Чернівецький національний університет імені Юрія Федьковича
Факультет математики та інформатики
Кафедра прикладної математики та інформаційних технологій

“ЗАТВЕРДЖУЮ”
Декан факультету математики та інформатики
_____ Ольга МАРТИНЮК
“ 28 ” серпня 2026 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Створення власної мови програмування

(назва навчальної дисципліни)

Вибіркова навчальна дисципліна
(вказати: обов’язкова / вибіркова)

Освітньо-професійна програма: Технології програмування та комп’ютерне моделювання
(назва програми)

Спеціальність: F1 «Прикладна математика»
(вказати: код, назва)

Галузь знань: F – Інформаційні технології
(вказати: шифр, назва)

Рівень вищої освіти: перший (бакалаврський)
(вказати: перший бакалаврський / другий магістерський)

Факультет: факультет математики та інформатики
(назва факультету/інституту, на якому здійснюється підготовка фахівців за вказаною освітньо-професійною програмою)

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

Чернівці – 2026

Робоча програма із вибіркової навчальної дисципліни «Створення власної мови програмування» складена відповідно до освітньо-професійної програми «Технології програмування та комп'ютерне моделювання» першого (бакалаврського) рівня вищої освіти за спеціальністю «F1 Прикладна математика» галузі знань «F Інформаційні технології», затвердженої Вченою радою Чернівецького національного університету імені Юрія Федьковича 28 квітня 2025 р., протокол №5.

Розробник: Тетяна Миколаївна Сопронюк, доцент, канд. фіз.-мат. наук, доцент
(П.І.Б. автора, посада, науковий ступінь, вчене звання)

Викладач, що забезпечує читання даної навчальної дисципліни: Тетяна Миколаївна Сопронюк, доцент, канд. фіз.-мат. наук, доцент
(П.І.Б. викладача, посада, науковий ступінь, вчене звання)

Погоджено з гарантом ОП _____ Василь МАЦЕНКО
(підпис)

Затверджено на засіданні кафедри прикладної математики та інформаційних технологій

Протокол № 2 від « 28 » серпня 2026 року

Завідувач кафедри _____ Ярослав БІГУН
(підпис)

Схвалено методичною радою факультету математики та інформатики

Протокол № 1 від « 28 » серпня 2026 року

Голова методичної ради _____ Тоня ФРАТАВЧАН
(підпис)

Мета навчальної дисципліни

Опанувати основні принципи побудови компіляторів, елементи теорії формальних мов та теорію скінченних автоматів, навчитися будувати спрощені лексичні і синтаксичні аналізаторів, продемонструвати застосування основних принципів побудови компіляторів на прикладі розробки інтерпретатора простої мови програмування SPL (інтерпретатор має обмежені можливості, і це дозволяє охопити весь процес його побудови).

Пререквізити: системне програмування (теорія скінченних автоматів, формальні граматики, регулярні вирази).

Результати навчання. У результаті вивчення навчальної дисципліни студент повинен:

знати: основні фази компіляції, принципи побудови лексичних і синтаксичних аналізаторів та компіляторів.

вміти: застосовувати теорію формальних мов до вирішення задач аналізу, програмувати алгоритми перетворення скінченних автоматів, праволінійних граматик та регулярних виразів; розробляти власні найпростіші аналізатори та інтерпретатори.

Студенти повинні оволодіти програмним матеріалом, застосувати вивчені алгоритми до модельних прикладів, запрограмувати частину алгоритмів, виконати залікову тестову контрольну роботу.

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

загальні компетентності:

ЗК01. Здатність учитися і оволодівати сучасними знаннями.

ЗК02. Здатність застосовувати знання у практичних ситуаціях.

ЗК03. Здатність генерувати нові ідеї (креативність).

ЗК06. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК10. Навички у використанні інформаційних і комунікаційних технологій.

фахові компетентності:

ФК01. Здатність використовувати й адаптувати математичні теорії, методи та прийоми для доведення математичних тверджень і теорем.

ФК06. Здатність розв'язувати професійні задачі за допомогою комп'ютерної техніки, комп'ютерних мереж та Інтернету, в середовищі сучасних операційних систем, з використанням стандартних офісних додатків.

ФК07. Здатність експлуатувати та обслуговувати програмне забезпечення автоматизованих та інформаційних систем різного призначення.

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

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

ФК17. Здатність до використання новітніх інформаційно-комунікаційних технологій.

та отримуються такі програмні результати навчання:

ПРН11. Вміти застосовувати сучасні технології програмування та розроблення програмного забезпечення, програмної реалізації чисельних і символьних алгоритмів.

ПРН14. Виявляти здатність до самонавчання та продовження професійного розвитку.

ПРН16. Демонструвати навички взаємодії з іншими людьми, уміння працювати в команді.

Опис навчальної дисципліни

Загальна інформація

Назва навчальної дисципліни: Створення власної мови програмування												
Форма навчання	**Рік підготовки**	**Семестр**	**Кількість кредитів**	**Кількість годин**	**Змістових модулів**	**Лекції**	**Практичні**	**Семінарські**	**Лабораторні**	**Самостійна робота**	**Індивідуальні завдання**	**Вид підсумкового контролю**
Денна	4	7	**3**	**90**	2	30	—	—	16	44	—	**Залік**

Структура змісту навчальної дисципліни

Назви змістових модулів і тем	**Усього годин (денна форма)**	**Лекції (л)**	**Практичні (п)**	**Лабораторні (лаб)**	**Інд. завд. (інд)**	**Самост. робота (с.р.)**
1	**2**	**3**	**4**	**5**	**6**	**7**
Змістовий модуль 1. Лексичний і синтаксичний аналіз						

Назви змістових модулів і тем	Усього годин (денна форма)	Лекції (л)	Практичні (п)	Лабораторні (лаб)	Інд. завд. (інд)	Самост. робота (с.р.)
1	2	3	4	5	6	7
Тема 1. Лексичний аналіз і скінченні автомати. Поняття лексичного аналізатора. Побудова лексичних аналізаторів, що розпізнають лексему одного типу. Побудова лексичного аналізатора, що розпізнає декілька типів лексем у тексті. Програмна реалізація детермінованого скінченного автомата з декількома типами заключних станів. Скінченні автомати для розпізнавання ключових слів у тексті.	**12**	4	—	4	—	4

Назви змістових модулів і тем	Усього годин (денна форма)	Лекції (л)	Практичні (п)	Лабораторні (лаб)	Інд. завд. (інд)	Самост. робота (с.р.)
1	2	3	4	5	6	7
Тема 2. Синтаксичний аналіз і контекстно-вільні граматики. Розбір та інтерпретація математичних формул. Граматики Хомського і LA(1)-граматики. Синтаксичний аналізатор розпізнавання виразів. Середовище розробки аналізаторів ANTLRWorks.	**12**	4	—	4	—	4
Тема 3. Побудова інтерпретатора математичних формул. Побудова спрощеного синтаксичного аналізатора для виконання числових обчислень і обчислення значення функції в точці. Лексичний аналіз формули. Синтаксичний аналіз формули методом рекурсивного спуску.	**8**	2	—	—	—	6

Назви змістових модулів і тем	Усього годин (денна форма)	Лекції (л)	Практичні (п)	Лабораторні (лаб)	Інд. завд. (інд)	Самост. робота (с.р.)
1	2	3	4	5	6	7
Тема 4. Обчислення регулярних виразів над формалізмами автоматних мов. Створення ієрархії класів для формалізмів автоматних мов. Робота з об’єктами класів “Регулярний вираз”, “Праволінійна граматика”, “Недетермінований скінчений автомат”. Обчислення глобального регулярного виразу методом рекурсивного спуску	**8**	2	—	—	—	6
Разом за змістовим модулем 1	**40**	**12**	**—**	**8**	**—**	**20**
Змістовий модуль 2. Створення інтерпретатора мови SPL						
Тема 1. Можливості інтерпретатора мови програмування SPL. Неформальний опис мови SPL . Приклади SPL-програм. Склад інтерпретатора	**10**	4	—	2	—	4

Назви змістових модулів і тем	Усього годин (денна форма)	Лекції (л)	Практичні (п)	Лабораторні (лаб)	Інд. завд. (інд)	Самост. робота (с.р.)
1	2	3	4	5	6	7
Тема 2. Фаза лексичного аналіза. Типи лексем та їх кодування. Таблиця ідентифікаторів. Приклад лексичного аналізу SPL-програми	**8**	2	—	1	—	5
Тема 3. Синтаксичний аналіз мови SPL. Формальний синтаксис мови SPL. Аналіз об’яв. Аналіз операторів і виразів. Приклад граматичного розбору одного оператора мови SPL. Застосування середовища розробки аналізаторів ANTLRWorks для опису граматики мови SPL.	**10**	4	—	2	—	4
Тема 4. Інтерпретація. SPL-процесор. Виконання SPL-програми. Створення SPL-програми. Приклад проміжного коду для SPL програми. Розробка компілятора SPL для платформи .NET	**12**	4	—	2	—	6

Назви змістових модулів і тем	Усього годин (денна форма)	Лекції (л)	Практичні (п)	Лабораторні (лаб)	Інд. завд. (інд)	Самост. робота (с.р.)
1	2	3	4	5	6	7
Тема 5. Генерація проміжного коду. Робота з таблицями. Розширення аналізу описів. Розширення аналізу операторів та виразів.	**10**	4	—	1	—	5
Разом за змістовим модулем 2	**50**	**16**	**—**	**8**	**—**	**26**
Усього годин	**90**	**30**	**—**	**16**	**—**	**44**

Тематика та питання лекційних занять

Змістовий модуль 1. Лексичний і синтаксичний аналіз

1. Тема 1.1. Лексичний аналіз і скінченні автомати. Поняття лексичного аналізатора. Побудова лексичних аналізаторів, що розпізнають лексему одного типу. Побудова лексичного аналізатора, що розпізнає декілька типів лексем у тексті. Програмна реалізація детермінованого скінченного автомата з декількома типами заключних станів. Скінченні автомати для розпізнавання ключових слів у тексті.

2. Тема 1.2. Синтаксичний аналіз і контекстно-вільні граматики. Розбір та інтерпретація математичних формул. Граматики Хомського і LA(1)-граматики. Синтаксичний аналізатор розпізнавання виразів. Середовище розробки аналізаторів ANTLRWorks.

3. Тема 1.3. Побудова інтерпретатора математичних формул. Побудова спрощеного синтаксичного аналізатора для виконання числових обчислень і обчислення значення функції в точці. Лексичний аналіз формули. Синтаксичний аналіз формули методом рекурсивного спуску.

4. Тема 1.4. Обчислення регулярних виразів над формалізмами автоматних мов. Створення ієрархії класів для формалізмів автоматних мов. Робота з об’єктами класів “Регулярний вираз”, “Праволінійна граматика”, “Недетермінований скінчений автомат”. Обчислення глобального регулярного виразу методом рекурсивного спуску

Змістовий модуль 2. Створення інтерпретатора мови SPL

5. Тема 2.1. Можливості інтерпретатора мови програмування SPL. Неформальний опис мови SPL . Приклади SPL-програм. Склад інтерпретатора

6. Тема 2.2. Фаза лексичного аналіза. Типи лексем та їх кодування. Таблиця ідентифікаторів. Приклад лексичного аналізу SPL-програми

7. Тема 2.3. Синтаксичний аналіз мови SPL. Формальний синтаксис мови SPL. Аналіз об’яв. Аналіз операторів і виразів. Приклад граматичного розбору одного оператора мови SPL. Застосування середовища розробки аналізаторів ANTLRWorks для опису граматики мови SPL.

8. Тема 2.4. Інтерпретація. SPL-процесор. Виконання SPL-програми. Створення SPL-програми. Приклад проміжного коду для SPL програми. Розробка компілятора SPL для платформи .NET

9. Тема 2.5. Генерація проміжного коду. Робота з таблицями. Розширення аналізу описів. Розширення аналізу операторів та виразів.

Тематика та питання (завдання) лабораторних занять

№ з/п	Назва теми	Кількість годин
1	Графічне середовище розробки граматик ANTLRWorks	2
2	Побудова спрощенного лексичного аналізатора	4
3	Побудова спрощенного синтаксичного аналізатора	4
4	Елементи компіляції	4

Завдання для самостійної роботи

№ з/п	Назва алгоритмів для самостійного вивчення (п. 1) і завдань для самостійного виконання (п. 2)	Кількість годин
Форма навчання	Рік підготовки	Семестр
Форма навчання	Рік підготовки	Семестр
Денна	4	7

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 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**	зараховано
70–79	**C**	зараховано
60–69	**D**	зараховано
50–59	**E**	зараховано
35–49	**FX**	не зараховано з можливістю повторного складання
0–34	**F**	не зараховано з обов’язковим повторним вивченням дисципліни

Засоби оцінювання

Під час проведення лекцій використовуються пасивний та активний методи навчання, консультаційна робота. Під час виконання студентами лабораторних робіт використовуються активні методи навчання.

Форми поточного та підсумкового контролю

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

Розподіл балів, які отримують студенти

Поточне оцінювання та самостійна робота (80 балів)									Залік (тести)	Сума
Змістовий модуль №1 (40 балів)				Змістовий модуль № 2 (40 балів)						
Т1	Т2	Т3	Т4	Т1	Т2	Т3	Т4	Т5		
10	10	10	10	10	10	10	5	5	**20**	**100**

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. 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. A. 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-TheTheoryofParsing,Translation,andCompiling (Volume 1): Parsing by Alfred V. Aho & Jeffrey D. Ullman. 1973– 562 p.

3. http://www.unicyb.kiev.ua/Library/PROG/Zmist.htm – А.Б.Ставровський. Посібник з програмування (факультет кібернетики Київського національного університету)

4. http://uk.wikipedia.org/wiki/Формальні_граматики

5. http://courses.cs.vt.edu/~cs4114/lectures/ – Formal Languages and Automata Theory Cource Lecture notes

6. http://khpi-iip.mipk.kharkiv.edu/library/datastr/book_sod/kgsu/oglav.html – Мова програмування С++. Динамічні структури даних

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// Буковинський 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 університету імені Юрія Федьковича» (https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/polityka-vykorystannia-shtuchnoho-intelektu-v-chernivetskomu-natsionalnomu-universyteti-imeni-yuriia-fedkovycha/).

Використання здобувачами вищої освіти систем штучного інтелекту під час виконання завдань допускається виключно як допоміжного інструменту за умови обов’язкового зазначення факту такого використання та критичного осмислення отриманих результатів. Подання згенерованого штучним інтелектом коду чи тексту без належного посилання та розуміння його роботи вважається порушенням академічної доброчесності.