

Міністерство освіти і науки України
Харківський національний університет імені В.Н. Каразіна
Кафедра прикладної математики

“ЗАТВЕРДЖУЮ”

Декан факультету
математики і інформатики

Григорій ЖОЛТКЕВИЧ

2023 р.



РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Алгоритми і структури даних

(назва навчальної дисципліни)

рівень вищої освіти бакалавр

галузь знань 11 – Математика та статистика

спеціальність 113 – Прикладна математика
(шифр і назва)

освітня програма Прикладна математика
(шифр і назва)

спеціалізація _____
(шифр і назва)

вид дисципліни обов'язкова
(обов'язкова / за вибором)

факультет математики і інформатики

2023 / 2024 навчальний рік


Програму рекомендовано до затвердження вченою радою факультету математики і інформатики
 “29” серпня 2023 року, протокол №8

РОЗРОБНИКИ ПРОГРАМИ:

Ігнатович Світлана Юрївна, доктор фізико-математичних наук, доцент, професор
 кафедри прикладної математики.


Програму схвалено на засіданні кафедри Прикладної математики
 Протокол від “28” серпня 2023 року №10

Завідувач кафедри Прикладної математики


 (підпис) Валерій КОРОБОВ
 (прізвище та ініціали)

Програму погоджено з гарантом освітньо-професійної програми (керівником проектної
 групи) Прикладна математика
 назва освітньої програми

Гарант освітньо-професійної програми
 (керівник проектної групи) Прикладна математика


 (підпис) Сергій ПОСЛАВСЬКИЙ
 (прізвище та ініціали)

Програму погоджено науково-методичною комісією факультету математики
і інформатики Харківського національного університету імені В.Н. Каразіна
 назва факультету, для здобувачів вищої освіти якого викладається навчальна дисципліна

Протокол від “29” серпня 2023 року, протокол №1

Голова науково-методичної комісії факультету математики і інформатики


 (підпис) Ольга АНОЩЕНКО
 (прізвище та ініціали)

ВСТУП

Програма навчальної дисципліни «**Алгоритми і структури даних**» складена відповідно до освітньо-професійної програми підготовки

бакалавр

(назва рівня вищої освіти)

спеціальності (напрямку) 113 – Прикладна математика

спеціалізації _____

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.

1.3. Кількість кредитів – 4

1.4. Загальна кількість годин – 120

1.5. Характеристика навчальної дисципліни

Обов'язкова	
Денна форма навчання	Заочна (дистанційна) форма навчання
Рік	
2-й	
Се	
3-й	
Л	
32 год.	.
Практичні, семінарські заняття	
32 год.	.
Лабораторні	
Самостійна робота	
56 год.	.
у тому числі індивідуальні завдання	
6 год.	

1.6. Заплановані результати навчання

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.

2. Тематичний план навчальної дисципліни.**Розділ 1. Алгоритми***Тема 1. Вступ до мови Python.*

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

Тема 2. Аналіз алгоритмів

Запис алгоритму у вигляді псевдокоду. Обчислювальна складність алгоритмів, знаходження обчислювальної складності. Коректність алгоритмів. Методи доведення коректності.

Тема 3. Рекурсивні та ітеративні алгоритми.

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

Тема 4. Пошук і сортування

Алгоритми пошуку і сортування. Аналіз складності, приклади.

Розділ 2. Структури даних*Тема 5. Список, дерево, стек, черга.*

Основні структури даних: список, дерево, стек, черга. Особливості і приклади використання.

Тема 6. Множина, словник, хеш-таблиця, граф.

Структури даних множина, словник, хеш-таблиця. Особливості і приклади використання.

2. Структура навчальної дисципліни

Назви модулів і тем	Кількість годин					
	Денна форма					
	Усього	у тому числі				
л		п	лаб	інд	сп	
1	2	3	4	5	6	7
Розділ 1. Алгоритми						
Тема 1. Вступ до мови Python	16	4	4			8
Тема 2. Аналіз алгоритмів	24	8	8			8
Тема 3. Рекурсивні та ітеративні алгоритми	20	8	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	88	24	24			40

Розділ 2. Структури даних						
Тема 5. Список, дерево, стек, черга	14	4	4			6
Тема 6. Множина, словник, хеш-таблиця, граф	14	4	4			6
<i>Підготовка до заліку</i>	4					4
Разом за розділом 3	32	8	8			16
Усього годин	120	32	32			56

3. Теми практичних занять

№ з/п	Назва теми	Кількість годин
1	Знайомство з мовою Python, написання найпростіших програм.	4
2	Приклади алгоритмів, їх запис, складання програм. Цикли. Реалізація алгоритмів на мові Python з використанням списків (list).	4
3	Аналіз обчислювальної складності алгоритмів і доведення їх коректності	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.	2
8	Бібліотека Turtle і візуалізація алгоритмів сортування.	4
9	Реалізація типів даних за допомогою вбудованих типів даних у мові Python.	2
10	Класи у мові Python. Створення і використання класів для моделювання структур даних.	2
11	Графи і їх застосування для розв'язання задач пошуку і оптимізації.	4
Разом		32

4. Завдання для самостійної роботи

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

5. Індивідуальні завдання

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

6. Методи навчання

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

7. 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.

8. Схема нарахування балів

Поточний контроль, самостійна робота, індивідуальні завдання				Залікова робота	Сума
Розділ 1		Розділ 2	Контрольна робота		
T1–T2	T3–T4	T5–T6			
10	10	10	10	20	40

T1 – T6 – теми розділів.

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. Максимальна оцінка – 10 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 – 20 балів. Бали нараховуються за реалізацію кожного методу окремо; по 10 балів за кожний метод:

- Написана програма з помилками, які студент не виправив: 0–2 бали.
- Написана програма з помилками, які студент виправив під час здачі завдання: 3–4 бали.
- Сортування написане коректно, але візуалізація не реалізована: 3–6 балів.
- Візуалізація працює, але сортування некоректне: 3–6 балів.
- Сортування коректне, але візуалізація неповна (не всі етапи сортування видно і т.д.): 7–9 балів.
- Сортування і візуалізація повні і коректні: 10 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. Коротєєва, Алгоритми та структури даних: навч. посібник. Львів, 2014.
3. D. Knuth, The Art of Computer Programming, Vol. 1-3.
4. N. Wirth, Algorithms + Data Structures = Programs.

Допоміжна література

1. 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>.