

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Львівський національний університет імені Івана Франка
Факультет прикладної математики та інформатики
Кафедра інформаційних систем

Затверджено

на засіданні кафедри інформаційних систем
факультету прикладної математики та інформатики
Львівського національного університету імені Івана Франка
(протокол №__ від __ 2025 р.)

В.о. завідувача кафедри Горлач В.М..

Силабус з навчальної дисципліни
“Розробка компіляторів для предметно-орієнтованих мов”,
що викладається в межах ОПШ «Інформатика»
першого (бакалаврського) рівня вищої освіти для здобувачів з
спеціальності F3 – Комп’ютерні науки

Львів 2025 р.

Назва дисципліни	Розробка компіляторів для предметно-орієнтованих мов
Адреса викладання дисципліни	Головний корпус ЛНУ ім. І. Франка м. Львів, вул. Університетська 1
Факультет та кафедра, за якою закріплена дисципліна	Факультет прикладної математики та інформатики
Галузь знань, шифр та назва спеціальності	F – інформаційні технології F3 – комп'ютерні науки
Викладачі дисципліни	Жолткевич Григорій Миколайович, професор кафедри інформаційних систем, доктор техн. наук. (лекції та лабораторні заняття) Василишин Б.Б., асистент кафедри інформаційних систем (лабораторні заняття)
Контактна інформація викладачів	grygoriy.zholtkevych@lnu.edu.ua ; Головний корпус ЛНУ ім. І. Франка, каб. 260. м. Львів, вул. Університетська, 1
Консультації з питань навчання по дисципліні відбуваються	Консультації в день проведення лекцій/лабораторних занять (а також за розкладом консультацій кафедри).
Інформація про дисципліну	Дисципліна “Розробка компіляторів для предметно-орієнтованих мов” є вибірковою дисципліною для спеціальності F3 – комп'ютерні науки для освітньої програми Інформатика, яка викладається в X-му семестрі в обсязі X-ти кредитів (за Європейською Кредитно-Трансферною Системою ECTS).
Мета та цілі дисципліни	<i>Метою навчальної дисципліни є опанування у студентами теоретичних знань та практичних навичок у сфері розробки мов програмування та компіляторів, з особливим акцентом на предметно-орієнтовані мови (DSL).</i> <i>Завданнями</i> <ul style="list-style-type: none"> ● опанування алгоритмічних основ обробки текстів; ● вивчення основ теорії формальних мов та граматики; ● формування навичок роботи з інструментарієм розробки мов на прикладі textX; ● вивчення методів реалізації семантичних обмежень та трансляції; ● створення компілятора через виконання навчального проекту.
Коротка анотація дисципліни	Навчальна дисципліна присвячена комплексному вивченню теоретичних засад та практичних методів створення сучасних мовних процесорів, з особливим акцентом на розробку предметно-орієнтованих мов (DSL). Програма курсу структурована таким чином, щоб провести здобувача через усі етапи проектування компілятора: від базових алгоритмів обробки рядків до генерації цільового коду та візуалізації моделей. Початковий етап навчання охоплює фундаментальні алгоритми пошуку, що є критично важливими для лексичного аналізу. Сюди входить вивчення наївного пошуку підрядків, а також високоефективних методів, таких як алгоритм Кнута-Морісона-Прата та багатошаблонний алгоритм Ахо-Корасик. Ці знання закладають підґрунтя для переходу до більш складних структур даних та методів обробки текстової інформації.

Центральна частина курсу присвячена теорії формальних мов і граматики. Студенти опановують роботу з регулярними виразами та генеративними граматиками, вчать розрізняти контекстно-вільні граматики та будувати синтаксичні дерева. Особлива увага приділяється методу лівого рекурсивного спуску та граматикам виразів розбору (PEG), які є основою для сучасних інструментів синтаксичного аналізу. Практична реалізація цих концепцій здійснюється за допомогою інструменту **textX**, що дозволяє створювати метамоделі мов та описувати їхню структуру на високому рівні абстракції.

Програма також детально розглядає питання семантичного аналізу та трансляції. Курс передбачає вивчення технік навігації за моделями, реалізацію семантичних обмежень та використання об'єктних процесорів. Важливою складовою є вивчення шаблону проектування **Visitor**, який застосовується для обробки абстрактних синтаксичних дерев, що дозволяє виконувати задачі інтерпретації, візуалізації та безпосередньої трансляції моделей.

Логічним завершенням курсу є розробка та захист індивідуального міні-проєкту, де студенти демонструють навички проектування повноцінного компілятора. Це включає специфікацію власної мови, реалізацію її семантики та механізмів трансляції. Загальний обсяг дисципліни становить **120 годин**, розподілених на 16 лекційних годин, 32 години лабораторних занять та 72 години самостійної роботи протягом 16 навчальних тижнів. Даний курс забезпечує підготовку фахівців, здатних створювати спеціалізовані мовні інструменти для автоматизації розробки програмного забезпечення.

Література для вивчення дисципліни

Основна література:

1. **A. Aho, M. Lam, R. Sethi, J. Ullman.** *Compilers: Principles, Techniques, and Tools*, 2nd ed. 2007.
2. **F. Tomassetti.** *The complete guide to (external) Domain Specific Languages*. 2017.
3. **I. Dejanović et al.** *textX*. <https://github.com/textX/>
4. **Thomas H. Cormen et al.** *Introduction to Algorithms*.

Додаткова література:

5. **S. Kelly, Juha-Pekka Tolvanen.** *Domain-specific modeling: enabling full code generation*. 2008.
6. **K. Cooper, L. Torczon.** *Engineering a Compiler* 2nd ed. 2012.

Обсяг курсу

Загальний обсяг: 120 годин. Аудиторних занять: 48 год., з них 16 години лекцій та 32 години лабораторних занять. Самостійної роботи: 120 годин.

Очікувані результати навчання

У результаті вивчення навчальної дисципліни студент повинен

знати:

- Граматики виразів розбору (PEG) та їх відмінності від класичних підходів.
- Принципи обробки текстів шляхом застосування наївного алгоритму пошуку підрядка, алгоритму Кнута-Морісона-Прата та алгоритму Ахо-Корасик для пошуку рядків в тексті.
- Призначення та структуру абстрактних синтаксичних дерев (AST).
- Поняття неоднозначності граматики та методи її усунення.
- Метамодель мови в textX та принципи її побудови.
- Техніку лексичного та синтаксичного аналізу засобами спеціалізованих бібліотек.
- Способи навігації за моделлю, роботу з 'процесорами моделей та об'єктів.
- Методи специфікації та реалізації семантичних обмежень.
- Шаблон проектування **Visitor** та його роль у обробці синтаксичних дерев.
- Методи візуалізації, інтерпретації та трансляції моделей.
- Принципи реалізації цільової моделі

вміти:

- Використовувати алгоритми пошуку та обробки рядків.
- Проектувати граматики та провадити синтаксичний аналіз.
- Розробляти та налаштовувати мовні процесори у textX.
- Застосовувати шаблон проектування Visitor для обробки та трансформації абстрактних синтаксичних дерев.
- Реалізовувати задачі візуалізації, інтерпретації та трансляції розроблених моделей у цільові формати.

Курс забезпечує набуття таких компетентностей: ІК, ЗК 1, ЗК 2, ЗК 6, СК 3, СКФ 7, СК 12 та програмних результатів навчання: ПРН 1-5.

Ключові слова

мови програмування, предметно-орієнтовані мови (DSL), компіляція, фаза компіляції, метамодель, регулярний вираз, формальна мова, контекстно-вільна граматики, граматики виразів розбору (PEG), синтаксичне дерево, лівий рекурсивний спуск, лексичний аналіз, синтаксичний аналіз, семантичне обмеження, трансляція, інтерпретація

Формат курсу

Очний, дистанційний.

Проведення лекцій, лабораторних робіт і консультацій.

Теми

Тема 1. Мови в програмуванні

Тема 2. Задача пошуку підрядка в рядку

Тема 3. Предметно-орієнтовані мови та їх компіляція: загальні концепції

Тема 4. Регулярні вирази, формальні мови та граматики

Тема 5. Лексичний та синтаксичний аналіз в textX

Тема 6. Специфікація та обробка семантичних обмежень в textX

Тема 7. Шаблон Visitor. Трансляція

Тема 8. Міні-проект з розробки компілятора

Підсумковий контроль, форма

Залік у кінці семестру викладання.

Пререквізити

Для вивчення дисципліни студенти потребують базові знання, що формуються при підготовці рівня бакалавр з математичної логіки, дискретної математики, основ функціонального програмування, теорії алгоритмів.

Навчальні методи та техніки, які будуть використовувати ся під час викладання курсу

Презентації, лекції, модульний контроль

Індивідуальні завдання

Індивідуальні мініпроекти

Необхідне обладнання

Для проведення занять використовується програмне забезпечення Python та textX, доступне на сайтах <http://python.org> та <https://textx.github.io/textX/>, яке використовується для будь-якої програмної платформи (Microsoft, MacOS, Linux).

Критерії оцінювання (окремо для кожного виду навчальної діяльності)

Оцінювання проводиться за 100-бальною шкалою. Підсумкова максимальна кількість балів 100.

Оцінка за шкалою ECTS		Оцінка в балах
A	Відмінно	90 - 100
B	Дуже добре	81 - 89
C	Добре	71 - 80
D	Задовільно	61 - 70
E	Достатньо	51 - 60
FX(F)	Незадовільно	0 - 50

Поточне оцінювання: впродовж семестру студент може отримати 100 балів. З них за виконання лабораторних робіт – 90 балів (5 лабораторних робіт – по 18 б). Для кожного завдання встановлено терміни здачі. Роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку (кожне практичне заняття: 1-14б.).

Додатково може бути нараховано 10 б за активність на заняттях.

Академічна доброчесність: Очікується, що роботи студентів будуть їх самостійними дослідженнями чи міркуваннями. Відсутність посилань на використані джерела, фабрикування джерел, списування, втручання в роботу інших студентів становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в письмовій роботі студента є підставою для її незарахування викладачем, незалежно від масштабів плагіату чи обману.

Відвідування занять є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції та лабораторні заняття курсу. Студенти повинні інформувати викладача про неможливість відвідати заняття. У будь-якому випадку студенти зобов'язані дотримуватися термінів, визначених для виконання всіх лабораторних робіт та індивідуальних завдань, передбачених курсом.

Література. Уся література, яку студенти не зможуть знайти самостійно, буде надана викладачем виключно в освітніх цілях без права її передачі третім особам. Студенти заохочуються до використання також й іншої літератури та джерел, яких немає серед

рекомендованих.

Політика виставлення балів. Враховуються бали, отримані при поточному опитуванні, самостійній роботі та бали підсумкового тестування. При цьому обов'язково враховуються присутність на заняттях та активність студента під час лабораторного заняття; недопустимість пропусків та запізнь на заняття; користування мобільним телефоном, планшетом чи іншими мобільними пристроями під час заняття в цілях, не пов'язаних з навчанням; списування та плагіат; несвоєчасне виконання поставленого завдання і т. ін.

Жодні форми порушення академічної доброчесності не толеруються.

Екзамен не передбачений

Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенні курсу.

Питання до екзамену Опитування

Схема курсу

Тиж.	Тема, план, когроткі тези	Форма діяльності (заняття)	Література, ресурси в Інтернеті	Завдання, год.	Термін виконання
1	Тема 1. Мови в програмуванні: роль мов у програмуванні, їх класифікація, особливості мов різних класів)	лекція			
	Тема 2. Задача пошуку підрядка в рядку: наївний алгоритм та алгоритм Кнута-Морісона- Прата пошуку підрядка в рядку та їх програмна реалізація	лаб. робота самостійна робота		4	тиждень 2
2	Тема 2. Задача пошуку підрядка в рядку: алгоритм Ахо-Корасик пошуку підрядків в рядку та його програмна реалізація	лаб. робота самостійна робота		4	тиждень 3
3	Тема 3. Пред-метно-орієнтовані мови та їх компіляція: загальні концепції: задача компіляції, її фази, структура компілятора та робочі формати даних, приклад розробки простого компілятора з використанням textX	лекція		4	тиждень 4
	Тема 4. Регулярні вирази, формальні			4	тиждень 4

	мови та граматики: поняття про регулярні вирази та рядки, що їм відповідають	лаб. робота самостійна робота			
4	генеративні граматики, контекстно-вільні граматики, синтаксичні дерева, неоднозначність граматик; синтез регулярних виразів за зразками	лекція лаб. робота самостійна робота		4	тиждень 5
5	Побудова синтаксичного дерева	лаб. робота самостійна робота		4	тиждень 6
6	Побудова синтаксичного дерева лівим рекурсивним спуском.	лаб. робота самостійна робота		4	тиждень 7
7	Граматики виразів розбору (PEG), метамодель мови в textX; приклад побудови метамodelей в textX	лекція лаб. робота самостійна робота		4	тиждень 8
8	Тема 5. Лексичний та синтаксичний аналіз в textX: техніка побудови метамodelей та моделей в textX	лаб. робота самостійна робота		4	тиждень 9
9	Тема 6. Специфікація та обробка семантичних обмежень в textX: навігація за textX-моделлю; процесори моделей і об'єктів	лекція лаб. робота самостійна робота		4	тиждень 10
10	прикладні реалізації семантичних обмежень в textX	лаб. робота самостійна робота		4	тиждень 11
11	Тема 7. Шаблон Visitor. Трансляція: шаблон проектування Visitor, його застосування для обробки абстрактних синтаксичних дерев - візуалізація, інтерпретація, трансляція - засобами textX; задача візуалізації textX-моделі	лекція лаб. робота самостійна робота		4	тиждень 12

12	задачі інтерпретації та трансляції textX-моделі	лаб. робота самостійна робота		4	тиждень 13
13	Тема 8. Міні-проект з розробки компілятора: розробка та реалізація цільової моделі міні-проєкту;	лекція лаб. робота самостійна робота		6	тиждень 14
14	визначення та реалізація семантичних обмежень моделі	лаб. робота самостійна робота		6	тиждень 15
15	постановка задач трансляції, їх реалізація	лаб. робота самостійна робота		6	тиждень 16
16	завершення розробки та захист міні-проектів	лаб. робота самостійна робота		6	тиждень 16