



Алгоритмізація та програмування. Частина 2.

Процедурне програмування

Робоча програма навчальної дисципліни (Силабус)

- Реквізити навчальної дисципліни

Рівень вищої освіти	Перший (бакалаврський)
Галузь знань	12 Інформаційні технології
Спеціальність	F3 Комп'ютерні науки
Освітня програма	Цифрові технології в енергетиці
Статус дисципліни	Нормативна
Форма навчання	Очна (денна)
Рік підготовки, семестр	1 курс 2 семестр
Обсяг дисципліни	4 кредити ЕКТС / 120 годин: лекції- 30 год., лабораторні роботи – 14 год., практичні роботи – 14 год., самостійна робота – 62 год.
Семестровий контроль/ контрольні заходи	Залік, мкр
Розклад занять	http://rozklad.kpi.ua
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектор: старший викладач Беспала Ольга Миколаївна Лабораторні роботи: старший викладач Беспала Ольга Миколаївна bespala.olha@iit.kpi.ua
Розміщення курсу	https://campus.kpi.ua

- Програма навчальної дисципліни

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

Дисципліна «Алгоритмізація та програмування. Частина 2. Процедурне програмування» викладається у другому навчальному семестрі першого курсу та є важливою складовою у підготовці студентів спеціальності F3 – «Комп'ютерні науки». Даний курс з вивчення базових основ процедурного програмування та проектування програм базується на мові програмування C. В результаті вивчення курсу студент повинен опанувати засоби проектування програм та отримати навички, необхідні для самостійного проектування та супроводження програмних продуктів.

Метою дисципліни «Алгоритмізація та програмування. Частина 2. Процедурне програмування» є оволодіння студентами теоретичних знань та практичних навичок основ розробки ефективних алгоритмів розв'язання задач та формування у студентів компетентностей у відповідності до ОПП:

ЗК 01	Здатність до абстрактного мислення, аналізу та синтезу.
ЗК 02	Здатність застосовувати знання у практичних ситуаціях.
ЗК 06	Здатність вчитися і оволодівати сучасними знаннями.
ФК 03	Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та

	нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем
ФК 8	Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

Предмет дисципліни – вивчення основ мови програмування С, базових типів даних, основних операцій, математичних функцій, операторів управління, масивів та вказівників, структур, списків, динамічного виділення пам'яті, роботи з файлами, способів передачі параметрів при виклику функцій.

В результаті засвоєння кредитного модуля студенти мають продемонструвати такі програмні результати навчання:

ПРН 01	Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.
ПРН 05	Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій.
ПРН 09	Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.

Результати навчання. Згідно з вимогами програми навчальної дисципліни в результаті вивчення курсу студенти повинні

Знати:

- структуру та оператори управління С-програми,
- масиви та вказівники,
- структурний тип даних,
- динамічного виділення пам'яті та функції роботи з файлами,
- способи виклику функцій.

Вміти:

- розробляти алгоритми вирішення типових задач, що пов'язані з розробкою прикладного програмного забезпечення,
- реалізовувати програмні алгоритми засобами мови програмування С,
- виконувати налаштування та тестування програмного забезпечення.

Набути досвід:

- проектування, розробки та тестування програмного забезпечення,
- застосування мови програмування С для взаємодії з операційною системою,
- застосування набутих знань під час розробки курсової роботи.

2. Пререквізити та постреквізити дисципліни

У структурно-логічній схемі навчання дисципліна «Алгоритмізація та програмування. Частина 2. Процедурне програмування» викладається у 2-му семестрі. Отримані студентами знання та

компетенції при вивченні дисциплін «Алгоритмізація та програмування. Частина 2. Процедурне програмування» використовуються ними при виконанні курсової роботи з дисципліни «Алгоритмізація та програмування. Курсова робота». Ця дисципліна формує базові знання для вивчення наступних дисциплін: «Об'єктно-орієнтоване програмування», «Об'єктно-орієнтоване програмування. Курсова робота», які викладаються в наступних семестрах.

3. Зміст навчальної дисципліни

Лекційні заняття

Розділ 1. Поглиблення базових структур і пам'яті

Тема 1.1. Динамічне виділення пам'яті

Тема 1.2 Динамічні масиви (1D, 2D, багатовимірні)

Тема 1.3. Методи роботи з динамічними масивами

Розділ 2. Структури даних у процедурному підході

Тема 2.1. Структурний тип даних

Тема 2.2. Вказівники на структури

Тема 2.3. Вкладені структури і складні дані

Тема 2.4 Динамічні структури: списки, стек, черга

Тема 2.5. Хеш-таблиці та дерева

Розділ 3. Робота з файлами

Тема 3.1. Файлові вказівники і робота з файлами (stdio)

Тема 3.2. Форматований і бінарний файловий ввід-вивід

Тема 3.3 Низькорівневі функції роботи з файлами

Розділ 4. Сучасні підходи та передові методи у роботі з динамічними структурами та файлами

Тема 4.1. Вказівники на функції та callback

Тема 4.2. Бібліотеки і повторне використання коду

Тема 4.3. Оптимізація та профілювання коду

Тема 4.4. Процедурне програмування у великих системах

Лабораторні роботи

1. Динамічні масиви та робота з пам'яттю
2. Структури та вказівники
3. Динамічні структури даних (списки, стек, черга)
4. Робота з файлами (stdio + low-level)
5. Міні-проєкт «Інформаційна система»

4. Навчальні матеріали та ресурси

Основна література

1. Керніган, Б. В., Рітчі, Д. М. Мова програмування C / пер. з англ. — Київ : Видавнича група BHV, 2018. — 288 с.
2. Таненбаум, Е. Сучасні операційні системи / пер. з англ. — Київ : Вид-во «Діалектика», 2020. — 1136 с.
3. Дейтел, Г., Дейтел, П. Як програмувати мовою C. — 8-ме вид. — Київ : Діалектика, 2021. — 1056 с.
4. Шилдт, Г. С. Повний довідник. — 4-те вид. — Київ : Видавнича група BHV, 2019. — 896 с.
5. Prata, S. C Primer Plus. — 6th ed. — Boston : Addison-Wesley, 2014. — 984 p.
6. Bryant, R., O'Hallaron, D. Computer Systems: A Programmer's Perspective. — 3rd ed. — Boston : Pearson, 2016. — 1100 p.
7. Zelle, J. Python Programming: An Introduction to Computer Science. — 3rd ed. — Franklin, Beedle & Associates, 2017. — 552 p. (для розуміння алгоритмічних основ і порівняння з C).

Додаткова література

1. Wirth, N. Algorithms + Data Structures = Programs. — Prentice Hall, 1976. — 366 p. (класика структурного програмування).
2. Kernighan, B. The Practice of Programming. — Addison-Wesley, 1999. — 267 p. (про стиль і якість програмування).
3. Downey, A. Think C: How to Think Like a Computer Scientist. — Green Tea Press, 2012. — 300 p. (безкоштовна у відкритому доступі).
4. Крячок О.С. Алгоритмізація та програмування: Навчальний посібник до виконання комп'ютерного практикуму (адаптовано під Linux-системи) для студентів спеціальності 122 «Комп'ютерні науки» / О.С. Крячок; Київ: КПІ ім. Ігоря Сікорського, 2019. – 69 с. <https://ela.kpi.ua/handle/123456789/42794>.
5. Крячок О.С. Базові концепції програмування [електронний ресурс]: Навчальний посібник до виконання комп'ютерного практикуму для студентів спеціальності 122 – «Комп'ютерні науки», освітньо-професійної програми «Комп'ютерний моніторинг та геометричне моделювання процесів і систем» / О.С. Крячок; Київ: КПІ ім. Ігоря Сікорського, 2021. – 65 с. <https://ela.kpi.ua/handle/123456789/42795>.

- Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Лекційні заняття

№ з/п	Назва теми лекції та перелік основних питань	Кільк. ауд.год
Розділ 1. Поглиблення базових структур і пам'яті		
Тема 1.1. Динамічне виділення пам'яті		
1	Лекція 1. Динамічне виділення пам'яті: стек, купа, управління пам'яттю та практичні кейси.	2
Тема 1.2. Динамічні масиви (1D, 2D, багатовимірні)		
2	Лекція 2. Динамічні масиви та багатовимірні структури даних: реалізація, адаптація та застосування у наукових обчисленнях і графіці»	2
Тема 1.3. Методи роботи з динамічними масивами		
3	Лекція 3. Методи роботи з динамічними масивами: індексування, оптимізація пам'яті та обробка великих наборів даних	2
Розділ 2. Структури даних у процедурному підході		
Тема 2.1. Структурний тип даних		
4	Лекція 4. Структурні типи даних: створення, масиви структур та застосування для опису користувачів і баз даних у пам'яті	2
Тема 2.2. Вказівники на структури		

№ з/п	Назва теми лекції та перелік основних питань	Кільк. ауд.год
5	Лекція 5. Вказівники на структури і динамічне створення з прикладами списків та дерев	2
Тема 2.3. Вкладені структури і складні дані		
6	Лекція 6. Вкладені структури та складні дані з динамічними полями на прикладі JSON-подібних структур	2
Тема 2.4. Динамічні структури: списки, стек, черга		
7	Лекція 7. Динамічні структури списки стек черга на прикладі обробки черги задач	2
Тема 2.5. Хеш-таблиці та дерева		
8	Лекція 8. Хеш-таблиці та бінарні дерева пошуку на прикладі словників і баз даних у пам'яті	2
Розділ 3. Робота з файлами		
Тема 3.1. Файлові вказівники і робота з файлами (stdio)		
10	Лекція 9. Файлові вказівники і робота з файлами stdio для зберігання структурованих даних	2
Тема 3.2. Форматований і бінарний файловий ввід-вивід		
11	Лекція 10. Форматований і бінарний файловий ввід-вивід з серіалізацією масивів та структур	2
Тема 3.3. Низькорівневі функції роботи з файлами		
12	Лекція 11. Низькорівневі функції роботи з файлами дескриптори буферизація та приклади Unix Linux	2
Розділ 4. Сучасні підходи та передові методи у роботі з динамічними структурами та файлами		
Тема 4.1. Вказівники на функції та callback		
13	Лекція 12. Вказівники на функції та callback таблиці функцій і емульований поліморфізм	2
Тема 4.2. Бібліотеки і повторне використання коду		
14	Лекція 13. Бібліотеки і повторне використання коду статичні та динамічні бібліотеки	2
Тема 4.3. Оптимізація та профілювання коду		
15	Лекція 14. Оптимізація та профілювання коду аналіз складності та оптимізація пам'яті	2
Тема 6.1. Процедурне програмування у великих системах		
16	Лекція 15. Процедурне програмування у великих системах, використання API та взаємодія з ОС	2

Практичні роботи:

№	Назва лабораторної роботи	Кільк. ауд.год
1	Динамічні масиви та робота з пам'яттю. Обробка числових та текстових даних у динамічних масивах	2
2	Структури та вказівники Управління даними через структури та вказівники	2
3	Динамічні структури даних: списки, стек, черга. Моделювання прикладних задач з використанням динамічних структур	2
4	Робота з файлами (текстові та бінарні) Зберігання та обробка структурованих даних у файлах	2
5	Міні-проєкт "Інформаційна система". Створення консольної інформаційної системи з процедурним програмуванням	4
Модульна контрольна робота		2

Лабораторні роботи:

№	Назва лабораторної роботи	Кільк. ауд.год
1	Динамічні масиви та робота з пам'яттю. Виділення та звільнення динамічної пам'яті, створення одномірних та двовимірних масивів, обробка даних в масивах (пошук, сортування, обчислення функцій над елементами).	2
2	Структури та вказівники Оголошення та ініціалізація структури, робота з масивами структур та вказівниками на структури, сортування, пошук і обробку структурованих даних.	2
3	Динамічні структури даних (списки, стек, черга) Створення однозв'язних та двозв'язних списків, реалізація стеку і черги, вставка, видалення та пошук елементів, моделювання прикладні задачі на цих структурах.	2
4	Робота з файлами (stdio + low-level) Використання файлових вказівників, форматоване читання та запис даних, дескриптори файлів, серіалізацію структур та масивів, обробку даних у файлах.	4
5	Міні-проєкт «Інформаційна система» Створення повноцінної прикладної системи на основі процедурного програмування: динамічні масиви та структури, вказівники, робота з файлами, реалізація CRUD-операцій (додавання, видалення, редагування, пошук), сортування та навігація через меню.	4

6. Самостійна робота студента

Самостійна робота студента (62 години) передбачає підготовку до аудиторних занять та контрольних заходів, проведення розрахунків та підготовка вхідних даних до роботи.

Розподіл годин СРС: підготовка до заліку – 6 годин; підготовка до лабораторного практикуму – 1,5 година; підготовка до МКР – 3 години; підготовка до лекції – 1 година.

- Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Відвідування лекційних та лабораторних занять є обов'язковим за винятком поважних причин (хвороби, форс-мажорних обставин).

В разі пропущення занять з поважних причин викладач надає можливість студенту виконати усі або деякі лабораторні завдання (винятком є виконання деяких завдань у зв'язку із закінченням навчального процесу).

В разі пропущення занять без поважних причин, а також через порушення граничного терміну виконання завдання (deadline) студент може отримати 80% від максимальної оцінки відповідне завдання.

При захисті практичних робіт студент має продемонструвати розроблений програмний код та результати його виконання на тестових прикладах, як заздалегідь підготованих, так і запропонованих викладачем. У випадку дистанційної форми навчання захист відбувається на відповідній конференції шляхом демонстрації екрана.

Протягом семестру студенти:

- виконують та захищають лабораторні роботи у відповідні терміни,
- пишуть модульну контрольну роботу, контрольні роботи пишуться без застосування допоміжних засобів (мобільні телефони, планшети та ін.); результат пересилається у файлі викладачу;
- повинні позитивно закрити дві атестації,
- заохочувальні бали виставляються за: активну участь на лекціях; участь у факультетських та інститутських олімпіадах з навчальних дисциплін, участь у конкурсах робіт, підготовка оглядів наукових праць; презентацій по одній із тем СРС дисципліни тощо. Кількість заохочуваних балів не більше 10.
- по закінченні навчального процесу складають залік.

Політика та принципи академічної доброчесності визначені у розділі 3 Кодексу честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського». Детальніше: <https://kpi.ua/code>.

Норми етичної поведінки Норми етичної поведінки студентів і працівників визначені у розділі 2 Кодексу честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського». Детальніше: <https://kpi.ua/code>.

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

1. Поточний контроль (виконання робіт в семестрі) оцінюється у 100 балів. Розподіл балів наведено в таблиці:

Роботи	Максимальна кількість балів за виконання однієї роботи
Лабораторна робота 1	14
Лабораторна робота 2	14
Лабораторна робота 3	16
Лабораторна робота 4	16
Лабораторна робота 5	20
Модульна контрольна робота	20
Загальний бал	100

Штрафні бали віднімаються за:

1. неоптимальний алгоритм – 10% від максимальної кількості балів;

2. неоптимальні структури представлення інформації – 10% від максимальної кількості балів;
3. ненадану або невірну відповідь на запитання – 20% від максимальної кількості балів при захисті лабораторної роботи або 100% - на контрольній роботі.

2. Календарний контроль проводиться двічі на семестр як моніторинг поточного стану виконання вимог силябусу:

Критерій	Перший календарний контроль	Другий календарний контроль
Термін календарного контролю	Тиждень 7-8	Тиждень 14-15
Умови отримання позитивної оцінки	≥ 24 бали	≥ 45 балів

3. Умови допуску до заліку: відсутність заборгованостей з лабораторних робіт 1 - 5.

Студенти, які виконали умови допуску до заліку та мають рейтингову оцінку 60 і більше балів, отримують відповідну до набраного рейтингу оцінку без додаткових випробувань.

Зі студентами, які виконали умови допуску до заліку та мають рейтингову оцінку менше 60 балів, а також з тими студентами, хто бажає підвищити свою рейтингову оцінку, на останньому за розкладом занятті з дисципліни в семестрі проводиться семестровий контроль у вигляді заліку.

4. Відповідь на заліку оцінюється у 100 балів. Залікова робота складається з одного теоретичного питання та одного практичного завдання. Ваговий бал теоретичного питання – 50 балів, завдання – 50 балів.

Теоретична частина оцінюється таким чином:

- правильна чітко викладена, повна відповідь – (не менше 90% потрібної інформації) – 45-50 балів;
- достатньо повна відповідь (не менше 75% потрібної інформації) – 38-44 бали;
- неповна відповідь (не менше 60% потрібної інформації) – 30-37 балів;
- незадовільна відповідь – 0 балів.

Практичне завдання оцінюється таким чином:

- повне, безпомилкове розв'язування завдання – 45-50 балів;
- повне, розв'язування завдання із несуттєвими невідповідностями – 38-44 бали;
- завдання виконане з певними недоліками – 30-37 балів;
- завдання не виконано – 0 балів.

5. Рейтингова оцінка за освітній компонент за бажанням студента визначається одним з таких способів:

- 1) кількість балів, отриманих за поточний контроль, або
- 2) результат виконання залікової контрольної роботи (тоді не враховуються бали, отримані в семестрі).

Таблиця переведення рейтингових балів у оцінку за університетською шкалою:

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

Дистанційне навчання Можливе синхронне дистанційне навчання з використанням платформ для відео-конференцій та освітньої платформи для дистанційного навчання в університеті.

Інклюзивне навчання Допускається

9. Додаткова інформація з дисципліни (освітнього компонента)

Стартові бали можуть бути перезараховані як результати навчання, одержані в неформальній освіті, за наявності у студента сертифікату проходження дистанційних чи онлайн курсів за відповідною тематикою, викладач закриває відповідну частину курсу (лабораторні чи лекції). В якості прикладу, теми розділу 1-2 можуть бути зараховані за наявності сертифікату з проходження курсу «Спеціалізація C++: Data Structures and Algorithms» за посиланням <https://www.coursera.org/specializations/codio-cpp-dsa>; теми розділу 4 можуть бути зараховані за наявності сертифікату з проходження курсу «Спеціалізація Programming in C++: A Hands-on Introduction» за посиланням <https://www.coursera.org/specializations/hands-on-cpp>.

Робочу програму навчальної дисципліни (силабус):

Складено старшим викладачем Беспалою Ольгою Миколаївною

Ухвалено кафедрою ЦТЕ (протокол № 22 від 25.06.2025)

Погоджено Методичною радою НН ІАТЕ КПІ ім. Ігоря Сікорського (протокол № 9 від 27.06.2025)