

Структура згідно методичних рекомендацій	«Чекліст» компонентів, які обов'язково мають бути продемонстровані (позначені як [_]), а опціональні — як [X] (також усі блок-схеми алгоритмів)	Допоміжні матеріали (ВНС або назва теки на репозиторії), надані протягом семестру на практичних заняттях
ТИТУЛЬНА СТОРІНКА		
ЗАВДАННЯ НА КУРСОВЕ ПРОЄКТУВАННЯ	«Скріншот» варіantu завдання (а) та текстовий опис (б): а[_], б[_].	варіанти завдань (ВНС)
АНОТАЦІЯ		
ЗМІСТ		
ВСТУП		
1. ОГЛЯД МЕТОДІВ ТА СПОСОБІВ ПРОЄКТУВАННЯ ТРАНСЛЯТОРІВ		
2. ФОРМАЛЬНИЙ ОПИС ВХІДНОЇ МОВИ ПРОГРАМУВАННЯ 2.1. Деталізований опис вхідної мови в термінах розширеної нотації Бекуса-Наура 2.2. Опис термінальних символів та ключових слів	EBNF: [_]. Код (boost::spirit), що <u>пострічково відповідає EBNF</u> та коректно опрацьовує тестові програми: [_].	EBNFVerify+GrammarVerify EBNFVerify+GrammarVerify verify_syntax_by_EBNF_2025
3. РОЗРОБКА ТРАНСЛЯТОРА З ВХІДНОЇ МОВИ ПРОГРАМУВАННЯ		
3.1. Вибір технології програмування 3.2. Проектування таблиць транслятора та вибір структур даних		

<p>3.3. Розробка лексичного аналізатора</p> <p>3.3.1. Розробка блок-схеми алгоритму</p> <p>3.3.2. Опис програми реалізації лексичного аналізатора</p>	<p>Початковий етап:</p> <p>Регулярний вираз для токенізатора: [_].</p> <p>Регулярний вираз підмови ключових слів: [_].</p> <p>Регулярний вираз підмови ідентифікаторів: [_].</p> <p>Регулярний вираз підмови беззнакових літералів із фіксованою комою: [_].</p> <p>Код на основі регулярних виразів коректно опрацьовує тестові програми: [_].</p> <p>Безпосередній етап:</p> <p>Детальний опис процесу перетворення регулярних виразів у недетерміновані скінченні automati (NFA): [X].</p> <p>Детальний опис процесу видалення λ-переходів для отриманих NFA: [X].</p> <p>Детальний опис процесу перетворення NFA (після видалення λ-переходів) у детермінований скінчений автомат (DFA): [X].</p>	lexica_2025 lexica_part_dfa_2025 lexica_2025 lexica_part_dfa_2025 lexica_2025 lexica_part_dfa_2025 lexica_2025 lexica_part_dfa_2025 lexica_2025 lexica_part_dfa_2025 lexica_2025 lexica_part_dfa_2025 lexica_2025 lexica_part_dfa_2025 (KN1)/sp2019_soft_12345/ addon_09_11_2024/1 (KN1)/sp2019_soft_12345/ addon_09_11_2024/2 (KN1)/sp2019_soft_12345/ addon_09_11_2024/3
---	---	---

	<p>Детальний опис процесу мінімізації (ключовий етап) DFA: [X].</p> <p>Результативний етап:</p> <p>Повністю детермінований скінчений автомат для токенізації (а: двоеквірний масив для програмного моделювання; б: орієнтований граф з усіма переходами, окрім тих, що ведуть у <u>«мертвий» стан</u>, для графічного відображення): a[_], б[_].</p> <p>Модель токенізатора на основі детермінованого скінченого автомату, оскільки це нетривіальне завдання теорії автоматів- акцепторів: [_].</p> <p>Повністю детермінований скінчений автомат для розпізнавання підмови ключових слів у вигляді таблиці переходів (а: двоеквірний масив для програмного моделювання; б: орієнтований граф з усіма переходами, окрім тих, що ведуть у <u>«мертвий» стан</u>, для графічного відображення): a[_], б[_].</p> <p>Повністю детермінований скінчений автомат для розпізнавання підмови ідентифікаторів у вигляді таблиці переходів (а: двоеквірний масив для програмного моделювання; б: орієнтований граф з усіма переходами, окрім тих, що ведуть у <u>«мертвий» стан</u>, для графічного відображення): a[_], б[_].</p> <p>Повністю детермінований скінчений автомат для розпізнавання підмови беззнакових літералів із фіксованою комою (а: двоеквірний масив для програмного моделювання; б: орієнтований граф з усіма переходами, окрім тих, що ведуть у</p>	(KN1/)sp2019_soft_12345/ addon_09_11_2024/4
--	--	--

	<p><u>«мертвий» стан, для графічного відображення):</u> a[_], б[_].</p>	dfa_generator_2025
<p>3.4. Розробка синтаксичного та семантичного аналізатора</p> <p>3.4.1. Розробка дерев граматичного розбору</p> <p>3.4.2. Розробка блок-схеми алгоритму</p> <p>3.4.3. Опис програми реалізації синтаксичного та семантичного аналізатора</p>	<p><i>Початковий етап посилається на попередній розділ (2.1), у якому формувалися EBNF та еквівалентний (пострічково) код для його перевірки.</i></p> <p>Безпосередній етап:</p> <p>Синтез граматики відповідно до EBNF: [_].</p> <p>Синтез множин FIRST та FOLLOW для отриманої граматики: [X].</p> <p>Специфікація lookahead у правилах виведення граматики для моделі LL(2)-аналізатора: [_].</p> <p>Формування параметрів моделі LL(2)-аналізатора: [_].</p> <p>Результативний етап:</p> <p>Формування моделі LL(2)-аналізатора у вигляді недетермінованого автомата з магазинною пам'яттю (NPDA) (а: формальний опис; б: програмна модель): a[_], б[_].</p> <p>Формування моделі LL(2)-аналізатора у вигляді детермінованого автомату з магазинною пам'яттю (DPDA): [X].</p>	<p>EBNFVerify+GrammarVerify</p> <p>EBNFVerify+GrammarVerify</p> <p>EBNFVerify+GrammarVerify</p> <p>EBNFVerify+GrammarVerify</p> <p>dpda1_for_ll2_generator_2025</p> <p>dpda1_for_ll2_generator_2025</p>

<p>3.5. Розробка генератора коду</p> <p>3.5.1. Розробка блок-схеми алгоритму</p> <p>3.5.2. Опис програми реалізації генератора коду</p>	<p>У цьому розділі виконується опис:</p> <p>алгоритму формування польського інверсного запису [_],</p> <p>алгоритму виконання польського інверсного запису з використанням стеку [_],</p> <p><i>Сам код генерації універсальний.</i></p> <p><i>Додатково можна описати процес формування SSA та інші аспекти [X].</i></p>	<p><code>cw_sp2_2025_2026/src/implementation/preparer</code></p> <p><code>cw_sp2_2025_2026/src/implementation/generator</code></p> <p><code>28_11_2025 (+інші теки)</code></p>
<p>4. НАЛАГОДЖЕННЯ ТА ТЕСТУВАННЯ РОЗРОБЛЕНГО ТРАНСЛЯТОРА</p> <p>4.1. Опис інтерфейсу та інструкції користувачу</p> <p>4.2. Виявлення лексичних і синтаксичних помилок</p> <p>4.3. Перевірка роботи транслятора за допомогою тестових задач</p>	<p>Описати консольний інтерфейс компілятора [_]</p> <p>Три тестові програми згідно варіанту [_]</p> <p>Коректна компіляція трьох тестових програм* [_]</p> <p><i>*у записці наводиться тестування лише трьох заданих тестових програм, але під час захисту потрібно вміти написати вашою мовою програмування будь-яку просту тестову програму</i></p>	<p><code>cw_sp2_2025_2026/src/implementation/cli</code></p> <p><code>base_test_programs_2025</code></p> <p><code>cw_sp2_2025_2026 (запуск)</code></p>
<p>ВІСНОВКИ</p> <p>СПИСОК ЛІТЕРАТУРНИХ ДЖЕРЕЛ</p> <p>ДОДАТКИ</p> <p>А. Таблиці лексем для тестових прикладів</p> <p>Б. С код (або код на асемблері), отриманий на виході транслятора для тестових прикладів;</p> <p>Б. Документований текст програмних модулів (лістинги)</p>	<p>Оскільки це курсовий проект (а не курсова робота), то окремо у форматі А3 (що має бути оформлено як стандартне креслення) подається один графічний матеріал. Це може бути приміром загальна блок-схема роботи транслятора: [_]</p>	