

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ

БАКАЛАВР
(освітній ступінь)

Кафедра інформаційних технологій

«Затверджую»

Голова НМР факультету автоматизації і
інформаційних технологій

_____ / Олександр ТЕРЕНТЬЄВ/

« ____ » _____ 2024 року

РОБОЧА ПРОГРАМА ОСВІТНЬОЇ КОМПОНЕНТИ

ОК 5 «Алгоритмізація та програмування»

(шифр та назва освітньої компоненти)

Шифр	назва спеціальності, освітньої програми
F7	<i>Комп'ютерна інженерія, Комп'ютерні системи та мережі</i>

Мова викладання: українська

Розробник:

Олександр ПОПЛАВСЬКИЙ, кандидат технічних наук, доцент

(ім'я та прізвище, науковий ступінь, звання)

(підпис)

Робоча програма затверджена на засіданні кафедри інформаційних технологій,
протокол № ____ від « ____ » **2024 року**

Завідувачка кафедри ІТ _____
(підпис)

Тетяна ГОНЧАРЕНКО

Схвалено гарантом освітньої програми

Гарант ОП _____
(підпис)

Шабала Є.Є.

Розглянуто на засіданні науково-методичної комісії спеціальності F7
«Комп'ютерна інженерія»

Протокол № ____ від « ____ » **2024 року.**

ВИТЯГ З РОБОЧОГО НАВЧАЛЬНОГО ПЛАНУ НА 2024-2025 НАВЧАЛЬНИЙ РІК

Шифр	Назва спеціальності, освітньої програми	Форма здобуття ВО: денна											Форма контролю	Семестр	Погодження заступником декана факультету
		Кількість кредитів	Кількість годин						Кількість індивідуальних робіт						
			Всього	Разом	Аудиторних			Самостійна робота							
					Лекції	Лабораторні	Практичні								
F7	Комп'ютерна інженерія, Комп'ютерні системи та мережі	4,5	135	60	30	30		75			1		Екз	1	
F7	Комп'ютерна інженерія, Комп'ютерні системи та мережі	4,0	120	60	30	30		60			1		Залік	2	

Анотація. Мета та завдання освітньої компоненти

Посилання на сторінку електронного навчально-методичного комплексу дисципліни: [Курс: https://org2.knuba.edu.ua/course/view.php?id=254](https://org2.knuba.edu.ua/course/view.php?id=254)

Навчальна дисципліна **«Алгоритмізація та програмування»** має на меті забезпечити здобувачів вищої освіти базовими теоретичними знаннями та практичними навичками з алгоритмізації та програмування, необхідними для розв'язання прикладних інженерних задач і створення коректного, надійного та ефективного програмного забезпечення для комп'ютерних систем і компонентів обчислювальної техніки. Дисципліна спрямована на формування алгоритмічного та системного мислення, розвиток логічного аналізу задач і опанування фундаментальних принципів програмування, що є основою професійної підготовки фахівців у галузі комп'ютерної інженерії.

Завдання навчальної дисципліни **«Алгоритмізація та програмування»** полягає у формуванні у здобувачів вищої освіти базових знань і вмінь, необхідних для побудови алгоритмів та їх програмної реалізації з урахуванням вимог коректності, надійності та ефективності програмного коду. У процесі вивчення дисципліни студенти ознайомлюються з поняттям алгоритму, його властивостями, основними структурами даних і класичними алгоритмами, що застосовуються під час розв'язання прикладних задач у комп'ютерній інженерії.

Важливим аспектом навчання є розвиток навичок аналізу та оптимізації алгоритмів, а також формування вмінь розв'язання задач шляхом програмної реалізації алгоритмів. Особлива увага приділяється засвоєнню основних методів і прийомів програмування, зокрема вивченню синтаксису та семантики мови програмування C++, використанню базових конструкцій мови, функцій, масивів, структур даних і файлів.

Навчальна дисципліна передбачає ознайомлення із сучасними середовищами розробки програмного забезпечення, набуття навичок використання засобів налагодження та тестування програм, а також формування початкових умінь аналізу програмних помилок і забезпечення коректності виконання програм. У контексті підготовки фахівців з комп'ютерної інженерії акцент робиться на дотриманні принципів структурованого програмування, раціональній організації програмного коду та підвищенні його ефективності.

Окремим завданням дисципліни є розвиток логічного та критичного мислення здобувачів вищої освіти. Студенти опановують методи декомпозиції задач, що дозволяє розв'язувати складні прикладні задачі шляхом їх поділу на підзадачі. У процесі навчання також формується вміння самостійної та командної роботи під час виконання практичних завдань і розрахунково-графічної роботи.

Важливим результатом вивчення дисципліни є формування основ алгоритмічного мислення. Студенти навчаються використовувати програмний код, блок-схеми та інші формальні засоби для проєктування алгоритмів, а також ознайомлюються з базовими підходами до програмування, зокрема структурним та об'єктно-орієнтованим. Це створює підґрунтя для подальшого вивчення професійно орієнтованих дисциплін з комп'ютерної інженерії та розробки

програмного забезпечення для комп'ютерних систем і обчислювальних комплексів.

Вивчення дисципліни «Алгоритмізація та програмування» сприяє формуванню у здобувачів вищої освіти компетентностей, необхідних для подальшої професійної діяльності у галузі комп'ютерної інженерії.

Компетентності здобувачів освітньої програми, що формуються в результаті засвоєння освітньої компоненти

Код	ЗМІСТ КОМПЕТЕНТНОСТІ
Інтегральна компетентність	
ІК	Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми під час професійної діяльності в комп'ютерній галузі або навчання, що передбачає застосування теорій та методів комп'ютерної інженерії і характеризується комплексністю та невизначеністю умов.
Загальні компетентності	
Z1	Здатність до абстрактного мислення, аналізу і синтезу.
Z2	Здатність вчитися і оволодівати сучасними знаннями.
Z3	Здатність застосовувати знання у практичних ситуаціях.
Z7	Вміння виявляти, ставити та вирішувати проблеми.
Спеціальні (фахові, предметні) компетентності	
P2	Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.
P3	Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж.
P15	Здатність аргументувати вибір методів розв'язування спеціалізованих задач, критично оцінювати отримані результати, обґрунтовувати та захищати прийняті рішення.

Програмні результати здобувачів освітньої програми, що формуються в результаті засвоєння освітньої компоненти

Код	ПРОГРАМНІ РЕЗУЛЬТАТИ
Знання	
N1	Знати і розуміти наукові положення, що лежать в основі функціонування комп'ютерних засобів, систем та мереж.
N3	Знати новітні технології в галузі комп'ютерної інженерії.
Уміння	
N6	Вміти застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей.
N7	Вміти розв'язувати задачі аналізу та синтезу засобів, характерних для спеціальності.

N8	Вміти системно мислити та застосовувати творчі здібності до формування нових ідей.
N11	Вміти здійснювати пошук інформації в різних джерелах для розв'язання задач комп'ютерної інженерії.
N16	Вміти оцінювати отримані результати та аргументовано захищати прийняті рішення.
Комунікація	
N17	Спілкуватись усно та письмово з професійних питань українською мовою та однією з іноземних мов (англійською, німецькою, італійською, французькою, іспанською).

ЗМІСТ КУРСУ

Змістовий модуль 1. Основи програмування та алгоритмізації

Лекція 1. Комп'ютер як основний інструмент програмування. Будова пам'яті комп'ютера. Програмне забезпечення.

Лекція 2. Мови програмування. C, C++.

Лекція 3. Поняття алгоритму та правила його складання.

Лекція 4. Блок-схеми алгоритмів. Застосування циклів з лічильником

Лекція 5. Структура програми на мові C, C++.

Лекція 6. Елементи мов програмування C, C++. Оператори та операнди.

Лекція 7. Змінні, оператори, модулі.

Лекція 8. Операції мов програмування C, C++. Арифметичні операції.

Лекція 9. Операції порівняння та логічні операції.

Лекція 10. Порозрядні (побітові) операції. Унарні операції.

Лекція 11. Функції вводу-виводу C, C++. Бібліотека `conio.h`.

Лекція 12. C++ введення виведення потоками.

Лекція 13. Циклічні процеси. Цикл `while`. Цикл `do`. Цикл `for`.

Лекція 14. Масиви. Застосування масивів.

Лекція 15. Масиви. Сортування.

Змістовий модуль 2: Основи роботи з мовою C++

Лекція 16. Вказівники та масиви.

Лекція 17. Функції користувача. Підпрограми. Види підпрограм. Функції на мові C/C++.

Лекція 18. Формальні і фактичні параметри.

Лекція 19. Масиви як параметри функцій. Класи пам'яті.

Лекція 20. Структури. Використання структур. Оголошення і визначення структур. Доступ до членів структур. Ініціалізація структур. Масиви структур.

Лекція 21. Графіка VS C++. Малювання у консольному вікні. Функції малювання. Малювання геометричних фігур.

Лекція 22. Вказівники

Лекція 23. Тип даних «текстовий рядок». Дії з рядками.

Лекція 24. Визначення та види рекурсії.

Лекція 25. Прості фігури. Малювання із застосуванням класів.

Лекція 26. Робота з файлами. Робота з текстовими файлами за допомогою файлових потоків. Створення потоків. Перевірка помилок виконання файлових операцій. Символьне введення / виведення.

Лекція 27 Вказівники на функції

Лекція 28 Створення проєктів. Початок роботи над проєктом.

Лекція 29. Компоненти. Події.

Лекція 30. Функція обробки події.

Теми лабораторних занять

№	Назва теми	К-сть годин	К-сть балів
Змістовий модуль 1.			
1.	Робота з IDE Microsoft Visual Studio. Програмування алгоритмів лінійної структури, компіляція та відлагодження програми	4	4
2.	Програма, що розгалужується	4	4
3.	Циклічні структури	4	4
4.	Одновимірні масиви	4	4
5.	Функції. Варіативні функції.	4	4
6.	Багатовимірні масиви	4	4
7.	Структуровані типи даних	4	4
8.	Малювання в Visual Studio	2	2
	Разом за 1 семестр	30	30
Змістовий модуль 2.			
8.	Малювання в Visual Studio	2	2
9.	Показчики	4	2
10.	Рядки типу char*	4	2
11.	Файли	4	2
12.	Рекурсивні функції в мові програмування C/C++	4	2
13.	Використання динамічних числових масивів	4	4
14.	Розробка багатофайлових проєктів	4	4
15.	Використання масивів типу char і рядків	4	2
	Разом за 2 семестр	30	20

Розподіл годин самостійної роботи здобувачів

№	Назва теми	К-сть годин
	Змістовний модуль 1.	
1	Ознайомлення з налаштуваннями IDE для роботи з проєктами.	5

	Вивчення стандартних налаштувань компіляції та налагодження.	
2	Налагодження програм у середовищі Visual Studio. Використання точок зупину (breakpoints), покрокове виконання програм, перегляд значень змінних під час виконання.	4
3	Аналіз помилок компіляції та виконання програм. Типові синтаксичні та логічні помилки у програмах мовою C/C++. Методи їх виявлення та усунення.	4
4	Вивчення умовних операторів (if-else, switch). Побудова блок-схем для програм із розгалуженнями.	4
5	Опрацювання роботи циклів for, while, do-while. Приклади застосування вкладених циклів.	4
6	Сортування масивів різними алгоритмами (бульбашкове, вибір, вставки). Робота з пошуком мінімуму/максимуму в масивах.	4
7	Типи параметрів функцій (формальні, фактичні). Використання функцій із передачею масивів.	4
8	Реалізація програм для обробки матриць (додавання, множення). Використання вкладених циклів для роботи з багатовимірними масивами.	4
	Виконання РГР	12
	Підготовка до екзамену	30
	Разом за 1 семестр	75
	Змістовний модуль 2	
9	Різновиди структурованих типів даних у C++. Практичне використання структур і масивів структур.	3
10	Робота з динамічною пам'яттю та покажчиками. Використання покажчиків для роботи з масивами та функціями.	4
11	Використання функцій для роботи з рядками (додавання, порівняння, копіювання). Особливості роботи з рядками у C++ в порівнянні з бібліотекою std::string.	4
12	Основи роботи з текстовими файлами за допомогою файлових потоків. Методи перевірки помилок при роботі з файлами.	3
13	Основи роботи з графікою у VS C++. Реалізація простих	4

	графічних програм із малюванням геометричних фігур.	
14	Підготовка базових компонентів і подій для роботи з проектами. Реалізація функцій обробки подій.	4
	Виконання РГР	12
	Підготовка до заліку	26
	Разом за 2 семестр	60
	Всього	135

Індивідуальні завдання: Розрахунково-графічна робота

Виконання РГР згідно індивідуального варіанту для закріплення теоретичних знань і практичних навичок з Алгоритмізації та програмування у 1 та 2 семестрах.

Для забезпечення об'єктивності та прозорості оцінювання РГР включає 40 балів за перший семестр та 50 балів за другий семестр у загальній підсумковій оцінці з можливістю отримання додаткових 10 балів за особливі досягнення такі як участь Здобувача у міжнародній або всеукраїнській науково-практичній конференції з публікацією у матеріалах конференції тез виступу (доповіді) на одну з тем, дотичних до змісту освітньої компоненти, або публікація статті на одну з таких тем в інших наукових виданнях. Оцінювання здійснюється за чітко визначеними критеріями, що дозволяють врахувати всі аспекти виконаної роботи, від коректності вирішення завдання до якості коду та документації.

РГР підлягає захисту здобувачем на заняттях, які призначаються додатково. Під час захисту Здобувач повинен продемонструвати роботу розробленого програмного забезпечення.

РГР повинна вміщувати пояснювальну записку, яка повинна мати обсяг від 18 до 24 сторінок А4 тексту (кегель Times New Roman, шрифт 14, інтервал 1,5), включно з планом, структурою основної частини тексту відповідно до плану, висновками та списком літератури, складеним відповідно до ДСТУ 8302:2015.

В основній частині повинно бути проведено дослідження предметної області згідно варіанту, сформовано вимоги до програмного забезпечення, розроблена його архітектура, проаналізована його структура та проведена специфікація програмних модулів, наведено принтскріни готового програмного забезпечення та його код (у додатку).

Література, що рекомендується для виконання РГР, наведена у цій робочій програмі, а в електронному вигляді вона розміщена на освітньому сайті КНУБА, на сторінці кафедри.

Також як частину виконання РГР за рішенням викладача може бути зарахована участь Здобувача у міжнародній або всеукраїнській науково-практичній конференції з публікацією у матеріалах конференції тез виступу (доповіді) на одну з тем, дотичних до змісту дисципліни, або публікація статті на одну з таких тем в інших наукових виданнях.

Розподіл балів для РГР (1 семестр)

Аналіз та проектування	Розробка програми	Оформлення та захист роботи	Сума
~20	~10	~10	40

Розподіл балів для РГР (2 семестр)

Аналіз та проектування	Розробка програми	Оформлення та захист роботи	Сума
~10	~20	~20	50

Варіанти РГР

1. Розробка програми для обліку посібників з навчальних дисциплін.
2. Розробка програми калькуляції вартості страв на основі даних про кулінарні рецепти.
3. Розробка програми продажу квитків на поточні авіарейси.
4. Розробка програми роботи з відомостями про маршрути громадського транспорту.
5. Розробка програма – утилізатора файлів.
6. Розробка програми для підтримки процесу курсового проектування.
7. Розробка програми ведення ділового щоденника.
8. Розробка програми ведення обліку земельних ділянок населених пунктів.
9. Розробка програми ведення обліку учбового навантаження викладачів кафедри (“Вчений секретар”).
10. Розробка програми ведення розрахунків за комунальні платежі ЖЕКа.
11. Розробка програми графічного редактора зі збереженням зображення, списку об’єктів та їх параметрів у файлі.
12. Розробка програми для ведення довідника про випускників університету.
13. Розробка програми для контролю виконання студентами програми певної дисципліни.
14. Розробка програми для обліку пацієнтів ветеринарної клініки.
15. Розробка програми каталогізації носіїв інформації.
16. Розробка програми обліку відвідування занять студентами.
17. Розробка програми обліку зайнятості авто в агенції по прокату автомобілів.
18. Розробка програми обліку замовлень в агенції таксі.
19. Розробка програми обліку ліків в аптеці.
20. Розробка програми обліку міжміських та міжнародних телефонних переговорів.
21. Розробка програми обліку постояльців в готелі.
22. Розробка програми обліку спортивних змагань.
23. Розробка програми обліку товарів, що зберігаються на складах.
24. Розробка програми продажу путівок туристичної агенції.
25. Розробка програми роботи з об’єктами вулично-дорожньої мережі міста.
26. Розробка програми формування меню ресторану.
27. Розробка програми формування програми телепередач на каналах.
28. Розробка програми формування сітки мовлення телевізійного каналу.

Методи контролю та оцінювання знань

Загальне оцінювання здійснюється через вимірювання результатів навчання у формі проміжного (модульного) та підсумкового контролю (залік, захист індивідуальної роботи тощо) відповідно до вимог зовнішньої та внутрішньої системи забезпечення якості вищої освіти.

Політика щодо академічної доброчесності

Тексти індивідуальних завдань (в т.ч. у разі, коли вони виконуються у формі презентацій або в інших формах) можуть перевірятись на плагіат. Для цілей захисту індивідуального завдання оригінальність тексту має складати не менше 70%.

Виключення становлять випадки зарахування публікацій Здобувачів у матеріалах наукових конференціях та інших наукових збірниках, які вже пройшли перевірку на плагіат.

Списування під час тестування та інших опитувань, які проводяться у письмовій формі, заборонені (в т.ч. із використанням мобільних девайсів). У разі виявлення фактів списування з боку здобувача він отримує інше завдання. У разі повторного виявлення призначається додаткове заняття для проходження тестування.

Політика щодо відвідування

Здобувач, який пропустив аудиторне заняття з поважних причин, має продемонструвати викладачу та надати до деканату факультету документ, який засвідчує ці причини.

За об'єктивних причин (хвороба, міжнародне стажування, наукова та науково-практична конференція (круглий стіл) тощо) навчання може відбуватись в он-лайн формі за погодженням із керівником курсу.

Методи контролю

Основні форми участі здобувачів у навчальному процесі, що підлягають поточному контролю: виступ на практичних заняттях; доповнення, опонування до виступу, рецензія на виступ; участь у дискусіях; аналіз першоджерел; письмові завдання (тестові, індивідуальні роботи у формі рефератів); та інші письмові роботи, оформлені відповідно до вимог. Кожна тема курсу, що винесена на лекційні та практичні заняття, відпрацьовується Здобувачами у тій чи іншій формі, наведеній вище. Обов'язкова присутність на лекційних заняттях, активність впродовж семестру, відвідування/відпрацювання усіх аудиторних занять, виконання інших видів робіт, передбачених навчальним планом з цієї дисципліни.

Під час оцінювання рівня знань здобувача аналізу підлягають:

- характеристики відповіді: цілісність, повнота, логічність, обґрунтованість, правильність;
- якість знань (ступінь засвоєння фактичного матеріалу): осмисленість, глибина, гнучкість, дієвість, системність, узагальненість, міцність;
- ступінь сформованості умінь поєднувати теорію і практику під час розгляду ситуацій, практичних завдань;
- рівень володіння розумовими операціями: вміння аналізувати, синтезувати, порівнювати, абстрагувати, узагальнювати, робити висновки з проблем, що розглядаються;
- досвід творчої діяльності: вміння виявляти проблеми, розв'язувати їх, формувати гіпотези;
- самостійна робота: робота з навчально-методичною, науковою, допоміжною вітчизняною та зарубіжною літературою з питань, що розглядаються, вміння отримувати інформацію з різноманітних джерел (традиційних; спеціальних періодичних видань, ЗМІ, Internet тощо).

Тестове опитування може проводитись за одним або кількома змістовими модулями. В останньому випадку бали, які нараховуються Здобувачу за відповіді на тестові питання, поділяються між змістовими модулями.

Результати **поточного контролю** заносяться до журналу обліку роботи. Позитивна оцінка поточної успішності Здобувачів за відсутності пропущених та невідпрацьованих практичних занять та позитивні оцінки за індивідуальну роботу є підставою для допуску до підсумкової форми контролю. Бали за аудиторну роботу відпрацьовуються у разі пропусків.

Підсумковий контроль здійснюється під час проведення залікової сесії з урахуванням підсумків поточного та модульного контролю. Під час семестрового контролю враховуються результати здачі усіх видів навчальної роботи згідно зі структурою кредитів.

Оцінювання проводиться за 100-бальною шкалою.

Розподіл балів для дисципліни з формою контролю екзамен (семестр 1)

Поточне оцінювання		Екзамен	Сума
Змістовий модуль № 1	РГР		
30	40	30	100

Розподіл балів для дисципліни з формою контролю залік (семестр 2)

Поточне оцінювання		Залік	Сума
Змістовий модуль № 2	РГР		
20	50	30	100

Політика щодо академічної доброчесності

Тексти індивідуальних завдань (в т.ч. у разі, коли вони виконуються у формі презентацій або в інших формах) можуть перевірятись на плагіат. Для цілей захисту індивідуального завдання оригінальність тексту має складати не менше 70%. Виключення становлять випадки зарахування публікацій Здобувачів у матеріалах наукових конференціях та інших наукових збірниках, які вже пройшли перевірку на плагіат.

Списування під час тестування та інших опитувань, які проводяться у письмовій формі, заборонені (в т.ч. із використанням мобільних девайсів). У разі виявлення фактів списування з боку здобувача він отримує інше завдання. У разі повторного виявлення призначається додаткове заняття для проходження тестування.

Політика щодо відвідування

Здобувач, який пропустив аудиторне заняття з поважних причин, має продемонструвати викладачу та надати до деканату факультету документ, який засвідчує ці причини.

За об'єктивних причин (хвороба, міжнародне стажування, наукова та науково-практична конференція (круглий стіл) тощо) навчання може відбуватись в он-лайн формі за погодженням із керівником курсу.

Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою
90-100	A	Відмінно
82-89	B	Добре
74-81	C	
64-73	D	Задовільно
60-63	E	
35-59	FX	Не зараховано з можливістю повторного складання
0-34	F	Не зараховано з обов'язковим повторним вивченням дисципліни

Рекомендована література Базова

1. Іванов Є.О., Ліндер Я.М., Жереб К.А. **Основи мови програмування C++**: навчальний посібник. – К.: Логос, 2020. – 90 с.;
2. Stroustrup B. **A Tour of C++** .– Addison-Wesley Professional, 2018.– 256p.
3. Зеленський О.С., Лисенко В.С. **Основи програмування на C++**: навчальний посібник.– К.Р.: ДУЕТ, 2023.– 269с.

Допоміжна

1. Grigoryan V., Wu S., Expert C++: Become a proficient programmer by learning coding best practices with C++17 and C++20's latest features, 2nd Edition. – Packt Publishing, 2020. – 606p
2. Davidson J., Gregory K., Beautiful C++: 30 Core Guidelines for Writing Clean, Safe, and Fast Code. – Addison-Wesley Professional, 2021. – 352p.
3. O. Poplavskyi, "Systems development for processing and analysis of biomedical data based on artificial intelligence," in Proc. Ninth International Scientific-Practical Conference «Management of the Development of Technologies», Kyiv, Ukraine, 2022, pp. 46-47.
4. В. В. Гилюк, В. В. Литвиненко, О. А. Поплавський, П. Крук, "Хмарні технології", на III міжнародна науково-практична конференція «Розподілені програмні системи і технології», Київ, Україна, 2022.
5. Програмування та алгоритмічні мови : методичні вказівки до виконання лабораторних робіт / Київ. нац. ун-т буд-ва і архіт. ; укладач О. А. Поплавський., І. В. Босенко – Київ : КНУБА, 2024. – 55 с. : іл. - Бібліогр. : с. 54.
<https://repository.knuba.edu.ua/handle/123456789/13612>

Інформаційні ресурси

1. <http://library.knuba.edu.ua/>
2. <http://org.knuba.edu.ua/>
3. <https://visualstudio.microsoft.com/ru/free-developer-offers/>