



ПОБУДОВА КОМПІЛЯТОРІВ

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	другий (магістерський)
Галузь знань	12 Інформаційні технології
Спеціальність	121 Інженерія програмного забезпечення
Освітня програма	Інженерія програмного забезпечення інформаційних систем
Статус дисципліни	Нормативна
Форма навчання	заочна
Рік підготовки, семестр	1-й курс, осінній семестр
Обсяг дисципліни	6 кредитів, 180 годин (10 годин — Лекції, 10 годин — Лабораторні, 160 годин - СРС)
Семестровий контроль/ контрольні заходи	Екзамен
Розклад занять	<i>перший семестр</i>
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектор: к.т.н., доцент, Статівка Ю. І., statyvka-yu@lil.kpi.ua Комп'ютерний практикум: к.т.н., доцент, Статівка Ю. І., statyvka-yu@lil.kpi.ua
Розміщення курсу	https://classroom.google.com/c/NzE0MTEyNDAyMjI0 https://ecampus.kpi.ua

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

В курсі розглядаються як фундаментальні принципи створення і імплементації мов програмування загального призначення, так і практичні аспекти автоматизації розробки компіляторів з використанням новітніх технологій та інструментів.

Значна увага приділяється проектуванню мов програмування, їх специфікації, математичним основам представлення та опрацювання формальних мов та їх імплементації.

Розглядаються загальні та спеціалізовані засоби побудови компіляторів та інтерпретаторів: алгоритми лексичного, синтаксичного та семантичного аналізу, генерування коду (низького, проміжного чи високого рівня), засобів виконання коду цільової мови, такі засоби автоматизації, зокрема, як ANTLR4 тощо.

Метою дисципліни є опанування студентами теоретичних знань та набуття практичного досвіду проєктування і імплементації мов програмування .

Предмет дисципліни — методи та алгоритми побудови, специфікації та імплементації мов програмування високого рівня.

Завдання. В результаті вивчення дисципліни у студентів повинні сформуватися наступні компетентності:

загальні:

- ЗК 01 - Здатність до абстрактного мислення, аналізу та синтезу.

фахові:

- ФК 2 - Здатність розробляти і реалізовувати наукові та/або прикладні проєкти у сфері інженерії програмного забезпечення.
- ФК 4 - Здатність розвивати і реалізовувати нові конкурентоспроможні ідеї в інженерії програмного забезпечення.
- ФК 5 - Здатність розробляти, аналізувати та застосовувати специфікації, стандарти, правила і рекомендації в сфері інженерії програмного забезпечення.
- ФК 7 - Здатність критично осмислювати проблеми у галузі інформаційних технологій та на межі галузей знань, інтегрувати відповідні знання та розв'язувати складні задачі у широких або мультидисциплінарних контекстах.
- ФК 12 - Здатність до використання базових ідей та методів для створення сучасних компіляторів

Після засвоєння навчальної дисципліни студенти мають продемонструвати такі програмні результати навчання:

- ПРН 2 - Оцінювати і вибирати ефективні методи і моделі розроблення, впровадження, супроводу програмного забезпечення та управління відповідними процесами на всіх етапах життєвого циклу.
- ПРН 6 - Розробляти і оцінювати стратегії проєктування програмних засобів; обґрунтовувати, аналізувати і оцінювати варіанти проєктних рішень з точки зору якості кінцевого програмного продукту, ресурсних обмежень та інших факторів.
- ПРН 7 - Аналізувати, оцінювати і застосовувати на системному рівні сучасні програмні та апаратні платформи для розв'язання складних задач інженерії програмного забезпечення.
- ПРН 9 - Обґрунтовано вибирати парадигми і мови програмування для розроблення програмного забезпечення; застосовувати на практиці сучасні засоби розроблення програмного забезпечення.
- ПРН 17 - Збирати, аналізувати, оцінювати необхідну для розв'язання наукових і прикладних задач інформацію, використовуючи науково-технічну літературу, бази даних та інші джерела.
- ПРН 18 - Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Пререквізити дисципліни. Знання, отримані при вивченні дисциплін: «Основи програмування» «Об'єктно-орієнтований аналіз та конструювання програмних систем», «Комп'ютерна дискретна математика», «Алгоритми та структури даних».

Постреквізити дисципліни. Компетенції, отримані студентами в процесі вивчення дисципліни «Побудова компіляторів» можуть бути використані при виконанні дипломної роботи.

3. Зміст навчальної дисципліни

Розділ 1. Основи теорії формальних мов

- Тема 1.1. Вступ. Формальні мови. Компіляція. Фази компіляції.
- Тема 1.2. Елементи теорії формальних мов.
- Тема 1.3. Скінченні автомати.

Розділ 2. Лексичний, синтаксичний та семантичний аналіз програм

- Тема 2.1. Лексичний аналіз програм.
- Тема 2.2. Автомат з магазинною пам'яттю (МП-автомат).
- Тема 2.3. Призначення та види синтаксичного аналізу. Метод рекурсивного спуску.
- Тема 2.4. Семантичний аналіз програм.
- Тема 2.5 Приклад реалізації семантичного аналізатора.

Розділ 3. Трансляція та виконання програм

- Тема 3.1. Проміжні форми подання програм.
- Тема 3.2. Трансляція у ПОЛІЗ і виконання ПОЛІЗ-програм.
- Тема 3.3. Віртуальні стекові машини. CLR. JVM.
- Тема 3.4. Переклад на CIL та Java-асемблер.

Розділ 4. Автоматизація розробки трансляторів.

- Тема 4.1. Засоби автоматизації розробки трансляторів
- Тема 4.2. ANTLR4 як компілятор компіляторів.
- Тема 4.3. Варіанти використання ANTLR4.

Розділ 5. Окремі питання розробки трансляторів.

- Тема 5.1. Генерування коду для регістрових машин.
- Тема 5.2. Оптимізація цільового коду.
- Тема 5.3. Розробка трансляторів. Підсумки.

4. Навчальні матеріали та ресурси

Основна література

1. Стативка Ю.І. Формальні мови: Основні концепти і представлення [Текст]: навч. посіб. / Ю. І. Стативка. – Київ: КПІ ім. Ігоря Сікорського, 2023. – 87 с.
2. Медведєва В.М. Транслятори: лексичний та синтаксичний аналізатори [Текст] : навч. посіб. / В.М. Медведєва, В.А. Третяк. – К. : НТУУ «КПІ», 2012. – 148с.
3. Медведєва В.М. Транслятори: внутрішнє подання програм та інтерпретація [Текст] : навч. посіб. / В.М. Медведєва, В.А. Третяк. . – К. : Текст, 2015. – 144 с.
4. Aho, Alfred, Lam, Monica, Sethi, Ravi, Ullman, Jeffrey Compilers: Principles, Techniques, and Tools, 2nd edition. - Addison Wesley, 2006. - 1040 p.

Додаткова література

5. Hopcroft J. E., Motwani R., Ullman J. D. Introduction to Automata Theory, Languages, and Computation (3rd ed.). Pearson, 2013. 560 p.
6. A. V. Aho and J. D. Ullman, The Theory of Parsing, Translation, and Compiling, Vol. 1, Parsing. Prentice Hall, 1972. 1030 p.
7. Winskel G. The formal semantics of programming languages: an introduction. Cambridge, Massachusetts, London: MIT Press, 1993. 384 p.

Інформаційні ресурси

8. <https://www.antlr.org/>

9. <https://github.com/antlr/antlr4/blob/master/doc/getting-started.md>
10. <https://github.com/antlr/antlr4/blob/4.6/doc/index.md>
11. EBNF Visualizer: веб-сайт. URL: <http://dotnet.jku.at/applications/visualizer/>
12. EBNF 2 RailRoad: веб-сайт. URL: <https://github.com/matthijsgroen/ebnf2railroad>
13. Railroad Diagram Generator: веб-сайт. URL: <https://bottlecaps.de/rr/ui>
14. JFLAP Version 7.1 RELEASED July 27, 2018: веб-сайт. URL: <https://www.jflap.org/>
15. Rodger S. H., Finley T. W. JFLAP – An Interactive Formal Languages and Automata Package Jones & Bartlett Publishers, Sudbury, 2005. 212 p. – URL: <https://www2.cs.duke.edu/csed/jflap/jflapbook/jflapbook2006.pdf>
16. <https://learn.microsoft.com/en-us/dotnet/framework/tools/ilasm-exe-il-assembler>
17. <https://learn.microsoft.com/en-us/dotnet/framework/tools/ildasm-exe-il-disassembler>
18. <https://github.com/drstrng/Krakatau-noff>

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Лекційні заняття

Розділ 1. Основи теорії формальних мов

Лекція 1. Вступ. Формальні мови. Компіляція. Фази компіляції.

Місце та призначення компілятора в програмному забезпеченні комп'ютера. Основні блоки та їх взаємодія. Структура компілятора.

Лекція 2. Елементи теорії формальних мов.

Основні поняття: мова, граматики, виведення, сентенційна форма, рекурсивні правила.

Приведення граматики. Трансформація граматики.

Форми подання граматики мов програмування: БНФ, РБНФ. Синтаксичне дерево і абстрактне синтаксичне дерево. Синтаксичні діаграми.

Скінченні автомати. Поняття розпізнавача. Види розпізнавачів. Скінченні автомати. Визначення, правила функціонування, форми подання. Детерміновані скінченні автомати (ДСА). Недетерміновані скінченні автомати (НСА).

Розділ 2. Лексичний, синтаксичний та семантичний аналіз програм

Лекція 3. Лексичний, синтаксичний та семантичний аналіз програм

Лексичний аналізатор. Призначення, вхід, вихід. Діаграма станів. Лексичний аналіз програм на основі діаграми станів (переходів).

Призначення, види синтаксичного розбору. Низхідний розбір. Алгоритм рекурсивного спуску.

Семантика мови і поведінка програми. Аналіз відповідності програми семантиці, декларованій у специфікації мови.

Розділ 3. Трансляція та виконання програм

Лекція 4. Проміжні форми подання програм і генерування коду.

Призначення проміжних форм, їх види. Польський інверсний запис. Алгоритм Дейкстри побудови ПОЛІЗ. Пріоритети. Віртуальна стекова машина.

Розділ 4. Автоматизація розробки трансляторів

Лекція 5. Засоби автоматизації розробки трансляторів

Генератор парсерів: Yacc, Bison, Lex, Flex, Harpu, BNFC, ANTLR4.

ANTLR4: Структура, встановлення, використання. Засоби проектування та часу виконання.

Нотація. Приклад використання.

Лабораторні заняття

Лабораторне заняття 1. Елементи теорії формальних мов.

Формальні мови. Ланцюжки, операції над ними. Формальні граматики. Виведення. Побудова граматики за описом мови. Побудова опису мови за граматиною.

Лабораторне заняття 2. Специфікація формальних мов.

Структура специфікації — лексика, синтаксис, семантика. Ревю специфікацій.

Лабораторне заняття 3. Скінченні автомати і лексичний аналіз

Скінченні автомати. Діаграма станів. Лексичний аналіз.

Лабораторне заняття 4. Синтаксичний і семантичний аналіз

Метод рекурсивного спуску. Реалізація. Семантика мови і поведінка програми.

Лабораторне заняття 5. Генерування коду і виконання програм.

Генерування postfix- та il-коду. Postfix Stack Machine. CLR.

5.1 Тематика робіт комп'ютерного практикуму/лабораторних робіт

1. Специфікація мови програмування.
2. Лексичний аналіз програм.
3. Синтаксичний та семантичний аналіз програм.
4. Трансляція у ПОЛІЗ та виконання програм.
5. Трансляція для віртуальної стекової машини (CLR/JVM).
6. Автоматизація побудови компіляторів.

6. Самостійна робота студента

№ з/п	Розділ	Тема	Джерела	Контроль
1	1	Однопрохідний компілятор.	[2, розд. 1.2]	на практ. заняттях
2	1	Робота з граматиною у JFLAP.	[1, розд. 1.5; 15, розд. 3]	на практ. заняттях
3	1	Класифікація граматик Чомські.	[1, розд. 1.3.4; 2, розд. 2.1.2]	на практ. заняттях
4	1	Усунення лівої рекурсії.	[1, розд. 1.3.5]	на МКР
5	1	Ebnf-visualizer (чи аналогічні інструменти) для генерування синтаксичних діаграм.	[1, розд. 1.2.4]	на практ. заняттях, РГР

6	1	Робота з регулярними виразами у JFLAP.	[15, розд. 4]	на практич. заняттях
7	1	Робота з скінченними автоматами у JFLAP.	[15, розд. 4]	на практич. заняттях, РКП 1, МКР
8	2	Лексичний аналіз методом розбору до роздільника.	[1, розд. 3.2]	на практич. заняттях,
9	2	Робота з МП-автоматами у JFLAP.	[1, розд. 1.3.4; 15, розд. 5]	на практич. заняттях, МКР
10	2	LL(k) і LR(k)– аналіз.	[4, розд. 4.4.3, 4.6.1]	на практич. заняттях, на екзамені
11	3	Використання ILASM і ILDASM.	[16; 17]	на практич. заняттях, екзамен, РКП 5
12	3	Використання java-асемблерів третіх сторін: Krakatau.	[18]	на практич. заняттях, РКП 5

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Система вимог при вивченні дисципліни:

1. правила відвідування занять: заборонено оцінювати присутність або відсутність здобувача на аудиторному занятті, в тому числі нараховувати заохочувальні або штрафні бали. Відповідно до РСО даної дисципліни бали нараховують за відповідні види навчальної активності на лекційних та практичних заняттях.
2. правила поведінки на заняттях: студент має можливість отримувати бали за відповідні види навчальної активності на лекційних та практичних заняттях, передбачені РСО дисципліни. Використання засобів зв'язку для пошуку інформації на гугл-диску викладача, в інтернеті, в дистанційному курсі на платформі Сікорський здійснюється за умови вказівки викладача;
3. політика дедлайнів та перескладань: якщо студент не проходив або не з'явився на МКР (без поважної причини), його результат оцінюється у 0 балів. Перескладання результатів МКР не передбачено;
4. політика щодо академічної доброчесності: Кодекс честі Національного технічного університету України «Київський політехнічний інститут» <https://kpi.ua/files/honorcode.pdf> встановлює загальні моральні принципи, правила етичної поведінки осіб та передбачає політику академічної доброчесності для осіб, що працюють і навчаються в університеті, якими вони мають керуватись у своїй діяльності, в тому числі при вивченні та складанні контрольних заходів з дисципліни «Побудова компіляторів»;
5. при використанні цифрових засобів зв'язку з викладачем (мобільний зв'язок, електронна пошта, переписка на форумах та у соцмережах тощо) необхідно дотримуватись загальноприйнятих етичних норм, зокрема бути ввічливим та обмежувати спілкування робочим часом викладача.

8. Види контролю та рейтингова система оцінювання результатів навчання (РСО)

Система рейтингових (вагових) балів та критерії оцінювання

Поточний контроль: активність — питання і відповіді на лекційних заняттях, питання і відповіді та участь у ревію на лабораторних заняттях; вправи на лабораторних заняттях;

МКР; виконання завдань до лабораторних занять; виконання та представлення робіт комп'ютерного практикуму.

1) Активність

Активність (частота, змістовність) участі студента у процесі обговорення відповідних тем на заняттях — питання і відповіді на лекційних заняттях, питання і відповіді та участь у ревію на практичних заняттях, виконання вправ на практичних заняттях оцінюється, максимум, 10 балами, які може отримати кожен студент за семестр.

2) Роботи комп'ютерного практикуму

Максимальна кількість балів за виконання кожної роботи — 5 балів, за захист — 5 балів. Отже виконання та захист однієї роботи може становити 10 балів.

Тоді максимальна кількість балів за усі виконані роботи комп'ютерного практикуму становить 60 балів.

Критерії оцінювання:

Виконання та захист робіт комп'ютерного практикуму:

- правильне функціонування розробленого модуля є необхідною умовою зарахування виконаної роботи;
- якісно оформлений код та звіт;
- виконана і представлена частково (не представлена) — відповідна частка від максимальної кількості балів;
- захист передбачає:
 - коротку змістовну доповідь (до 3 хвилин);
 - змістовні відповіді на питання чи зауваження;
 - вміння вносити зміни до коду (необхідні або для розгляду альтернативних варіантів).

3) Модульна контрольна робота

Максимальна кількість балів за модульну контрольну роботу дорівнює 10 балів.

Якість виконання роботи:

- виконана у повному обсязі з необхідними текстовими поясненнями дій та результатів – максимальна кількість балів;
- виконана частково з поясненнями — відповідна частка від максимальної кількості балів;
- виконана без текстових пояснень дій та результатів – не більше 5 балів;

4) Складання іспиту

Максимальний ваговий бал $r_{\text{ісп}}=20$

На іспиті студент виконує письмово роботу, яка містить два теоретичних питання. Теоретичні питання оцінюються максимально по 10 балів.

Умови позитивної проміжної атестації

Для отримання „зараховано” з першої проміжної атестації студент матиме не менше ніж 10 балів.

Для отримання „зараховано” з другої проміжної атестації студент матиме не менше ніж 30 бали.

Умови допуску до іспиту

Необхідною умовою допуску до іспиту є виконання чотирьох лабораторних робіт з п'яти перших (тобто з робіт ЛР1-ЛР5), а також стартовий рейтинг (R_c) не менше 40 балів.

Розрахунок шкали (R) рейтингу:

Сума вагових балів контрольних заходів протягом семестру (шкала рейтингу) складає:

$$R = r_{\text{акт}} + r_{\text{практ}} + r_{\text{мкр}} + r_{\text{ісп}} = 10 + 60 + 10 + 20 = 100 \text{ балів.}$$

Максимальний стартовий рейтинг становить $R_c = r_{\text{акт}} + r_{\text{практ}} + r_{\text{тв}} + r_{\text{мод}} = 80$ балів.

Рейтинг іспиту дорівнює 20 балів. Мінімальний рейтинг допуску до іспиту становить 40 балів.

Таким чином, рейтингова шкала з кредитного модуля складає

$$R = 80 + 20 = 100 \text{ балів.}$$

Для отримання студентом відповідних оцінок рейтингова оцінка студента переводиться згідно таблиці:

Бали	Оцінка
95 - 100	Відмінно
85 - 94	Дуже добре
75 - 84	Добре
65 - 74	Задовільно
60 - 64	Достатньо
Менше 60	Незадовільно
R < 40 або виконано менше чотирьох робіт комп'ютерного практикуму з переліку ЛР1-ЛР5	Не допущено

9. Додаткова інформація з дисципліни (освітнього компонента)

Робочу програму навчальної дисципліни (силабус) «Побудова компіляторів»:

Склав доцент кафедри ІІІ к.т.н., доц. Стативка Юрій Іванович

Ухвалено кафедрою ІІІ (протокол № 16 від 29.05.2024 р.)

Погоджено Методичною комісією факультету¹ (протокол № 10 від 21.06.2024 р.)

¹

Методичною радою університету – для загальноуніверситетських дисциплін.