-інструменти для зворотного зв'язку, обговорення завдань та підтримки студентів у процесі виконання проєктів. Важливо мати регулярні консультації та можливість допомогти студентам у разі виникнення питань.

6. Самостійна робота здобувача вищої освіти

Види самостійної роботи для студентів денної форми навчання (66 годин):

• підготовка до лекцій шляхом вивчення попереднього лекційного матеріалу, а також літературних джерел, на яких ґрунтується матеріал попередніх лекцій (1 година х 18 лекцій = 18 годин);

• підготовка до лабораторної роботи через ознайомлення із завданням та методичними вказівками до лабораторних робіт, включаючи вивчення теоретичного матеріалу, необхідного для відповіді на контрольні запитання до лабораторних робіт, складання і програмування алгоритму, компіляцію та відлагодження програми, розгортання програмного середовища та завантаження вихідних кодів програмного забезпечення для виконання лабораторних робіт, робота з системою контролю версій Git, а також оформлення звіту. Крім того, створення власного програмного інструменту для представлення графів. Кожна лабораторна робота вимагає для підготовки та виконання двох тижнів і витрат до 11 годин часу (5 х 11 годин = 55 годин);

• підготовка до МКР (6 годин).

Теми на самостійне опрацювання

Тема 1: Рекурсивні функції.

Рекурсивна функція в математичних обчисленнях.

Тема 2: Зв'язані списки.

Стек у граматичному аналізі текстів. Мови програмування, орієнтовані на стек. Стекові процесори. Визначення черги. Циклічний буфер як черга. Список як черга. Черга з пріоритетом. Дек.

Тема 3: Алгоритми на графах.

Ейлерові графи і алгоритми мандрівника-продавця. Кліки, незалежні множини. Алгоритм Магу.

Тема 4: Асоціативна пам'ять.

Алгоритми стиснення даних на основі словника.

Матеріали для самостійного опрацювання, теми, питання для самоконтролю та контрольні запитання знаходяться на платформі дистанційного навчання «Сікорський» (Classroom). *https://classroom.google.com/c/NDIyNTg2NDY3NTY4?cjc=whbtqcx*

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Система вимог до студентів:

• студент зобов'язаний відвідувати лекції та заняття комп'ютерного практикуму, активно працювати над засвоєнням викладеного на них матеріалу;

• на лекції лектор використовує власний презентаційний матеріал;

• лабораторні роботи захищаються у два етапи: перший етап - студенти виконують завдання, складають електронний звіт та надсилають викладачу; другий етап - захист лабораторних робіт у лабораторії. Контроль знань на лабораторних роботах здійснюється шляхом перевірки звіту про лабораторні роботи, а також питань з матеріалів робіт.

• модульна контрольна робота пишеться як тест з використанням усіх наявних матеріалів, але без використання комп'ютерних засобів.

Дедлайни. Для виконання лабораторних робіт встановлюються дедлайни. Дедлайни уточнюються і оприлюднюються викладачем на початку семестру.

Заохочувальні бали (бонуси) виставляються за: активну участь на лекціях та лабораторних заняттях, ведення конспекту, підготовку повідомлення з презентацією по одній із тем СРС дисципліни тощо. Кількість заохочуваних балів не більше 5.

Лабораторні роботи. Виконанню кожної лабораторної роботи передує виконання індивідуального завдання і оформлення його у вигляді протоколу. Протокол виконання лабораторного завдання має бути вислано на пошту викладача не пізніше ніж за добу до лабораторного завдання. До протоколу додаються посилання на Git з кодом програм (за необхідності) і посилання на всі необхідні супровідні матеріали для захисту, зокрема на відеозапис виконання лабораторного завдання (за необхідності, в дистанційному режимі або за використання віддаленого устаткування, тощо). Студент, який відвідав заняття без підготовленого протоколу до захисту лабораторної роботи не допускається. Першим етапом студент захищає результати виконання індивідуального завдання до лабораторної роботи, на другому етапі – захищає теорію шляхом усного опитування або тестування. В дистанційному режимі більшість лабораторних робіт супроводжуються тестами для оцінки вивченого теоретичного та лабораторного матеріалу замість усного опитування. Бали отримані за виконання лабораторної роботи - за тест та за протокол складають оцінку за лабораторну роботу. Тестування проводиться на лабораторному занятті після перевірки результатів виконання індивідуальної роботи. Студент, який не виконав індивідуальне завдання до лабораторної роботи до тесту не допускається.

Виконання лабораторних робіт та МКР є обов'язковими для допуску до семестрового контролю (заліку). Загальною умовою допуску до семестрового контролю (заліку) є зарахування всіх лабораторних робіт, МКР та загальний рейтинг не менше 60 балів.

Оцінка, яку студент може отримати за виконання кожної лабораторної роботи наведені у розділі 8 силабуса.

Здобувачі, які виконали всі умови допуску до заліку та мають рейтингову оцінку менше 60 балів, а також здобувачі, які бажають підвищити свою рейтингову оцінку, на останньому за розкладом занятті складають залік у вигляді виконання залікового завдання та співбесіди.

З метою посилення зацікавленості здобувачів у якісному виконанні індивідуальних семестрових завдань, передбачених індивідуальним навчальним планом здобувача, рейтингова (семестрова) оцінка, у разі виконання залікової контрольної роботи, визначається, як оцінка за залікову контрольну роботу та сума балів за індивідуальні семестрові завдання (що відповідають балам за лабораторні роботи протягом семестру). Всі інші бали, які набрані протягом семестру (окрім балів за лабораторні роботи) анулюються.

Модульна контрольна робота є обов'язковою для допуску до семестрового контролю. Модульна контрольна робота пишеться на лекційному занятті без застосування допоміжних засобів (мобільні телефони, планшети та ін.); результат пересилається до відповідної директорії Google -диску через Google форму.

У разі виконання модульної контрольної роботи рейтингова (семестрова) оцінка визначається як сума балів за модульну контрольну роботу, балів за індивідуальні семестрові завдання (лабораторні роботи).

До індивідуальних семестрових завдань вноситься індивідуальна робота студента що стосується виконання лабораторних робіт. Кількість балів за індивідуальну роботу за семестр дорівнює від 38 до 60 балів. Максимальна оцінка за МКР дорівнює 40 балів.

Дистанційний режим навчання. У разі запровадження обмежень на відвідування університету, пов'язаних з введенням карантину або режиму воєнного стану в державі, освітній процес здійснюється у дистанційному режимі відповідно до Положення про дистанційне навчання в КПІ ім. Ігоря Сікорського (https://osvita.kpi.ua/index.php/node/188), Регламенту організації освітнього процесу в дистанційному режимі (https://profkom.kpi.ua/reglament-organizatsiyi-osvitnogo-protsesu-v-distantsiynomu-rezhimi) та Регламенту проведення семестрового контролю в дистанційному режимі (https://osvita.kpi.ua/node/148). У режимі дистанційного навчання заняття відбуваються у вигляді онлайн-конференції на платформах BigBlueButton, Google Meet, Zoom. Посилання на конференцію видається на початку семестру і розміщується в АС «Електронний кампус». З метою забезпечення якісної підготовки здобувачів, дистанційний курс дисципліни розміщено на Платформа дистанційного навчання «Сікорський» (https://www.sikorsky-distance.org). Результати оцінювання висвітлюють у АС «Електронний кампус» на особистій сторінці здобувача (https://ecampus.kpi.ua).

Правила поведінки на заняттях. На заняттях слід дотримуватись норм етичної поведінки визначених у Кодексі честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (https://kpi.ua/code). На території університету здобувачі мають поводити себе відповідно до Правил внутрішнього розпорядку (https://kpi.ua/admin-rule).

Визнання результатів навчання, набутих у неформальній/інформальній освіті. Положення про визнання результатів навчання, набутих у неформальній / інформальній освіті (https://osvita.kpi.ua/index.php/node/179) регламентує визнання результатів навчання, набутих у неформальній/інформальній освіті.

Неформальна освіта. У разі проходження здобувачем онлайн курсу чи іншого елементу неформальної освіти здобувачеві/здобувачці можуть бути зараховані окремі змістовні модулі або теми дисципліни. В такому разі здобувач звільняється від виконання відповідних лабораторних робіт, отримуючи за них максимальний бал відповідно до рейтингової системи оцінювання. У разі навчання на неформальній освіті за самостійного її обрання проходиться процедура валідації, що передбачає подання здобувачем заяви на ім'я декана, декларації підтверджувальних документів. Рішення про визнання чи не визнання приймається комісією у складі завідувача кафедри, викладача, гаранта освітньо-професійної програми.

Політика використання штучного інтелекту. Використання штучного інтелекту регламентується «Політикою використання штучного інтелекту для академічної діяльності в КПІ ім. Ігоря Сікорського» (https://osvita.kpi.ua/node/1225). Усі завдання, як під час виконання навчальних завдань з дисципліни, так і індивідуальні завдання, мають бути результатом власної оригінальної роботи здобувача. Використання ШІ для автоматичної генерації відповідей без подальшого їх аналізу та доопрацювання заборонено.

Порушення академічної доброчесності: використання плагіату. У разі виявлення фактів використання плагіату під час виконання навчальних завдань, студент втрачає право на подальше виконання поточних завдань у межах навчальної дисципліни до з'ясування обставин порушення. Студент зобов'язаний аргументовано пояснити причини використання плагіату, а також засвідчити готовність дотримуватись принципів академічної доброчесності в подальшому навчанні; повторно виконати завдання або, за потреби, виконати додаткове завдання відповідно до бачення викладача з урахуванням вимог щодо самостійності.

Зазначені дії регламентуються Кодексом честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (https://kpi.ua/code)

8. Види контролю та рейтингова система оцінювання результатів навчання (РСО)

На першому занятті студенти знайомляться з політикою оцінювання, яка ґрунтується на Положенні про систему оцінювання результатів навчання https://document.kpi.ua/files/2020_1-273.pdf. Рейтинг студента на курсі складається з балів, які він/вона отримує за захищені лабораторні роботи (R1) та модульну контрольну роботу (R2).

Rs = R1+R2 = 100 балів

В результаті максимальний середній бал дорівнює:

5 лабораторних робіт х 12 балів = 60 балів

модульна контрольна робота = 40 балів.

Згідно з університетськими правилами щодо моніторингу успішності студента (https://kpi.ua/document_control), існує два тижні оцінювання (атестація), зазвичай протягом 7/8 та 14/15 тижнів семестру, коли оцінюється прогрес у навчанні студентів та відповідність до критеріїв політики оцінювання курсу.

Умовою першої атестації є отримання не менше 10 балів (на момент атестації). Умова другої атестації - набрати не менше 30 балів (на момент атестації).

Критеріями оцінки є:

• Виконання лабораторних робіт:

- бездоганна робота оцінюється у 11 балів, правильні відповіді на усі питання при захисті;
- є певні недоліки в оформленій роботі - 10-9 балів, є неточні відповіді на питання;
- є деякі недоліки у виконанні робочої програми - 8-6 балів, окремі відповіді на питання є неправильними;

Робота не виконана або не захищена - 0 балів.

За роботу, подану вчасно, діє заохочення - 1 бал (сумарно не більше - 5 балів).

• Модульна контрольна робота оцінюється 40 балами. Контрольна робота складається з 16 тестових питань, а також практичного завдання (скласти програму) зі списку, наведеного в додатку до робочої програми.

За кожну правильну відповідь на тестове питання нараховується 2 бали. Відповідь на практичне завдання оцінюється 8 балами за такими критеріями:

- "відмінно" - правильний текст програми з коментарями - 8 - 7 балів;
- "добре" - текст програми, загалом правильний, але не використані прийоми оптимізації - 6-5 балів;
- "задовільно" - є деякі принципові помилки в тексті програми - 4 - 3 бали;
- "незадовільно" - незадовільна відповідь - 0 балів.

Студенти, які нарешті наберуть необхідну кількість балів (≥60), можуть:

- отримати остаточну залікову оцінку відповідно до рейтингового балу;
- виконати залікову контрольну роботу, щоб підвищити оцінку.

Студенти можуть отримати до 6 заохочувальних балів за виконання творчих робіт з кредитного модуля (складання тез, участь у конкурсах, у дослідженнях тощо).

Студенти, чий остаточний бал за успішність Rs нижче 60 балів, але більше за 30, повинні виконати залікову контрольну роботу. Студенти, чий остаточний бал за успішність Rs вище або дорівнює 60 балів, можуть виконати залікову контрольну роботу, щоб підвищити оцінку. В обох випадках застосовується вимога - попередній рейтинг студента складає: 0.7R1. Студенти, чий бал нижче 30, не допускаються до заліку.

Максимальна оцінка R3 за залікову контрольну роботу складає 60 балів. Контрольна робота складається з 25 тестових питань, а також практичного завдання (скласти програму). За кожну правильну відповідь на тестове питання нараховується 2 бали. Відповідь на практичне завдання оцінюється 10 балами за такими критеріями:

- "відмінно" - правильний текст програми з коментарями - 10 - 9 балів;
- "добре" - текст програми, загалом, правильний, але не використані прийоми оптимізації - 8-6 балів;
- "задовільно" - є деякі принципові помилки в тексті програми - 5 - 4 балів;
- "незадовільно" - незадовільна відповідь - 0 балів.

Рейтингова оцінка разом з оцінкою за залікову контрольну роботу дорівнює

$Rs = 0.7R1 + R3$.

Таблиця. Остаточна оцінка RS приймається за системою оцінювання університету

Кількість балів	*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

Робочу програму навчальної дисципліни (силабус):

Складено д.т.н., проф. А. М. Сергієнко

Ухвалено кафедрою обчислювальної техніки (протокол №18 від 24.06.2026)

Погоджено Методичною комісією ФІОТ (протокол №12 від 26.06.2026)