Чернівецький національний університет імені Юрія Федьковича

факультет математики та інформатики
Прикладної математики та інформаційних технологій

СИЛАБУС
навчальної дисципліни

Програмування мовних процесорів

Вибіркова

Освітньо-професійні програми:
«Технології програмування та комп'ютерне моделювання»
Спеціальності:
113 «Прикладна математика»,
Галузі знань:
11 «Математика та статистика»,

Рівень вищої освіти *перший (бакалаврський)*
Факультет математики та інформатики
Мова навчання *українська*
Розробник:
Сопронюк Т. М., доцент кафедри прикладної математики та інформаційних технологій, кандидат фізико-математичних наук

Профайл викладача https://amit.chnu.edu.ua/pro-kafedru/personalii/soproniuk-tetiana-mykolaivna/
E-mail: t.sopronyuk@chnu.edu.ua

Сторінка курсу в Moodle: : https://moodle.chnu.edu.ua/course/view.php?id=71
Консультації Онлайн-консультації: Середа з 18.00 до 19.00.

1. **Анотація дисципліни (призначення навчальної дисципліни).** У курсі вивчаються основні принципи побудови компіляторів, елементи теорії формальних мов та скінченних автоматів.

2. Мета навчальної дисципліни: 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).

2. **Пререквізити.** Для ефективності засвоєння курсу здобувач вищої освіти має вивчити дисципліну «Системне програмування» (регулярні вирази, формальні граматики, скінченні автомати), «Програмування».

4. 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 наступні

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:

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 наступні **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.

5. Опис навчальної дисципліни

5.1. Загальна інформація

Назва навчальної дисципліни_____													
Форма навчання	**Рік підготовки**	**Семестр**	**Кількість**			**Кількість годин**							**Вид підсумкового контролю**
			кредити	**годин**	**змістових**	**лекції**	**практичн**	**семінарс**	**лаборато**	**самостій роботта**	**індивідуа**		
Денна	4	7	3	90	2	15	-	-	30	45	-		Залік

5.2. Дидактична карта 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
Модуль 1						
Змістовий модуль 1. Лексичний і синтаксичний аналіз						
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.	10	2	-	4	-	4
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. Генератор лексичних аналізаторів Flex	10	2	-	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.	10	2	-	4	-	4
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	10	2	-	4	-	4
Разом за змістовим модулем 1	40	8	-	16	-	16
Змістовий модуль 2. Створення інтерпретатора мови SPL						
Тема 1. Можливості інтерпретатора мови програмування SPL. Неформальний опис мови SPL . Приклади SPL-програм. Склад інтерпретатора	10	1	-	2	-	7
Тема 2. Фаза лексичного аналіза. Типи лексем та їх кодування. Таблиця ідентифікаторів. Приклад лексичного аналізу SPL-програми	10	1	-	2	-	7
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	2	-	4	-	4
Тема 4. Інтерпретація. SPL-процесор. Виконання SPL-програми. Створення SPL-програми. Приклад проміжного коду для SPL програми. Розробка компілятора SPL для платформи .NET	10	2	-	4	-	4
Тема 5. Генерація проміжного коду. Робота з таблицями. Розширення аналізу описів. Розширення аналізу операторів та виразів.	10	1	-	2	-	7
Разом за змістовим модулем 2	50	7	-	14	-	29
Усього годин	90	15	-	30	-	45

5.3. Самостійна робота

№ з/п	Назва алгоритмів для самостійного вивчення (п.1) і завдань для самостійного виконання (п.2)	Кількість годин
1	1. Побудова множин FIRST і FOLLOW 2. Обчислення FOLLOW 3. Побудова праволінійної граматики за скінченим автоматом 4. Видалення лівої рекурсії 5. Ліва факторизація	25

	6. Алгоритми приведення граматики до нормальної форми Хомського **6.1. Алгоритм видалення безпосередніх лівих рекурсій за допомогою е-правил.** **6.2. Алгоритм видалення безпосередніх лівих рекурсій за допомогою розширених правил.** **6.3. Алгоритм видалення безкорисних символів і правил** **6.4. Алгоритм видалення недосяжних символів** **6.5. Алгоритм видалення е-правил** **6.6. Алгоритм усунення ланцюгових правил** 7. Нормальна форма Хомського 8. Алгоритм перетворення до нормальної форми Хомського 9. Опис програмної реалізації алгоритмів **9.1. Програмна реалізація алгоритму видалення лівої рекурсії за допомогою е-правил** **9.2. Програмна реалізація алгоритму видалення лівої рекурсії за допомогою розширених правил**	
2	1. Розширення можливостей спрощеного синтаксичного аналізатора для виконання символьних обчислень з використанням стандартних функцій. 2. Генерація проміжного коду для SPL-програми. Робота з таблицями. Розширення аналізу описів. Розширення аналізу операторів та виразів. Виконання коду. 3. Аналіз проміжних файлів роботи SPL-інтерпретатора, які утворюються для конкретної програми з лабораторної роботи №6. 4. Автоматне програмування. Конструкції while-switch-case. Перетворення ітераційних алгоритмів в автоматні. 5. Виконання домашніх завдань, які регулярно задаються під час викладання лекцій, для закріплення матеріалу і засвоєння наведених алгоритмів. 6. Оформлення лабораторних робіт. 7. Знайомство з матеріалами магістерських, дипломних і курсових робіт, виконаних на кафедрі ПМ по тематиці “Системне програмування”. 8. Робота з посібниками і додатковими матеріалами, які винесені на сайт дистанційної освіти	20
	Разом	45

5.4. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	**Графічне середовище розробки граматик ANTLRWorks**	6
2	**Побудова спрощеного лексичного аналізатора**	8
3	**Побудова спрощеного синтаксичного аналізатора**	8
4	**Елементи компіляції**	8

5.5. Система контролю та оцінювання
Види та форми контролю

Методи навчання

1. Лекційні заняття із застосуванням презентацій.
2. Лабораторні комп'ютерні заняття
3. Самостійна робота по вивченню алгоритмів
4. Розробка програм, їх відладка і тестування
5. Виконання деяких частин лабораторних робіт у робочому зошиті
6. Вивчення матеріалів, винесених на дистанційне навчання

Методи контролю

1. Перевірка засвоєних алгоритмів під час прийому лабораторних робіт
2. Перевірка розроблених студентами програм з різними вхідними даними
3. Перевірка робочих зошитів
4. Тестування

Розподіл балів, які отримують студенти

Поточне оцінювання та самостійна робота									Залік	Сума
Змістовий модуль №1				Змістовий модуль № 2					Тести	
Т1	Т2	Т3	Т4	Т1	Т2	Т3	Т4	Т5		
10	10	10	10	10	10	10	5	5	20	100

Критерії оцінювання результатів навчання з навчальної дисципліни

Сума балів за всі види навчальної діяльності	Оцінка ЄКТС	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
90 – 100	**A**	відмінно	зараховано
80-89	**B**	добре	
70-79	**C**		
60-69	**D**	задовільно	
50-59	**E**		
35-49	**FX**	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0-34	**F**	незадовільно з обов’язковим повторним вивченням дисципліни	не зараховано з обов’язковим повторним вивченням дисципліни

Захист та критерії оцінювання лабораторної роботи

- Здача лабораторної роботи 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.

7. **Рекомендована література**

Базова

1. Сопронюк Т.М. Елементи теорії компіляції: Навчальний посібник. – Чернівці: ЧНУ, 2008. – 84 с.
2. Сопронюк Т.М. Елементи теорії компіляції: Навчальний посібник. – Чернівці: ЧНУ, 2008. – 84 с.
3. 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.
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
5. P. J. Denning, J. B. Dennis, and J. E. Qualitz. Machines, Languages,and Computation. Englewood Cliffs, N.J.: Prentice Hall. 3. M., 1978.
6. M. A. Harrison. Introduction to Formal Language Theory. Reading,Mass.: Addison-Wesley, 1978.
7. W. Homenda, W. Pedrycz. Automata theory and formal languages, De Gruyter, 2022.
8. J. E. Hopcroft and J. D. Ullman. Introduction to Automata Theory, Languages and Computation. Reading, Mass.: Addison-Wesley, 1979.
9. R. Hunter. The Design and Construction of Compilers. Chichester, New York: John Wiley, 1981.
10. Z. Kohavi and N. K. Jha. Switching and Finite Automata Theory. Third Edition. New York: Cambridge University Press, 2010.
11. P. Linz. An Introduction to Formal Languages and Automata-Peter Linz Univ. of California at Davis. Jones & Bartlett Learning, 2016.
12. A. Pettorossi. Automata theory and formal languages: fundamental notions, theorems, and techniques. Springer, 2022.
13. G. E. Revesz. Introduction to Formal Languages. New York: McGrawHill, 1983.
14. С.Ю Гавриленко,А.М. Клименко, Н.Ю. Любченко. Теорія цифрових автоматів та формальних мов. Вступний курс: навчальний посібник– Харків: НТУ ХПІ, 2007. – 176 с.
15. Олег Гутік. Формальні мови та автомати. Ел. Посібник. Львів, 2022.
16. Олег Гутік. Коди та автомати: основи алгебраїчної теорії. Ел. Посібник. Львів, 2021.

8. Інформаційні ресурси

17. http://www-db.stanford.edu/~ullman/ialc.html – Introduction to Automata Theory, Languages, and Computation. Slides and Lecture Notes. (Stanford University)

18. https://ela.kpi.ua/handle/123456789/45710 – Основи проектування трансляторів. Конспект лекцій [Електронний ресурс] : навчальний посібник для здобувачів ступеня бакалавра за спеціальністю 123 Комп'ютерна інженерія / О. І. Марченко ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 2,71 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2021. – 108 с.

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

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

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

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

Додаток

Методичне забезпечення

1. Системне програмування: Методичні рекомендації та завдання для лабораторних робіт/ Укл. Т.М. Сопронюк.– Чернівці: ЧНУ, 2003. – 33 с.
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. Тестові завдання (Система Moodle)
5. Презентації лекцій (Система Moodle)
6. Відео-лекції на Google диску
7. Сертифікат про закінчення курсів професора Стенфордського університету Дж.Ульмана “Automata” (https://drive.google.com/file/d/0B-cxrxmP0J7XRUl2YlNGem1wb3c/view?resourcekey=0--NXsRlO20hZHdxuVKlDQ4Q)