Чернівецький національний університет імені Юрія Федьковича
(повне найменування закладу вищої освіти)

Факультет математики та інформатики
(назва факультету/навчально-наукового інституту)

Кафедра прикладної математики та інформаційних технологій
(назва кафедри, що забезпечує викладання)

"ЗАТВЕРДЖУЮ"
Декан факультету
математики та інформатики
_____ Ольга МАРТИНЮК
"28" 08 2026 року

РОБОЧА ПРОГРАМА
навчальної дисципліни
Системне програмування
(назва навчальної дисципліни)

обов'язкова
(вказати: обов'язкова)

Освітньо-професійна програма Інформаційні технології та управління проектами
(назва програми)

Спеціальність 122 Комп'ютерні науки
(вказати: код, назва)

Галузь знань 12 Інформаційні технології
(вказати: шифр, назва)

Рівень вищої освіти перший (бакалаврський)
(вказати: перший (бакалаврський) / другий (магістерський) / третій (освітньо-науковий)

Факультет математики та інформатики
(назва факультету/ навчально-наукового інституту, на якому здійснюється підготовка фахівців за вказаною освітньою програмою)

Мова навчання українська
(вказати: на якій мові читається дисципліна)

Чернівці 2026 рік

Робоча програма навчальної дисципліни «*Системне програмування*» складена відповідно до освітньо-професійної програми «Інформаційні технології та управління проектами»

Розробник:
Сопронюк Тетяна Миколаївна, доцент кафедри прикладної математики та інформаційних технологій, кандидат фіз.-мат. наук, доцент

Викладач, що забезпечує читання даної навчальної дисципліни:
Сопронюк Тетяна Миколаївна, доцент кафедри прикладної математики та інформаційних технологій, кандидат фіз.-мат. наук, доцент

Погоджено з гарантом ОП _____ **Ігор ЮРЧЕНКО**

Затверджено на засіданні кафедри прикладної математики та інформаційних технологій
Протокол № 2 від «28» серпня 2026 року

Завідувач кафедри _____ **Ярослав БІГУН**

Схвалено методичною радою факультету математики та інформатики
Протокол № 1 від «28» серпня 2026 року

Голова методичної ради _____ **Тоня ФРАТАВЧАН**

Мета навчальної дисципліни

Студенти повинні опанувати основні принципи побудови компіляторів, елементи теорії формальних мов (регулярні вирази, формальні граматики, зокрема, граматики Хомського, розпізнавачі) та теорію скінченних автоматів.

Результати навчання

У результаті вивчення навчальної дисципліни студент повинен

знати: типи мовних процесорів, основні фази компіляції, роботу з хеш-таблицями, поняття автоматних і неавтоматних мов та засоби їх представлення, алгоритми перетворення різних форм подання формальних мов, алгоритми перетворення автоматів (недетермінованого скінченного автомата в детермінований, алгоритми мінімізації та інші);

вміти: розв’язувати задачу належності для граматик, що розпізнають; генерувати ланцюжки граматик, що породжують; застосовувати алгоритми перетворення скінченних автоматів, праволінійних граматик та регулярних виразів.

Студенти повинні оволодіти програмним матеріалом, застосувати вивчені алгоритми до модельних прикладів, запрограмувати частину алгоритмів, виконати контрольні роботи, здати іспит.

Під час вивчення дисципліни, відповідно до освітньо-професійної програми, формуються наступні

загальні компетентності:

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу;

ЗК2. Здатність застосовувати знання у практичних ситуаціях;

ЗК3. Знання та розуміння предметної області та розуміння професійної діяльності.

ЗК6. Здатність вчитися й оволодівати сучасними знаннями.

ЗК7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ЗК8. Здатність генерувати нові ідеї (креативність).

ЗК9. Здатність працювати в команді.

ЗК10. Здатність бути критичним і самокритичним.

ЗК11. Здатність приймати обґрунтовані рішення.

ЗК12. Здатність оцінювати та забезпечувати якість виконуваних робіт.

фахові компетентності:

ФК1. Здатність до математичного формулювання та досліджування неперервних та дискретних математичних моделей, обґрунтовування вибору методів і підходів для розв’язування теоретичних і прикладних задач у галузі комп’ютерних наук, аналізу та інтерпретування

ФК3. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проєктування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв’язності та нерозв’язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.

ФК8. Здатність проєктувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об’єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

ФК12. Здатність забезпечити організацію обчислювальних процесів в інформаційних системах різного призначення з урахуванням архітектури, конфігурування, показників результативності функціонування.

та отримуються наступні **програмні результати навчання**:

ПРН1. Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.

ПРН9. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв’язання задач в галузі комп’ютерних наук.

ПРН13. Володіти мовами системного програмування та методами розробки програм, що взаємодіють з компонентами комп’ютерних систем, знати мережні технології, архітектури комп’ютерних мереж, мати практичні навички технології адміністрування комп’ютерних мереж та їх програмного забезпечення.

Опис навчальної дисципліни
Загальна інформація

Форма навчання	Рік	Семестр	Кількість		Кількість годин						Вид підсумкового контролю
			кредитів	годин	лекції	практичні	семінарсь	лаборатор	самостійна робота	індивідуальна робота	
Денна	3	6	4	120	30	-	-	30	60	-	залік

Структура змісту навчальної дисципліни

Назви змістових модулів і тем	Кількість годин					
	денна форма					
	усього	у тому числі				
		л	п	лаб	інд	с.р.
1	2	3	4	5	6	7
Змістовий модуль 1. Обчислювальні та регулярні вирази. Вирази з метасимволами						
Тема 1. **Розробка мовних процесорів мов програмування (загальний огляд)** Поняття мовного процесора. Типи мовних процесорів. Основні фази мовного процесора (лексичний аналіз, робота з таблицями, синтаксичний аналіз, генерація проміжного коду, оптимізація коду, генерація машинного коду, аналіз помилок). Спрощена модель компілятора. Проходи компілятора.	16	4	-	4		8
Тема 2. **Організація таблиць імен** Таблиці розміщення. Схеми хешування. Хешування зі						

списками. Алгоритм обчислення адреси хеш-таблиці розміщення в одновимірному випадку. Первинні і вторинні функції розміщення.	12	3	-	3		6
Тема 3. **Формальні мови і граматики** Алфавіт. Ланцюжки. Означення формальної мови. Способи визначення мов. Граматики, що породжують і розпізнають. Задача належності. Регулярні операції над мовами. Метамова БНФ. Розширення БНФ. Граматики Хомського. Спосіб визначення мови за допомогою граматик. Вивід ланцюжка в граматиці G і його аналіз. Ієрархія граматик Хомського.	16	4	-	4		8
Тема 4. **Регулярні множини і регулярні вирази** Регулярні множини та регулярні вирази. Тотожності над регулярними виразами. Системи з регулярними коефіцієнтами. Алгоритми перетворення праволінійної граматики в регулярний вираз і навпаки.	16	4	-	4		8
Разом за змістовим модулем 1	**60**	**15**	**-**	**15**		**30**
Змістовий модуль 2. Скінченні автомати і контекстно-вільні граматики						
Тема 5. **Скінченні автомати** Розпізнавачі (структура, конфігурація). Мова, що дозволяється розпізнавачем. Способи завдання скінченних автоматів. Детерміновані та недетерміновані скінченні автомати. Функція переходів. Алгоритм вилучення недосяжних станів ДСК. Розширена (узагальнена) функція переходів. Перевірка еквівалентності станів. Побудова таблиці нееквівалентності станів. Перевірка еквівалентності регулярних мов. Алгоритм перетворення недетермінованого скінченого автомата в детермінований. Мінімізація скінчених автоматів.	16	4	-	4		8

Тема 6. **Побудова спрощеного лексичного аналізатора** Етапи побудови лексичного аналізатора. Зв'язок між регулярними множинами, скінченими автоматами та праволінійними граматиками. Побудова лексичних аналізаторів на основі скінчених автоматів. Алгоритм побудови недетермінованого скінченного автомата за регулярним виразом. Програмування скінчених автоматів та лексичних аналізаторів на прикладі лексичного аналізатора, що розпізнає одну лексему – дійсне число.	12	3	-	3		6
Тема 7. **Синтаксичний аспект в мовах програмування** Породжуючі граматики. Контестно-вільні граматики. Дві ідеї аналізу. Дерево виводу. Лівостороння та правостороння стратегії виводу. Означення та властивості LA(1)-граматики. Множини FIRST і FOLLOW та алгоритми їх побудови. Ліворекурсивні та розширені рекурсії. Синтаксичні діаграми. Застосування алгоритму LA(1)-аналізу.	16	4	-	4		8
Тема 8. **Автоматні мови та регулярні вирази** Регулярні і нерегулярні мови. Лема про накачку. Приклади доведення нерегулярності мов. Алгоритм перетворення НСА в регулярний вираз за допомогою рекурентних формул. Алгоритм перетворення ДСА в регулярний вираз методом вилучення станів.	16	4	-	4		8
Разом за змістовим модулем 2	60	15	-	15		30
Усього годин	120	30	-	30		60

Тематика та питання лекційних занять

Тема 1. Розробка мовних процесорів мов програмування (загальний огляд).

Поняття мовного процесора. Типи мовних процесорів. Основні фази мовного процесора (лексичний аналіз, робота з таблицями, синтаксичний аналіз, генерація проміжного коду, оптимізація коду, генерація машинного коду, аналіз помилок). Спрощена модель компілятора. Проходи компілятора.

Тема 2. Організація таблиць імен.

Таблиці розміщення. Схеми хешування. Хешування зі списками. Алгоритм обчислення адреси хеш-таблиці розміщення в одновимірному випадку. Первинні і вторинні функції розміщення.

Тема 3. Формальні мови і граматики

Алфавіт. Ланцюжки. Означення формальної мови. Способи визначення мов. Граматики, що породжують і розпізнають. Задача належності. Регулярні операції над мовами. Метамова БНФ. Розширення БНФ. Граматики Хомського. Спосіб визначення мови за допомогою граматик. Вивід ланцюжка в граматиці G і його аналіз. Ієрархія граматик Хомського.

Тема 4. Регулярні множини і регулярні вирази

Регулярні множини та регулярні вирази. Тотожності над регулярними виразами. Системи з регулярними коефіцієнтами. Алгоритми перетворення праволінійної граматики в регулярний вираз і навпаки.

Тема 5. Скінченні автомати

Розпізнавачі (структура, конфігурація). Мова, що дозволяється розпізнавачем. Способи завдання скінченних автоматів. Детерміновані та недетерміновані скінченні автомати. Функція переходів. Алгоритм вилучення недосяжних станів ДСК. Розширена (узагальнена) функція переходів. Перевірка еквівалентності станів. Побудова таблиці нееквівалентності станів. Перевірка еквівалентності регулярних мов. Алгоритм перетворення недетермінованого скінченого автомата в детермінований. Мінімізація скінчених автоматів.

Тема 6. Побудова спрощеного лексичного аналізатора

Етапи побудови лексичного аналізатора. Зв’язок між регулярними множинами, скінченими автоматами та праволінійними граматиками. Побудова лексичних аналізаторів на основі скінчених автоматів. Алгоритм побудови недетермінованого скінченного автомата за регулярним виразом. Програмування скінчених автоматів та лексичних аналізаторів на прикладі лексичного аналізатора, що розпізнає одну лексему – дійсне число.

Тема 7. Синтаксичний аспект в мовах програмування

Породжуючі граматики. Контестно-вільні граматики. Дві ідеї аналізу. Дерево виводу. Лівостороння та правостороння стратегії виводу. Означення та властивості LA(1)-граматики. Множини FIRST і FOLLOW та алгоритми їх побудови. Ліворекурсивні та розширені рекурсії. Синтаксичні діаграми. Застосування алгоритму LA(1)-аналізу.

Тема 8. Автоматні мови та регулярні вирази

Регулярні і нерегулярні мови. Лема про накачку. Приклади доведення нерегулярності мов. Алгоритм перетворення НСА в регулярний вираз за

допомогою рекурентних формул. Алгоритм перетворення ДСА в регулярний вираз методом вилучення станів.

Тематика лабораторних занять

Розподіл балів і годин

Модуль 1. Обчислювальні та регулярні вирази. Вирази з метасимволами (37 бали)

Елемент модуля	Кількість балів	Час виконання
Лабораторна робота №1: Розпізнавання ланцюжків по заданих шаблонах в текстових файлах	8	**18.02**
Індивідуальне практичне завдання №1: Виконання всіх етапів компіляції для оператора присвоєння	6(4+2)	**25.02**
Лабораторна робота №1а: Генерація ланцюжків по заданих шаблонах	5	**4.03**
Лабораторна робота №2: Створення діалогових вікон Find, Replace, Delete з шаблонами для обробки текстових документів	8	**11.03**
Індивідуальне практичне завдання №2: Граматики, що розпізнають ланцюжки	4(2+2)	**18.03**
Індивідуальне практичне завдання №3: Граматики Хомського, регулярні вирази і скінченні автомати	6	**01.04**

Модуль 2. Скінченні автомати і контекстно-вільні граматики (28 балів)

Елемент модуля	Кількість балів	Час виконання
Індивідуальне практичне завдання №4: Моделювання скінченних автоматів	5(3+2)	**8.04**
Лабораторна робота №3: Побудова скінченного автомата і праволінійної граматики	13	15.04, 22.04
Лабораторна робота №4: Побудова аналізаторів методом рекурсивного спуску	7	6.05
Індивідуальне практичне завдання №5: Побудова регулярного виразу по скінченному автомату	3	13.05

Підсумковий модуль (тести). 25 балів

Перелік питань для самоконтролю та контролю знань студентів

1. Поняття мовного процесора. Типи мовних процесорів.
2. Основні фази мовного процесора. Лексичний аналіз. Робота з таблицями.
3. Основні фази мовного процесора. Синтаксичний аналіз. Генерація проміжного коду. Приклад.
4. Основні фази мовного процесора. Оптимізація коду. Генерація машинного коду.
5. Взаємодія етапів компіляції. Спрощена модель компілятора. Проходи компілятора. Аналіз помилок.
6. Таблиці розміщення. Схеми хешування. Хешування зі списками.
7. Хеш-таблиці. Алгоритм обчислення адреси таблиці розміщення в одновимірному випадку. Приклад.
8. Хеш-таблиці. Функції розміщення
9. Алфавіт. Ланцюжки. Означення формальної мови. Приклади.
10. Граматики, що породжують і розпізнають. Задача належності. Регулярні операції над мовами.
11. Способи визначення мов. Метамова БНФ. Приклади.
12. Способи визначення мов. Розширені БНФ. Приклади.
13. Граматики Хомського. Спосіб визначення мови за допомогою граматики.
14. Ієрархія граматик Хомського. Приклади.
15. Приклад доведення належності мови, що складається з ланцюжків символів a і b парної довжини, перша половина яких співпадає з другою, граматиці без обмежень (приклад граматики додається).
16. Детерміновані та недетерміновані розпізнавачі (структура, конфігурація). Мова, що дозволяється розпізнавачем.
17. Регулярні множини та регулярні вирази. Приклади. Тотожності над регулярними виразами. Теорема про зв’язок між регулярними множинами та праволінійними граматиками.
18. Системи рівнянь з регулярними коефіцієнтами. Приклад побудови регулярного виразу по праволінійній граматиці.
19. Алгоритм побудови праволінійної граматики по регулярному виразу. Приклад.
20. Поняття скінченного автомата. Конфігурація. Такт. Детерміновані та недетерміновані скінченні автомати. Мова, що дозволяється скінченним автоматом. Діаграми переходів.
21. Моделювання роботи автомата із заданими ланцюжками. Алгоритм розпізнавання ланцюжків, належних мові ДСА.
22. Алгоритм перетворення недетермінованого скінченного автомата в детермінований. 24. Поняття зведеного (мінімального) скінченного автомата. Приклад неформальної мінімізації.
23. Етапи мінімізації. Означення недосяжних станів. Алгоритм вилучення недосяжних станів. Приклад.
24. Означення еквівалентних станів і станів, що k-не розрізняються. Мінімізація детермінованого скінченного автомата методом побудови відношень еквівалентності.

25. Функція переходів і розширена (узагальнена) функція переходів недетермінованого скінченного автомата. Приклад. Алгоритм побудови розширеної функції переходів.
26. Означення мови, що дозволяється НСА, еквівалентних станів і станів, що розрізняються, через узагальнену функцію переходів. Мінімізація детермінованого скінченного автомата методом побудови таблиці нееквівалентних станів.
27. Алгоритм побудови недетермінованого скінченного автомата по регулярному виразу (для найпростіших регулярних виразів та операцій альтернативи й ітерації (*)).
28. Алгоритм побудови недетермінованого скінченного автомата по регулярному виразу (для найпростіших регулярних виразів та операцій конкатенації й ітерації (+)).
29. Поняття лексичного аналізатора. Можливий варіант реалізації лексичного аналізатора.
30. Побудова лексичного аналізатора, що розпізнає одну лексему – дійсне число.
31. Алгоритм и програмна реалізація детермінованого скінченного автомата (два способи).
32. Синтаксичний аналіз і контекстно-вільні граматики. Дерева розбору
33. LA(1)-граматики. Множини First і Follow
34. Ліворекурсивні та розширені продукції. Синтаксичні діаграми
35. Алгоритми видалення лівої рекурсії
36. Побудова алгоритму LA(1)-аналізу
37. Регулярні і нерегулярні мови. Лема про накачку.
38. Приклади доведення нерегулярності мов.
39. Форми подання регулярних мов. Перевірка еквівалентності регулярних мов.
40. Алгоритм перетворення НСА в регулярний вираз за допомогою рекурентних формул.
41. Алгоритм перетворення ДСА в регулярний вираз методом вилучення станів.
42. Перетворення праволінійної граматики у недетермінований скінченний автомат

Завдання для самостійної роботи

Самостійна робота студентів є важливою складовою навчального процесу і спрямована на поглиблення, узагальнення та закріплення знань, отриманих на лекційних і лабораторних заняттях, а також на формування навичок самостійного опрацювання теоретичного матеріалу, розроблення алгоритмів і програмних рішень.

До кожної теми навчальної дисципліни подаються види самостійної роботи, що враховують специфіку освітньої програми «Системний аналіз» і зміст навчальної дисципліни «Системне програмування».

Основні види самостійної роботи студентів:

- опрацювання конспектів лекцій, навчальної та наукової літератури з теми;

- виконання завдань, розміщених у системі дистанційного навчання (Moodle);
- підготовка до лабораторних і модульних робіт;
- розроблення індивідуальних алгоритмів і програмних модулів відповідно до завдань дисципліни;
- аналіз прикладів реалізації компіляторів, інтерпретаторів, скінченних автоматів, формальних граматик;
- підготовка до тестового контролю та підсумкового заліку.

Кількість годин, передбачених навчальним планом для самостійної роботи, розподіляється між темами дисципліни відповідно до їх змісту та складності.

Орієнтовний розподіл самостійної роботи за темами:

№	Назва теми	Орієнтовна кількість годин самостійної роботи
1	Розробка мовних процесорів мов програмування	8
2	Організація таблиць імен	6
3	Формальні мови і граматики	8
4	Регулярні множини і регулярні вирази	8
5	Скінченні автомати	8
6	Побудова спрощеного лексичного аналізатора	6
7	Синтаксичний аспект в мовах програмування	8
8	Автоматні мови та регулярні вирази	8
Разом:		**60 годин**

Примітка: конкретний зміст і форми самостійної роботи уточнюються викладачем з урахуванням рівня підготовки студентів, особливостей навчального процесу та використання системи дистанційного навчання Moodle.

Критерії оцінювання результатів навчання з навчальної дисципліни

Підсумкова оцінювання – це тестове завдання.

Тести відкриваються екзаменатором і оцінюється системою Moodle. Посилання на тест: (http://e-learning.fpm.chnu.edu.ua/mod/quiz/view.php?id=15692).

Оцінювання відбувається автоматично. Правильна відповідь на кожне питання оцінюється у 1 бал. Тести стосуються лекційного матеріалу та завдань, які давалися на протязі семестру. Послідовність питань вибирається випадковим чином із підготовленого банку питань.

Захист лабораторної/практичної роботи та критерії оцінювання

- Здача лабораторної роботи та практичного заняття завдання проводиться під час заняття згідно з календарними планом.
- Для захисту лабораторної роботи кожен студент має самостійно виконати лабораторну роботу і здати її викладачу **на занятті.**
- Не допускається заочне прийняття програм (електронною поштою) без запуску програм з різними вхідними даними.
- Під час здачі програми викладач зобов'язаний перевіряти здатність студента орієнтуватися у власній програмі, пропонуючи йому виконати нескладні зміни, розраховані на 5-10 хвилин поточного заняття.
- При необхідності виконання частини завдання або усього завдання у **робочому** зошиті, бали виставляти у зошиті, вказуючи число і підпис.
- Під час здачі лабораторної роботи студент повинен

- вміти пояснити постановку задач, які розв'язувались в лабораторній роботі; алгоритм розв'язування задач; програмну реалізацію завдання;

- продемонструвати розуміння програми та обґрунтувати зроблені висновки;

- відповісти на питання, які належать до виконання лабораторної роботи та додаткові теоретичні питання, якщо розданий перелік таких питань.

- Якщо студент не розуміє алгоритму розв'язання задачі, не орієнтується в програмній реалізації, але є у наявності правильно виконувана програма, то робота зараховується не більше як на 30%.
- Якщо студент розуміє задачу і алгоритм її виконання, але не орієнтується (слабо орієнтується) в практичній частині (програмній реалізації), то оцінка знижується до 50%.
- Якщо програма не працює, або працює частково і студент може пояснити алгоритм, роботу оцінювати частково, в залежності від об'єму і якості коду.
- Кількість балів за лабораторну роботу визначає викладач в процесі здачі. Оцінка повідомляється студенту.
- За невчасний захист лабораторних робіт у межах модуля допускається знімати по одному балу за кожне прострочене заняття, якщо робота оцінюється до 10 балів, і по 1,5-2 бали, якщо робота оцінюється в межах від 11 до 20 балів, але не більше половини балів.

Шкала оцінювання: національна та ЄКТС

Сума балів за всі види навчальної діяльності	Оцінка ЄКТС	Оцінка за національною шкалою
		для заліку
90 – 100	**A**	зараховано
80-89	**B**	
70-79	**C**	
60-69	**D**	
50-59	**E**	
35-49	**FX**	не зараховано з можливістю повторного складання
0-34	**F**	не зараховано з обов'язковим самостійним опрацюванням освітнього компоненту до перескладання

Засоби оцінювання

Під час проведення лекцій використовуються пасивний та активний методи навчання. Консультаційна робота. Під час виконання студентами лабораторних робіт використовується активні методи навчання. Проведення модульних лабораторних та практичних робіт та навчальна робота під час їх прийому.

Форми поточного та ідсумкового контролю

1. Оцінювання на лабораторних заняттях в обох модулях.
2. Тестові завдання засобами системи Moodle.

Розподіл балів, які отримують студенти

	Поточне тестування та самостійна робота									Тести	Сума
Змістовий модуль №1						Змістовий модуль № 2					
Т1	Т2	Т2	Т3	Т4	Т4	Т5	Т6	Т7	Т8	25	100
8	6	5	8	4	6	5	13	7	3		

Т1, Т2 ... Т8 – теми змістових модулів.

Рекомендована література
Базова (основна)

1. Сопронюк Т.М. Мовні процесори та формальні мови: від теорії до практики : навч. посібник. Чернівці : Чернівец. нац. ун-т ім. Ю. Федьковича, 2025. 198 с. ISBN 978-966-423-942-1 https://archer.chnu.edu.ua/xmlui/handle/123456789/12079
2. Сопронюк Т.М. Елементи теорії компіляції: Навчальний посібник. – Чернівці: ЧНУ, 2008. – 84 с.
3. Сопронюк Т.М. Елементи теорії компіляції: Навчальний посібник. – Чернівці: ЧНУ, 2008. – 84 с.
4. A.V. Aho and J. D. Ullman. The Theory of Parsing, Translation, and Compiling. Vol. 1. Englewood Cliffs, N.J.: Prentice Hall, 1972. - 460 p. https://www.amazon.com/Theory-Parsing-Translation-Compiling/dp/0139145567
4. P. J. Denning, J. B. Dennis, and J. E. Qualitz. Machines, Languages,and Computation. Englewood Cliffs, N.J.: Prentice Hall. 3. M., 1978.
5. M. A. Harrison. Introduction to Formal Language Theory. Reading,Mass.: Addison-Wesley, 1978.
6. W. Homenda, W. Pedrycz. Automata theory and formal languages, De Gruyter, 2022.
7. J. E. Hopcroft and J. D. Ullman. Introduction to Automata Theory, Languages and Computation. Reading, Mass.: Addison-Wesley, 1979.
8. R. Hunter. The Design and Construction of Compilers. Chichester, New York: John Wiley, 1981.
9. Z. Kohavi and N. K. Jha. Switching and Finite Automata Theory. Third Edition. New York: Cambridge University Press, 2010.

10. P. Linz. An Introduction to Formal Languages and Automata-Peter Linz Univ. of California at Davis. Jones & Bartlett Learning, 2016.
11. Pettorossi. Automata theory and formal languages: fundamental notions, theorems, and techniques. Springer, 2022.
12. G. E. Revesz. Introduction to Formal Languages. New York: McGrawHill, 1983.
13. Теорія цифрових автоматів та формальних мов. Вступний курс: навчальний посібник С.Ю Гавриленко,А.М. Клименко, Н.Ю. Любченко та ін. – Харків: НТУ ХПІ, 2007. – 176 с.
14. Олег Гутік, Формальні мови та автомати. Ел. Посібник. Львів, 2022.
15. Олег Гутік, Коди та автомати: основи алгебраїчної теорії. Ел. Посібник. Львів, 2021.
16. Сопронюк Т. М., Сопронюк А. Ю., Дробот А. В. Фази побудови мовного процесора для платформи .NET// Буковинський матем. журнал. — 2023. — Т.11, No2. — С. 71–84.
17. Формальні мови, граматики та автомати. [Електронний ресурс]. – Режим доступу: https://repository.kpi.kharkov.ua/server/api/core/bitstreams/5847efdd-6ff5-4f7f-9de8-6f6555ad4cc0/content
18. Online compiler and debugger [Електронний ресурс] Режим доступу: https://www.onlinegdb.com
19. Вікіпедія. Формальні граматики. Доступно за посиланням: http://uk.wikipedia.org/wiki/ Формальні_граматики. [Дата звернення: 20 січня 2025].
20. Вікіпедія. Скінченний автомат. Доступно за посиланням: https://uk.wikipedia.org/wiki/Скінченний_автомат. [Дата звернення: 20 січня 2025].
21. Сопронюк Т. М., Сопронюк А. Ю., Дробот А. В. Фази побудови мовного процесора для платформи .NET// Буковинський матем. журнал. — 2023. — Т.11, No2. — С. 71–84.[Електронна версія: DOI: https://doi.org/10.31861/bmj2023.02.07]

Інформаційні ресурси

1. http://www-db.stanford.edu/~ullman/ialc.html – Introduction to Automata Theory, Languages, and Computation. Slides and Lecture Notes. (Stanford University)
2. https://www.pdfdrive.com/the-theory-of-parsing-translation-and-compiling-volume-1-parsing-e157156882.html - The Theory of Parsing, Translation, and Compiling (Volume 1): Parsing by Alfred
3. V. Aho & Jeffrey D. Ullman. 1973– 562 p.
4. http://www.unicyb.kiev.ua/Library/PROG/Zmist.htm – А.Б.Ставровський. Посібник з програмування (факультет кібернетики Київського національного університету)
5. http://uk.wikipedia.org/wiki/Формальні_граматики
6. http://courses.cs.vt.edu/~cs4114/lectures/ – Formal Languages and Automata Theory Cource Lecture notes
7. http://khpi-iip.mipk.kharkiv.edu/library/datastr/book_sod/kgsu/oglav.html – Мова програмування С++. Динамічні структури даних

Додатково

Методичне забезпечення

1. Сопронюк Т.М. Мовні процесори та формальні мови: від теорії до практики : навч. посібник. Чернівці : Чернівец. нац. ун-т ім. Ю. Федьковича, 2025. 198 с. ISBN 978-966-423-942-1 https://archer.chnu.edu.ua/xmlui/handle/123456789/12079
2. Сопронюк Т.М. Елементи теорії компіляції: Навчальний посібник. – Чернівці: ЧНУ, 2008. – 84 с.
3. Сопронюк Т.М. Елементи теорії формальних мов: Навчальний посібник. – Чернівці: ЧНУ, 2008. – 84 с.
4. Сопронюк Т. М., Сопронюк А. Ю., Дробот А. В. Фази побудови мовного процесора для платформи .NET// Буковинськиӈ матем. журнал. — 2023. — Т.11, No2. — С. 71–84.
5. Тестові завдання (Система Moodle)
6. Презентації лекцій (Система Moodle)
7. Відео-лекції на Google диску
8. Сертифікат про закінчення курсів професора Стенфордського університету Дж.Ульмана “Automata” (https://drive.google.com/file/d/0B-cxrxmP0J7XRUl2YlNGem1wb3c/view?resourcekey=0--NXsRlO20hZHdxuVKlDQ4Q)

Політика академічної доброчесності

Дотримання політики щодо академічної доброчесності учасниками освітнього процесу при вивченні освітнього компонента регламентовано чинними документами ЧНУ:

- "Кодекс академічної доброчесності та етики Чернівецького національного університету імені Юрія Федьковича" https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/kodeks-akademichnoi-dobrochesnosti-ta-etyky-chernivetskoho-natsionalnoho-universytetu-imeni-yuriia-fedkovycha/
- "Положення про виявлення та запобігання академічного плагіату у Чернівецькому національному університеті імені Юрія Федьковича" https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/polozhennia-pro-vyiavlennia-ta-zapobihannia-akademichnomu-plahiatu-u-chernivetskomu-natsionalnomu-universyteti-imeni-yuriia-fedkovycha/
- "Політика використання штучного інтелекту в Чернівецькому національному університеті імені Юрія Федьковича" https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/polityka-vykorystannia-shtuchnoho-intelektu-v-chernivetskomu-natsionalnomu-universyteti-imeni-yuriia-fedkovycha-zi-zminamy-zatverdzhenymy-vchenoiu-radoiu-chnu-imeni-yuriia-fedkovycha/

Використання здобувачами освіти інструментів штучного інтелекту (ШІ) під час виконання навчальних завдань:

- *допускається* за умови дотримання принципів академічної доброчесності (здобувач освіти несе відповідальність за зміст, достовірність і результати поданої роботи, а у разі використання ШІ зобов'язаний зазначити факт і спосіб його застосування);
- *не допускається* використання ШІ як заміни самостійного виконання завдань або з метою порушення вимог академічної доброчесності;
- *додаткові обмеження чи вимоги* щодо використання ШІ *можуть визначатися* викладачем в освітньому процесі для конкретних видів навчальної діяльності відповідно до особливостей дисципліни.