

Котиков Д.С.,

студент

Научный руководитель: Зайцева Т.В., доц, к.т.н

Белгородский государственный национальный исследовательский

университет, г. Белгород, Российская Федерация

**ИСПОЛЬЗОВАНИЕ PYTHON LEX-YACC (PLY) В РАЗРАБОТКЕ
ЛАБОРАТОРНОГО ПРАКТИКУМА ДЛЯ ОБУЧЕНИЯ
ПРОЕКТИРОВАНИЮ ТРАНСЛЯТОРОВ**

Аннотация. В статье исследуется использование Python Lex-Yacc (PLY) для создания лабораторного практикума по проектированию трансляторов. Описаны принципы работы PLY и разработка заданий, включая пример транслятора арифметических выражений с применением схем. Показана ценность практикума для обучения студентов основам создания трансляторов.

Ключевые слова: транслятор, PLY, Python Lex-Yacc, лабораторный практикум, синтаксический анализ, обучение программированию.

Kotikov D.S.,

student

Academic Advisor: Zaitseva T.V., PhD of Engineering Sciences,

Associate Professor

Belgorod State National Research University,

Russia, Belgorod

**IMPLEMENTATION OF PYTHON LEX-YACC (PLY) IN THE
DEVELOPMENT OF A LABORATORY PRACTICUM FOR TRAINING
IN TRANSLATOR DESIGN**

Abstract. This article examines the use of Python Lex-Yacc (PLY) in developing a laboratory practicum for translator design. It describes PLY's

operating principles and task development, including an example of an arithmetic expression translator using diagrams. The practicum's value for teaching students translator creation fundamentals is highlighted.

Key words: translator, PLY, Python Lex-Yacc, laboratory practicum, syntax analysis, programming education.

Трансляторы играют ключевую роль в современных программных системах, преобразуя код из одного языка в другой – от компиляторов до интерпретаторов и анализаторов текста. Освоение их проектирования является важной частью подготовки специалистов в области программирования, однако сложность теоретических основ часто становится препятствием для студентов. Лабораторные практикумы помогают преодолеть этот барьер, позволяя применять знания на практике. В этом контексте библиотека Python Lex-Yacc (PLY) представляет собой эффективный инструмент, сочетающий простоту использования с мощными возможностями создания трансляторов.

PLY реализует классические подходы Lex и Yacc, адаптированные для языка Python. Процесс проектирования транслятора с ее помощью включает два основных этапа: лексический анализ, преобразующий исходный текст в последовательность токенов (числа, операторы, ключевые слова), и синтаксический анализ, строящий дерево разбора на основе заданных грамматических правил. Открытый исходный код, отсутствие внешних зависимостей и интеграция с Python делают PLY удобным выбором для образовательных целей.

Разработка лабораторного практикума на основе PLY представляет собой набор из четырех лабораторных работ, последовательность которых обеспечивает постепенное усложнение задач. Первая работа носит вводный характер и знакомит студентов с основами работы PLY на простом уровне – например, распознавание базовых токенов в духе

программы «Hello World». Последующие работы углубляют материал: от выделения токенов из текста до создания транслятора арифметических выражений, таких как « $2 + 3 * 4$ ». На первом этапе исходный текст разбивается на токены, что иллюстрирует схема лексического анализа (рис. 1). Далее токены обрабатываются парсером, который учитывает приоритет операций и формирует результат, как показано на схеме синтаксического анализа (рис. 2). Завершающая работа включает учет более сложных структур, таких как вложенные выражения, что закрепляет понимание студентами ключевых этапов проектирования трансляторов.

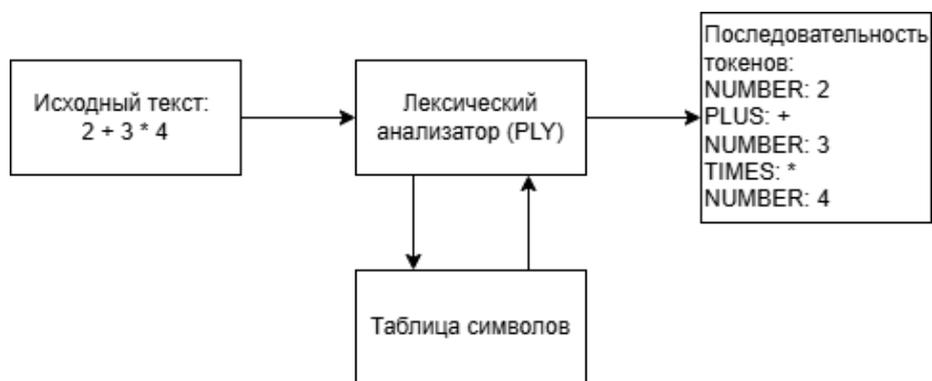


Рис. 1. Схема процесса лексического анализа в PLY

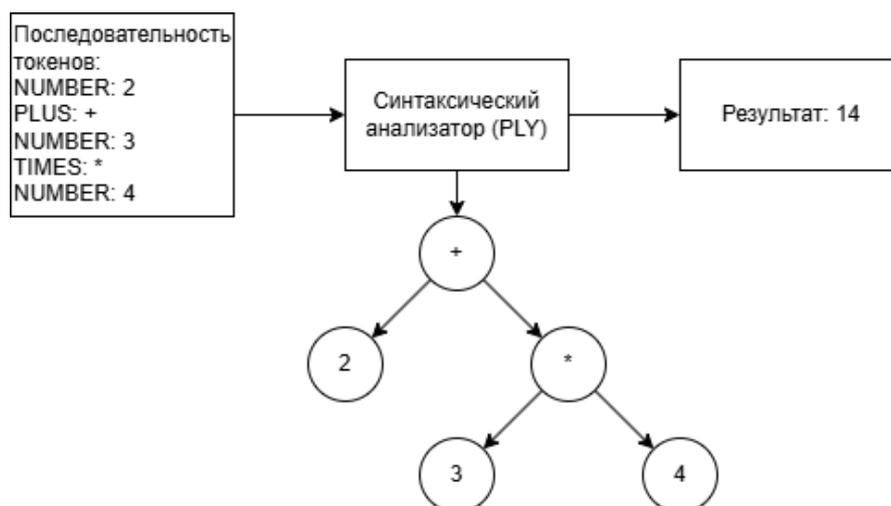


Рис. 2. Схема процесса синтаксического анализа в PLY

Использование PLY в таком практикуме позволяет студентам не только понять теоретические основы лексического и синтаксического

анализа, но и приобрести практические навыки реализации трансляторов. Это способствует формированию целостного представления о роли данных процессов в создании компиляторов и интерпретаторов. В процессе выполнения лабораторных работ студенты развивают навыки работы с грамматиками и алгоритмами разбора, что особенно важно для будущей профессиональной деятельности в области разработки программного обеспечения. Кроме того, интерактивный характер заданий, поддерживаемый визуализацией процессов через схемы, способствует более глубокому усвоению материала, особенно для начинающих программистов.

Перспективы применения PLY в образовании выходят за рамки текущего практикума. Например, можно разработать дополнительные задания, связанные с анализом более сложных языков, таких как мини-языки программирования с условными операторами или циклами. Это позволит студентам углубить знания в области теории компиляции и приобрести опыт, востребованный в индустрии. Кроме того, интеграция PLY с другими инструментами Python, такими как системы автоматической проверки заданий, может повысить эффективность обучения, предоставляя студентам мгновенную обратную связь. Таким образом, PLY выступает не только инструментом для текущих образовательных целей, но и платформой для дальнейшего развития учебных программ в области проектирования трансляторов.

Использованные источники:

1. Kuznetsov, M. V. Methods and tools for teaching compiler design in higher education [Электронный ресурс] // Procedia Computer Science. – 2020. – Vol. 176. – P. 1234–1242. – URL: <https://doi.org/10.1016/j.procs.2020.09.134> (дата обращения: 02.03.2025).

2. Иванова, О. П. Обучение технологиям разработки компиляторов: современные подходы // Вестник ЮУрГУ. Серия «Образование. Педагогические науки». – 2019. – Т. 11, № 3. – С. 45–52. – URL: <https://elibrary.ru/item.asp?id=41123456> (дата обращения: 02.03.2025).
3. Parr, T. Language Implementation Patterns: Create Your Own Domain-Specific and General Programming Languages. – Pragmatic Bookshelf, 2010. – URL: <https://www.pragprog.com/book/tpdsl/language-implementation-patterns> (дата обращения: 03.03.2025).
4. Смирнов, А. В. Инструменты автоматизации проектирования трансляторов в образовательных целях // Информатика и образование. – 2021. – № 6. – С. 78–85. – URL: <https://elibrary.ru/item.asp?id=46378912> (дата обращения: 03.03.2025).
5. Beazley, D. PLY (Python Lex-Yacc): An Implementation of Lex and Yacc in Python [Электронный ресурс] // International Python Conference Proceedings. – 2001. – URL: <http://www.dabeaz.com/ply/ply.html> (дата обращения: 03.03.2025).