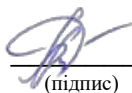


Міністерство освіти і науки України
Національний аерокосмічний університет
«Харківський авіаційний інститут»

Кафедра систем управління літальних апаратів (№ 301)

ЗАТВЕРДЖУЮ

Гарант освітньої програми



Олена ГАВРИЛЕНКО

(підпис)

« 29 » _____ серпня 2025 р.

**СИЛАБУС ОBOB'ЯЗKОВОЇ
HABЧАЛЬНОЇ ДИСЦИПЛІНИ**

Алгоритмізація та програмування

(назва навчальної дисципліни)

Галузь знань: G «Інженерія, виробництво та будівництво»
(шифр і найменування галузі знань)

Спеціальність: G7 «Автоматизація, комп'ютерно-інтегровані
технології та робототехніка»
(код і найменування спеціальності)

Освітня програма: Інженерія мобільних додатків
(найменування освітньої програми)

Рівень вищої освіти: перший (бакалаврський)

Силабус введено в дію з 01.09.2025

Харків – 2025 р.

Розробник:

Олена ГАВРИЛЕНКО, доцент кафедри систем управління
літальних апаратів (№301), к.т.н., доцент

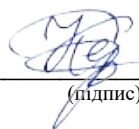


(підпис)

Силабус навчальної дисципліни розглянуто на засіданні кафедри систем
управління літальних апаратів (№ 301)

Протокол № 1 від “28” серпня 2025 р.

Завідувач кафедри 301 к.т.н., доцент


(підпис)

Костянтин ДЕРГАЧОВ

Погоджено з представником здобувачів освіти:

здобувач групи 360


(підпис)

Олександр ГОРБАЧ
(ім'я та прізвище)

1. Загальна інформація про викладача



ПІБ: Гавриленко Олена Володимирівна

Посада: доцент кафедри Систем управління літальних апаратів

Науковий ступінь: кандидат технічних наук

Вчене звання: доцент

Перелік дисциплін, які викладає:

Алгоритмізація та програмування. Об'єктно-орієнтоване проектування систем управління. Математичні основи цифрових систем. Автоматизація інформаційно-управляючих процесів

Напрями наукових досліджень: інформаційні технології, інтелектуальні навчаючі системи;

Контактна інформація:

o.havrylenko@khai.edu

2. Опис навчальної дисципліни

Форма здобуття освіти	Денна
Семестр	1, 2
Мова викладання	Українська
Тип дисципліни	Обов'язкова
Обсяг дисципліни: кредити ЄКТС/ кількість годин	<u>денна:</u> 1-й семестр: 6,5 кредитів ЄКТС / 195 годин (72 аудиторних, з яких: лекції – 32, лабораторні – 40; СРЗ – 123); 2-й семестр: 6,5 кредитів ЄКТС / 195 годин (80 аудиторних, з яких: лекції – 32, лабораторні – 24, практичні – 24; СРЗ – 115)
Види навчальної діяльності	Лекції, практичні та лабораторні заняття, виконання розрахункової роботи, самостійна робота
Види контролю	Поточний контроль, модульний контроль, захист розрахункової роботи, семестровий контроль – іспит (семестри 1 і 2)
Пререквізити	немає

3. Мета та завдання навчальної дисципліни, переліки компетентностей та очікуваних результатів навчання

Мета – освоєння студентами методів і засобів проектування і реалізації алгоритмів обробки даних, а також структурного підходу до побудови програмного забезпечення комп'ютеризованих систем.

Завдання – вивчення методів проектування алгоритмів, засвоєння синтаксичних конструкцій в мові програмування високого рівня, а також здобуття навичок реалізації програмного забезпечення і розробки UML діаграм на прикладі вирішення спрощених інженерних задач.

Компетентності, які набуваються:

Інтегральна компетентність:

Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми у певній галузі професійної діяльності або у процесі навчання, що передбачає застосування певних теорій та методів відповідної науки і характеризується комплексністю та невизначеністю умов.

Загальні компетентності (ЗК)

ЗК1. Здатність застосовувати знання у практичних ситуаціях.

ЗК2. Здатність спілкуватися державною мовою як усно, так і письмово.

ЗК3. Здатність спілкуватися іноземною мовою.

ЗК4. Навички використання інформаційних і комунікаційних технологій.

ЗК5. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ЗК8. Здатність працювати в команді.

Спеціальні компетентності (ФК)

ФК7. Здатність обґрунтовувати вибір технічної структури та вміти розробляти прикладне програмне забезпечення для мікропроцесорних систем керування на базі локальних засобів автоматизації, промислових логічних контролерів та програмованих логічних матриць і сигнальних процесорів.

ФК9. Здатність вільно користуватись сучасними комп'ютерними та інформаційними технологіями для вирішення професійних завдань, програмувати та використовувати прикладні та спеціалізовані комп'ютерно-інтегровані середовища для вирішення задач автоматизації із застосуванням інженерії мобільних додатків.

Програмні результати навчання (ПРН):

ПРН3. Вміти застосовувати сучасні інформаційні технології та мати навички розробляти алгоритми та комп'ютерні програми з використанням мов високого рівня та технологій об'єктно-орієнтованого програмування, створювати бази даних та використовувати інтернет-ресурси.

ПРН10. Вміти обґрунтовувати вибір структури та розробляти прикладне програмне забезпечення для мікропроцесорних систем управління на базі локальних засобів автоматизації, промислових логічних контролерів та програмованих логічних матриць і сигнальних процесорів.

ПРН12. Вміти використовувати різноманітне спеціалізоване програмне забезпечення для розв'язування типових інженерних задач в галузі інженерії мобільних додатків, зокрема, математичного моделювання, автоматизованого проектування, керування базами даних, методів комп'ютерної графіки.

4. Зміст навчальної дисципліни

МОДУЛЬ 1. Базові алгоритми та основні синтаксичні конструкції мови C++

Змістовний модуль 1. Послідовні алгоритми для математичних обчислень

ТЕМА 1. Вступ до навчальної дисципліни

Стисла анотація:

Етапи створення програмного забезпечення. Постановка задачі . Аналіз, формальна постановка і вибір методу рішення. Проектування. Реалізація. Модифікація. Класифікація мов програмування. Мови низького та високого рівня. Компілятор. Лінковщик. Основи роботи у середовищі QtCreator [Д2, Д5].

Теми лекцій: Огляд методів і засобів розробки програмного забезпечення. Робота на навчальній платформі і оформлення звітності.

Лабораторна робота 1 - Введення-виведення даних мовою C++

Самостійна робота: Корегування, компіляція та запуск програми у середовищі QTCreator. Вивчення додаткових матеріалів, прикладів коду.

ТЕМА 2. Основи алгоритмізації та базові поняття мови C++

Стисла анотація:

Консольний ввід та вивід. Концепція типу даних. Прості типи даних. Цілий типу (int). Символьний тип (char). Логічний тип (bool). Типи з плаваючою крапкою (float, double). Змінні. Ключові слова. Константи та літерали. Операції. Арифметичні операції. Логічні операції. Інші операції. Пріоритет операцій. Вираження. Порядок обчислення складових вираження. Перетворення типів даних. Директиви препроцесора.

Теми лекцій: Базовий синтаксис і поняття мови C++. Засоби обчислення і організації програмного коду.

Лабораторна робота 2 - Математичні обчислення мовою C++

Самостійна робота: Математичні функції бібліотеки <cmath>. Вивчення додаткових матеріалів, прикладів коду.

ТЕМА 3. Алгоритми розгалуження

Стисла анотація:

Алгоритми та форми їх подання. Складові елементи алгоритмів. UML діаграми активності. Основні елементи. Засоби побудови UML діаграм. Інструкції розгалуження. Умовна інструкція (if, if-else). Інструкція вибору (switch) [Д1, Д3].

Теми лекцій: Синтаксис команд розгалуження. Подання алгоритмів у графічній формі (UML).

Лабораторна робота 3 - Реалізація алгоритмів з розгалуженням мовою C++

Самостійна робота: Тема 3. Елементи UML діаграм алгоритмів. Приклади побудови UML діаграм.

Модульний контроль 1

Змістовний модуль 2. Алгоритми розгалуження та повторення ТЕМА 4. Структурування програми на підпрограми

Стисла анотація:

Вказівники та посилання в C++. Функції. Прототипи функцій. Тип void. Інструкція return. Передача параметрів у функцію. Локальні змінні і область видимості. Глобальні змінні. Підпрограми у UML діаграмах [Д1].

Теми лекцій: Синтаксис опису, реалізації і виклику функцій.

Лабораторна робота 4 - Структурування програм з використанням функцій

Самостійна робота: Тема 4. Вивчення додаткових матеріалів, прикладів коду.

ТЕМА 5. Циклічні алгоритми

Стисла анотація:

Циклічні структури. UML діаграми циклічних алгоритмів. Цикл з передумовою (while). Цикл з післяумовою (do while). Цикл з параметром (for). Інструкції керування циклами (break, continue) [Д1].

Теми лекцій: Синтаксис команд організації циклів.

Лабораторна робота 5 - Реалізація циклічних алгоритмів мовою C++

Самостійна робота: Тема 5. Вивчення додаткових матеріалів, прикладів коду.

ТЕМА 6. Обробка масивів (1D)

Стисла анотація:

Одновимірні масиви. Введення-виведення елементів масиву. Доступ до елементів масиву. Передача масивів у функції. Типові задачі з використанням масивів.

Теми лекцій: Декларація, введення/виведення і обробка одновимірних масивів в C++.

Лабораторна робота 6 - Реалізація алгоритмів обробки одновимірних масивів мовою C++

Самостійна робота: Типові алгоритми на одновимірних масивах. Вивчення додаткових матеріалів, прикладів коду.

Модульний контроль 2

МОДУЛЬ 2. Алгоритми обробки структурованих даних та їх візуалізація

Змістовний модуль 3. Обробка послідовних структур даних

ТЕМА 7. Обробка масивів (2D)

Стисла анотація:

Двовимірні масиви (матриці). Введення-виведення матриць. Доступ до елементів матриці. Передача матриць у функції. Типові задачі з використанням матриць.

Теми лекцій: Декларація, введення/виведення і обробка двовимірних масивів в C++.

Лабораторна робота 7 - Реалізація алгоритмів обробки двовимірних масивів мовою C++

Самостійна робота: Математичні перетворення матриць. Вивчення додаткових матеріалів, прикладів коду.

ТЕМА 8. Робота з файлами і алгоритми сортування

Стисла анотація:

Типи файлів з точки зору мови програмування. Організація роботи з файлами засобами C. Організація роботи з файлами засобами C++. Алгоритми сортування. Сортування вибором. Сортування вставками. Сортування обміном.

Теми лекцій: Алгоритм і засоби роботи з файлами. Підходи і алгоритми сортування.

Лабораторна робота 8 - Реалізація алгоритмів сортування та робота з файлами мовою C++

Самостійна робота: Бінарні файли, імена файлів у різних файлових системах. Удосконалені методи сортування (бінарні вставки, шейкерна).

ТЕМА 9. Алгоритми обробки рядків

Стисла анотація:

Оголошення символьних рядків у програмах. Використання символу NULL. Ініціалізація символьного рядка. Передача рядків в функції. Тип даних string. Методи. Алгоритми пошуку. Послідовний пошук. Бінарний пошук. Пошук підрядка в рядку [Д2, Д4].

Теми лекцій: Обробка рядків посимвольно і за допомогою стандартної бібліотеки.

Лабораторна робота 9 - Робота з рядками символів мовою C++

Самостійна робота: Розрахунково-графічна робота. Системи числення. Перетворення до бінарної та шістнадцяткової систем.

Модульний контроль 3

Змістовний модуль 4. Створення власних структур та використання бібліотечних класів

ТЕМА 10. Типи даних користувача і динамічні структури даних

Стисла анотація:

Структури/записи. Масиви структур. Передача структур у функції. Функції в структурах. Відмінність структур і класів. Перерахування. Об'єднання. Динамічні структури даних. Контейнери STL (vector, list). Структурування програми на модулі.

Теми лекцій: Структури як тип даних і як засіб організації динамічних структур.

Лабораторна робота 10 - Робота зі структурами в C++ та розбиття програми на модулі

Самостійна робота: Розрахунково-графічна робота. Арифметичні дії з бінарними числами.

ТЕМА 11. Десктоп-застосунки (проєктування GUI)

Стисла анотація:

Архітектура крос-платформових застосунків. Qt Framework. Створення форми інтерфейсу користувача у середовищі Qt Creator. Компоновка елементів інтерфейсу за допомогою Qt Designer. Механізм сигналів-слотів для організації взаємодії між об'єктами. Використання стандартних діалогових вікон Qt.

Теми лекцій: Проєктування і реалізація віконних застосунків.

Лабораторна робота 11 - Розробка десктоп-застосунків в середовищі QtCreator

Самостійна робота: Розрахунково-графічна робота. Побудова UML діаграми програми та написання коду для переводу з десяткової системи до бінарної.

ТЕМА 12. Аналіз складності алгоритмів та рекурсивні алгоритми

Стисла анотація:

Поняття складності. О-оцінка. Підрахунок інструкцій. Аналіз найбільш несприятливого випадку. Асимптотична поведінка. Визначення складності. Складність рекурсивних алгоритмів. Проста рекурсія. Багаторазова рекурсія. Загальні функції оцінки складності.

Теми лекцій: Оцінка складності алгоритмів для оптимізації коду.

Самостійна робота: Вивчення прикладів рекурсивних алгоритмів.

Модульний контроль 4

5. Індивідуальні завдання

Виконання розрахункової роботи на тему «Формування алгоритмів обробки багатовимірних масивів та оцінювання їх складності».

6. Методи навчання

Словесні: лекція, пояснення, навчальна дискусія. Наочні: презентації. Практичні: лабораторні та практичні роботи. Індивідуальні консультації. Виконання розрахункової роботи. Самостійна робота з текстами лекцій та додатковими матеріалами і навчальними відео.

7. Методи контролю

Проведення поточного (на практичних заняттях) та модульного контролю, оформлення та захист звітів з лабораторних робіт, захист розрахункової роботи, фінальний семестровий контроль у вигляді іспиту.

8. Критерії оцінювання та розподіл балів, які отримують здобувачі освіти

Таблиця 8.1 – Розподіл балів, які отримують здобувачі освіти

Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
1-й семестр			
Змістовний модуль 1			
Робота на лекціях	0...1	8	0...8
Виконання і захист лабораторних робіт	0...10	3	0...30
Модульний та поточний контроль	0...12	1	0...12
Змістовний модуль 2			
Робота на лекціях	0...1	8	0...8
Виконання і захист лабораторних робіт	0..10	3	0..30
Модульний та поточний контроль	0...12	1	0...12
Усього за семестр			0...100
2-й семестр			
Змістовний модуль 3			
Робота на лекціях	0...1	8	0...8
Виконання і захист лабораторних робіт	0...10	3	0...30
Модульний та поточний контроль	0...12	1	0...12

Змістовний модуль 4			
Робота на лекціях	0...1	4	0...4
Виконання і захист лабораторних робіт	0...10	2	0...20
Виконання і захист РР	0 ...10	1	0...10
Модульний контроль	0...16	1	0...16
Усього за семестр			0...100

Семестровий контроль (іспит) проводиться у разі відмови студента від балів поточного тестування й за наявності допуску до іспиту/заліку. Під час складання семестрового іспиту/заліку студент має можливість отримати максимум 100 балів.

Білет для іспиту складається з одного теоретичного питання (30 балів), одного практичного питання (30 балів) та одного лабораторного завдання, яке необхідно виконати на комп'ютері (40 балів).

Таблиця 8.2 – Шкали оцінювання: бальна і традиційна

Сума балів	Оцінка за традиційною шкалою	
	Іспит, диференційний залік	Залік
90 – 100	Відмінно	Зараховано
75 – 89	Добре	
60 – 74	Задовільно	
0 – 59	Незадовільно	Не зараховано

Критерії оцінювання роботи здобувача освіти протягом семестру

Необхідний обсяг знань для одержання позитивної оцінки: базові структури алгоритмів та методи їх побудови; синтаксис мови програмування високого рівня для реалізації алгоритмів збереження, обробки та відображення даних; структуру та можливості бібліотек математичних обчислень, стандартних алгоритмів та структур, роботи з файлами та графічним інтерфейсом користувача;

Необхідний обсяг вмінь для одержання позитивної оцінки: аналіз постановки задачі та проектування алгоритму її вирішення; реалізація алгоритмів послідовної, розгалуженої та циклічної структур на мові програмування високого рівня; структурування коду програми з використанням підпрограм, структур даних та модулів; створення консольних додатків та додатків з графічним інтерфейсом для візуалізації даних.

Задовільно (60÷74 бали):

Здобувач слабо володіє теоретичним матеріалом, має мінімум знань та умінь, допускає помилки у вирішенні практичних завдань. Захистив всі лабораторні завдання та розрахункове завдання, виконав усі модульні

завдання, має не впевнені практичні навички написання програм на C++. Зменшення кількості балів в межах оцінки можливе за неточні та неповні відповіді на теоретичні та практичні запитання.

Добре (75÷89 балів):

Здобувач має достатньо глибокі знання з теоретичної частини дисципліни. Захистив всі практичні, лабораторні завдання та індивідуальне завдання, виконав усі модульні завдання з оцінкою «добре», має практичні навички написання програм на C++. Правильно будує UML діаграми, його відповіді не є чіткими. Зменшення кількості балів в межах оцінки можливе при неповних відповідях на теоретичні або практичні запитання.

Відмінно (90÷100 балів):

Здобувач твердо знає: базові поняття і принципи, що відносяться до дисципліни «Алгоритмізація та програмування». Захистив всі практичні, лабораторні завдання та індивідуальне завдання, виконав усі модульні завдання з оцінкою «відмінно», має тверді практичні навички написання програм на C++. Вільно користується навчальною та науково-технічною літературою з питань дисципліни. Вміє логічно і чітко скласти свою відповідь, вирішити лабораторне завдання та скласти UML діаграму. Зменшення кількості балів в межах оцінки можливе при неточних формулюваннях у відповідях на додаткові запитання, які були поставлені перед ним.

9. Політика навчального курсу

Відвідування занять. Регуляція пропусків. Інтерактивний характер курсу передбачає обов'язкове відвідування практичних занять. Здобувачі освіти, які за певних обставин не можуть відвідувати практичні заняття регулярно, повинні протягом тижня узгодити із викладачем графік індивідуального відпрацювання пропущених занять. Окремі пропущені заняття мають бути відпрацьовані на найближчій консультації протягом тижня після їх пропуску. Відпрацювання занять здійснюється усно у формі співбесіди за питаннями, визначеними планом заняття. В окремих випадках дозволяється письмове відпрацювання пропущених занять шляхом виконання індивідуального письмового завдання.

Дотримання вимог академічної доброчесності здобувачами освіти під час вивчення навчальної дисципліни. Під час вивчення навчальної дисципліни здобувачі освіти мають дотримуватися загальноприйнятих морально-етичних норм і правил поведінки, вимог академічної доброчесності, передбачених Положенням про академічну доброчесність Національного аерокосмічного університету «Харківський авіаційний інститут» (<https://khai.edu/assets/files/polozhennya/polozhennya-pro-akademichnu-dobrochesnist.pdf>). Очікується, що роботи здобувачів освіти будуть їх оригінальними дослідженнями або міркуваннями. Відсутність посилань на використані джерела, фабрикування джерел, списування, втручання в роботу інших здобувачів освіти становлять, але не обмежують, приклади можливої

академічної недоброчесності. Виявлення ознак академічної недоброчесності в письмовій роботі здобувача освіти є підставою для її незарахування викладачем незалежно від масштабів плагиату чи обману.

Вирішення конфліктів. Порядок і процедури врегулювання конфліктів, пов'язаних із корупційними діями, зіткненням інтересів, різними формами дискримінації, сексуальними домаганнями, міжособистісними стосунками та іншими ситуаціями, що можуть виникнути під час навчання, а також правила етичної поведінки регламентуються Кодексом етичної поведінки в Національному аерокосмічному університеті «Харківський авіаційний інститут» (<https://khai.edu.ua/university/normativna-baza/ustanovchi-dokumenti/kodeks-etichnoi-povedinki/>).

10. Методичне забезпечення

Посилання на НМКД дисципліни у системі дистанційного навчання Ментор:
<https://mentor.khai.edu/course/view.php?id=3066> – семестр 1
<https://mentor.khai.edu/course/view.php?id=1240> – семестр 2

12. Рекомендована література

Базова

1. С++. Основи програмування. Теорія та практика: підручник / [О.Г. Трофименко, Ю.В. Прокоп, І.Г. Швайко, Л.М. Буката та ін.] ; за ред. О.Г. Трофименко. – Одеса: Фенікс, 2019. – 544 с.
2. Т. П. Караванова. Інформатика. Основи алгоритмізації та програмування: 777 задач з рекомендаціями та прикладами. Електронний ресурс: <https://informatik.pp.ua/pidruchniki/programuvannya/karavanova-777-zadach>.
3. Б. Пекарський. Основи програмування. Кондор, 2018, - 364 с.
4. Жуковський С.С., Вакалюк Т.А. Об'єктно-орієнтоване програмування мовою С++. Навчально-методичний посібник – Житомир: Вид-во ЖДУ, 2016. – 100 с.

Допоміжна

1. Гавриленко О. В., Гавриленко О. І. Об'єктно-орієнтований аналіз і програмування: навч. посіб. Харків : ХАІ, 2007. – 62 с. – теми 3,4,5 (діаграми UML)
2. Structural Diagnosis Method for Computer Programs Developed by Trainees / D. Gaydachuk, O. Havrylenko, J. P. Martínez Bastida, A. Chukhray. // ICTERI2019 ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer – PP 501-506. – тема 1 (лексеми С++), тема 9 (динамічні структури даних)
3. A Graphical Environment for Algorithms Training / Markovych, S., Chukhray, A., Lukashov, V., Havrylenko, O., Novytska, O. //Lecture Notes in

Networks and Systems, 2021. – V. 188. – pp. 186–205. – тема 3 (способи подання алгоритмів)

4. Chukhray, A., Havrylenko, O. The engineering skills training process modeling using dynamic bayesian nets [Текст] // «Радіоелектронні і комп'ютерні системи. – № 2/98. – 2021. – С. 87–96. – тема 9 (динамічні структури даних)

5. Models and methods for computer support of adaptive training of algorithmic tasks solution / Mygal V., Mygal G., Chukhray A., Havrylenko, O. // 16th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. ICTERI'2020: proceedings, 6–10 Oct. 2020, Kharkiv. – Kharkiv, 2020. – Vol. I. – P. 408–415. – тема 1 (алгоритмічні задачі та алгоритмічне мислення)

12. Інформаційні ресурси

1. Сайт кафедри 301: <http://k301.khai.edu/>
2. Офіційний сайт Microsoft Visual Studio:
<https://visualstudio.microsoft.com/>
3. Електронний підручник.
https://sites.google.com/site/zsuelearning/c_plus
4. Електронний підручник. Мова програмування C++:
<http://cppstudio.com/cat/274/>
5. Бібліотека стандартних шаблонів (eng):
<https://msdn.microsoft.com/en-us/library/csc687y.aspx>