

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

Кафедра математичного моделювання та штучного інтелекту (№ 304)

ЗАТВЕРДЖУЮ

Керівник проектної групи


(підпис)

Д. І. Чумаченко

(ініціали та прізвище)

« 31 » 08 2021 р.

**РОБОЧА ПРОГРАМА *ОБОВ'ЯЗКОВОЇ*
НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

Теорія алгоритмів

(назва навчальної дисципліни)

Галузь знань: 12 «Інформаційні технології»

(шифр і найменування галузі знань)

Спеціальність: 122 «Комп'ютерні науки»

(код і найменування спеціальності)

Освітня програма: "Інтелектуальні системи та технології"

(найменування освітньої програми)

Форма навчання: денна

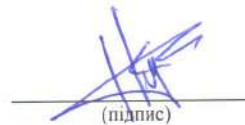
Рівень вищої освіти: перший (бакалаврський)

Харків 2021 рік

Робоча програма «Теорія алгоритмів»
(назва дисципліни)
для студентів за спеціальностями 122 "Комп'ютерні науки"
освітніми програмами "Інтелектуальні системи та технології"

« 24 » 08 2021 р., – 12 с.

Розробник: Коробчинський К. П., доц., к.т.н.
(прізвище та ініціали, посада, науковий ступінь і вчене звання)

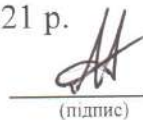

(підпис)

Робочу програму розглянуто на засіданні кафедри математичного моделювання та штучного інтелекту (№ 304)

(назва кафедри)

Протокол № 2 від « 27 » 08 2021 р.

Завідувач кафедри д.т.н., проф.
(науковий ступінь і вчене звання)


(підпис)

А. Г. Чухрай
(ініціали та прізвище)

1. Опис навчальної дисципліни

Найменування показника	Галузь знань, спеціальність, освітня програма, рівень вищої освіти	Характеристика навчальної дисципліни (денна форма навчання)
Кількість кредитів – 4	Галузь знань 12 "Інформаційні технології" (шифр і найменування) Спеціальність 122 «Комп’ютерні науки» (код і найменування) Освітня програма "Інтелектуальні системи та технології" (найменування) Рівень вищої освіти: перший (бакалаврський)	Цикл загальної підготовки
Кількість модулів – 2		Навчальний рік
Кількість змістовних модулів – 3		2021/2022
Індивідуальне науково-дослідне завдання – не передбачено		Семестр
Загальна кількість годин – 72*/ 120		2-й
		Лекції*
Кількість тижневих годин для денної форми навчання: аудиторних – 4 самостійної роботи студента – 3		32 годин
		Практичні, семінарські*
		40 годин
		Лабораторні*
		-
		Самостійна робота
		48 годин
		Вид контролю
	модульний контроль, іспит	

Співвідношення кількості годин аудиторних занять до самостійної роботи становить: 72 / 48.

* Аудиторне навантаження може бути зменшене або збільшене на одну годину залежно від розкладу занять.

2. Мета та завдання навчальної дисципліни

Мета вивчення: викладення основних понять алгоритмізації і техніки застосування у програмуванні базових алгоритмічних структур (організація програм) і базових структур даних (організація даних).

Для досягнення мети поставлені такі **основні завдання:**

- вивчення основних етапів процесу проектування програмного забезпечення і визначення принципів процедурного програмування щодо розробки програм мовою C++;
- вивчення типових підходів до розробки і аналізу найбільш розповсюджених алгоритмів рішення економіко-математичних задач;
- здійснення аналізу можливостей сучасних інструментальних середовищ розробки програм (на прикладі середовища Visual Studio.NET);
- визначення концепцій і вивчення основних принципів організації програм у ОС Windows.

Предметом навчальної дисципліни є теорія і практика застосування у програмуванні базових алгоритмічних структур і базових структур даних на базі сучасних технологій розробки програмного забезпечення.

Дисципліна "Теорія алгоритмів" є базовою для бакалаврів галузі знань "Комп'ютерні науки".

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

ЗК 3. Знання та розуміння предметної області та професійної діяльності.

ЗК 5. Здатність оцінювати та забезпечувати якість виконуваних робіт.

ФК 2. Ґрунтовна підготовка в області програмування, володіння алгоритмічним мисленням, методами програмної інженерії для реалізації програмного забезпечення з урахуванням вимог до його якості, надійності, виробничих характеристик.

Програмні результати навчання:

ПРН 4. Знання сучасних методів побудови та аналізу ефективних алгоритмів і вміння їх реалізувати в конкретних застосуваннях.

ПРН 9. Знання принципів структурного програмування, сучасних процедурно-орієнтованих мов, основних структур даних і вміння їх застосовувати під час програмної реалізації алгоритмів професійних завдань.

ПРН 11. Знання сучасних технологій та інструментальних засобів розробки програмних систем, уміння їх застосовувати на всіх етапах життєвого циклу.

Міждисциплінарні зв'язки:

Необхідна навчальна база перед початком вивчення дисципліни: для успішного вивчення дисципліни необхідні базові знання, отримані студентами в об'ємі шкільної програми, а також поточні знання при паралельному освоєнні ними дисциплін "Архітектура ЕОМ", "Вища математика".

У результаті вивчення навчальної дисципліни студенти повинні **знати:**

- основні етапи процесу проектування програмного забезпечення;

- типові алгоритмічні конструкції;
- принципи процедурного і структурованого програмування;
- особливості застосування сучасних базових інструментальних програмних засобів, призначених для вирішення економічних задач;
- базові типи даних;
- похідні типи даних: переліки, покажчики, посилання, масиви, структури, об'єднання;
- оператори управління програмою;
- команди перед процесорної обробки;
- правила роботи з функціями;
- систему введення-виведення C++;
- основні принципи роботи з файлами;
- правила роботи із шаблонами;
- принципи розробки Windows-додатків;
- основу побудови програм на керованому C++;

вміти: складати програми мовою C++, забезпечуючи:

- рішення задач з курсу вищої математики;
- створення і обробку структур, масивів структур; найпростішу обробку файлів;
- використання функцій;
- використання основних елементів призначеного для користувача інтерфейсу;
- використання сучасного інструментального програмного забезпечення;
- користуватися раніше складеними програмами і здійснювати супровід програм, вносити зміни в програму, виконувати відладку програм за допомогою вбудованих інструментальних засобів.

мати уявлення:

- про основні етапи створення об'єктно-орієнтованого програмного додатку;
- про фундаментальні принципи класичного об'єктно-орієнтованого підходу до програмування комп'ютерної системи;
- про сучасні способи використання мови програмування C++.

3. Програма навчальної дисципліни

Модуль 1. Процедурне програмування. Рядки.

Змістовий модуль 1. Організація похідних типів даних та алгоритми їх оброблення.

Тема 1. Абстракція даних. Складені структури даних.

Поняття основних структур даних. Основні характеристики базових структур даних. Алгоритми для обробки базових структур даних.

Тема 2. Масиви. Масиви, їх опис, ініціалізація.

Масиви. Оголошення масивів та розміщення у пам'яті. Ініціалізація масивів. Обробка одновимірних масивів даних економічного характеру. Алгоритми сортування масивів. Багатовимірні масиви. Ініціалізація багатовимірного масиву. Типові приклади обробки матриць.

Тема 3. Обробка символьної інформації.

Символьний тип даних, символьна константа. Рядки, константа-рядок. Бібліотека `string.h` функцій обробки рядків. Введення-виведення рядків та масивів символів. Розташування у пам'яті. Обробка рядків: різноманітний пошук у рядку, вставлення та видалення символів, аналіз слів, копіювання та поєднання рядків. Алгоритми та демонстраційні програми роботи з текстами. Рядки як масиви символів. Операції з рядками. Тип даних `string`.

Тема 4. Файли.

Загальні відомості про систему вводу-виводу даних. Текстовий файл. Типи файлів і оголошення їх змінних. Відкриття файлу для запису /читання. Вказівник файлу. Переміщення вказівника. Розміщення даних у файлі. Запис у файл. Читання із файлу. Закриття файлу.

Тема 5. Моделі та класи пам'яті.

Моделі пам'яті. Класи пам'яті. Змінні глобальні, локальні, автоматичні, розміщення у пам'яті. Особливості використання пам'яті. Вказівники на різні об'єкти. Робота з вказівниками. Приклади типового використання покажчиків.

Тема 6. Змінні різних типів пам'яті.

Змінні різних типів пам'яті у програмах. Приклади.

Змістовий модуль 2. Розробка функцій та їх типи.

Тема 7. Функції.

Загальні відомості про функції. Структура функції. Значення, параметри і аргументи, що повертаються. Оголошення функції. Прототипи функцій. Визначення функції. Виконання функції. Локальні і глобальні змінні. Правило видимості змінних. Приведення типів аргументів функцій. Правила автоматичного (неявного) приведення типів. Явні перетворення типів. Правила роботи з функціями. Класи пам'яті. Список параметрів функції. Параметри за умовчанням. Способи передачі параметрів. Способи повернення значення. Функції, що підставляються.

Створення власних заголовних файлів. Перевантаження функцій. Рекурсія. Робота функцій. Розбиття пам'яті. Стек і функції. Модифікатори функцій.

Тема 8. Робота з функціями.

Масиви як параметри функцій. Робота з функціями.

Типові алгоритми і їх реалізація. Приклади.

Тема 9. Базові засоби процедурного програмування.

Процедурно-орієнтоване програмування. Рекурсія.

Процедури, методи, підпрограми, функції. Поняття процедурно-орієнтованого програмування. Рекурсія в програмуванні. Приклади рекурсії.

Тема 10. Особливості процедурного програмування на базі C++.

Методології розробки програм. Використання процедур. Процедурний підхід.

Процедурна декомпозиція. Особливості процедурного програмування. Використання умовних операторів і оператора безумовного переходу.

Основні методології розробки програм. Низхідне та висхідне проектування, модульне програмування.

Тема 11. Шаблони

Основи апарату шаблонів. Шаблони функцій. Перевантаження шаблонів функцій. Шаблони функцій сортування.

Стандартна бібліотека шаблонів (STL). Призначення та склад STL. Контейнери. Робота з векторами.

Модуль 2. Введення в ООП.

Змістовий модуль 3. Структурні типи даних та робота з ними.

Тема 12. Структурні типи даних.

Структури, об'єднання, перелічення: їх опис, розміщення у пам'яті, ініціалізація, використання у програмах.

Структури. Структури з бітовими полями. Вкладені структури. Доступ до елементів структур. Операції з структурами. Структури як параметри функцій.

Масиви структур. Показчики на структури. Передача з посилання масивів структур. Об'єднання. Операції з об'єднаннями. Переліки. Функції роботи з датою та часом.

Тема 13. Основи об'єктно-орієнтованого підходу до створення програм.

Парадигми програмування: об'єктно-орієнтована парадигма (інкапсуляція, поліморфізм, наслідування).

Тема 14. Основні властивості ООП.

Реалізація об'єктно-орієнтованого підходу до програмування засобами мови програмування C++.

Класи і їх опис в C++. Екземпляри класів або об'єкти. Поля (атрибути) та методи. Специфікатори доступу (private, protected, public). Вказівник this. Перевантаження методів. Конструктори і деструктори класів. Перевантаження операцій. Наслідування в C++

Тема 15. Відносини та зв'язки між класами

Основи UML. Сфера застосування. Основні поняття та принципи. Зв'язки між класами (асоціація, агрегація, композиція, узагальнення).

UML: діаграми класів. UML: діаграми взаємодії. UML: діаграми послідовності. Шаблони проектування.

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. Процедурне програмування. Рядки.						
Змістовий модуль 1. Організація похідних типів даних та алгоритми їх оброблення.						
Тема 1. Абстракція даних. Складені структури даних.	8	2	2			3
Тема 2. Масиви. Масиви, їх опис, ініціалізація.	8	2	2			3
Тема 3. Обробка символьної інформації.	8	2	2			3
Тема 4. Файли.	7	2	2			3
Тема 5. Моделі та класи пам'яті.	7	2	2			3
Тема 6. Змінні різних типів пам'яті.	7	2	2			3
Змістовий модуль 2. Розробка функцій та їх типи.						
Тема 7. Функції.	7	2	3			3
Тема 8. Робота з функціями.	7	2	3			3
Тема 9. Базові засоби процедурного програмування.	7	2	3			3
Тема 10. Особливості процедурного програмування на базі C++	7	2	3			3
Тема 11. Шаблони	7	2	3			3
Модульний контроль	2		3			
Контрольний захід	1	1				
Разом за модулем 1	83	23	30	0	0	33
Модуль 2. Введення в ООП.						
Змістовий модуль 3. Структурні типи даних та робота з ними.						
Тема 24. Структурні типи даних.	9	2	2			3
Тема 25. Основи ООП до створення програм.	9	2	2			4
Тема 26. Основні властивості ООП.	8	2	2			4
Тема 27. Відносини та зв'язки між класами	8	2	2			4
Модульний контроль	2		2			
Контрольний захід	1	1				
Разом за модулем 2	37	9	10	0	0	15
Усього годин навчальної дисципліни	120	32	40	0	0	48

5. Теми практичних занять

№ з/п	Назва теми	Кількість годин
1	Алгоритми лінійних процесів.	4
2	Алгоритми та розгалужених процесів.	4
3	Алгоритми циклічних процесів	4
4	Модульна структура програм	4
5	Базові конструкції структурного програмування	4

№ з/п	Назва теми	Кількість годин
6	Рекурсія в програмуванні	5
7	Низхідне та висхідне проектування, модульне програмування	5
8	Перевантаження операцій.	5
9	UML: діаграми класів, діаграми взаємодії, діаграми послідовності.	5
	Разом	40

6. Самостійна робота

№ з/п	Назва теми	Кількість годин
1.	Функції	6
2.	Алгоритми для обробки базових структур даних	7
3.	Рекурсія в програмуванні. Приклади рекурсії.	7
4.	Процедурна декомпозиція.	7
5.	Похідні типи (структури)	7
6.	Принцип інкапсуляції. Класи	7
7.	Зв'язки між класами	7
	Разом	56

7. Методи навчання

1. Пояснювально-ілюстративний (інформаційно-рецептивний) метод та метод проблемного виконання (лекційні заняття).
2. Репродуктивний (лабораторні роботи).
3. Частково-пошуковий (евристичний) та дослідницький (самостійна робота та виконання розрахункової та розрахунково-графічної робіт).

8. Методи контролю

Визначення рівня засвоєння студентом навчального матеріалу дисципліни здійснюється шляхом проведення поточних і підсумкових контролів. У завдання поточного контролю входить систематична перевірка розуміння й засвоєння студентом програмного матеріалу, виконання лабораторних робіт, умінь самостійно проробляти тексти складання конспектів, написання звітів, здатності усно або письмово представляти певний матеріал. Перед підсумковим контролем ставиться завдання перевірки глибини засвоєння студентом програмного матеріалу дисципліни, логіки й взаємозв'язки між її окремими розділами, здатності творчо використати придбані знання, умінь сформулювати своє відношення до проблеми, що впливає зі змісту дисципліни.

9. Критерії оцінювання та розподіл балів, які отримують студенти

9.1. Розподіл балів, які отримують студенти (кількісні критерії оцінювання)

Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
Модуль 1			
Робота на лекціях	0...1	5	0...5
Виконання і захист лабораторних (практичних) робіт	0...5	4	0...20
Модульний контроль	0...15	1	0...15
Модуль 2			
Робота на лекціях	0...1	5	0...5
Виконання і захист лабораторних (практичних) робіт	0...5	4	0...20
Модульний контроль	0...15	1	0...15
Виконання і захист РГР (РР, РК)	0...20	1	0...20
Усього за семестр			0...100

Семестровий контроль (іспит/залік) 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/заліку складається з 4 *теоретичних та практичних питань та максимальна кількість балів за кожне питання 25 балів (сума – 100 балів)*.

9.2. Якісні критерії оцінювання

Необхідний обсяг **знань** для одержання позитивної оцінки:

- основні етапи процесу проектування програмного забезпечення;
- типові алгоритмічні конструкції;
- принципи процедурного і структурованого програмування;
- особливості застосування сучасних базових інструментальних програмних засобів, призначених для вирішення економічних задач;
- базові типи даних;
- похідні типи даних: переліки, покажчики, посилання, масиви, структури, об'єднання;
- оператори управління програмою;
- правила роботи з функціями;
- систему введення-виведення C++;
- основні принципи роботи з файлами;

Необхідний обсяг **вмінь** для одержання позитивної оцінки:

- складати програми мовою C++, забезпечуючи:
- рішення задач з курсу вищої математики;
- створення і обробку структур, масивів структур; найпростішу обробку файлів;

- використання функцій;
- використання основних елементів призначеного для користувача інтерфейсу;
- використання сучасного інструментального програмного забезпечення.

9.3 Критерії оцінювання роботи студента протягом семестру

Задовільно (60-74). Показати мінімум знань та умінь. Захистити всі індивідуальні завдання та здати тестування. Знати основні етапи процесу проектування програмного забезпечення, типові алгоритмічні конструкції, принципи процедурного і структурованого програмування; особливості застосування сучасних базових інструментальних програмних засобів, призначених для вирішення економічних задач. Уміти складати програми мовою C++, забезпечуючи: рішення задач з курсу вищої математики, створення і обробку структур, масивів структур; найпростішу обробку файлів.

Добре (75-89). Твердо знати мінімум, захистити всі індивідуальні завдання, виконати всі КР, здати тестування та поза аудиторну самостійну роботу. Уміти використовувати похідні типи даних: переліки, покажчики, посилання, масиви, структури, об'єднання. Володіти операторами управління програмою та використовувати як базові типи даних, так і розроблені користувачем. Знати правила роботи з функціями та основні принципи роботи з файлами. Вміти використовувати функції, основні елементи призначеного для користувача інтерфейсу.

Відмінно (90-100). Здати всі контрольні точки з оцінкою «відмінно». Досконально знати всі теми та уміти застосовувати їх.

Шкала оцінювання: бальна і традиційна

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

10. Методичне забезпечення

Електронний ресурс, на якому розміщено навчально-методичний комплекс дисципліни – http://library.khai.edu/library/fulltexts/doc/_002__Programuvannya.pdf, який включає в себе:

- робоча програма дисципліни;
- конспект лекцій, підручники (навчальні посібники), в тому числі в електронному вигляді, які за змістом повністю відповідають робочій програмі дисципліни;
- методичні вказівки та рекомендації для виконання курсових робіт та проектів, розрахункових та розрахунково-графічних робіт, лабораторних та практичних робіт, а також рекомендації для самостійної підготовки;
- тематики індивідуальних завдань;
- приклади розв'язування типових задач чи виконання типових завдань;
- питання, тести для контрольних заходів;
- каталоги інформаційних ресурсів.

11. Рекомендована література

Базова

1. А.Ю. Соколов, М.Л. Угрюмов, В.А. Халтурин, Ю.К. Чернышев. Информатика [Текст] : конспект лекцій – Харьков : Нац. аерокосм. ун-т «Харк. авіац. ін-т», 2003. – 191 с.
2. Соколов, О.Ю. Информатика для інженерів [Текст] / О.Ю. Соколов, І.Т. Зарецька, Г.М. Жолткевич, О.В. Ярова. – Харків : Факт, 2005. – 423с.
1. Юрченко І.В. Информатика та програмування. Частина 1. Навчальний посібник.– Чернівці: Книги–ХХІ, 2011.– 203 с.
2. Юрченко І.В., Сікора В.С. Информатика та програмування. Частина 2.– Чернівці: Видавець Яворський С.Н., 2015.– 210 с.
3. Хознер С. Visual C++. Учебный курс. СПб.: Питер, 2007. 570 с. (С. 169 – 375).

Допоміжна

1. Караванова Т.П. Информатика: методи побудови алгоритмів та їх аналіз. Необчислювальні алгоритми: Навч. посіб. для 9-10 кл. із поглибл. вивч. інформатики. – К.: Генеза. – 2007.- 216 с.: іл.
2. Караванова Т.П. Информатика: методи побудови алгоритмів та їх аналіз. Обчислювальні алгоритми: Навч. посіб. для 9-10 кл. із поглибл. вивч. інформатики – К.: Генеза. – 2008.- 333 с.: іл.

12. Інформаційні ресурси

1. <https://stm.khai.edu> – Електронний курс Національного аерокосмічного університету ім. М.Є. Жуковського «ХАІ» курс “Основи програмування (Мова C++)”
2. <https://mentor.khai.edu> – Електронний курс Національного аерокосмічного університету ім. М.Є. Жуковського «ХАІ» курс “Основи програмування (Мова C++)”
3. <http://www.twirpx./files/informatics/languages/Visual C++/> - Книги мовою C++
4. <http://progbook.ru/Visual C++/> - Книги на языке C++.
5. <http://citforum.ru/> – Форум. Новини, статті, розсилки, форуми.
6. <http://programming.in.ua/> - Програмування українською.