

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ПРИКАРПАТСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАСИЛЯ СТЕФАНІКА



Факультет математики та інформатики
Кафедра алгебри та геометрії

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Методи розробки компіляторів

Освітня програма _____ Математика комп'ютерних технологій
Спеціальність(ості) _____ 111 — Математика
Галузь знань _____ 11 — Математика і статистика
Рівень освіти _____ Магістр

Затверджено на засіданні кафедри

Протокол №1 від 22.08.2023

ЗМІСТ

1. Загальна інформація
2. Контактна інформація
3. Опис дисципліни
4. Структура курсу
5. Система оцінювання курсу
6. Політика курсу
7. Рекомендована література

1. ЗАГАЛЬНА ІНФОРМАЦІЯ

Назва дисципліни	Методи розробки компіляторів	
Освітня програма	Математика комп'ютерних технологій	
Спеціальність	111 — Математика	
Галузь знань	11 — Математика і статистика	
Освітній рівень	магістр	
Статус дисципліни	вибіркова	
Рік підготовки / семестр	2-й / 3-й	
Обсяг дисципліни	6 кредитів	
Розподіл за видами занять	Лекції:	30
	Практичні:	
	Лабораторні:	30
	Самостійна робота:	120
Мова викладання	українська	
Посилання на сайт дистанційного на- вчання	d-learn.pnu.edu.ua	

2. КОНТАКТНА ІНФОРМАЦІЯ

Кафедра	алгебри та геометрії
Викладач(-і)	Гаврилків В.М.
Контактний теле- фон викладача	59-60-16
E-mail викладача	volodymyr.gavrylkiv@pnu.edu.ua
Консультації	Вівторок, 16 ⁰⁰

3. ОПИС КУРСУ

3.1. Анотація до навчальної дисципліни. Навчальна дисципліна “Методи розробки компіляторів” займає одне із центральних місць у системі професійної науково-предметної підготовки фахівця з математики комп’ютерних технологій. Зміст дисципліни розкривається через такі змістові модулі: “Лексичний аналіз”, “Синтаксичний аналіз”. У першому модулі вивчаються методи розробки лексичних аналізаторів мов програмування на основі регулярних виразів, праволінійних граматики, недетермінованих і детермінованих скінченних автоматів, а також програмна реалізація лексичних аналізаторів, використовуючи генератор лексичних аналізаторів Flex. У другому модулі вивчаються контекстно-вільні граматики і автомати з магазинною пам’яттю та їх використання при проектуванні синтаксичних аналізаторів мов програмування, програмна реалізація синтаксичних аналізаторів з використанням генераторів Yacc та Bison, а також методи спільного використання Flex і Yacc (Bison) для розробки компілятора мови програмування. Курс потребує базових знань з таких курсів як дискретна математика, теорія формальних мов та скінченних автоматів, інструментальні засоби аналізу та обробки текстів.

3.2. Мета та завдання навчальної дисципліни. Основною метою спецкурсу є ознайомлення студентів з методами розробки лексичних і синтаксичних аналізаторів та компіляторів мов програмування; формування компетентного спеціаліста в даній області, здатного застосовувати і розвивати основні положення дисципліни у науковій і навчальній діяльності, використовувати унікально багаті можливості регулярних та контекстно-вільних мов у різних прикладних задачах математики та комп’ютерних наук, зокрема при побудові компіляторів мов програмування.

Завданням курсу є засвоєння методів побудови лексичних аналізаторів з допомогою регулярних виразів, праволінійних граматик, недетермінованих і детермінованих скінченних автоматів; програмна реалізація лексичного аналізатора на мові програмування C++; реалізація лексичного аналізатора з використанням генератора Flex; побудова контекстно-вільних LALR-граматик для опису синтаксичної структури мови програмування; реалізація синтаксичних аналізаторів з використанням генератора Yacc (Bison).

У результаті вивчення навчальної дисципліни студент повинен знати:

- поняття формальної мови, регулярної мови та регулярного виразу;
- основні метасимволи, конструкції, квантифікатори та стандартні символні класи для запису регулярних виразів;
- суть лексичного аналізу з застосуванням регулярних виразів, детермінованих скінченних автоматів і праволінійних граматик;
- можливості і принципи роботи генератора лексичних аналізаторів Flex;
- поняття граматики, типи граматик, різні способи подання граматик;
- поняття і методи побудови автоматів з магазинною пам'яттю;
- суть синтаксичного аналізу з застосуванням контекстно-вільних граматик і автоматів з магазинною пам'яттю;
- нотацію Бекуса-Наура для опису вхідної мови;
- можливості і принципи роботи генераторів синтаксичних аналізаторів Bison і Yacc.
- методи спільного використання Flex і Yacc (Bison) для розробки компілятора мови програмування.

вміти:

- задавати класи лексем з допомогою регулярних виразів, детермінованих скінченних автоматів і праволінійних граматик;

- створити на мові програмування C++ програми для реалізації детермінованих скінченних автоматів, які розпізнають різні типи лексем вхідної мови;
- створити лексичний аналізатор нескладної мови програмування, використовуючи генератор лексичних аналізаторів Flex;
- будувати однозначні контекстно-вільні граматики для опису синтаксичної структури вхідної мови і записувати їх у нотації Бекуса-Наура;
- створити синтаксичний аналізатор, використовуючи генератор синтаксичних аналізаторів Yacc.
- розробити і реалізувати компілятор нескладної мови програмування, спільно використовуючи генератори Flex і Yacc.

4. СТРУКТУРА КУРСУ

Тематика дисципліни						
Назви змістових модулів і тем	Кількість годин					
	вс.	лек.	пр.	лаб.	інд.	сам.
Семестр 3						
Змістовий модуль 1. Лексичний аналіз.						
Тема 1. <i>Поняття мовного процесора. Типи мовних процесорів. Основні фази мовного процесора. Моделі компілятора.</i> [5, 6]	10	2				8
Тема 2. <i>Моделювання лексики формальної мови на основі регулярних виразів, недетермінованих і детермінованих скінченних автоматів та праволінійних граматик. Методи переходу від одного способу подання лексем до іншого.</i> [4, 5, 6]	10	2		2		6

Тематика дисципліни						
Назви змістових модулів і тем	Кількість годин					
	вс.	лек.	пр.	лаб.	інд.	сам.
Тема 3. Основні прийоми синтезу детермінованих скінченних автоматів для розпізнавання різних типів лексем: ідентифікаторів, вісімкох, десяткових і шістнадцяткових цілочислових констант, десяткових дійсних чисел з плаваючою крапкою в звичайній і експоненціальній формі. [4, 5, 6]	9	2		1		6
Тема 4. Програмна реалізація різних типів детермінованих скінченних автоматів для розпізнавання лексем на мові програмування C++. [3, 5, 6]	12	2		2		8
Тема 5. Моделювання та програмна реалізація на мові програмування C++ лексичного аналізатора для розпізнавання основних типів лексем, який ігнорує пропуски та виводить повідомлення про наявність у вхідному тексті помилок, які можуть виникати на етапі лексичного аналізу. [3, 5, 6]	14	2		2		10
Тема 6. Програма Flex для генерування лексичних аналізаторів. Опис структури вхідних flex-файлів. Задання лексем вхідної мови програмування з допомогою регулярних виразів у форматі flex. [3, 7, 8]	12	2		2		8
Тема 7. Розробка і реалізація лексичних аналізаторів з допомогою генератора лексичних аналізаторів Flex. [3, 7, 8]	14	2		2		10
Тема 8. Контрольна робота.	2			2		
Всього за модуль:	83	14		13		56
Змістовий модуль 2. Синтаксичний аналіз.						
Тема 9. Формальні породжувальні граматики. Типи граматики. Класифікація Хомського. [4, 9]	11	2		2		7

Тематика дисципліни						
Назви змістових модулів і тем	Кількість годин					
	вс.	лек.	пр.	лаб.	інд.	сам.
Тема 10. <i>Контекстно-вільні мови і граматики та їх зв'язок з синтаксичними аналізаторами мов програмування. Подання контекстно-вільної граматики у формі Бекуса-Наура. [4, 9]</i>	11	2		2		7
Тема 11. <i>Дерева синтаксичного розбору. Однозначні і неоднозначні граматики. Граматичні розбори зверху вниз і знизу вгору. [4, 5, 9]</i>	11	2		2		7
Тема 12. <i>Автомати з магазинною пам'яттю і контекстно-вільні мови. Синтез МП-автоматів. Приклад формальної мови, яка не є контекстно-вільною. [4, 5, 9]</i>	12	2		2		8
Тема 13. <i>Синтаксичний аналізатор як автомат з магазинною пам'яттю. [4, 5, 7]</i>	10	2		1		7
Тема 14. <i>Генератори синтаксичних аналізаторів Yacc та Bison. Опис структури входних yacc- та bison-файлів. Можливості та принципи роботи генераторів. [5, 7, 8]</i>	12	2		2		8
Тема 15. <i>Побудова контекстно-вільних LALR-граматик для застосування у програмі Yacc. Приклади розробки синтаксичних аналізаторів з допомогою генераторів синтаксичних аналізаторів Yacc та Bison. [4, 7, 8]</i>	14	2		2		10
Тема 16. <i>Методи спільного використання Flex і Yacc (Bison) для розробки компілятора нескладної мови програмування на підставі її неформального опису. [5, 7, 8]</i>	14	2		2		10
Тема 17. <i>Контрольна робота.</i>	2			2		
Всього за модуль:	97	16		17		64
Всього за семестр:	180	30		30		120
Усього годин:	180	30		30		120

5. СИСТЕМА ОЦІНЮВАННЯ ДИСЦИПЛІНИ

Володіння матеріалом дисципліни студенти виявляють при написанні аудиторної контрольної роботи та домашньої контрольної роботи. Максимальна оцінка за кожну з 2-х контрольних робіт становить 40 балів. Домашня контрольна робота передбачає усний захист, включно з формулюваннями використаних означень і фактів.

За активну і змістовну участь у розв'язуванні завдань на лабораторних заняттях здобувач може набрати щонайбільше 20 балів.

Сума балів за семестр визначає підсумкову оцінку згідно поданої нижче таблиці.

Сума балів за всі види навчальної діяльності протягом семестру	Оцінка ЄКТС	Оцінка за національною шкалою
90 – 100	A	відмінно
80 – 89	B	добре
70 – 79	C	добре
60 – 69	D	задовільно
50 – 59	E	достатньо
1 – 49	FX	незадовільно

6. ПОЛІТИКА КУРСУ

Самостійне виконання завдань поточного та підсумкового контролю результатів навчання здійснюється під керівництвом викладача, який веде заняття, із наступним їх захистом (для осіб з особливими освітніми потребами ця вимога застосовується з урахуванням їхніх індивідуальних потреб і можливостей). Важливим є надання достовірної інформації про результати власної навчальної (наукової, творчої) діяльності та посилення на джерела інформації у разі використання ідей, відомостей, розробок. Плагіат та інші

види академічної недоброчесності не принесуть позитивного результату, а тому не рекомендуються.

Відвідування занять є обов'язковим. Засвоєння пропущеної теми лекції з поважної причини перевіряється під час складання підсумкового контролю. Пропуск лекції з неповажної причини відпрацьовується студентом відповідно вимог кафедри (співбесіда, реферат тощо). Пропущені лабораторні заняття, незалежно від причини пропуску, студент відпрацьовує згідно з графіком консультацій. Поточні “незадовільно”, отримані студентом під час засвоєння відповідної теми на лабораторному занятті, перескладаються викладачеві до складання підсумкового контролю з обов'язковою відміткою у журналі обліку роботи академічних груп.

7. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Белов Ю.А. Інструментальні засоби програмування: навчальний посібник / Ю.А. Белов, В.С. Проценко, П.Й. Чаленко. – К.: Либідь, 1993. – 248 с.
2. Бондаренко М.Ф. Комп'ютерна дискретна математика: підручник / М.Ф. Бондаренко, Н.В. Білоус, А.Г. Руткас. – Харків: «Компанія СМІТ», 2004. – 480 с.
3. Гаврилків В.М. Регулярні вирази у програмних продуктах: навчальний посібник / В.М. Гаврилків. – Івано-Франківськ: Голіней О.М., 2012. – 72 с.
4. Гаврилків В.М. Формальні мови та алгоритмічні моделі: навчальний посібник (вид. друге) / В.М. Гаврилків. – Івано-Франківськ, 2023. – 180 с.
5. Гаврилків В.М. Практичні методи розробки компіляторів: навчальний посібник / В.М. Гаврилків. – Івано-Франківськ, 2023. – 71 с.
6. Сопронюк Т.М. Системне програмування. Частина II. Елементи теорії компіляції: Навчальний посібник у двох частинах / Т.М. Сопронюк. – Чернівці: ЧНУ, 2008. – 84 с.

7. Brown D. *lex and yacc*, 2nd Edition / Doug Brown, John Levine, Tony Mason. – O'Reilly Media, 2012. – 388 p.
8. Levine J.R. *Flex and bison* / J.R. Levine. — O'Reilly Media, Inc., 2009. – 274 p.

Додаткова література

9. Нікольський Ю.В. *Дискретна математика* / Ю.В. Нікольський, В.В. Пасічник, Ю.М. Щербина. – К.: Видавнича група BHV, 2007. – 368 с.
10. Vambenek J. *grep Pocket Reference* / John Vambenek and Agnieszka Klus. – O'Reilly Media, 2009 – 75 p.
11. Goyvaerts J. *Regular Expressions Cookbook* / Jan Goyvaerts and Steven Levithan. – O'Reilly Media, 2009 – 494 p.

Викладач

Гаврилків В.М.