

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ




РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Алгоритми та структури даних



Галузь знань 12 – Інформаційні технології
 Спеціальність 121 – Інженерія програмного забезпечення
 Рівень вищої освіти – Перший бакалаврський
 Освітньо-професійна програма – Інженерія програмного забезпечення
 Обсяг дисципліни – 7 кредитів ЄКТС, Шифр дисципліни – ОЗП.15
 Статус дисципліни: обов'язкова, Мова навчання Англійська, українська
 Факультет – Інформаційних технологій
 Кафедра – Інженерії програмного забезпечення

Форма навчання	Курс	Семестр	Загальне навантаження		Кількість годин						Вид семестрового контролю			
			Європ. кредити	Години	Аудиторні заняття				Індивідуальна робота студента	Самостійна робота	Курсовий проект	Курсова робота	Залік	Іспит
					Всього	Лекції	Лабораторні роботи	Практичні заняття						
Денна	2	4	7	210	72	36	36			138				+

Робоча програма складена на основі Стандарту вищої освіти, освітньої програми підготовки бакалаврів 2023 року та навчального плану

Програма складена  Леонід БЕДРАТЮК

Схвалено на засіданні кафедри ПЗ
 протокол № 7 від 29 05 2023 р.

Зав. кафедри інженерії програмного забезпечення  Леонід БЕДРАТЮК
 Робоча програма розглянута та схвалена Вченою радою факультету інформаційних технологій
 Голова Вченої ради  Олег САВЕНКО
 Підпис Ініціали, прізвище

АЛГОРИТМИ ТА СТРУКТУРИ ДАНИХ

Тип дисципліни	Обов'язкова
Рівень вищої освіти	Перший (бакалаврський)
Мова викладання	Українська, Англійська
Семестр	4
Обсяг кредитів ЄКТС	7
Форми здобуття освіти	Очна (денна)

Результати навчання. Студент, який успішно завершив вивчення дисципліни, має: Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань. *Вміти* формалізувати обчислювальні та логічні задачі; *досліджувати* ефективність алгоритмів; *використовувати* основні структури даних (масиви, множини, стеки, дерева); *використовувати* способи конструювання складних структур даних на базі простих структур даних, *шукати* інформацію в структурах даних використовуючи різні способи пошуку; *вибрати* оптимальну структуру даних та оптимальний алгоритм розв'язання задачі.

Пререквізити - Математичний аналіз, Теорія ймовірностей та математична статистика, Об'єктно-орієнтоване програмування, Дискретна математика

Кореквізити –Методи оптимізації, Моделювання та оцінка програмного забезпечення,

Зміст навчальної дисципліни. Алгоритми та їх аналіз. Алгоритми сортування. Прості структури даних. Хеш-таблиці. Зв'язні структури даних. Рекурсія. Алгоритми пошуку. Алгоритми на графах. Обчислювальна геометрія.

Запланована навчальна діяльність. Лекції – 36 год, лабораторні заняття –36 год., самостійна робота – 138 год., разом – 210 год.

Форми (методи) навчання: лекції (з використанням методів проблемного навчання і візуалізації); лабораторні заняття, самостійна робота.

Форми оцінювання результатів навчання: захист лабораторних робіт, тестування; письмова підсумкова робота.

Форма семестрового контролю: іспит

Навчальні ресурси:

1. Томас Г. Кормен, Чарлз Е. Лейзерсон, Роналд Л. Рівест, Кліффорд Стайн. Вступ до алгоритмів. Переклад з англійської третього видання. Київ, К.І.С., 2019, 1288 стор.
2. Крєневич А.П. Алгоритми і структури даних. Підручник. – К.: ВПЦ "Київський Університет", 2021. – 200 с..
3. I. Ahmad, 40 Algorithms Every Programmer Should Know, Packt Publishing, 2020
4. Васильєв О., Програмування мовою Python. –Навчальна книга – Богдан 2019 – 504 стр.
5. Електронна бібліотека університету. Доступ до ресурсу: <http://library.khmnu.edu.ua>
6. Репозитарій ХНУ:<https://elar.khmnu.edu.ua/home>

Викладач: доктор фізико-математичних наук, професор Леонід БЕДРАТЮК

3. ПОЯСНЮВАЛЬНА ЗАПИСКА

Дисципліна "Алгоритми та структури даних" є однією з дисциплін загальної підготовки і посідає важливе місце у підготовці фахівців освітнього рівня «бакалавр» за освітньо-професійною програмою «Інженерія програмного забезпечення».

Пререквізити - Математичний аналіз, Теорія ймовірностей та математична статистика, Об'єктно-орієнтоване програмування, Дискретна математика

Кореквізити –Методи оптимізації, Моделювання та оцінка програмного забезпечення,

Мета дисципліни. Дисципліна "Алгоритми та структури даних" має метою здобуття студентами загальних і фахових компетенцій в області алгоритмізації та реалізації базових алгоритмів та структур даних.

Предмет дисципліни. Типові алгоритми та структури даних які зустрічаються в програмуванні.

Завдання дисципліни. Надати студентам знання і практичні навички з проектування оптимальних алгоритмів та вибору оптимальних структур даних для вирішення типових алгоритмічних задач.

Відповідно до *Стандарту вищої освіти* із та освітньої програми дисципліна сприяє забезпеченню:

компетентностей:

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК2. Здатність застосовувати знання у практичних ситуаціях.

ЗК5. Здатність вчитися і оволодівати сучасними знаннями.

ЗК6. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ФК7. Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних.

ФК8. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення

ФК14. Здатність до алгоритмічного та логічного мислення.

програмних результатів навчання:

ПРН5 Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення.

ПРН7 Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.

ПРН13 Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.

Результати навчання. Студент, який успішно завершив вивчення дисципліни, має: Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань. *Вміти* формалізувати обчислювальні та логічні задачі; *досліджувати* ефективність алгоритмів; *використовувати* основні структури даних (масиви, множини, стеки, дерева); *використовувати* способи конструювання складних структур даних на базі простих структур даних, *шукати* інформацію в структурах даних використовуючи різні способи пошуку; *вибрати* оптимальну структуру даних та оптимальний алгоритм розв'язання задачі.

Політика дисципліни Організація освітнього процесу з дисципліни відповідає вимогам положень про організаційне і навчально-методичне забезпечення освітнього процесу, освітній програмі та навчальному плану. Студент зобов'язаний відвідувати лекції, практичні заняття, лабораторні роботи, тощо, згідно з розкладом, не запізнюватися на заняття, виконувати усі завдання та контрольні точки відповідно до графіка. Пропущені практичні заняття і лабораторні роботи студент зобов'язаний опрацювати самостійно у повному обсязі і відзвітувати перед викладачем не пізніше, ніж за тиждень до чергової атестації. До лабораторних робіт студент має підготуватися за відповідною темою і проявляти активність. Набутті особою знання з дисципліни або її окремих розділів у неформальній освіті зараховуються відповідно до Положення про порядок перерахування результатів навчання та визначення академічної різниці у ХНУ

4. СТРУКТУРА ЗАЛКОВИХ КРЕДИТІВ ДИСЦИПЛІНИ

Назва теми	Кількість годин, відведених на:		
	Лекції	Лаб роб.	Сам. роб.
2 семестр			
Тема 1. Алгоритми та їх аналіз	4	4	13
Тема 2. Алгоритми сортування	6	6	13
Тема 3. Прості структури даних	4	4	14
Тема 4. Хеш-таблиці	2	2	14
Тема 5. Зв'язні структури даних	2	2	14
Тема 6. Рекурсія	2	2	14
Тема 7. Алгоритми пошуку	6	6	14
Тема 8. Алгоритм швидкого перетворення Фур'є	2	2	14
Тема 9. Алгоритми на графах	6	6	14
Тема 10. Обчислювальна геометрія	4	6	14
Всього семестр	36	36	138

5. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

5.1. Зміст лекційного курсу*

Номер лекції	Перелік тем лекцій, їх анотації	Кількість годин
	<i>Другий семестр</i>	
1-2	<p>Тема 1. Алгоритми та їх аналіз</p> <p><i>Лекція 1. Алгоритми та їх властивості.</i> Означення алгоритму та його властивості. Порівняння алгоритмів, часова та просторова складність. Розмір вхідних даних, час роботи алгоритму. Час роботи в найгіршому, найкращому та середньому випадку.</p> <p><i>Лекція 2. Аналіз алгоритмів</i> Теоретичний та експериментальний аналіз часу роботи. Порядок росту, асимптотичний аналіз порядку росту часу роботи алгоритму. Позначення O, Ω-велике. Складності $O(n)$, $O(n \log(n))$, $O(n^2)$.</p>	4

3-5	<p>Тема 2. Алгоритми сортування</p> <p><i>Лекція 3. Алгоритми сортування квадратичної складності.</i> Алгоритми сортування на основі порівняння та їхній аналіз: Сортування вставленням, бульбашкове сортування, сортування виділенням.</p> <p><i>Лекція 4. Метод розділяй і володарюй.</i> Алгоритми сортування лінеологарифмічної складності. Сортування злиттям, швидке сортування</p> <p><i>Лекція 5. Алгоритми сортування лінійної складності.</i> Алгоритми сортування без використання порівняння. Порозрядне сортування, сортування підрахунком, коміркове сортування.</p>	6
6-7	<p>Тема 3. Прості структури даних</p> <p><i>Лекція 6. Стек, черга, дек.</i> Абстрактні структури даних, Стек та його реалізації. Черга, дек та їхні реалізації. Типові алгоритми в яких застосовуються прості структури даних.</p> <p><i>Лекція 7. Дерева.</i> Означення дерева, типи дерев та їх реалізація. Використання дерев в алгоритмах: алгоритм пірамідального сортування, черга з пріоритетом.</p>	4
8	<p>Тема 4. Хеш-таблиці</p> <p><i>Лекція 8. Хеш-таблиці.</i> Хешування та вирішення колізій. Хеш-функції та хеш-коди. Відкрита адресація та ланцюжок. Коефіцієнт навантаження. Реалізація хеш-таблиці</p>	2
9	<p>Тема 5. Зв'язні структури даних</p> <p><i>Лекція 9. Зв'язні структури даних.</i> Зв'язні структури: Однозв'язні структури та двозв'язні структури. Операції над однозв'язними структурами: обхід, пошук, заміна, вставка, видалення. Циркулярні та двозв'язні структури.</p>	2
10	<p>Тема 6. Рекурсія</p> <p><i>Лекція 10. Рекурсія.</i> Рекурсивні функції та їх властивості. Рекурсивне дерево викликів. Застосування рекурсії - пошук з поверненням, задача про ханойські вежі, рекурентні послідовності.</p>	2
10-12	<p>Тема 7. Алгоритми пошуку</p> <p><i>Лекція 11. Алгоритми пошуку в масивах і деревах.</i> Лінійний, бінарний інтерполяційний пошук. Бінарні дерева пошуку. Червоно-чорні дерева пошуку</p> <p><i>Лекція 12. Алгоритми пошуку в рядках.</i> Алгоритм Рабіна-Карпа. Алгоритм Моріса-Кнута-Пратта. Алгоритм Ахо-Корасик та алгоритм Боєра-Мура-Хорспула</p>	4
13	<p>Тема 8. Алгоритм швидкого перетворення Фур'є</p> <p><i>Лекція 13. Множення многочленів та алгоритм швидкого перетворення Фур'є.</i> Алгоритми множення многочленів та їхня складність. Дискретне та швидке перетворення Фур'є та їхні застосування.</p>	2
14-15	<p>Тема 8. Алгоритми на графах.</p> <p><i>Лекція 14. Алгоритми обходу графів.</i> Подання графів. Пошук в глибину, пошук в ширину. Топологічне сортування.</p>	6

	<p><i>Лекція 15. Мінімальне остовне дерево. Побудова мінімального каркасного дерева. Алгоритми Крускала, Пріма,</i></p> <p><i>Лекція 16. Найкоротші шляхи в графах. Побудова найкоротшого шляху. Алгоритми Дейкстри і Флойда, застосування.</i></p>	
17-18	<p>Тема 10. Обчислювальна геометрія</p> <p><i>Лекція 17. Найпростіші задачі обчислювальної геометрії. Поворот відрізка, перетин двох відрізків. Належність точки до фігури. Метод вимітаючої прямої. Пошук найближчої точки</i></p> <p><i>Лекція 18. Випукла оболонка, триангуляція, діаграми Вороного. Випуклої оболонки множини точок та її властивості. Алгоритми побудови випуклої оболонки - обхід за Джарвісом, сканування за Грехемом. Триангуляція, діаграми Вороного.</i></p>	4
	Разом за другий семестр:	36

5.2 Зміст лабораторних занять

Номер теми	Тема лабораторного заняття	Кількість годин
1	Алгоритми та їх аналіз	4
2	Встановлення Python, середовища Jupyter notebook. Типи даних, оператори та функції мови Python.	6
3	Вбудовані структури даних: списки, множини, кортежі.	4
4	Стеки, черги, деки	4
5	Хеш-таблиці.	2
6	Алгоритми пошуку підрядків, модуль re.	4
7	Алгоритми на графах.	6
8	Обчислювальна геометрія.	6
Разом за семестр		36

5.3 Зміст самостійної роботи

Номер тижня	Зміст самостійної роботи	Кількість годин
1	Опрацювання лекційного матеріалу. . Підготовка до виконання лабораторної роботи №1	9
2	Опрацювання лекційного матеріалу. . Підготовка до виконання лабораторної роботи №1	9
3	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи №2	9
4	Опрацювання лекційного матеріалу. . Підготовка до виконання лабораторної роботи №2	9
5	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи №3	9
6	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи №3 Захист лабораторної роботи №3	9
7	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи №4	9
8	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи №4	9
9	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи №5	9
10	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи №5.	9
11	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи № 6. Захист лабораторної роботи №6	8

12	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи №6	8
13	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи №6.	8
14	Опрацювання лекційного матеріалу. Підготовка до виконання та захисту лабораторної роботи №6	8
15	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи №7	8
16	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи №7	8
17	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи №8 Захист лабораторної роботи №8	8
18	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи №8	8
	Разом за семестр	138

6. МЕТОДИ НАВЧАННЯ

Процес навчання з дисципліни ґрунтується на використанні традиційних та сучасних технологій, зокрема: лекції (з використанням методів проблемного навчання і візуалізації); лабораторні (з використанням методів інформаційних технологій та сучасних інтегрованих середовищ програмування, майстер-класів), самостійна робота, і які мають за мету – здобуття загальних і фахових компетенцій в області алгоритмізації та реалізації базових алгоритмів та структур даних.

7. ФОРМИ І МЕТОДИ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Контроль за засвоєнням навчального матеріалу здійснюється на основі поточного контролю. При оцінюванні знань студентів здійснюються різні засоби контролю, зокрема поточний контроль (опрацювання теоретичного матеріалу), тестові завдання. Проміжний контроль полягає у перевірці знань студентів по темам. Одержані студентом оцінки під час поточного, проміжного контролю не перепадають. Кожний вид роботи оцінюється за чотирибальною шкалою. Підсумкова оцінка за семестр виставляється із урахуванням всіх оцінок, одержаних студентом за семестр та результатів заліку. Вагові коефіцієнти змінюються залежно від структури дисципліни. Кожний вид роботи з дисципліни оцінюється за **чотирибальною** шкалою. Семестрова підсумкова оцінка визначається як середньозважена з усіх видів навчальної роботи, виконаних і зданих **позитивно** з врахуванням вагових коефіцієнтів. Студент, який набрав позитивний середньозважений бал за поточну роботу і не здав підсумковий контрольний захід, вважається невстигаючим.

При оцінюванні знань студентів здійснюються різні засоби контролю, зокрема поточний контроль здійснюється під час лабораторних занять, при чому враховуються різні види роботи. Окремо здійснюється контроль за виконанням самостійної та індивідуальної роботи.

Пропущене заняття студент повинен відпрацювати у встановлений викладачем термін. Протягом семестру студент повинен отримати не менше 4 оцінок на лабораторних заняттях.

При оцінюванні знань студентів викладач керується такими критеріями.

Оцінку „відмінно” отримує студент за глибоке і повне опанування змісту навчального матеріалу, в якому він легко орієнтується, понятійного апарату, за уміння зв'язувати теорію з практикою, вирішувати практичні завдання, висловлювати і обґрунтовувати свої судження і конструктивні

рішення. Відмінна оцінка передбачає грамотний, логічний виклад відповіді (як в усній, так і в письмовій формі), якісне зовнішнє оформлення.

Оцінку „добре” отримує студент за повне засвоєння навчального матеріалу, володіння понятійним апаратом, орієнтування в вивченому матеріалі, свідоме використання знань для вирішення практичних завдань, грамотний виклад відповіді, але у змісті і формі відповіді мали місце окремі неточності (похибки), нечіткі формулювання закономірностей тощо. Відповідь студента має будуватись на основі самостійного мислення.

Оцінки "задовільно" заслуговує студент, який виявив знання основного навчально-програмного матеріалу в обсязі, необхідному для подальшого навчання та практичної діяльності за професією, що справляється з виконанням практичних завдань, передбачених програмою.

Оцінки "задовільно" заслуговує студент за неповне опанування програмного матеріалу.

Оцінка „незадовільно” виставляється, коли студент має розрізнені, безсистемні знання, не вміє виділяти головне і другорядне, допускається помилок у визначенні понять, перекручує їх зміст, хаотично і невпевнено викладає матеріал, не може використовувати знання при вирішенні практичних завдань.

На основі результатів поточного контролю виставляється підсумкова семестрова оцінка.

Структурування дисципліни за видами робіт і оцінювання результатів навчання студентів за ваговими коефіцієнтами

Аудиторна робота			Семестровий контроль	
Лабораторні роботи	Тестовий контроль			Іспит
№ 1 – 8	TK1	TK2	TK3	
0,5	0,1			0,4

Співвідношення вітчизняної шкали оцінювання і шкали оцінювання ЄКТС

Оцінка ЄКТС	Інтервальна шкала балів	Вітчизняна оцінка	
A	4,75–5,00	5	Відмінно – глибоке і повне опанування навчального матеріалу і виявлення відповідних умінь та навиків
B	4,25–4,74	4	Добре – повне знання навчального матеріалу з кількома незначними помилками
C	3,75–4,24	4	Добре – в загальному правильна відповідь з двома-трьома суттєвими помилками
D	3,25–3,74	3	Задовільно – неповне опанування програмного матеріалу, але достатнє для практичної діяльності за професією
E	3,00–3,24	3	Задовільно – неповне опанування програмного матеріалу, що задовольняє мінімальні критерії оцінювання
FX	2,00–2,99	2	Незадовільно – безсистемність одержаних знань і неможливість продовжити навчання без додаткових знань з дисципліни
F	0,00–1,99	2	Незадовільно – необхідна серйозна подальша робота і повторне вивчення дисципліни

Екзамен виставляється при отриманні студентом з дисципліни від 3,00 до 5,00 балів. При цьому за вітчизняною шкалою ставиться «зараховано», а за шкалою ECTS - оцінка, що відповідає набраній студентом кількості балів.

8. ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

1. Поняття про алгоритм, властивості алгоритмів
2. Задача сортування, сортування вставкою, аналіз алгоритму сортування вставкою.
3. Розмір вхідних даних, час роботи алгоритму.
4. Час роботи алгоритму в найкращому, середньому і найгіршому випадках.
5. Алгоритм сортування Merge Sort та його аналіз.
6. Порядок росту функції. Асимптотичні позначення.
7. Алгоритм множення Карацуби та його аналіз.
8. Структура даних піраміда та її властивості.
9. Структура даних піраміда. Алгоритм створення піраміди та алгоритм підтримки властивостей піраміди.
10. Алгоритм пірамідального сортування HeapSort та його аналіз.
11. Динамічні множини та операції в динамічних множинах.
12. Структура даних стек та її властивості.
13. Структура даних список та її властивості.
14. Структура даних черга та її властивості.
15. Структура даних черга з пріоритетом та її властивості.
16. Структура даних бінарне дерево пошуку та його властивості.
17. Структура даних бінарне дерево. Алгоритми вставки та видалення вузла.
18. Алгоритм сортування на основі бінарного дерева та його аналіз.
19. Структура даних хеш-таблиця та її властивості.
20. Хеш-функції.
21. Алгоритм сортування Quick Sort та його аналіз.
22. Алгоритм сортування підрахунком Counting Sort та його аналіз.
23. Алгоритм порозрядного сортування Radix Sort та його аналіз.
24. Алгоритм карманного сортування Bucket Sort та його аналіз.
25. Наївний алгоритм пошуку підрядків та його аналіз.
26. Алгоритм пошуку підрядків Рабіна-Карпа та його аналіз.
27. Алгоритм пошуку підрядків Кнута-Моріса-Пратта та його аналіз.
28. Структура даних зв'язний список та її властивості
29. Алгоритми Пріма, Борувки, Краскала
30. Пошук в ширину і глибину
31. Алгоритм Флойда
32. Рекурсія та їх властивості
33. Циркулярні та двозв'язні списки.
34. Алгоритми пошуку в масиві
35. Бінарний та інтерполяційний пошук
36. Найпростіші задачі обчислювальної геометрії
37. Алгоритми обходу графів. Пошук в глибину.

38. Алгоритми обходу графів. пошук в ширину.
39. Алгоритм топологічного сортування графів.
40. Мінімальне остовне дерево. Алгоритми Крускала, Пріма
41. Мінімальне остовне дерево. Алгоритми Пріма
42. Найкоротші шляхи в графах. Алгоритми Дейкстри
43. Найкоротші шляхи в графах. Алгоритми Флойда.
44. Найпростіші задачі обчислювальної геометрії. Поворот відрізка, перетин двох відрізків.
45. Найпростіші задачі обчислювальної геометрії. Належність точки до фігури
46. Алгоритми побудови випуклої оболонки - обхід за Джарвісом
47. Алгоритми побудови випуклої оболонки - сканування за Грехемом.
48. Триангуляція,
49. Діаграми Вороного.

9. МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

Освітній процес з дисципліни «Алгоритми та структури даних» повністю і в достатній кількості забезпечений необхідною навчально-методичною літературою.

Algorithms and Data Structures: Guidelines for performing laboratory works for students of the first (bachelor's) level of higher education, specialty 121 "Software Engineering"/ L.P. Bedratyuk, A.I. Bedratyuk – Khmelnytskyi: KhNU, 2023 – 114 p.

10. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна

1. Томас Г. Кормен, Чарлз Е. Лейзерсон, Роналд Л. Рівест, Кліфорд Стайн. Вступ до алгоритмів. Переклад з англійської третього видання. Київ, К.І.С., 2019, 1288 стор.
2. Крєневич А.П. Алгоритми і структури даних. Підручник. – К.: ВПЦ "Київський Університет", 2021. – 200 с.
3. I. Ahmad, 40 Algorithms Every Programmer Should Know, Packt Publishing, 2020
4. Васильєв О., Програмування мовою Python. – Навчальна книга – Богдан 2019 – 504 стр.
5. Шаховська Н. Б., Голощук Р. О. Алгоритми і структури даних. Навчальний посібник. Львів : Магнолія, 2018. 216 с

Додаткова

6. Крєневич А.П. Python у прикладах і задачах. Частина 1. Структурне програмування, Навчальний посібник, – К.: ВПЦ "Київський Університет", 2017. – 206 с
7. Коваль В.С., Струбицький П.Р. Алгоритми і структури даних. – Навчальний посібник – Тернопіль: ФОП Шпак В. Б. – 2017. – 74 с.
8. Ткачук В.М., Алгоритми і структура даних: Навчальний посібник, - ІваноФранківськ : Видавництво ПНУ, 2016.-286 с
9. Т.О. Гришанович, Курс лекцій з дисципліни «Алгоритми та структури даних», ВНУ імені Лесі Українки, Луцьк , 2021. – 110 с.
10. В.М. Ільман, О.П. Іванов, Л.О. Панік., Алгоритми, дані і структури. навч. посіб., Дніпропет. нац. ун-т залізн. трансп.ім. акад. В. Лазаряна. – Дніпро, 2019. – 134 с.
11. О. О. Петрова, Г. В. Солодовник, Алгоритмічні задачі та їх вирішення : навч. посібник Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2021. – 105 с
12. О.Д. Воробйова, Л.В. Глазунова, Алгоритми та структури даних: конспект лекцій. Частина 2. Алгоритми пошуку, стиснення даних, внутрішнього та зовнішнього сортування, алгоритми на графах, – Одеса:ОНАЗ ім. О.С. Попова, 2017. – 52 с.

13. Коротєєва Т.О. Алгоритми та структури даних: навч.посібник / Т.О.Коротєєва.- Львів: Видавництво Львівської політехніки, 2014. - 280 с.
14. M. Goodrich, R. Tamassia, M.Goldwasser., Data Structures and Algorithms in Python, – N.Y.: John Wiley & Sons, Inc., 2013. – 768 p.
15. I. Ahmad, 40 Algorithms Every Programmer Should Know, Packt Publishing, 2020
16. Kent D. Lee, Steve Hubbard, Data Structures and Algorithms with Python, Springer, 2015
17. Karumanchi N., Data Structures And Algorithmic Thinking With Python, CareerMonk Publications, 2016
18. Бедратюк Л.П., Бедратюк Г.І., Гурман І.В., Метод збільшення окремого зображення на основі багатовиходової регресії та моментної метрики, "Вимірювальна та обчислювальна техніка в технологічних процесах", 2023,№1, с.149-157

11. ІНФОРМАЦІЙНІ РЕСУРСИ

1. Модульне середовище для навчання. Доступ до ресурсу: <https://msn.khmnu.edu.ua>.
2. Електронна бібліотека університету. Доступ до ресурсу: <http://library.khmnu.edu.ua>
3. Репозитарій ХНУ:<https://elar.khmnu.edu.ua/home>
4. Молчановський О. Розробка та аналіз алгоритмів та структур даних. Дистанційний курс Prometheus URL:<https://courses.prometheus.org.ua/courses/KPI/Algorithms101/>

ALGORITHMS AND DATA STRUCTURES

Type of Discipline	Compulsory
Level of Higher Education	First (Bachelor's)
Language of Instruction	Ukrainian, English
Semester	4
ECTS Credits	7
Course study mode	Full-time (Daytime)

Learning outcomes. According to the Standard of higher education and the educational program, the discipline must provide:

competences: Ability to abstract thinking, analysis and synthesis. Ability to apply knowledge in practical situations. Ability to learn and master modern knowledge. Ability to search, process and analyze information from different sources. Knowledge of data information models, ability to create software for data storage, extraction and processing. Ability to apply fundamental and interdisciplinary knowledge to successfully solve software engineering tasks. Ability to algorithmic and logical thinking.

program learning outcomes : Analyzing, purposefully searching for and choosing the necessary information and reference resources and knowledge for solving professional tasks, taking into account the modern achievements of science and technology. Know and apply methods of developing algorithms, designing software and data and knowledge structures. Motivated to choose programming languages and development technologies to solve the tasks of creating and maintaining software.

Course content. Algorithms and their analysis. Sorting algorithms. Simple data structures. Hash tables. Related data structures. Recursion . Search algorithms. Algorithms on graphs. Computational geometry.

Planned academic activity. Lectures - 36 hours, laboratory classes - 36 hours, independent work - 138 hours, together - 210 hours.

Teaching forms (methods). lectures (using methods of problem-based learning and visualization); laboratory classes, independent work.

Assessment forms and methods: protection of laboratory works, thematic online testing; written final work.

Form of semester control: exam

Educational resources:

1. Томас Г. Кормен, Чарлз Е. Лейзерсон, Роналд Л. Рівест, Кліфорд Стайн. Вступ до алгоритмів. Переклад з англійської третього видання. Київ, К.І.С., 2019, 1288 стор.
2. Крєневич А.П. Алгоритми і структури даних. Підручник. – К.: ВПЦ "Київський Університет", 2021. – 200 с..
3. I. Ahmad, 40 Algorithms Every Programmer Should Know, Packt Publishing, 2020
4. Васильєв О., Програмування мовою Python. –Навчальна книга – Богдан 2019 – 504 стр.
5. MOODLE Learning Platform. Access to the resource <https://msn.khmnu.edu.ua>.
6. University Electronic Library. Access to the resource: <http://library.khmnu.edu.ua>.

Lecturer: DSc, Professor Leonid BEDRATYUK

3. EXPLANATORY NOTE

The discipline "Algorithms and data structures" is one of the disciplines of general training and occupies a leading place in the training of specialists of the "bachelor" educational level under the educational and professional program "Software Engineering".

The goal of the discipline . The discipline "Algorithms and Data Structures" aims to provide students with general and professional competencies in the field of algorithmization and implementation of basic algorithms and data structures.

Subject of discipline. Typical algorithms and data structures found in programming.

Tasks of the discipline . To provide students with knowledge and practical skills in designing optimal algorithms and selecting optimal data structures for solving typical algorithmic problems.

According to the Standard of higher education and the educational program, the discipline must ensure:

General competences :

GC1. Ability to abstract thinking, analysis and synthesis.

GC2. Ability to apply knowledge in practical situations.

GC5. Ability to learn and master modern knowledge.

GC6. Ability to search, process and analyze information from various sources.

Professional competences.

PC7. Knowledge of data information models, ability to create software for data storage, extraction and processing.

PC8. Ability to apply fundamental and interdisciplinary knowledge to successfully solve software engineering tasks

PC14. Ability to algorithmic and logical thinking .

Program learning outcomes

PLO1 Analyze, purposefully search for and select information and reference resources and knowledge necessary for solving professional tasks, taking into account modern achievements of science and technology.

PLO13 Know and apply methods of developing algorithms, designing software and data and knowledge structures.

PLO15 Motivated to choose programming languages and development technologies to solve the tasks of creating and maintaining software .

Learning outcomes . A student who has successfully completed the study of the discipline has *be able* to formalize computational and logical problems; *present* algorithms of computational and logical problems; *to investigate* the effectiveness of algorithms; *use* the main ones data structures (arrays, sets, stacks, trees); *use* methods of constructing complex data structures based on simple data structures, *search for* information in data structures using various search methods; *choose* the optimal data structure and the optimal algorithm for solving the problem.

Discipline Policy. The organization of the educational process for the discipline complies with the requirements of the provisions on organizational and instructional-methodological support of the educational process, the educational program, and the curriculum. Students are required to attend lectures, practical classes, laboratory work, etc., according to the schedule, not to be late for classes, and to complete all tasks and checkpoints according to the schedule. Missed practical classes and laboratory work must be independently completed by the student in full and reported to the instructor no later than one week before the next assessment. For practical classes and laboratory work, students must prepare on the relevant topic and demonstrate active participation. Knowledge acquired by an individual in the discipline or its specific sections through informal education is credited according to the Regulation on the procedure for transferring learning outcomes and determining academic differences at KhNU.

4 . COURSE CREDIT STRUCTURE

Topic name	The number of hours allocated to:		
	Lectures	lab.wors	Individual works
2 semester			
Topic 1. Algorithms and their analysis	4	4	13
Topic 2 . Sorting algorithms	6	6	13
Topic 3. Simple data structures	4	4	14
Topic 4. Hash tables	2	2	14
Topic 5. Related data structures	2	2	14
Topic 6. Recursion	2	2	14
Topic 7. Search algorithms	6	6	14
Topic 8. Algorithm of fast Fourier transformation	2	2	14
Topic 9. Algorithms on graphs	6	6	14
Topic 10. Computational geometry	4	6	14
Total semester	36	3 6	138

5 . COURSE PROGRAM

5.1. Content of lectures

Lectures numbers	List of topics of lectures, their annotations	Number hours
	<i>Second semester</i>	
1 -2	<p>Topic 1. Algorithms and their analysis</p> <p><i>Lecture 1. Algorithms and their properties.</i> Definition of the algorithm and its properties. Comparison of algorithms, time and space complexity. The size of the input data, the running time of the algorithm. Worst case, best case, and average uptime.</p> <p><i>Lecture 2. Analysis of algorithms.</i> Theoretical and experimental analysis of working time. Order of growth, asymptotic analysis of the order of growth of the running time of the algorithm. The designation O is Ωlarge . Difficulties $O(n)$, $O(n \log(n))$, $O(n^2)$.</p>	4
3 -5	<p>Topic 2. Sorting algorithms</p> <p><i>Lecture 3 . Sorting algorithms of quadratic complexity.</i> Sorting algorithms based on comparison and their analysis: Insertion sort, bubble sort, selection sort.</p> <p><i>Lecture 4.</i> Divide and conquer method. Sorting algorithms of lineo - logarithmic complexity. Merge sort, quick sort</p> <p><i>Lecture 5. Sorting algorithms of linear complexity .</i> Sorting algorithms without comparison. Bitwise sorting, counting sorting, cell sorting.</p>	6
6 -7	<p>Topic 3. Simple data structures</p>	4

	<p><i>Lecture 6. Stack, queue, dec . Abstract data structures, Stack and its implementation. Queue, deck and their implementations. Typical algorithms in which simple data structures are used.</i></p> <p><i>Lecture 7. Trees . Definition of a tree, types of trees and their implementation. Use of trees in algorithms : pyramidal sorting algorithm, priority queue.</i></p>	
8	<p>Topic 4. Hash tables</p> <p><i>Lecture 8. Hash tables. Hashing and collision resolution. Hash functions and hash codes. Open Addressing and Chaining. Load factor. Implementation of a hash table</i></p>	2
9	<p>Topic 5. Related data structures</p> <p><i>Lecture 9. Connected data structures. Linked structures: Single-linked structures and double-linked structures. Operations on single-connected structures: browsing, searching, replacement, insertion, deletion. Circular and double bond structures.</i></p>	2
10	<p>Topic 6. Recursion</p> <p><i>Lecture 10 . Recursion. Recursive functions and their properties . Recursive call tree. Application of recursion - search with return, problem about the towers of Hanoi, recursive sequences.</i></p>	2
10 -1 2	<p>Topic 7. Search algorithms</p> <p><i>Lecture 11. Search algorithms in arrays and trees. Linear, binary interpolation search . Binary search trees . Red and black search trees</i></p> <p><i>Lecture 1 2. Search algorithms in lines. Rabin -Karp algorithm . Maurice -Knuth- Pratt algorithm . Aho-Korasyk algorithm and an algorithm Boyer - Moore - Horspool</i></p>	4
13	<p>Topic 8. Algorithm of fast Fourier transformation</p> <p><i>Lecture 1 3. Multiplication of polynomials and the algorithm of fast Fourier transformation . Polynomial multiplication algorithms and their complexity. Discrete and fast Fourier transforms and their applications.</i></p>	2
1 4-15	<p>Topic 8. Algorithms on graphs.</p> <p><i>Lecture 14. Graph traversal algorithms. Presentation of graphs. Search in depth, search in breadth. Topological sorting .</i></p> <p><i>Lecture 15. Minimal spanning tree. Construction of a minimal frame tree. Algorithms of Kruskal , Prima ,</i></p> <p><i>Lecture 16. Shortest paths in graphs. Construction of the shortest way Algorithms of Dijkstra and Floyd, application .</i></p>	6
17-18	<p>Topic 10. Computational geometry</p> <p><i>Lecture 17. The simplest problems of computational geometry . Segment rotation, intersection of two segments. Belonging of a point to a shape . The sweeping line method . Search for the nearest point</i></p> <p><i>Lecture 18. Convex shell, triangulation , Voronoi diagrams. The convex hull of a set of points and its properties. Algorithms for constructing a convex hull - traversing according to Jarvis, scanning according to Graham. Triangulation , Voronoi diagrams.</i></p>	4
	Total for the second semester:	3 6

5.2 Content of laboratory classes

Number topics	The topic of the laboratory session	Number hours
1	Algorithms and their analysis	4
2	Installing Python , Jupyter environment notebook . Python data types, operators, and functions .	6
3	Built-in data structures : lists, sets, tuples.	4
4	Stacks, queues, decks	4
5	Hash tables.	2
6	Algorithms for searching substrings , module re .	4
7	Algorithms on graphs .	6
8	Computational geometry .	6
Total per semester		36

5.3 Content of independent work

Number of the week	Content of independent work	Number of hours
1	Processing of lecture material. . Preparation for laboratory work No. 1	9
2	Processing of lecture material. . Preparation for laboratory work No. 1	9
3	Processing of lecture material. Preparation for laboratory work No. 2	9
4	Processing of lecture material. . Preparation for laboratory work No. 2	9
5	Processing of lecture material. Preparation for laboratory work No. 3	9
6	Processing of lecture material. Preparation for laboratory work No. 3 Protection of laboratory work No. 3	9
7	Processing of lecture material. Preparation for laboratory work No. 4	9
8	Processing of lecture material. Preparation for laboratory work No. 4	9
9	Processing of lecture material. Preparation for laboratory work No. 5	9
10	Processing of lecture material. Preparation for laboratory work No. 5 .	9

11	Processing of lecture material. Preparation for laboratory work No. 6 . Protection of laboratory work No. 6	8
12	Processing of lecture material. Preparation for laboratory work No. 6	8
13	Processing of lecture material. Preparation for laboratory work No. 6 .	8
14	Processing of lecture material. Preparation for performance and defense of laboratory work No. 6	8
15	Processing of lecture material. Preparation for laboratory work No. 7	8
16	Processing of lecture material. Preparation for laboratory work No. 7	8
17	Processing of lecture material. Preparation for laboratory work No. 8 Protection of laboratory work No. 8	8
18	Processing of lecture material. Preparation for laboratory work No. 8	8
	Total per semester	138

6 . TEACHING METHODS

The teaching process in the discipline is based on the use of traditional and modern technologies, in particular: lectures (using methods of problem-based learning and visualization); laboratory (using methods of information technologies and modern integrated programming environments, master classes), independent work, and have the goal of acquiring general and professional competencies by students in the field of algorithmization and implementation of basic algorithms and data structures.

7. ASSESSMENT FORMS AND METHODS

Control over learning material is carried out on the basis of current control. When assessing students' knowledge, various means of control are implemented, in particular current control (processing of theoretical material), test tasks. Intermediate control consists in checking students' knowledge on topics. The grades received by the student during the current, intermediate control are not transferred . Each type of work is evaluated on a four-point scale. The final grade for the semester is issued taking into account all the grades received by the student for the semester and the results of the assessment. Weighting factors vary depending on the structure of the discipline.

Each type of work in the discipline is evaluated on *a four-point* scale. The semester final grade is defined as a weighted average of all types of academic work completed and passed *positively* , taking into account the weighting factors . A student who scored a positive weighted average score for the current work and did not pass the final test is considered to have failed.

When assessing students' knowledge, various means of control are carried out, in particular, current control is carried out during laboratory classes, taking into account different types of work. Independent and individual work is monitored separately.

The student must make up the missed lesson within the deadline set by the teacher. During the semester, the student must receive at least 4 grades in laboratory classes.

When evaluating students' knowledge, the teacher is guided by the following criteria.

The grade "excellent" is given to the student for a deep and complete mastery of the content of the educational material in which he can easily navigate, conceptual apparatus, for the ability to connect theory with practice, solve practical tasks, express and justify his judgments and constructive decisions. An excellent assessment implies a competent, logical presentation of the answer (both orally and in writing), high-quality external design.

The grade "good" is given to the student for complete assimilation of the educational material, mastery of the conceptual apparatus, orientation in the studied material, conscious use of knowledge to solve practical tasks, competent presentation of the answer, but there were some inaccuracies (errors) in the content and form of the answer, unclear formulations of regularities etc. The student's answer should be based on independent thinking.

"satisfactory" grade is awarded to a student who has demonstrated knowledge of the basic curriculum material to the extent necessary for further study and practical work in a profession that copes with the implementation of practical tasks provided for by the program.

"satisfactory" grade for incomplete mastery of the program material.

The grade "unsatisfactory" is assigned when the student has scattered, unsystematic knowledge, does not know how to distinguish the main and secondary, makes mistakes in the definition of concepts, distorts their meaning, presents the material chaotically and uncertainly, cannot use knowledge when solving practical tasks.

Based on the results of the current control, a final semester grade is issued.

Structuring the course by types of work and assessing learning outcomes for full-time students in the semester according to weighing coefficients

Auditory work			Semester control
Laboratory work	Test control		exam
No. 1 - 8	TC1	TC2	TC3
0.5	0.1		0.4

Correspondence of the national and ECTS grading scales

<i>ECTS grade</i>	<i>Institutional score scale</i>	<i>Institutional grade</i>	<i>Assessment criteria</i>
A	4,75-5,00	5	Passed
B	4,25-4,74	4	
C	3,75-4,24	4	
D	3,25-3,74	3	
E	3,00-3,24	3	
FX	2,00-2,99	2	Failed
F	0,00-1,99	2	

The exam is given when the student receives from 3.00 to 5.00 points in the discipline. At the same time, according to the national scale, "credited" is given, and according to the ECTS scale, the grade corresponding to the number of points scored by the student.

8. QUESTIONS FOR STUDENTS' SELF-CONTROL

1. Concept of algorithm, properties of algorithms
2. Sorting problem, insertion sort, analysis of insertion sort algorithm.
3. The size of the input data, the running time of the algorithm.
4. The running time of the algorithm in the best, average and worst cases.
5. Merge sorting algorithm Sort and its analysis.
6. The order of growth of the function. Asymptotic notations.
7. Karatsuba multiplication algorithm and its analysis.
8. Pyramid data structure and its properties.
9. Pyramid data structure. The algorithm for creating a pyramid and the algorithm for maintaining pyramid properties.
10. HeapSort pyramidal sorting algorithm and its analysis.
11. Dynamic sets and operations in dynamic sets.
12. Stack data structure and its properties.
13. The list data structure and its properties.
14. Queue data structure and its properties.
15. priority queue data structure and its properties.
16. Binary search tree data structure and its properties.
17. The data structure is a binary tree. Node insertion and deletion algorithms.
18. Binary tree-based sorting algorithm and its analysis.
19. Hash table data structure and its properties.
20. Hash functions.
21. Quick sorting algorithm Sort and its analysis.
22. Counting sorting algorithm Sort and its analysis.
23. Radix bitwise sorting algorithm Sort and its analysis.
24. Bucket sorting algorithm _ Sort and its analysis.
25. Naive substring search algorithm and its analysis.
26. Substring search algorithm Rabina -Karp and his analysis.
27. Knuth-Maurice- Pratt substring search algorithm and its analysis.
28. Linked list data structure and its properties
29. Algorithms of Prim, Boruvka , Kraskala
30. Search in breadth and depth
31. Floyd's algorithm

32. Recursion and their properties
33. Circular and doubly linked lists.
34. Array search algorithms
35. Binary and interpolation search
36. The simplest problems of computational geometry
37. Graph traversal algorithms. Search in depth.
38. Graph traversal algorithms. wide search
39. Algorithm of topological sorting of graphs.
40. Minimum spanning tree. Algorithms of Kruskal , Prima
41. Minimum spanning tree. Prim's algorithms
42. Shortest paths in graphs. Dijkstra's algorithms
43. Shortest paths in graphs. Floyd's algorithms.
44. The simplest problems of computational geometry. Segment rotation, intersection of two segments.
45. The simplest problems of computational geometry. Belonging of a point to a figure
46. Algorithms for constructing a convex hull - Jarvis bypass
47. Algorithms for constructing a convex hull - scanning according to Graham.
48. Triangulation ,
49. Voronoi diagrams.

9. TEACHING AND LEARNING MATERIALS

The educational process in the discipline "Algorithms and data structures" is fully and in sufficient quantity provided with the necessary educational and methodological literature.

Algorithms and Data Structures: Guidelines for performing laboratory works for students of the first (bachelor's) level of higher education, specialty 121 "Software Engineering"/ L.P. Bedratyuk, A.I. Bedratyuk – Khmelnytskyi: KhNU, 2023 – 114 p.

11. RECOMMENDED BOOKS

Main

1. Thomas G. Cormen , Charles E. Leiserson , Ronald L. Rivest , Clifford Stein. Introduction to algorithms . Translation from English the third publication _ Kyiv , K.I.S., 2019, 1288 pages .
2. Krenevich A.P. Algorithms and data structures. Textbook. – K.: Kyiv University, 2021. – 200 p.
3. I. Ahmad , 40 Algorithms Every Programmer Should Know , Packt Publishing , 2020
4. Vasiliev O., Programming in the Python language . – Educational book – Bohdan 2019 – 504 pages .
5. Shakhovska N. B., Holoshchuk R. O. Algorithms and data structures. Tutorial. Lviv: Magnolia, 2018. 216 p

Additional

6. Krenevich A.P. Python in examples and problems. Part 1. Structured programming, Study guide, K.: VOC "Kyiv University", 2017. – 206 p.
7. Koval V.S., Strubyskyi P.R. Algorithms and data structures. – Study guide – Ternopil: FOP Shpak V. B. – 2017. – 74 p.
8. Tkachuk , Algorithms and data structure: Study guide , - IvanoFrankivsk : PNU Publishing House, 2016.-286 p.
9. THEN. Hryshanovych , Course of lectures on the discipline "Algorithms and data structures",

Lesya Ukrainka University, Lutsk, 2021. - 110 p.

10. V.M. Ilman , O.P. Ivanov, L.O. Panik ., Algorithms, data and structures. teach _ help _ Dnipropetrovsk . National University of Iron . transp.im _ Acad. V. Lazaryan. - Dnipro, 2019. - 134 p.
11. O. O. Petrova, G. V. Solodovnyk, Algorithmic problems and their solution: academic . manual Kharkiv. national _ city university _ farm named after O. M. Beketova. – Kharkiv: XNUMX named after O. M. Beketova, 2021. - 105 p
12. O.D. Vorobyova, L.V. Glazunova, Algorithms and data structures: lecture notes. Part 2. Search algorithms, data compression, internal and external sorting, algorithms on graphs, – Odesa: ONAZ named after O.S. Popova, 2017. – 52 p.
13. Koroteeva T.O. Algorithms and data structures: study guide / T.O. Koroteeva .- Lviv: Publishing House of Lviv Polytechnic, 2014. - 280 p.
14. M. Goodrich , R. Tamassia , M. Goldwasser , Data Structures and Algorithms in Python , – NY: John Wiley & Sons , Inc. , 2013. - 768 p.
15. I. Ahmad, 40 Algorithms Every Programmer Should Know, Packt Publishing, 2020
16. Kent D. Lee, Steve Hubbard, Data Structures and Algorithms with Python, Springer, 2015
17. Karumanchi N. , Data Structures And Algorithmic Thinking With Python, CareerMonk Publications, 2016
18. Bedratyuk L.P., Bedratyuk G.I., Gurman I.V., Method of increasing individual image based _ multi-output regression and moment metrics, " Measurement and calculation technique in technological processes ", 2023 , No. 1, pp. 149-157

12 . INFORMATION RESOURCES

1. MOODLE Learning Platform. Access to the resource <https://msn.khmnu.edu.ua>.
2. University Electronic Library. Access to the resource: <http://library.khmnu.edu.ua>.
3. University Repository. Access to the resource <https://elar.khmnu.edu.ua/home>.
4. Molchanovsky O. Development and analysis algorithms and data structures . Distance course Prometheus URL: [https://courses.prometheus.org.ua/courses /KPI/Algorithms101/](https://courses.prometheus.org.ua/courses/KPI/Algorithms101/)