



ОСНОВИ РОЗРОБКИ ТРАНСЛЯТОРІВ

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	Перший (бакалаврський)
Галузь знань	12 Інформаційні технології
Спеціальність	121 Інженерія програмного забезпечення
Освітня програма	Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці
Статус дисципліни	Нормативна
Форма навчання	очна (денна)
Рік підготовки, семестр	3 курс осінній семестр
Обсяг дисципліни	7.5 кредитів ECTS /225 годин (36 години лекцій, 54 години практичних занять, 135 годин — самостійна робота студента)
Семестровий контроль/ контрольні заходи	Екзамен, МКР, РГР, представлення робіт комп'ютерного практикуму
Розклад занять	http://rozklad.kpi.ua
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектор: к.т.н., доцент, Статівка Юрій Іванович, statyvka-yu@ill.kpi.ua Практичні: к.т.н., доцент, Статівка Юрій Іванович, statyvka-yu@ill.kpi.ua
Розміщення курсу	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 трансляторів.

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

Розглядаються загальні та спеціалізовані засоби побудови компіляторів та інтерпретаторів: алгоритми лексичного, синтаксичного та семантичного аналізу, генерування коду, засобів виконання коду цільової мови.

Метою дисципліни є опанування студентами теоретичних знань та набуття практичного досвіду проектування і імплементації імперативних 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:

фахові:

- здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування. (ФК 02);
- здатність розробляти архітектури, модулі та компоненти програмних систем (ФК 03);
- здатність застосовувати транслятори мов програмування при реалізації програмних систем (ФК 19).

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

- аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки (ПРН 01);
- знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення (ПРН 05);
- знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення (ПРН 07);
- знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань (ПРН 13);
- знати і вміти використовувати фундаментальний математичний інструментарій при побудові алгоритмів та розробленні сучасного програмного забезпечення (ПРН 25);
- застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення (ПРН 32).

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

Пререквізити дисципліни. Знання, отримані при вивченні дисциплін: «Компоненти програмної інженерії», «Основи програмування», «Комп'ютерна дискретна математика», «Алгоритми та структури даних».

Постреквізити дисципліни. Компетенції, отримані студентами в процесі вивчення дисципліни 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/](http://dotnet.jku.at/applications/visualizer/)

12. [EBNF 2 Railroad: веб-сайт. URL: https://github.com/matthijsgroen/ebnf2railroad](https://github.com/matthijsgroen/ebnf2railroad)
13. [Railroad Diagram Generator: веб-сайт. URL: https://bottlecaps.de/rr/ui](https://bottlecaps.de/rr/ui)
14. [JFLAP Version 7.1 RELEASED July 27, 2018: веб-сайт. URL: https://www.jflap.org/.](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.](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, граматика, виведення, сентенційна форма, рекурсивні правила.

Приведення граматик. Трансформація граматик. Усунення ϵ -продукцій. Усунення ланцюгових правил. Усунення марних символів. Факторизація граматик. Нормальна форма Чомські.

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

Лекція 3. Скінченні автомати.

Поняття розпізнавача. Види розпізнавачів. Скінченні автомати. Визначення, правила функціонування, форми подання. Детерміновані скінченні автомати (ДСА). Недетерміновані скінченні автомати (НСА). Мінімізація ДСА. Автоматні мови. Еквівалентність множини регулярних і автоматних мов.

Розділ 2. Лексичний, синтаксичний та семантичний аналіз програм

Лекція 4. Лексичний аналіз програм.

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

Лекція 5. Автомат з магазинною пам'яттю (МП-автомат).

Визначення, правила функціонування, форми подання. МП-автомат і контекстно вільні граматики.

Лекція 6. Призначення та види синтаксичного аналізу. Метод рекурсивного спуску.

Призначення, види синтаксичного розбору. Низхідний розбір. Алгоритм рекурсивного спуску. Перетворення граматики для реалізації розбору без повернень та зациклювання.

Приклад реалізації парсера на основі алгоритму рекурсивного спуску.

Лекція 7. Семантичний аналіз програм.

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

Лекція 8 Приклад реалізації семантичного аналізатора.

Змінні: декларація, ініціалізація, l- і r-value. Типи змінних, констант, виразів, інструкцій (statements).

Розділ 3. Трансляція та виконання програм

Лекція 9. Проміжні форми подання програм.

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

Лекція 10. Трансляція у ПОЛІЗ і виконання ПОЛІЗ-програм. Postfix State Machine (PSM).

Переклад у ПОЛІЗ конструкцій мови високого рівня. Виконання ПОЛІЗ-програм: стекова машина Postfix State Machine (PSM). Розширення PSM.

Лекція 11. Віртуальні стекові машини. CLR. JVM.

Віртуальна стекова машина .Net. ILASM. ILDASM. Набір інструкцій CLR.

Віртуальна стекова машина Java. Набір інструкцій JVM. Засоби асемблювання та дизасемблювання для JVM. Асемблери та дизасемблери третіх сторін: Krakatau.

Лекція 12. Переклад на CIL та Java-асемблер.

Побудова транслятора на CIL. Асемблювання і виконання il-програм.

Побудова транслятора для JVM. Асемблювання і виконання j-програм.

Розділ 4. Автоматизація розробки трансляторів

Лекція 13. Засоби автоматизації розробки трансляторів

Огляд генератора парсерів: Yacc, Bison, Lex, Flex, Happy, BNFC, ANTLR4.

Лекція 14. ANTLR4 як компілятор компіляторів.

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

Лекція 15. Варіанти використання ANTLR4.

Варіанти використання ANTLR4. Інтегрування згенерованих компонент у Java-застосунок. Інтегрування згенерованих компонент у Python-застосунок.

Розділ 5. Окремі питання розробки трансляторів.

Лекція 16. Генерування коду для регістрових машин..

Тріади і тетради як проміжна форма подання програми.

Генерування триадресного коду.

Лекція 17. Оптимізація цільового коду.

Методи оптимізації цільового коду. Оптимізація лінійних фрагментів програм, логічних виразів, інших синтаксичних конструкцій.

Лекція 18. Розробка трансляторів. Підсумки.

Огляд курсу: концепти, технології, інструменти. Набутий досвід у контексті розробки і реалізації мов програмування. Сучасні тенденції розробки та імплементації мов програмування.

Практичні заняття

Практичне заняття 1. Мови програмування. Специфікація мови програмування: структура і зміст.

Мови програмування. Типологія мов програмування. Парадигми. Інструменти розробника. Компілятори та інтерпретатори.

Структура специфікації — лексика, синтаксис, семантика.

Практичне заняття 2. Елементи теорії формальних мов.

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

Практичне заняття 3. Приведення граматик. Форми подання граматик.

Усунення ϵ -продукцій. Усунення ланцюгових правил. Усунення марних символів. Факторизація граматик. Нормальна форма Чомські.

Нотація і розширена нотація Бекуса-Наура. Побудова граматик для елементів мови програмування у нотації РБНФ. Генератори синтаксичних діаграм. EBNF-Vizualizer.

Практичне заняття 4. Регулярні мови.

Регулярні вирази і мови. Операції над регулярними мовами. Граф регулярного виразу. JFLAP.

Практичне заняття 5. Скінченні автомати.

Детерміновані скінченні автомати (ДСА). Мінімізація ДСА. Недетерміновані скінченні автомати (НСА). Приведення НСА до ДСА. JFLAP. Еквівалентність множини регулярних і автоматних мов.

Практичне заняття 6. Лексичний аналіз програм.

ГраMATика мови програмування, необхідна для лексичного аналізу.

Таблиця символів мови (Таблиця лексем мови).

Діаграма станів. Представлення даних у кодї.

Лексичний аналізатор (сканер).

Практичне заняття 7. Представлення **КП 1**.

Представлення сканера для розробленої мови програмування.

Ревю сканера для розробленої мови програмування.

Практичне заняття 8. Автомат з магазинною пам'яттю (МП-автомат).

Детерміновані і недетерміновані МП-автомати. JFLAP. МП-автомат і контекстно вільні граматики.

Практичне заняття 9. Предиктивний синтаксичний аналіз методом рекурсивного спуску.

ГраMATика мови програмування.

Трансформація граматики для реалізації парсера.

*Практичне заняття 10. Представлення **КП 2**.*

Представлення синтаксичного аналізатора для розробленої мови програмування.

Ревю синтаксичного аналізатора для розробленої мови програмування.

Практичне заняття 11. Семантика мови і поведінка програми.

Типи і оголошення. Перевірка типів. Приведення типів. Семантика синтаксичних конструкцій.

Практичне заняття 12. Опис семантика розробленої мови.

Ревю семантичного компонента специфікації розробленої мови програмування.

*Практичне заняття 13. Представлення **КП 3**.*

Представлення семантичного аналізатора для розробленої мови програмування.

Ревю семантичного аналізатора для розробленої мови програмування.

Практичне заняття 14. Польський інверсний запис (ПОЛІЗ)

Алгоритми трансляції у ПОЛІЗ та виконання поліз програм.

Практичне заняття 15. Стекова машина для виконання ПОЛІЗ-програм (PSM)

Побудова та розширення PSM.

*Практичне заняття 16. Представлення **КП 4**.*

Представлення транслятора з розробленої мови у ПОЛІЗ.

Виконання ПОЛІЗ-програм у PSM.

Ревю транслятора з розробленої мови у ПОЛІЗ.

Практичне заняття 17. Віртуальна стекова машина .Net (CLR).

Огляд CLR та CIL.

Практичне заняття 18. Віртуальна стекова Java-машина (JVM).

Огляд JVM та Java-асемблера.

*Практичне заняття 19. Представлення **КП 5**.*

Представлення транслятора з розробленої мови на CLR та JVM.

Ревю транслятора з розробленої мови на CLR та JVM.

*Практичне заняття 20. Представлення **КП 5**.*

Представлення транслятора з розробленої мови на CLR та JVM.

Ревю транслятора з розробленої мови на CLR та JVM.

Практичне заняття 21. ANTLR4

ANTLR4: Встановлення, використання. Перший сеанс.

*Практичне заняття 22. Представлення **КП 6**.*

Ревю прикладної програми з інтегрованими компонентами, згенерованими засобами ANTLR4 для розробленої мови.

*Практичне заняття 23. Представлення **КП 6**.*

Ревю прикладної програми з інтегрованими компонентами, згенерованими засобами ANTLR4 для розробленої мови.

Практичне заняття 24. Генерування цільового коду.

Проектування генератора коду. Оптимізація коду.

Практичне заняття 25. Представлення **РГР**.

Ревю специфікації розробленої мови. Демонстрація того, що реалізація розробленої мови (КП 5) відповідає її специфікації (РГР).

Практичне заняття 26. Представлення **РГР**.

Ревю специфікації розробленої мови. Демонстрація того, що реалізація розробленої мови (КП 5) відповідає її специфікації (РГР).

Практичне заняття 27. Підсумкове заняття.

Обговорення досвіду в контексті розробки і реалізації мов програмування.

Обговорення шляхів вдосконалення структури/змісту дисципліни.

Підсумки академічної діяльності.

Перелік робіт комп'ютерного практикуму

1. Лексичний аналіз програм.
2. Синтаксичний аналіз програм.
3. Семантичний аналіз програм.
4. Трансляція у ПОЛІЗ та виконання ПОЛІЗ-програм.
5. Трансляція для віртуальної стекової машини (CLR/JVM).
6. Автоматизація побудови транслятора засобами ANTLR4.

Розрахунково-графічна робота

Індивідуальне завдання на **проектування та специфікацію простої імперативної мови програмування**.

Контрольна робота

Метою контрольної роботи є закріплення та перевірка теоретичних знань із освітнього компонента, набуття студентами практичних навичок самостійного вирішення задач.

Модульна контрольна робота (МКР) виконується після вивчення Розділів 1-2. Кожен студент отримує завдання одного з варіантів, відповідно до якого необхідно виконати побудову грамматики, скінченного або магазинного автомата, та виконати приведення грамматики.

Самостійна робота студента

Теми для самостійного опрацювання:

№ з/п	Розділ	Тема	Джерела	Контроль
1	1	Однопрохідний компілятор.	[2, розд. 1.2]	на практ. заняттях

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

Приблизний розподіл часу СРС

№ з/п	Вид самостійної роботи	Кількість	Кількість годин СРС
1	Виконання робіт комп'ютерного практикуму	6	42
2	Виконання РГР	1	12
3	Підготовка до практичних занять	27	27
4	Опрацювання тем, винесених на СРС	12	18
5	Підготовка до МКР		6
6	Підготовка до екзамену		30
Разом			135

Політика та контроль

6. Політика навчальної дисципліни (освітнього компонента)

Система вимог при вивченні дисципліни:

- 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.
- 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;

- 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;
- 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 «Основи розробки трансляторів»;
- 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.

7. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Система рейтингових (вагових) балів та критерії оцінювання

Поточний контроль: активність — питання і відповіді на лекційних заняттях, питання і відповіді та участь у ревію на практичних заняттях; вправи на практичних заняттях; МКР; виконання завдань до практичних занять; виконання та представлення робіт комп'ютерного практикуму, виконання та представлення РГР.

Вид поточного контролю	Кількість	Бали		Сума балів
РГР	5 етапів	виконання	2	14
	1	захист РГР	4	
Практ. та лекц. заняття	5 розділів	активність	2	10
Роботи КП	6	виконання	2+2	36
		представлення	2	
МКР	1		10	10
Сума вагових балів контрольних заходів				70

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 усі виконані роботи комп'ютерного практикуму становит 36 балів. Розподіл балів за виконання робіт практикуму:

№ з/п	Роботи комп'ютерного практикуму	Кількість балів
1	Лексичний аналіз програм	6
2	Синтаксичний аналіз програм	6

№ з/п	Роботи комп'ютерного практикуму	Кількість балів
3	Семантичний аналіз програм	6
4	Трансляція у ПОЛІЗ та виконання ПОЛІЗ-програм	6
5	Трансляція для CLR/JVM	6
6	Побудова синтаксичного аналізатора засобами ANTLR4	6
Разом:		36

Критерії оцінювання:

Виконання робіт комп'ютерного практикуму:

- правильне функціонування розробленого модуля є необхідною умовою зарахування виконаної роботи;
- виконана у повному обсязі і представлена робота – максимальна кількість балів $6 = 2 + 2 + 2$. Виконання у мінімальному обсязі — 2 бали, виконання у повному обсязі — $4 = 2 + 2$ бали, представлення (захист) роботи — 2 бали;
- виконана і представлена частково (не представлена) — відповідна частка від максимальної кількості балів.

3) Модульна контрольна робота

Максимальна кількість балів за модульну контрольну роботу дорівнює 15 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) Розрахунково-графічна робота (РГР)

Максимальна кількість балів — 14. Виконання передбачає 5 етапів, максимум 2 бали за кожний. Представлення фінальної версії — 4 бали.

Календарний контроль: проводиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу (“атестація”).

Умови позитивної проміжної атестації

Календарний контроль базується на поточній рейтинговій оцінці. Умовою позитивної атестації є значення поточного рейтингу студента не менше 50% від максимально можливого на час атестації. Бал, необхідний для отримання позитивного календарного контролю доводиться до відома студентів викладачем не пізніше ніж за 2 тижні до початку календарного контролю.

Семестровий контроль: *екзамен.*

Максимальний ваговий бал $r_{icn} = 30$

На усному екзамені студент обирає білет, який містить два теоретичні питання та одне практичне завдання. Кожне питання оцінюється максимум 8 балами.

Умови допуску до семестрового контролю: виконані роботи 1-4 комп'ютерного практикуму та РГР, стартовий рейтинг (R_c) — не менше 40 балів.

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Бали	Оцінка
95 - 100	Відмінно
85 - 94	Дуже добре
75 - 84	Добре
65 - 74	Задовільно
60 - 64	Достатньо
Менше 60	Незадовільно
$R < 40$ або не виконана одна з робіт № 1-4 комп'ютерного практикуму, або не виконана РГР	Не допущено

Розрахунок шкали (R) рейтингу:

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

$$R = r_{\text{КП}} + r_{\text{РГР}} + r_{\text{МКР}} + r_{\text{активн}} + r_{\text{ісп}} = 36 + 14 + 10 + 10 + 30 = 100 \text{ балів.}$$

Максимальний стартовий рейтинг становить $R_c = r_{\text{КП}} + r_{\text{РГР}} + r_{\text{МКР}} + r_{\text{активн}} = 70$ балів.

Рейтинг іспиту дорівнює 30 балів. Мінімальний рейтинг допуску до іспиту становить 40 балів.

Таким чином, рейтингова шкала з кредитного модуля складає

$$R = 70 + 30 = 100 \text{ балів.}$$

Робочу програму навчальної дисципліни (силабус) «Основи розробки трансляторів»:

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

Ухвалено кафедрою ІПЗЕ (протокол № 42 від 18 червня 2025 р.)

Погоджено Методичною комісією НН ІАТЕ ім.Ігоря Сікорського (протокол № 8 від 20 червня 2025 р.)