

Міністерство освіти і науки України

Харківський національний університет імені В.Н. Каразіна

Кафедра прикладної математики

“ЗАТВЕРДЖУЮ”

Декан факультету
математики і інформатики

Григорій ЖОЛТКЕВИЧ

“29” серпня 2024 р.



РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Алгоритми і структури даних

рівень вищої освіти _____ перший (бакалаврський) рівень _____

галузь знань _____ 11 – Математика та статистика _____

спеціальність _____ 113 Прикладна математика _____

освітня програма _____ Прикладна математика _____

спеціалізація _____

вид дисципліни _____ обов'язкова _____

факультет _____ математики і інформатики _____

2024 / 2025 навчальний рік

Програму рекомендовано до затвердження Вченою радою факультету математики і інформатики

“27” серпня 2024 року, протокол № 8

РОЗРОБНИКИ ПРОГРАМИ: *Ігнатович Світлана Юріївна*, доктор фіз.-мат. наук, доцент, професор закладу вищої освіти кафедри прикладної математики.

Програму схвалено на засіданні кафедри прикладної математики
Протокол від “26” серпня 2024 року № 8

Завідувач кафедри прикладної математики



Валерій КОРОБОВ

Програму погоджено з гарантом
освітньо-професійної програми «Прикладна математика»

Гарант освітньо-професійної програми «Прикладна математика»



Сергій ПОСЛАВСЬКИЙ

Програму погоджено науково-методичною комісією
факультету математики і інформатики

Протокол від “27” серпня 2024 року № 1

Голова науково-методичної комісії факультету математики і інформатики



Євген МЕНЯЙЛОВ

ВСТУП

Програма навчальної дисципліни «Алгоритми і структури даних» складена відповідно до освітньо-професійної програми підготовки бакалавра спеціальності 113 Прикладна математика

1. Опис навчальної дисципліни

1.1. Метою викладання навчальної дисципліни є надання студентам теоретичних знань та практичних навичок з основ розробки і аналізу алгоритмів.

1.2. Основні завдання вивчення дисципліни

1. Ознайомлення з основними поняттями і методами аналізу алгоритмів. Ознайомлення з деякими найпоширенішими алгоритмами.
2. Ознайомлення з основними типами структур даних та їх використанням для створення ефективних алгоритмів.
3. Застосування мови Python для розв'язання практичних задач зі створення і аналізу алгоритмів.

1.2.1. Формування наступних інтегральної та загальних компетентностей

ІК01. Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми прикладної математики у професійній діяльності або у процесі навчання, що передбачає застосування математичних теорій та методів і характеризується комплексністю та невизначеністю умов.

ЗК02. Здатність застосовувати знання у практичних ситуаціях.

ЗК06. Здатність до абстрактного мислення, аналізу та синтезу.

1.2.2. Формування наступних фахових компетентностей

ФК04. Здатність розробляти алгоритми та структури даних, програмні засоби та програмну документацію.

ФК08. Здатність використовувати сучасні технології програмування та тестування програмного забезпечення.

ФК09. Здатність до проведення математичного і комп'ютерного моделювання, аналізу та обробки даних, обчислювального експерименту, розв'язання формалізованих задач за допомогою спеціалізованих програмних засобів.

ФК18. Здатність оцінити рівень математичного обґрунтування методів, які застосовуються для розв'язання конкретних прикладних задач.

1.3. Кількість кредитів – 4

1.4. Загальна кількість годин – 120

1.5. Характеристика навчальної дисципліни	
Обов'язкова	
Денна форма навчання	Заочна (дистанційна) форма навчання
Рік підготовки	
2-й	
Семестр	

3-й	
Лекції	
32 год.	
Практичні, семінарські заняття	
32 год.	
Лабораторні заняття	
Самостійна робота	
56 год.	
у тому числі індивідуальні завдання	
6 год.	

1.6. Заплановані результати навчання

У результаті вивчення курсу студенти повинні

знати :

1. Методи створення і аналізу алгоритмів, типи структур даних, деякі алгоритми пошуку і сортування.
2. Основи синтаксису і особливості мови Python.

вміти:

1. Запропонувати алгоритм для розв'язання задачі, записати його за допомогою псевдокоду, провести аналіз складності алгоритму, довести його коректність.
2. Написати програму мовою Python, яка реалізує запропонований алгоритм.

Згідно з вимогами освітньо-професійної програми студенти повинні досягти таких програмних результатів навчання:

РН01. Демонструвати знання й розуміння основних концепцій, принципів, теорій прикладної математики і використовувати їх на практиці.

РН04. Виконувати математичний опис, аналіз та синтез дискретних об'єктів та систем, використовуючи поняття й методи дискретної математики та теорії алгоритмів.

РН09. Будувати ефективні щодо точності обчислень, стійкості, швидкодії та витрат системних ресурсів алгоритми для чисельного дослідження математичних моделей та розв'язання практичних задач.

2. Тематичний план навчальної дисципліни.

Розділ 1. Алгоритми

Тема 1. Аналіз алгоритмів

Запис алгоритму у вигляді блок-схеми і псевдокоду. Обчислювальна складність алгоритмів, знаходження обчислювальної складності. Коректність алгоритмів. Методи доведення коректності, інваріант циклу.

Тема 2. Вступ до мови Python

Особливості мови Python, вбудовані типи даних, особливості синтаксису, підключення модулів. Приклади програм.

Тема 3. Рекурсивні та ітеративні алгоритми

Рекурсивні алгоритми та їх аналіз. Особливості розподілу пам'яті при реалізації рекурсивних алгоритмів. Приклади рекурсивних алгоритмів. Хвостова рекурсія.

Порівняння рекурсивних і ітеративних алгоритмів. Жадібні алгоритми, динамічне програмування.

Тема 4. Пошук і сортування

Алгоритми пошуку і сортування: сортування бульбашкою, вставками, вибором. Сортування злиттям і швидке сортування. Аналіз обчислювальної складності алгоритмів сортування.

Розділ 2. Структури даних

Тема 5. Масив, список, стек, черга.

Структури даних масив, список, стек, черга. Особливості і приклади використання: синтаксичний розбір формул, зворотний польський запис.

Тема 6. Графи і дерева.

Структура даних граф, способи завдання графу. Деревя, кореневі дерева. Сортування купою. Деревя пошуку, AVL-деревя.

Тема 7. Хешування.

Хеш-функції, хеш-таблиці, методи розв'язання колізій.

3. Структура навчальної дисципліни

Назви модулів і тем	Кількість годин					
	Денна форма					
	Усього	у тому числі				
л		п	лаб	інд	ср	
1	2	3	4	5	6	7
Розділ 1. Алгоритми						
Тема 1. Аналіз алгоритмів	24	8	8			8
Тема 2. Вступ до мови Python	16	4	4			8
Тема 3. Рекурсивні та ітеративні алгоритми	18	6	4			8
<i>Контрольна робота</i>	4		2			2
Тема 4. Пошук і сортування	18	4	6			8
<i>Індивідуальне (розрахунково-графічне) завдання</i>	6					6
Разом за розділом 1	86	22	24			40
Розділ 2. Структури даних						
Тема 5. Масив, список, стек, черга	14	4	4			6
Тема 6. Графи і дерева	14	4	4			6
Тема 7. Хешування	4	2				2
<i>Підготовка до заліку</i>	2					2
Разом за розділом 2	34	10	8			16
Усього годин	120	32	32			56

4. Теми практичних занять

№ з/п	Назва теми	Кількість годин
1	Запис алгоритмів. Аналіз обчислювальної складності алгоритмів і доведення їх коректності	4
2	Знайомство з мовою Python, написання найпростіших програм.	4

3	Приклади алгоритмів, їх запис, складання програм. Цикли. Реалізація алгоритмів на мові Python з використанням списків (list).	4
4	Рекурсивні алгоритми. Реалізація рекурсивних алгоритмів на мові Python.	2
5	Деякі оптимізаційні алгоритми: жадібний алгоритм, метод динамічного програмування.	2
6	<i>Контрольна робота</i>	2
7	Алгоритми пошуку і сортування і їх реалізація за допомогою мови Python. Бібліотека Turtle і візуалізація алгоритмів сортування.	6
8	Реалізація стека і черги на мові Python. Задачі, де використовуються стек і черга.	4
9	Реалізація сортування купою. Дерева пошуку, AVL-дерева, балансування.	4
Разом		32

5. Завдання для самостійної роботи

№ з/п	Види, зміст самостійної роботи	Кількість годин
1	Виконання домашніх завдань з аналізу обчислювальної складності і доведення коректності алгоритмів.	8
2	Виконання домашніх завдань з ознайомлення з особливостями синтаксису і написання найпростіших програм на мові Python.	8
3	Виконання домашніх завдань з написання та аналізу програм мовою Python, які використовують рекурсивні функції.	8
4	Підготовка до контрольної роботи	2
5	Виконання домашніх завдань з алгоритмів пошуку і сортування.	6
6	Виконання домашнього завдання з використання модуля Turtle для візуалізації.	2
7	Виконання індивідуального розрахунково-графічного завдання	6
8	Виконання домашніх завдань з ознайомлення зі структурами даних стек і черга	6
9	Виконання домашніх завдань з ознайомлення зі структурою даних дерево	6
10	Виконання домашніх завдань з ознайомлення з поняттям хешування	2
11	Підготовка до заліку	2
Разом		56

6. Індивідуальні завдання

Обрати два методи сортування (на вибір), написати програму на мові Python, яка реалізує ці методи, з використанням модуля Turtle для візуалізації.

7. Методи навчання

Методи навчання: частково-пошуковий, пояснювально-ілюстративний. Студенти опановують значну частину теоретичного матеріалу шляхом самостійного написання комп'ютерних програм.

8. Методи контролю

1. Перевірка виконання домашніх завдань, поточне опитування за лекційним матеріалом, поточний контроль написання програм.

2. Перевірка контрольної роботи і індивідуального (розрахунково-графічного) завдання.
3. Перевірка залікової роботи.

9. Схема нарахування балів

Поточний контроль, самостійна робота, індивідуальні завдання					Разом	Залікова робота	Сума	
Розділ 1		Розділ 2		Контрольна робота				Розрахунково-графічне завдання
T1-T2	T3-T4	T5-T7						
10	10	10		15	15	60	40	100

Мінімальна кількість балів з навчальної дисципліни, яку здобувач вищої освіти повинен набрати під час поточного контролю, самостійної роботи, індивідуального завдання для допуску до складання підсумкового контролю (заліку), не передбачена програмою.

Критерії оцінювання навчальних досягнень

Поточний контроль: бали нараховуються за виконання домашніх завдань і активність під час практичних занять.

Зміст контрольної роботи: запис і читання блок-схем алгоритмів, запис алгоритмів на псевдокод, аналіз коректності і обчислювальної складності алгоритму. Студентам пропонується дві задачі: (1) прочитати блок-схему, пояснити, який алгоритм реалізується, довести його коректність, знайти його обчислювальну складність (максимальна оцінка 7 балів), (2) запропонувати алгоритм для розв'язання заданого завдання, записати його у вигляді блок-схеми або псевдокоду, довести його коректність, знайти його обчислювальну складність (максимальна оцінка 8 балів). Якщо пояснення відсутні або неповні, бал може бути знижений.

Зміст індивідуального завдання: обрати два методи сортування (на вибір), написати програму на мові Python, яка реалізує ці методи, з використанням модуля Turtle для візуалізації. Максимальна оцінка – 15 балів. Якщо завдання виконане не повністю, бали можуть бути знижені:

- написана програма з помилками, які студент не виправив: 0–3 бали.
- написана програма з помилками, які студент виправив під час здачі завдання: 4–10 бали.
- сортування написане коректно, але візуалізація не реалізована: 6–11 балів.
- візуалізація працює, але сортування некоректне: 5–10 балів.
- сортування коректне, але візуалізація неповна (не всі етапи сортування видно тощо): 9–14 балів.
- сортування і візуалізація повні і коректні: 15 балів.

Залікова робота передбачає письмову відповідь на два питання зі списку, який надається студентам. Питання включають теоретичний і практичний матеріал, який вивчався протягом семестру. До кожного питання обов'язково наводити пояснювальні приклади. Максимальна оцінка за роботу – 40 балів.

- Теоретичний зміст питань не розкритий, наведені приклади не відповідають змісту питань: 0–5 балів.
- Теоретичний зміст одного або обох питань в цілому розкритий, але наведені приклади не відповідають змісту питань: 10–20 балів.
- Теоретичний зміст одного або обох питань в цілому не розкритий, але наведені приклади відповідають змісту питань: 10–20 балів.
- Теоретичний зміст питань в цілому розкритий, наведені приклади в цілому відповідають змісту питань, але робота містить суттєві помилки: 20–30 балів.

- Теоретичний зміст питань розкритий, наведені приклади відповідають змісту питань, але робота містить несуттєві помилки: 30–35 балів.
- Теоретичний зміст питань повністю розкритий, наведені приклади повністю відповідають змісту питань: 40 балів.

Шкала оцінювання: дворівнева

Сума балів за всі види навчальної діяльності протягом семестру	Оцінка
50-100	зараховано
1-49	не зараховано

10. Рекомендована література

Основна література

1. А. П. Крєневич, Алгоритми і структури даних. Підручник. Київ, 2021.
2. Т. О. Коротеєва, Алгоритми та структури даних: навч. посібник. Львів, 2014.
3. Алгоритми та структури даних. Навчальний посібник. Укладач Ю.Є. Грудзинський. Київ, НТУУ "КПІ ім. Ігоря Сікорського", 2022.
<https://ela.kpi.ua/server/api/core/bitstreams/0db974f9-16fa-459c-9f19-fab0021222ed/content>

Допоміжна література

1. D. Knuth, The Art of Computer Programming, Vol. 1-3.
2. N. Wirth, Algorithms + Data Structures = Programs.
3. M. T. Goodrich, R. Tamassia, M. H. Goldwasser, Data Structures and Algorithms in Python. 2013.

11. Посилання на інформаційні ресурси в Інтернеті, відео-лекції, інше методичне забезпечення

1. Problem Solving with Algorithms and Data Structures using Python [Електронний ресурс] – Режим доступу до ресурсу:
<https://runestone.academy/runestone/books/published/pythonds/index.html>.
2. The Python Tutorial [Електронний ресурс] – Режим доступу до ресурсу:
<https://docs.python.org/3/tutorial/index.html>.
3. Leetcode platform <https://leetcode.com/problemset/>
4. О.І. Жмурко, Т.О. Охріменко. Олімпіадне програмування. Прості задачі. Умань, Візаві, 2020 р. <https://dspace.udpu.edu.ua/bitstream/123456789/12975/1/Olymp.pdf>