МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Забайкальский государственный университет»

(ФГБОУ ВО «ЗабГУ»)

Факультет энергетический

Кафедра информатика и вычислительная техника и прикладная математика

**КУРСОВАЯ РАБОТА**

по дисциплине «Теории языков программирования»

на тему: Программная реализация транслятора

Выполнил ст. гр. ИВТ-19

Лягоцкий Максим Владимирович

Проверил к. ф-м. н., доцент каф. ИВТ и ПМ

Батухтина Ирина Юрьевна

Чита 2022

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Забайкальский государственный университет»

(ФГБОУ ВО «ЗабГУ»)

Факультет энергетический

Кафедра информатика и вычислительная техника и прикладная математика

ЗАДАНИЕ

на курсовую работу

По дисциплине: Теория языков программирования и методы трансляции

Студенту Лягоцкому Максиму Владимировичу специальности 09.03.01 Информатика и вычислительная техника (Программное обеспечение вычислительной техники и автоматизированных систем)

специальности 09.03.01 Информатика и вычислительная техника

1. Тема курсовой работы: Программная реализация транслятора

2. Срок подачи студентом законченной работы: 23 декабря 2022 г.

3. Исходные данные к работе: литературные источники, источники в сети интернет

4. Перечень подлежащих в курсовой работе вопросов:

а) Постановка и анализ задачи;

б) Анализ данных;

в) Программная реализация;

5. Перечень графического материала (если имеется): -

Дата выдачи задания «9» сентября 2022г.

Руководитель курсовой работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Батухтина.И.Ю./

(подпись, расшифровка подписи)

Задание принял к исполнению

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

Подпись студента \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Лягоцкий М.В./

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Забайкальский государственный университет»

(ФГБОУ ВО «ЗабГУ»)

Факультет энергетический

Кафедра информатика и вычислительная техника и прикладная математика

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе

по 09.03.01 Информатика и вычислительная техника

на тему: «Программная реализация транслятора»

Выполнил студент группы ИВТ-19 Лягоцкий М.В.

Руководитель работы к. ф-м. н., доцент каф. ИВТ и ПМ, Батухтина Ирина Юрьевна

РЕФЕРАТ

Пояснительная записка 17 – страниц, 1 – приложений, 2 рисунка, 6 источников.

ТРАНСЛЯТОР, PYTHON, C++, .NET, ЛЕКСИЧЕСКИЙ АНАЛИЗАТОР, СИНТАКСИЧЕСКИЙ АНАЛИЗАТОР, ГРАММАТИКА, ЛЕКСЕМА.

В работе описывается создание транслятора с языка программирования Python на язык C++ и обратно. Разработка транслятора производится на языке python. Интерфейс реализован как консольное приложение. В работе рассматриваются понятие транслятора, принципы построения транслятора.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 6](#_Toc124240387)

[1 Постановка и анализ задачи 7](#_Toc124240388)

[1.1 Описание предметной области 7](#_Toc124240389)

[1.2 Постановка задачи 8](#_Toc124240390)

[1.3 Средства реализации 8](#_Toc124240391)

[1.4 Описание языка программирования Python 8](#_Toc124240392)

[1.5 Структура языка Python 9](#_Toc124240393)

[1.6 Описание языка программирования С++ 9](#_Toc124240394)

[1.7 Структура языка С++ 10](#_Toc124240395)

[2 Анализ данных 10](#_Toc124240396)

[2.1 Промежуточные данные 10](#_Toc124240397)

[3 Программная реализация 12](#_Toc124240398)

[ЗАКЛЮЧЕНИЕ 14](#_Toc124240399)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 15](#_Toc124240400)

[Приложение А 16](#_Toc124240401)

[Результаты тестирования программы. 16](#_Toc124240402)

# ВВЕДЕНИЕ

В настоящее время искусственные языки, использующие для описания предметной области текстовое представление, широко применяются не только в программировании, но и в других областях. С их помощью описывается структура всевозможных документов, трехмерных виртуальных миров, графических интерфейсов пользователя и многих других объектов, используемых в моделях и в реальном мире. Для того, чтобы эти текстовые описания были корректно составлены, а затем правильно распознаны и интерпретированы, используется специальный вид программ, именуемый трансляторами. Разработка трансляторов на сегодняшний момент является одной из самых актуальных областей в разработке программного обеспечения.

# 1 Постановка и анализ задачи

## 1.1 Описание предметной области

Транслятор — [программа](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0) или техническое средство, выполняющее трансляцию программы.

Трансляция программы— преобразование программы, представленной на одном из [языков программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), в программу на другом языке. Транслятор обычно выполняет также диагностику ошибок, формирует словари идентификаторов, выдаёт для печати текст программы и т. д.

Язык, на котором представлена входная программа, называется исходным языком, а сама программа — [исходным кодом](https://ru.wikipedia.org/wiki/%D0%98%D1%81%D1%85%D0%BE%D0%B4%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4). Выходной язык называется целевым языком.

Одна из важных ролей транслятора состоит в сообщении об ошибках в исходной программе, обнаруженных во время трансляции.

Процесс отображение входной программы в эквивалентную ей выходную программу состоит из двух частей: анализ и синтез.

Анализ разбивает входную программу на составные части и накладывает на них грамматическую структуру. Затем использует эту структуру для создания промежуточного представления входной программы. Если анализ обнаруживает, что входная программа неверно составлена синтаксически или семантически, он должен выдать информативные сообщения об этом, чтобы пользователь мог исправить обнаруженные ошибки. Анализ также собирает информацию об исходной программе и сохраняет её в структуре данных, называемой таблицей идентификаторов (символов), которая передаётся вместе с промежуточным представлением синтезу.

Синтез строит требуемую целевую программу на основе промежуточного представления и информации из таблицы идентификаторов.

Анализ часто называют начальной стадией, а синтез – заключительной.

Если посмотреть на процесс трансляции более детально, можно увидеть, что он представляет собой последовательность фаз, каждая из которых преобразует одно из представлений входной программы в другое. На практике некоторые фазы могут быть объединены, а межфазное промежуточное представление может не строиться явно. Таблица идентификаторов, в которой хранится информация обо всей входной программе, используется всеми фазами транслятора.

## 1.2 Постановка задачи

Разрабатываемый транслятор будет иметь возможность выполнять трансляцию с языка программирования Python на C++ и обратно.

При трансляции транслятор будет выполнять фазы лексического анализа, синтаксического анализа, а также генерации целевого исходного кода. Важно отметить, что на каждой из фаз транслятор должен осуществлять жёсткий контроль ошибок в анализируемом им исходном коде и по мере их нахождения уведомлять об этом пользователя.

## 1.3 Средства реализации

Реализация выполнена на языке Python. Интерфейс реализован как консольное приложение.

## 1.4 Описание языка программирования Python

Python - высокоуровневый язык программирования общего назначения с динамической типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ. Язык является полностью объектно-ориентированным в том плане, что всё является объектами.

## 1.5 Структура языка Python

Иерархия подпрограмм в Python основывается на табуляции. Знаки табуляции заменяют операторные скобки и нарушение в их постановке вызовет как синтаксические, так и логические ошибки

Идентификатор Python - это имя, используемое для идентификации переменной, функции, класса, модуля или любого другого пользовательского элемента. Идентификатор начинается с буквы от A до Z или от a до z или символа подчеркивания (\_), за которым следует ноль или несколько букв, символов подчеркивания и цифр (от 0 до 9).

Некоторые символы, например круглые скобки или точка с запятой в конце строки могут быть написаны, но требуются . Python - это язык программирования с учетом регистра.

## 1.6 Описание языка программирования С++

**C++** — компилируемый, статически типизированный язык программирования общего назначения. Поддерживает такие парадигмы программирования, как процедурное программирование, объектно-ориентированное программирование, обобщённое программирование.

Язык имеет богатую стандартную библиотеку, которая включает в себя распространённые контейнеры и [алгоритмы](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC), ввод-вывод, регулярные выражения, поддержку [многопоточности](https://ru.wikipedia.org/wiki/%D0%9C%D0%BD%D0%BE%D0%B3%D0%BE%D0%BF%D0%BE%D1%82%D0%BE%D1%87%D0%BD%D0%BE%D1%81%D1%82%D1%8C) и другие возможности. C++ сочетает свойства как [высокоуровневых](https://ru.wikipedia.org/wiki/%D0%92%D1%8B%D1%81%D0%BE%D0%BA%D0%BE%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D0%B5%D0%B2%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), так и [низкоуровневых языков](https://ru.wikipedia.org/wiki/%D0%9D%D0%B8%D0%B7%D0%BA%D0%BE%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D0%B5%D0%B2%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F). В сравнении с его предшественником — языком [C](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)) — наибольшее внимание уделено поддержке [объектно-ориентированного](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) и [обобщённого программирования](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D0%BE%D0%B1%D1%89%D1%91%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5).

## 1.7 Структура языка С++

Синтаксис C++ унаследован от языка [C](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)). Изначально одним из принципов разработки было сохранение совместимости с C.

Идентификатор Python - это имя, используемое для идентификации переменной, функции, класса, модуля или любого другого пользовательского элемента. Идентификатор начинается с буквы от A до Z или от a до z или символа подчеркивания (\_), за которым следует ноль или несколько букв, символов подчеркивания и цифр (от 0 до 9).

Блоки кода в языке отделаются фигурными скобками, что делает иерархию блоков однозначной и позволяет записать весь код в 1 строку. Команды в языке завершаются точкой с запятой.

# 2 Анализ данных

Данные, с которыми работает транслятор можно разделить на несколько групп. Входные данные – текст программы на Python или C++. Промежуточные данные – это данные, используемые транслятором во время работы. Выходные данные – текст программы на Python или C++.

## 2.1 Промежуточные данные

Трансляция одного языка в другого происходит постепенно. Сначала текст программы поступает на вход лексическому анализатору. После чего на выходе появляются таблицы идентификаторов и лексем. Данные таблицы поступают на вход синтаксическому анализатору, который генерирует дерево разбора.

Лексема - это структурная единица языка, состоящая из элементарных символов языка и не содержащая других структурных единиц языка. Лексемы языка программирования - это идентификаторы, константы, ключевые слова языка, знаки операций и т. п.

Таблица лексем содержит весь текст исходной программы, обработанный лексическим анализатором. В неё входят все возможные типы лексем, при этом, любая лексема может в ней встречаться любое число раз.

Таблица идентификаторов содержит только следующие типы лексем: идентификаторы и константы. В неё не попадают ключевые слова входного языка, знаки операций и разделители. Каждая лексема в таблице идентификаторов может встречаться только один раз.

Результат грамматического анализа. Дерево разбора отличается от [абстрактного синтаксического дерева](https://pvs-studio.com/ru/blog/terms/0004/) наличием узлов для тех синтаксических правил, которые не влияют на семантику программы.

# 3 Программная реализация

Транслятор выполняет перевод входной программы в эквивалентную ей исполняемую программу. Перевод разделён на две части: анализ и синтез.

Анализ разбивает входную программу на составные части и накладывает на них грамматическую структуру. Затем он использует эту структуру для создания промежуточного представления входной программы. Анализ также собирает информацию об исходной программе и сохраняет её в структуре данных, именуемой таблицей идентификаторов (символов), которая передаётся вместе с промежуточным представлением синтезу.

Синтез строит требуемую целевую программу на основе промежуточного представления и информации из таблицы идентификаторов.

Если рассмотреть процесс трансляции более детально, можно увидеть, что он представляет собой последовательность фаз, каждая из которых преобразует одно из представлений входной программы в другое. Таблица идентификаторов, в которой хранится информация обо всей входной программе, используется всеми фазами транслятора. Первая фаза трансляции называется лексическим анализом или сканированием. Лексический анализатор читает поток символов, составляющих входную программу, и группирует эти символы в значащие последовательности, называемые лексемами. Лексемы передаются последующей фазе транслятора, синтаксическому анализу.

Вторая фаза транслятора – синтаксический анализ или разбор. Во время выполнения этой фазы используются лексемы, полученные от лексического анализатора, для создания древовидного промежуточного представления программы, которое описывает грамматическую структуру потока лексем. Типичным промежуточным представлением является синтаксическое дерево, в котором каждый внутренний узел представляет операцию, а дочерние узлы – аргументы этой операции.

Генератор кода получает в качестве входных данных промежуточное представление исходной программы и отображает его в целевой язык. Важная функция транслятора состоит в том, чтобы записывать имена переменных входной программы и накапливать информацию о разных атрибутах каждого имени. Эти атрибуты могут предоставлять информацию о выделенной памяти для данного имени, его типе, области видимости (где именно в программе может использоваться его значение) и, в случае имён процедур, такие сведения, как количество и типы их аргументов, метод передачи каждого аргумента (например, по значению или по ссылке), а также возвращаемый тип. Таблица идентификаторов представляет собой структуру данных, содержащую записи для каждого имени переменной с полями для атрибутов имени. Структура данных должна быть разработана таким образом, чтобы позволять транслятору быстро находить запись для каждого имени, а также быстро сохранять данные в записи и получать их из неё. Каждая из фаз трансляции была реализованная в программе.

# ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы были получены навыки по проектированию программ, их отладке и документированию.

В результате работы была написана программа – транслятор с языка программирования Python на язык программирования C++ и наоборот.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Wikipedia. Транслятор [Электронный ресурс] // Электрон. текстовый документ – Режим доступа: [https://ru.wikipedia.org/wiki/Транслятор](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B0%D0%BD%D1%81%D0%BB%D1%8F%D1%82%D0%BE%D1%80) свободный
2. Wikipedia. Лексический анализ [Электронный ресурс] // Электрон. текстовый документ – Режим доступа: [https://ru.wikipedia.org/wiki/Лексический\_анализ](https://ru.wikipedia.org/wiki/%D0%9B%D0%B5%D0%BA%D1%81%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7) свободный
3. Wikipedia. Синтаксический анализатор [Электронный ресурс] // Электрон. текстовый документ – Режим доступа: [https://ru.wikipedia.org/wiki/Синтаксический\_анализатор](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BD%D1%82%D0%B0%D0%BA%D1%81%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%82%D0%BE%D1%80) свободный
4. Wikipedia. C++ [Электронный ресурс] // Электрон. текстовый документ – Режим доступа:  
   https://learn.microsoft.com/ru-ru/dotnet/csharp/tour-of-csharp/ свободный
5. Python.org. Документация Python Abstract Syntax Trees [Электронный реусурс] // Электрон. текстовый документ – Режим доступа: https://docs.python.org/3/library/ast.htmlсвободный
6. Молдованова О.В. Языки программирования и методы трансляции: учеб. пособие / О.В. Молдованова. – Новосибирск: ФГОБУ ВПО «СибГУТИ», 2012. – 39 с.

# Приложение А

(обязательное)

## Результаты тестирования программы.

Пример работы транслирования с языка Python на С++ представлен на рисунке А.1.

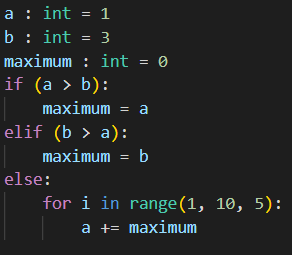
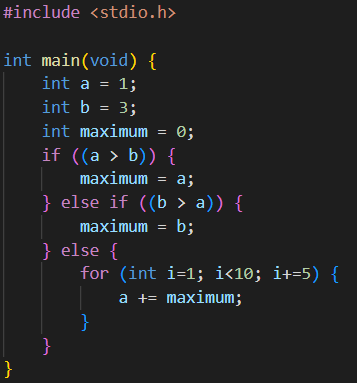
** **

Рисунок А.1 – Трансляция с языка Python на C++.

Пример работы транслирования с языка C++ на Python представлен на рисунке А.2.

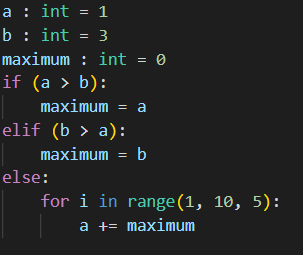
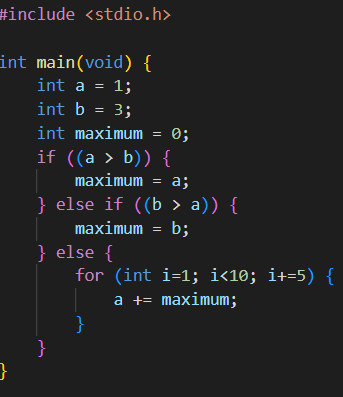
 

Рисунок А.2 – Трансляция с языка C++ на Python.