


Міністерство освіти і науки України
Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Кафедра систем управління літальних апаратів (№ 301)

ЗАТВЕРДЖУЮ

Гарант освітньої програми


(підпис) Сергій ПАСІЧНИК

« 25 » серпня 2023 р.

**РОБОЧА ПРОГРАМА ОBOB'ЯЗКОВОЇ
НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

Алгоритмізація та програмування
(назва навчальної дисципліни)

Галузь знань 27 Транспорт

Спеціальність: 272 Авіаційний транспорт

Освітня програма: Інтелектуальні транспортні системи

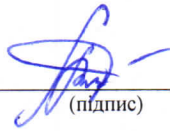
Форма навчання: денна

Рівень вищої освіти: перший (бакалаврський)

Харків 2023 рік

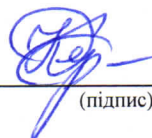
Розробник:

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


(підпис)

Робочу програму розглянуто на засіданні кафедри Систем управління літальних апаратів (№ 301)

Протокол № 1 від “25” серпня 2023 р.

Завідувач кафедри 301 к.т.н., доцент 
(підпис) Костянтин ДЕРГАЧОВ

1. Опис навчальної дисципліни

Найменування Показників	Галузь знань, спеціальність, освітня програма, рівень вищої освіти	Характеристика навчальної дисципліни	
		Денна форма навчання	
Кількість кредитів – 13,5	Галузь знань: 27 «Транспорт»	Обов’язкова дисципліна	
Кількість модулів – 2		Навчальний рік:	
Кількість змістових модулів – 4			
Індивідуальні завдання: Розрахункова робота у семестрі 2	Спеціальність: 272 «Авіаційний транспорт»	2023/2024	
		Семестр	
		1-й	2-й
Загальна кількість годин 405	Освітня програма: «Інтелектуальні транспортні системи»	Лекції	Лекції
Кількість тижневих годин для денної форми навчання: аудиторних – 3,75 самостійної роботи студента – 5,15		32 год.	32 год.
	Практичні	Практичні	
	–	24 год.	
	Лабораторні	Лабораторні	
	40 год.	24 год.	
	Самостійна робота	Самостійна робота	
	123 год.	130 год.	
	Вид контролю	Вид контролю	
іспит	іспит		
	Рівень вищої освіти: перший (бакалаврський)		

Примітка: Співвідношення кількості годин аудиторних занять до самостійної і індивідуальної роботи становить: 152/253.

2. Мета та завдання навчальної дисципліни

Мета вивчення – освоєння студентами методів і засобів проектування і реалізації алгоритмів обробки даних, а також структурного підходу до побудови програмного забезпечення систем та елементів авіаційного транспорту.

Завдання – вивчення методів проектування алгоритмів, засвоєння синтаксичних конструкцій в мовах програмування високого рівня, а також здобуття навичок проектування та реалізації програмного забезпечення систем та елементів авіаційного транспорту.

Згідно з вимогами освітньо-професійної програми студенти повинні досягти таких **компетентностей**:

Загальні:

ЗК3. Навички використання інформаційних і комунікаційних технологій.

ЗК7. Здатність працювати автономно.

ЗК9. Здатність до абстрактного мислення, аналізу та синтезу.

Фахові:

ФК19. Здатність реалізовувати та використовувати апаратні та програмно-алгоритмічні засоби при проектуванні систем управління для об'єктів авіаційної техніки.

Програмні результати навчання:

ПРН 03 Застосовувати сучасні інформаційні технології, технічну літературу, бази даних, інші ресурси та сучасні програмні засоби для розв'язання спеціалізованих складних задач авіаційного транспорту.

ПРН 26 Використовувати професійно-орієнтовані знання з математики, фізики, електротехніки, електроніки, обчислювальної техніки і програмування при проектуванні підсистем і приладів для об'єктів авіаційного транспорту.

ПРН 27 Виконувати аналіз і комп'ютерне моделювання підсистем і приладів об'єктів авіаційної техніки, синтез систем управління та вибір технічних засобів їх реалізації, використовуючи професійний математичний апарат та комп'ютерно-інтегровані технології і відповідні програмні середовища..

Міждисциплінарні зв'язки:

Передумови для вивчення даної дисципліни:

Шкільна математика: арифметичні та математичні операції і функції; дослідження функцій та побудова їх графіків; числові ряди; формули та параметри геометричних фігур на площині. Інформатика: базові навички

використання персонального комп'ютера (запуск програм, зокрема Інтернет браузера, робота з файлами, створення та редагування документу за допомогою офісних програм).

Дисципліна підтримує наступні курси:

Об'єктно-орієнтоване проектування авіаційних транспортних систем. Основи навігації. Навчальна практика. Ознайомча практика. Виробнича практика. Мікроконтролери в системах управління. Кваліфікаційна робота бакалавра.

3. Програма навчальної дисципліни

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

Змістовий модуль 1. Послідовні алгоритми для математичних обчислень

ТЕМА 1. Вступ до навчальної дисципліни

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 Visual Studio [Д2, Д5].

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

Консольний ввід та вивід. Концепція типу даних. Прості типи даних. Цілий тип (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.

Модульний контроль. Модульна робота 1.

Змістовий модуль 2. Алгоритми розгалуження та повторення

ТЕМА 3. Алгоритми розгалуження

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].

ТЕМА 4. Структурування програми на підпрограми

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].

ТЕМА 5. Циклічні алгоритми

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].

Модульний контроль. Модульна робота 2.

Модуль 2. Алгоритми обробки структурованих даних та їх візуалізація Змістовий модуль 3. Обробка послідовних структур даних

ТЕМА 6-7. Обробка масивів

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

ТЕМА 8. Робота з файлами і алгоритми сортування

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.

ТЕМА 9. Алгоритми обробки рядків

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].

Модульний контроль. Модульна робота 3.

Змістовий модуль 4. Створення власних структур та використання бібліотечних класів

ТЕМА 10. Типи даних користувача і динамічні структури даних

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.

ТЕМА 11. Desktop-застосунки Windows (проекування GUI)

Архітектура Windows-Застосунків. WinAPI. Створення форми Windows Forms. Компоновка елементів Windows Forms. Стандартні діалогові вікна.

ТЕМА 12. Аналіз складності алгоритмів та рекурсивні алгоритми

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.

Модульний контроль. Модульна робота 4.

4. Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин					
	денна форма					
	усього	у тому числі				
л		п	лаб	інд	с.р.	
1	2	3	4	5	6	7
Модуль 1						
Змістовий модуль 1. Послідовні алгоритми для математичних обчислень						
Тема 1. Вступ до навчальної дисципліни.	18	4	0	4	0	10
Тема 2. Основи алгоритмізації та базові поняття мови C++	22	4	0	4	0	14
Тема 3. Алгоритми розгалуження	24	8	0	4	0	12
Модульний контроль	2	0	0	0	0	2
Разом за змістовим модулем 1	66	16	0	12	0	38
Змістовий модуль 2. Алгоритми розгалуження та повторення						
Тема 4. Структурування програми на підпрограми	30	8	0	8	0	14
Тема 5. Циклічні алгоритми	20	4	0	4	0	12
Тема 6. Обробка масивів(1D)	17	4	0	8	0	5
Модульний контроль	2	0	0	0	0	2
Разом за змістовим модулем 2	69	16	0	20	0	33
Модуль 2						
Змістовий модуль 3. Обробка послідовних структур даних						
Тема 6. Обробка масивів (2D)	21	4	4	4	0	9
Тема 7. Робота з файлами і алгоритми сортування	26	8	4	4	0	14
Тема 8. Алгоритми обробки рядків	20	4	4	4	0	17
Модульний контроль	2	0	0	0	0	2
Разом за змістовим модулем 3	74	16	12	12	0	42
Змістовий модуль 4. Створення власних структур та використання класів						
Тема 9. Типи даних користувача і динамічні структури даних	24	2	4	4	0	19
Тема 10. Десктоп-застосунки Windows (проектування GUI)	26	4	4	8	0	19
Тема 11. Аналіз складності алгоритмів та рекурсивні алгоритми	14	2	4	0	0	12
Модульний контроль	2	0	0	0	0	2
Разом за змістовим модулем 4	77	16	12	12	0	53
Усього годин	405	64	24	56	0	253

5. Теми семінарських занять

№ з/п	Назва теми	Кількість годин
	Не заплановано	

6. Теми практичних занять

№ з/п	Назва теми	Кільк. год.
1	Вирішення задач з обробки масивів і матриць	4
2	Вирішення задач з алгоритмами сортування	4
3	Вирішення задач з обробки рядків	4
4	Приклади переведення чисел до двійкової системи	4
5	Приклади переведення чисел до шістнадцяткової системи	4
6	Приклади арифметичних дій у двійковій і шістнадцятковій системі	4
Разом		24

7. Теми лабораторних занять

№ з/п	Назва теми	Кільк. годин
1	Введення-виведення даних мовою C++	4
2	Математичні обчислення мовою C++	4
3	Реалізація алгоритмів з розгалуженням мовою C++	4
4	Структурування програм з використанням функцій	8
5	Реалізація циклічних алгоритмів мовою C++	4
6	Реалізація алгоритмів обробки одновимірних масивів мовою C++	8
7	Реалізація алгоритмів обробки двовимірних масивів мовою C++	4
8	Реалізація алгоритмів сортування та робота з файлами мовою C++	8
9	Робота з рядками символів мовою C++	4
10	Робота зі структурами в C++ та розбиття програми на модулі	8
11	Розробка десктоп-застосунків в середовищі Visual Studio	8
Разом		64

8. Самостійна робота

№ з/п	Назва теми	Кільк. годин
1	Тема 1. Корегування, компіляція та запуск програми у середовищі Visual Studio.	10
2	Тема 2. Константи, літерали. Математичні функції бібліотеки <cmath>. Вивчення прикладів коду	14
3	Тема 3. Елементи UML діаграм алгоритмів. Приклади побудови UML діаграм	14

4	Тема 4. Вивчення додаткових матеріалів, прикладів коду	14
5	Тема 5. Вивчення додаткових матеріалів, прикладів коду	14
6	Тема 6. Математичні перетворення матриць. Вивчення прикладів коду	14
7	Тема 7. Бінарні файли, імена файлів у різних файлових системах. Удосконалені методи сортування (бінарні вставки, шейкерна).	14
8	Розрахунково-графічна робота. Системи числення. Перетворення до бінарної та шістнадцяткової систем.	19
9	Розрахунково-графічна робота. Арифметичні дії з бінарними числами.	19
10	Розрахунково-графічна робота. Побудова UML діаграми програми та написання коду для переводу з десяткової системи до бінарної.	19
11	Тема 11. Рекурсивні алгоритми	14
Разом		165

9. Індивідуальні завдання

№ з/п	Назва теми	Кільк. годин
1	Не заплановано	-

10. Методи навчання

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

11. 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.

12. Критерії оцінювання та розподіл балів, які отримують студенти

12.1. Розподіл балів, які отримують студенти

1 семестр

Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
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 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
Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
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).

12.2. Якісні критерії оцінювання

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.

12.3. Критерії оцінювання роботи студента протягом семестру

1. Відмінно (90÷100 балів) виставляється студенту:

1.1 Який 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.

1.2 Зменшення кількості балів в межах оцінки можливе при неточних формулюваннях у відповідях на додаткові запитання, які були поставлені перед ним.

2. Добре (75÷89 балів) виставляється студенту:

2.1 Який 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.

2.2 Зменшення кількості балів в межах оцінки можливе при неповних відповідях на теоретичні або практичні запитання.

3. Задовільно (60÷74 бали) виставляється студенту:

3.1 Який 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++.

3.2 Зменшення кількості балів в межах оцінки можливе за неточні та неповні відповіді на теоретичні та практичні запитання.

Шкала оцінювання: національна та ECTS

Сума балів	Оцінка за традиційною шкалою	
	Іспит	Залік
90 – 100	відмінно	зараховано
75 – 89	добре	
60 -74	задовільно	
0 – 59	незадовільно	незараховано

13. Методичне забезпечення

1. Конспект лекцій з дисципліни «Алгоритмізація та програмування». 2023 р.
2. Слайди з презентаціями лекційних матеріалів з дисципліни «Алгоритмізація та програмування». 2023 р.
3. Методичні вказівки і завдання до виконання лабораторних робіт з дисципліни «Алгоритмізація та програмування». 2023 р.
4. Методичні вказівки і завдання до виконання розрахунково-графічної роботи. 2023 р.

Все методичне забезпечення в електронному вигляді розміщене на хмарному сховищі і відкрито для всіх користувачів. Автор розробок – доцент каф. 301 Гавриленко О.В.

Посилання для ознайомлення і скачування:

https://drive.google.com/open?id=0B0v3s_o3YMPmUUVVeUR6TIFoZG8

https://drive.google.com/drive/folders/1jnZsNnPK7gCSeD0E2cTogA0m7j_SjW91?usp=drive_link

14. 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)

15. 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>