

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЦЕНТРАЛЬНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Кафедра кібербезпеки та програмного забезпечення

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
ТЕХНОЛОГІЇ РОЗРОБКИ АЛГОРИТМІВ

Розглянуто на засіданні кафедри
Протокол № 9 від 9 лютого 2023 року

м. Кропивницький – 2023

ЗМІСТ

1. Загальна інформація
2. Анотація до дисципліни
3. Мета і завдання дисципліни
4. Формат дисципліни
5. Результати навчання
6. Обсяг дисципліни
7. Ознаки дисципліни
8. Пререквізити
9. Технічне і програмне забезпечення / обладнання
10. Політика курсу
11. Навчально-методична карта дисципліни
12. Система оцінювання та вимоги
13. Рекомендована література

1. Загальна інформація

Назва курсу	ТЕХНОЛОГІЇ РОЗРОБКИ АЛГОРИТМІВ
Викладачі та їх профайли	<i>Лектор</i> – Буравченко Костянтин Олегович, кандидат технічних наук, старший викладач кафедри кібербезпеки та програмного забезпечення http://kbpz.kntu.kr.ua/byravchenko-konstantin/ https://scholar.google.com.ua/citations?user=nvjzvagAAAAJ <i>Асистент</i> – Гермак Вікторія Сергіївна, викладач кафедри кібербезпеки та програмного забезпечення http://kbpz.kntu.kr.ua/germak-viktoriy/ https://scholar.google.com/citations?user=2IR18z4AAAAJ&hl=uk https://orcid.org/0000-0001-8473-4156 https://publons.com/researcher/1760991/victoriya-germak/
Контактний телефон	служб. (0522) 390-449– робочі дні з 8 ³⁰ до 14 ²⁰
Е-пошта	germak_vs@ukr.net
Консультації	<i>Очні</i> – відповідно до затвердженого графіку консультацій; <i>Онлайн</i> – е-листування, вебінари на платформі Zoom ⁰

2. Анотація до дисципліни

Курс «Технології розробки алгоритмів» призначений для набуття теоретичних знань стосовно базових алгоритмів розв’язання класичних задач програмування та методів їх побудови, а також практичних навичок реалізації даних алгоритмів шляхом написання ПЗ використовуючи сучасні мови програмування.

Даний курс присвячено розгляду теоретичних питань щодо математичного підґрунтя побудови комп’ютерних алгоритмів та актуальних питань застосування оптимальних алгоритмів для вирішення прикладних задач. Також даний курс призначений для формування практичних навичок використання апарату теорії алгоритмів для побудови ефективних алгоритмів та обрання оптимального алгоритму для розв’язання поставленої задачі в залежності від вимог, які висувуються до програмного забезпечення яке розробляється.

В сучасних умовах, зважаючи на різноманітність задач, які постають перед розробниками програмного забезпечення, вміння обирати оптимальний алгоритм та ефективні шляхи для його розробки є ключовим при розв’язанні широкого кола задач практичної діяльності.

3. Мета і завдання дисципліни

Метою викладання дисципліни “Технології розробки алгоритмів” є формування у здобувачів вищої освіти теоретичних знань

та практичних навичок реалізації щодо основних методів теорії алгоритмів; ґрунтовне опанування здобувачами основних технологій розробки алгоритмів; розвиток навичок дослідження математично формалізованих задач та опанування основними методами їх розв'язання. Також формування у здобувачів вищої освіти ґрунтовних уявлень про застосування для розв'язання практичних задач основних методів теорії алгоритмів; розвиток навичок творчого дослідження поставлених практичних задач з метою формування вмінь і навичок обирати ефективний шлях їх розв'язання в залежності від обставин та вимог до реалізації. Опанування здобувачами вищої освіти теорії та практики розв'язання типових задач дозволить на сучасному рівні застосовувати їх в інженерних задачах.

Завданнями дисципліни є набуття здобувачами таких програмних компетентностей:

1. Здатність використовувати сучасні методи (включаючи основні технології та методи розробки алгоритмів, такі як метод динамічного програмування, метод гілок і границь, евристичні методи, методи клітинних автоматів, користуючись висхідною і низхідною стратегіями побудови алгоритмів) і мови програмування (такі як C, C++, Python, FPC) для розроблення алгоритмічного та програмного забезпечення у вільних кросплатформових середовищах розроблення ПЗ Code::Blocks (<http://www.codeblocks.org/>), Lazarus (<https://www.lazarus-ide.org/>), PyCharm Community Edition (<https://www.jetbrains.com/pycharm/>) (на вибір здобувача).

2. Здатність аргументувати вибір методів розв'язування спеціалізованих задач, критично оцінювати отримані результати, обґрунтовувати та захищати прийняті рішення, з розумінням оперуючи поняттями часової, ємнісної, логічної складності алгоритму, складності задачі, а також володіючи методикою обчислення складності задачі та складності алгоритму з подальшим використанням отриманих результатів для вибору оптимального алгоритму розв'язання практичної задачі.

Набуття здобувачами вищої освіти вищевказаних компетентностей досягається шляхом послідовного засвоєння теоретичного матеріалу, виконання лабораторних робіт та ґрунтового опрацювання матеріалу, винесеного на самостійне вивчення.

4. Формат дисципліни

Для денної форми навчання:

Викладання курсу передбачає для засвоєння дисципліни традиційні лекційні заняття із застосуванням мультимедійних презентацій, у поєднанні з лабораторними заняттями.

Формат очний (offline / Face to face)

Для заочної форми навчання:

Під час сесії формат очний (offline / Face to face), у міжсесійний період – дистанційний (online).

5. Результати навчання

У результаті вивчення навчальної дисципліни “Технології розробки алгоритмів” студент буде:

знати

- наукові положення, що лежать в основі функціонування комп'ютерних засобів, серед яких поняття алгоритму, властивості алгоритму, формалізовані способи представлення алгоритму, складові складності алгоритму, складність задачі, класи складності, методи та технології розробки алгоритмів, типи даних, структури даних, алгоритми сортування, алгоритми пошуку, стратегії та методи оптимізації програмного коду;

- новітні технології в галузі комп'ютерної інженерії, мови програмування Python, Free Pascal;

вміти

- застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей, враховуючи особливості практичної задачі (вимоги до швидкодії результуючого ПЗ, точності отримуваних результатів, ресурсоемності тощо), виконувати розв'язання поставлених задач, застосовуючи методи побудови ефективних алгоритмів, знаходити та досліджувати оптимальні алгоритми для розв'язання конкретної практичної задачі;
- розв'язувати задачі аналізу та синтезу засобів, характерних для спеціальності, зокрема здійснювати аналіз часової складності обраного алгоритму та перевірку його на відповідність вимогам практичної задачі (забезпечення необхідної швидкодії та точності); розробляти алгоритми розв'язання практичних задач використовуючи класичні методи та технології розробки алгоритмів (методи динамічного програмування, перебірні, методи гілок та границь та інші); реалізовувати розроблені алгоритми на вибір здобувача мовою програмування Free Pascal у кросплатформовому середовищі Lazarus IDE, мовою програмування C або C++ у кросплатформовому середовищі Code::Blocks, мовою програмування Python у кросплатформовому середовищі PyCharm Community Edition;
- системно мислити та застосовувати творчі здібності до формування нових ідей під час аналізу практичної задачі, обрання оптимального алгоритму для її розв'язання в рамках вимог до точності отримуваних результатів, синтезу алгоритмів та реалізації програмних засобів;
- поєднувати теорію і практику, а також приймати рішення та виробляти стратегію діяльності для вирішення завдань спеціальності з урахуванням виробничих інтересів використовуючи вміння оцінювати розроблені алгоритми та тестувати результуюче програмне забезпечення з метою вибору оптимальної стратегії розв'язання практичної задачі, а також здатність розуміти наукові основи і практичні методи алгоритмізації обчислювальних процесів, методи оцінки складності алгоритмів та методи оптимізації алгоритмів;
- оцінювати отримані результати та аргументовано захищати прийняті рішення шляхом перевірки обраного для реалізації практичної задачі алгоритму на відповідність вимогам до результуючого ПЗ та тестування ПЗ з використанням розроблених тест-кейсів;

володіти соціальними навичками (Soft Skills):

- усвідомлення необхідності навчання впродовж усього життя з метою поглиблення набутих та здобуття нових фахових знань, удосконалення креативного мислення.

Набуття здобувачами вищої освіти вищевказаних навичок та програмних результатів вивчення дисципліни досягається шляхом послідовного засвоєння теоретичного матеріалу, виконання лабораторних робіт та ґрунтовного опрацювання матеріалу, винесеного на самостійне вивчення.

6.Обсяг дисципліни

Вид роботи	Кількість годин
Лекції	28

Лабораторні заняття	14
Самостійна робота	48
Екзамен	30
Разом	120

7. Ознаки дисципліни

Рік викладання	Курс (рік навчання)	Семестр	Кількість кредитів / годин	Кількість змістових модулів	Вид підсумкового контролю	Нормативна / вибіркова
2022	3	5	4/120	2	екзамен	вибіркова

8. Пререквізити

Ефективність засвоєння змісту дисципліни «Технології розробки алгоритмів» значно підвищиться, якщо здобувач попередньо опанував матеріал таких дисциплін як: «Структурне програмування», «Модульне програмування», «Програмування».

9. Технічне й програмне забезпечення / обладнання

Для викладання дисципліни «Технології розробки алгоритмів» застосовується потужна матеріально-технічна база кафедри кібербезпеки та програмного забезпечення: мультимедійний проектор Epson EB-X41, спеціалізовані комп'ютерні лабораторії з персональними комп'ютерами Athlon II 215x2 (10 шт.), Athlon 2.4, (15 шт.), AMD Sempron LE-1150 (18 шт.), AMD Duron 1,2 GHz (15 шт.), програмне забезпечення вільного кросплатформового середовища розроблення ПЗ Lazarus версії 2.0.8 / FPC 3.0.4 / (ліцензія GNU GPL), вільне кросплатформове середовище розроблення програм Code::Blocks версії 20.03 (ліцензія GNU GPLv3), вільне кросплатформове середовище розроблення програм PyCharm Community Edition версії: 2020.1 (ліцензія Apache License), відкрита бездротова мережа Wi-Fi, вільний доступ до Інтернету.

10. Політика дисципліни

Академічна доброчесність:

Очікується, що студенти будуть дотримуватися принципів академічної доброчесності, усвідомлювати наслідки її

порушення. Детальніше за посиланням URL : <http://www.kntu.kr.ua/doc/dobro.pdf>

Відвідування занять

Відвідування занять є важливою складовою навчання. Очікується, що всі студенти відвідають лекції і лабораторні заняття курсу. Пропущені заняття повинні бути відпрацьовані не пізніше, ніж за тиждень до залікової сесії.

Поведінка на заняттях

Недопустимість: запізнь на заняття, списування та плагіат, несвоєчасне виконання поставленого завдання.

При організації освітнього процесу в Центральньоукраїнському національному технічному університеті студенти, викладачі та адміністрація діють відповідно до: Положення про організацію освітнього процесу; Положення про організацію вивчення навчальних дисциплін вільного вибору; Положення про рубіжний контроль успішності і сесійну атестацію студентів ЦНТУ; Кодексу академічної доброчесності ЦНТУ.

11. Навчально-методична карта дисципліни

Тиждень, дата, академічні години	Тема, основні питання	Форма діяльності (заняття) /формат	Матеріали	Література, інформаційні ресурси	Завдання, години	Вага оцінки	Термін виконання
Змістовий модуль 1. Основи теорії алгоритмів.							
Тиж. 1, 2 (за розкладом) 4 акад. год.	Тема 1. Основи алгоритмізації. Історія виникнення та визначення алгоритму. Властивості, способи представлення та види алгоритмів. Складові складності алгоритму. Методика обчислення складності алгоритму.	Лекція / <i>Face to face</i>	Презентація	1, 2,10, 11, 17	Самостійно опрацювати матеріал: Сучасні напрямки розвитку теорії алгоритмів. 3 год.	5 балів	Самостійна робота до 2 тижня
Тиж. 1, 2 (за розкладом) 2 акад. год.	Тема 1. Основи алгоритмізації. Історія виникнення та визначення алгоритму. Властивості та способи представлення алгоритмів. Складові складності	Лабораторна робота / <i>Face to face</i>	Методичні рекомендації	16, 17, 18, 19	Сформулювати основні вимоги до алгоритмів, які можуть використовуватись при програмуванні систем реального часу. 3 год.	5 балів	Самостійна робота до 2 тижня

	алгоритму. Методика обчислення складності алгоритму. Обрання ефективного алгоритму в залежності від поставленої задачі.						
Тиж. 3, 4 (за розкладом) 4 акад. год.	Тема 2. Технології розробки алгоритмів. Види алгоритмів. Наближені алгоритми та методи їх побудови. Класичні наближені алгоритми розв'язання NP повних задач. Рекурсивні алгоритми.	Лекція / <i>Face to face</i>	Презентація	1, 2, 3, 4, 9, 10,17	Самостійно опрацювати матеріал: Сфера застосування жадібних алгоритмів. 3 год.	5 балів	Самостійна робота до 4 тижня
Тиж. 3, 4 (за розкладом) 2 акад. год.	Тема 2. Технології розробки алгоритмів. Види алгоритмів. Наближені алгоритми та методи їх побудови. Класичні наближені алгоритми розв'язання NP повних задач. Рекурсивні алгоритми.	Лабораторна робота / <i>Face to face</i>	Методичні рекомендації	16, 17, 18, 19	Проаналізувати алгоритми, які можуть бути використані для розв'язання задачі комівояжера. 3 год.	5 балів	Самостійна робота до 4 тижня
Тиж. 5, 6 (за розкладом) 4 акад. год.	Тема 3. Типи даних. Структури даних. Статичні та динамічні типи даних. Графи та дерева як структури даних. Бінарні дерева пошуку. Збалансовані дерева. Операції над елементами бінарного дерева.	Лекція / <i>Face to face</i>	Презентація	5, 9, 10, 12, 13, 14, 15,17	Самостійно опрацювати матеріал: Червоно-чорні дерева. 3 год.	5 балів	Самостійна робота до 6 тижня
Тиж. 5, 6	Тема 3. Типи даних.	Лабораторна	Методичні	16, 17, 18, 19	Розглянути практичне	5 балів	Самостійна

(за розкладом) 2 акад. год.	Структури даних. Статичні та динамічні типи даних. Графи та дерева як структури даних. Бінарні дерева пошуку. Збалансовані дерева. Операції над елементами бінарного дерева.	робота / <i>Face to face</i>	рекомендації		застосування збалансованих дерев. 3 год.		робота до 6 тижня
Максимальна кількість балів за змістовим модулем I 30 балів							
Змістовий модуль 2. Технології розробки алгоритмів.							
Тиж. 7,8 (за розкладом) 4 акад. год.	Тема 4. Сортування даних. Алгоритми внутрішнього сортування. Властивості алгоритмів сортування. Показники ефективності алгоритмів сортування. Внутрішнє сортування. Сортування бульбашкою та його модифікації. Швидке сортування. Пірамідальне сортування. Сортування за допомогою бінарного дерева.	Лекція / <i>Face to face</i>	Презентація	3, 4, 9,17	Самостійно опрацювати матеріал: Методи вдосконалення алгоритму Хоара. 4 год.	3 бали	Самостійна робота до 8 тижня
Тиж. 7,8 (за розкладом) 2 акад. год.	Тема 4. Сортування даних. Алгоритми внутрішнього сортування. Властивості алгоритмів сортування. Показники ефективності алгоритмів сортування. Внутрішнє сортування. Сортування	Лабораторна робота / <i>Face to face</i>	Методичні рекомендації	16, 17, 18, 19	Порівняти ефективність різних алгоритмів внутрішнього сортування в залежності від обсягу даних, які сортуються. 4 год.	5 балів	Самостійна робота до 8 тижня

	бульбашкою та його модифікації. Швидке сортування. Пірамідальне сортування. Сортування за допомогою бінарного дерева.						
Тиж. 9, 10 (за розкладом) 4 акад. год.	Тема 5. Алгоритми зовнішнього сортування. Зовнішнє сортування. Кишенькове сортування. Сортування злиттям. Пряме злиття (алгоритм Боуза-Нельсона). Природне (Нейманівське) злиття.	Лекція / <i>Face to face</i>	Презентація	3, 4, 7, 9,17	Самостійно опрацювати матеріал: Багатофазне сортування злиттям. 4 год.	3 бали	Самостійна робота до 10 тижня
Тиж. 9, 10 (за розкладом) 2 акад. год.	Тема 5. Алгоритми зовнішнього сортування. Зовнішнє сортування. Кишенькове сортування. Сортування злиттям. Пряме злиття (алгоритм Боуза-Нельсона). Природне (Нейманівське) злиття.	Лабораторна робота / <i>Face to face</i>	Методичні рекомендації	16, 17, 18, 19	Розглянути практичне застосування методів зовнішнього сортування. Порівняти ефективність та сфери застосування різних методів. 4 год.	5 балів	Самостійна робота до 10 тижня
Тиж. 11, 12 (за розкладом) 4 акад. год.	Тема 6. Пошук даних. Алгоритми пошуку. Загальна класифікація алгоритмів пошуку. Лінійний та бінарний пошук елемента в масиві. Алгоритми пошуку підрядка у тексті. Алгоритм Рабіна-Карпа. Алгоритм Кнута-Моріса-Пратта. Алгоритм Боуера-	Лекція / <i>Face to face</i>	Презентація	3, 6, 9,17	Самостійно опрацювати матеріал:Алгоритм Бойера - Мура - Хорспула. 4 год.	2 бали	Самостійна робота до 12 тижня

	Мура. Хеш-функції. Використання хешування в алгоритмах пошуку даних.						
Тиж. 11, 12 (за розкладом) 2 акад. год.	Тема 6. Пошук даних. Алгоритми пошуку. Загальна класифікація алгоритмів пошуку. Лінійний та бінарний пошук елемента в масиві. Алгоритми пошуку підрядка у тексті. Алгоритм Рабіна-Карпа. Алгоритм Кнута-Моріса-Пратта. Алгоритм Боуєра-Мура. Хеш-функції. Використання хешування в алгоритмах пошуку даних.	Лабораторна робота / <i>Face to face</i>	Методичні рекомендації	16, 17, 18, 19	Порівняти ефективність алгоритмів пошуку в залежності від сфери застосування. 4 год.	5 балів	Самостійна робота до 12 тижня
Тиж. 13, 14 (за розкладом) 4 акад. год.	Тема 7. Стратегії та методи оптимізації програмного коду. Показники якості програмного продукту. Відладка та тестування програмного продукту. Види тестування. Правила складання тест-кейсів. Комплексна відладка програмного засобу.	Лекція / <i>Face to face</i>	Презентація	6,7, 8, 9,17	Самостійно опрацювати матеріал: Тестування методом сендвіча. 3 год.	2 бали	Самостійна робота до 14 тижня
Тиж. 13, 14 (за розкладом) 2 акад. год.	Тема 7. Стратегії та методи оптимізації програмного коду. Показники якості програмного продукту.	Лабораторна робота / <i>Face to face</i>	Методичні рекомендації	16,17, 18, 19	Порівняти ефективність тестування за методами білої та чорної скриньки. 3 год.	5 балів	Самостійна робота до 14 тижня

Відладка та тестування програмного продукту. Види тестування. Правила складання тест-кейсів. Комплексна відладка програмного засобу.							
Максимальна кількість балів за змістовим модулем II 30 балів							
Максимальна кількість балів за екзамен 40 балів							

12 Система оцінювання та вимоги

Види контролю: поточний, підсумковий.

Методи контролю: спостереження за навчальною діяльністю здобувачів, усне опитування, письмовий контроль, тестовий контроль.

Форма підсумкового контролю: екзамен.

Контроль знань і умінь здобувачів (поточний і підсумковий) з дисципліни «Технології розробки алгоритмів» здійснюється згідно з кредитною трансферно-накопичувальною системою організації навчального процесу. Рейтинг студента із засвоєння дисципліни визначається за 100-бальною шкалою. Він складається з рейтингу навчальної роботи (засвоєння теоретичного матеріалу під час аудиторних занять та самостійної роботи, виконання лабораторних робіт), для оцінювання якої призначається 60 балів, та екзамену, максимальна оцінка за який складає 40 балів.

Розподіл балів, які отримують студенти при вивченні дисципліни «Технології розробки алгоритмів»

Поточний контроль та самостійна робота														Екзамен	Сума
Змістовий модуль 1						Змістовий модуль 2									
T1		T2		T3		T4		T5		T6		T7			
Л	ЛР	Л	ЛР	Л	ЛР	Л	ЛР	Л	ЛР	Л	ЛР	Л	ЛР	40	100
5	5	5	5	5	5	3	5	3	5	2	5	2	5		

Примітка: T1, T2, ..., T11 – тема, Л – теоретичні (лекційні) заняття, ЛР – лабораторні роботи.

Шкала оцінювання: національна та ЄКТС

Сума балів за всі		Оцінка за національною шкалою
-------------------	--	-------------------------------

види навчальної діяльності	Оцінка ЄКТС	для екзамену, курсового проекту (роботи), практики
90-100	A	відмінно
82-89	B	добре
74-81	C	
64-73	D	задовільно
60-63	E	
35-59	FX	незадовільно з можливістю повторного складання
1-34	F	незадовільно з обов'язковим повторним вивченням дисципліни

Критерії оцінювання. Еквівалент оцінки в балах для кожної окремої теми може бути різний, загальну суму балів за тему визначено в навчально-методичній карті. Розподіл балів між видами занять (лекції, лабораторні заняття, самостійна робота) можливий шляхом спільного прийняття рішення викладача і студентів на першому занятті:

оцінку «**відмінно**» (**90-100 балів, A**) заслуговує студент, який:

- всебічно, систематично і глибоко володіє навчально-програмовим матеріалом;
- вміє самостійно виконувати завдання, передбачені програмою, використовує набуті знання і вміння у нестандартних ситуаціях;
- засвоїв основну і ознайомлений з додатковою літературою, яка рекомендована програмою;
- засвоїв взаємозв'язок основних понять дисципліни та усвідомлює їх значення для професії, яку він набуває;
- вільно висловлює власні думки, самостійно оцінює різноманітні життєві явища і факти, виявляючи особистісну позицію;
- самостійно визначає окремі цілі власної навчальної діяльності, виявив творчі здібності і використовує їх при вивченні навчально- програмового матеріалу, проявив нахил до наукової роботи.

оцінку «**добре**» (**82-89 балів, B**) – заслуговує студент, який:

- повністю опанував і вільно (самостійно) володіє навчально-програмовим матеріалом, в тому числі застосовує його на практиці, має системні знання достатньому обсязі відповідно до навчально-програмового матеріалу, аргументовано використовує їх у різних ситуаціях;
- має здатність до самостійного пошуку інформації, а також до аналізу, постановки і розв'язування проблем професійного спрямування;
- під час відповіді допустив деякі неточності, які самостійно виправляє, добирає переконливі аргументи на підтвердження вивченого матеріалу;

оцінку «**добре**» (**74-81 бал, C**) - заслуговує студент, який:

- в загальному роботу виконав, але відповідає на екзамені з певною кількістю помилок;

- вміє порівнювати, узагальнювати, систематизувати інформацію під керівництвом викладача, в цілому самостійно застосовувати на практиці, контролювати власну діяльність;
- опанував навчально-програмовий матеріал, успішно виконав завдання, передбачені програмою, засвоїв основну літературу, яка рекомендована програмою;
- оцінку **«задовільно» (64-73 бали, D)** – заслуговує студент, який:
 - знає основний навчально-програмовий матеріал в обсязі, необхідному для подальшого навчання і використання його у майбутній професії;
 - виконує завдання, але при рішенні допускає значну кількість помилок;
 - ознайомлений з основною літературою, яка рекомендована програмою;
 - допускає на заняттях чи екзамені помилки при виконанні завдань, але під керівництвом викладача знаходить шляхи їх усунення.
- оцінку **«задовільно» (60-63 бали, E)** – заслуговує студент, який:
 - володіє основним навчально-програмовим матеріалом в обсязі, необхідному для подальшого навчання і використання його у майбутній професії, а виконання завдань задовольняє мінімальні критерії. Знання мають репродуктивний характер.
- оцінка **«незадовільно» (35-59 балів, FX)** – виставляється студенту, який:
 - виявив суттєві прогалини в знаннях основного програмового матеріалу, допустив принципові помилки у виконанні передбачених програмою завдань.
- оцінка **«незадовільно» (35 балів, F)** – виставляється студенту, який:
 - володіє навчальним матеріалом тільки на рівні елементарного розпізнавання і відтворення окремих фактів або не володіє зовсім;
 - допускає грубі помилки при виконанні завдань, передбачених програмою;
 - не може продовжувати навчання і не готовий до професійної діяльності після закінчення університету без повторного вивчення даної дисципліни.

13. Рекомендована література

Базова

1. Кормен Томас Г. Вступ до алгоритмів: Переклад з англійської третього видання : [укр.] = Introduction to algorithms : Third Edition : [пер. з англ.] / Томас Г. Кормен, Чарлз Е. Лейзерсон, Роналд Л. Рівест, Кліффорд Стайн, — К: К.І.С., 2019. — 1288 с.
2. Пришвидшений курс PYTHON. Практичний, проектно-орієнтований вступ до програмування [Текст] / Е. Маттес ; Пер. з англ. О. Белова: Львів : Вид-во Старого Лева, 2021, 600 с.
3. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Introduction to Algorithms. The MIT Press. 2022 1677 с.
4. Marcello La Rocca. Advanced Algorithms and Data Structures. Manning Publications Co. 2021. 769 с.
5. John Paul Mueller, Luca Massaron. Algorithms for Dummies. John Wiley & Sons, Inc. 2022. 451 с.
6. Tom Jenkyns, Ben Stephenson. Fundamentals of Discrete Math for Computer Science. Springer. 2018. 514 с.
7. Андруник В.А., Висоцька В.А., Пасічник В.В., Чирун Л.Б., Чирун Л.В. Чисельні методи в комп'ютерних науках: навчальний посібник – Львів: Видавництво «Новий світ – 2000», 2020. – 470 с.

Допоміжна

8. Jason Turner. C++ Best Practices. ISBN 979-8690792589. Independently published. 2022. 143 с.
9. Dzejla Medjedovic, Emin Tahirovic. Algorithms and Data Structures for Massive Datasets. Manning Publications Co. 2022. 280 с.
10. Steven C. Chapra, David E. Clough. Applied Numerical Methods with Python for Engineers and Scientists. McGraw Hill LLC. 2022. 1474 с.
11. Massimo Bertaccini. Cryptography Algorithms. Packt Publishing. 2022. 358 с.
12. Гермак В.С., Петренюк В.І., Про структуру 8-ми та 9-ти вершинних графів-обструкцій для тора, матеріали П'ятнадцятого Міжнародного науково – практичного семінару "Комбінаторні конфігурації та їх застосування", Кіровоград:ПП"Ексклюзив-Систем", 2013, с. 79-86.
13. Гермак В.С., Петренюк В.І., Доповнення до статті “Властивості графів мінімального базису проєктивної площини”, матеріали Шістнадцятого Міжнародного науково – практичного семінару "Комбінаторні конфігурації та їх застосування", Кіровоград:ПП"Ексклюзив-Систем", 2014, с. 111-114.
14. Гермак В.С., Петренюк В.І., Про двосторонній доступ та рід 9-ти вершинних графів, матеріали Сімнадцятого Міжнародного науково – практичного семінару "Комбінаторні конфігурації та їх застосування", Кіровоград: 2015, с. 27-32.
15. Гермак В.С., Петренюк В.І., Структура непроективних графів - обструкцій, матеріали Вісімнадцятого Міжнародного науково – практичного семінару "Комбінаторні конфігурації та їх застосування", Кіровоград: 2016, с. 103-127.

Методична

16. Методичні вказівки до виконання лабораторних робіт з навчальної дисципліни “Технології розробки алгоритмів” для студентів денної та заочної форми навчання спеціальностей 6.050102 та 123 “Комп’ютерна інженерія”, 6.170103 та 125 “Кібербезпека” / уклад. Гермак В.С.; Кропивницький: ЦНТУ – 2018.– 60 с. <http://dspace.kntu.kr.ua/jspui/handle/123456789/8425>

Інформаційні ресурси

17. Технології розробки алгоритмів: система дистанційної освіти ЦНТУ
18. Lazarus wiki [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://wiki.freepascal.org/>.
19. Wiki for Code::Blocks [Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу: http://wiki.codeblocks.org/index.php/Main_Page.