

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Дальневосточный федеральный университет»**

**ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ (ШКОЛА)**

**Департамент программной инженерии и искусственного интеллекта**

РАЗРАБОТКА ТРАНСЛЯТОРА, ПЕРЕВОДЯЩЕГО ПОДМНОЖЕСТВО ЯЗЫКА X\_LANG В ЭКВИВАЛЕНТНОЕ ПОДМНОЖЕСТВО ЯЗЫКА Y\_LANG

**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ**

по дисциплине «Теория языков программирования и компиляторы»

по образовательной программе подготовки бакалавров по направлению 09.03.04 Программная инженерия

Выполнили студенты

гр. Б9120-09.03.04прогин

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Name1 N.N.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Name2 N.N.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Name3 N.N.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Name4 N.N.

Руководитель:

ассистент департамента ПИиИИ

Глазырин М.А.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

(подпись)

г. Владивосток

2023 г

**Оглавление**

[Введение 4](#_Toc151905248)

[1 Неформальная постановка задачи 5](#_Toc151905249)

[2 Синтаксис входного языка 5](#_Toc151905250)

[3 Контекстные условия языка X\_LANG 5](#_Toc151905251)

[4 Контекстные условия языка Y\_LANG 5](#_Toc151905252)

[5 Таблица соответствия языков 5](#_Toc151905253)

[6 Проект лексического анализатора 6](#_Toc151905254)

[6.1 Модель данных 6](#_Toc151905255)

[6.2 Таблица ключевых слов 6](#_Toc151905256)

[6.3 Таблица зарезервированных имен 6](#_Toc151905257)

[6.4 Типы лексем 6](#_Toc151905258)

[6.5 Типы ошибок 6](#_Toc151905259)

[6.6 Конечный автомат лексического анализатора 7](#_Toc151905260)

[7 Проект семантического анализатора 8](#_Toc151905261)

[7.1 Модель данных 8](#_Toc151905262)

[7.2 Описание алгоритма X семантического анализа 8](#_Toc151905263)

[7.3 Пример дерева разбора 8](#_Toc151905264)

[7.4 Типы ошибок 8](#_Toc151905265)

[8 Проект семантического анализатора 9](#_Toc151905266)

[8.1 Модель данных (если есть) 9](#_Toc151905267)

[8.2 Методы проверки нарушений контекстных условия 9](#_Toc151905268)

[8.3 Типы ошибок 9](#_Toc151905269)

[9 Проект генератора кода 10](#_Toc151905270)

[9.1 Модель данных (если есть) 10](#_Toc151905271)

[9.2 Метод кодогенерации 10](#_Toc151905272)

[9.3 Примеры кодогенерации 10](#_Toc151905273)

[9.4 Оптимизация и методы оптимизации 10](#_Toc151905274)

[10 Тестирование 11](#_Toc151905275)

[10.1 Лексический анализатор 11](#_Toc151905276)

[10.2 Синтаксический анализатор 11](#_Toc151905277)

[10.3 Семантический анализатор 11](#_Toc151905278)

[10.4 Генератор кода 11](#_Toc151905279)

[Заключение 12](#_Toc151905280)

# Введение

Что такое транслятор, что такое язык, какие языки выбрали (что мы далее называем входным языком, что выходным), зачем мы это делаем. Это нужно написать в этих первых абзацах.

**Цель курсового проекта:** разработать проект транслятора, который преобразует программу, содержащую подмножество языка X\_LANG, в программу, содержащую подмножество языка Y\_LANG генерируя эквивалентный исходный код.

**Задачи:**

1. описать грамматику подмножества входного языка;
2. описать контекстные условия входного и выходного языков;
3. описать соответствие конструкций входного и выходного языков;
4. разработать проект лексического анализатора;
5. разработать проект синтаксического анализатора;
6. разработать проект семантического анализатора;
7. разработать проект генератора кода выходного языка;
8. реализовать транслятор входного языка в выходной;
9. выполнить тестирование разработанного транслятора.

# 1 Неформальная постановка задачи

Тут нужно написать про то, что нужно было сделать и что нужно сделать коротко. Говорим какое подмножество входного языка мы выбрали.

Подмножество языка X\_LANG включает в себя

1. типы данных: УКАЗАТЬ КАКИЕ (integer, boolean, …);

2. операторы: УКАЗАТЬ КАКИЕ – АРИФМЕТИЧЕСКИЕ КАКИЕ, ЛОГИЧЕСКИЕ КАКИЕ, ПРОЧИЕ КАКИЕ (+ and);

3. операторы циклов: УКАЗАТЬ КАКИЕ (while …);

4. операторы ветвления: УКАЗАТЬ КАКИЕ (if … then …);

5. стандартные функции: УКАЗАТЬ КАКИЕ (print, например).

6. и тд

# 2 Синтаксис входного языка

Тут нужно написать про какие обозначения используются ниже, например для переноса строки или табуляции. Описываем грамматику входного языка (<PROGRAM> ::= <INSTRUCT> | <INSTRUCTIONS> и тд и тп.). Можно добавлять подразделы для описания некоторых конструкций, если вам так удобнее.

# 3 Контекстные условия языка X\_LANG

Тут нужно написать про контекстные условия входного языка, мы это писали в первом семестре для одной из задач.

# 4 Контекстные условия языка Y\_LANG

Аналогично

# 5 Таблица соответствия языков

Тут нужно написать про то, о чем этот раздел. Этот раздел о том, как соответствуют отдельные лексемы одного языка другом, как соответствует структура одного языка другому (это нужно для того, чтобы показать что и куда транслятор переводит). Надо описать это наиболее общим способом. Ниже есть пример (кусочек).

Изображение выглядит как стол

Автоматически созданное описание

# 6 Проект лексического анализатора

Тут нужно написать что делает лексер, что такое лексический анализ, о разделах ниже и тд.

## 6.1 Модель данных

Какие классы/структуры использовали, почему. Диаграммы классов можно вставить, если есть. Описать как представлены лексемы и токены в памяти

## 6.2 Таблица ключевых слов

Что такое ключевые слова, чем они отличаются от остальных лексем и тд. Надо написать, что ниже есть таблица с ключевыми словами :) Ниже есть пример такой таблицы.

Изображение выглядит как стол

Автоматически созданное описание

## 6.3 Таблица зарезервированных имен

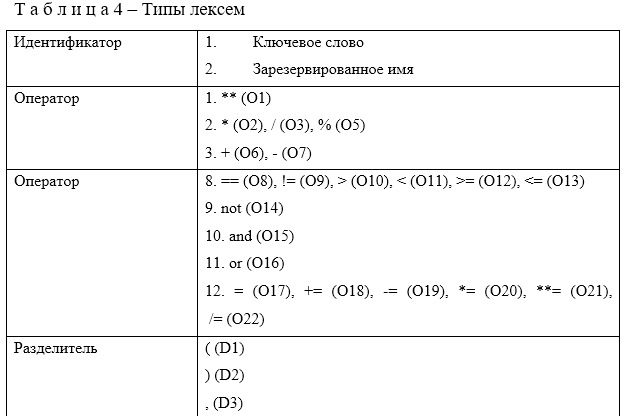
Что, зачем, в чем отличие от ключевых слов и других лексем. Ниже пример таблицы

Изображение выглядит как стол

Автоматически созданное описание

## 6.4 Типы лексем

Аналогично (что, зачем). Ниже пример таблицы (кусочек) (название токенов в первом столбце вы даете сами, а справа - представители токена)

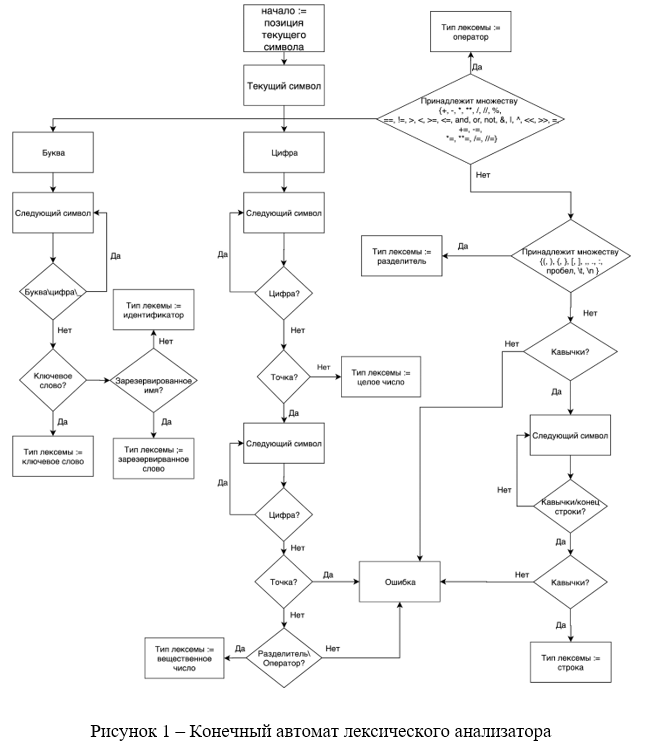
****

## 6.5 Типы ошибок

Все ошибки, которые может выдать лексер с примерами кода и сообщениями об ошибке

## 6.6 Конечный автомат лексического анализатора

Аналогично что зачем и диаграмму автомата, можно в виде блок-схемы как ниже, лучше в виде диаграммы конечного автомата – она будет короче. Еще нужно привести пример его работы на примере какой-нибудь (демонстрирующей возможности языка) программы на входном языке. Можно привести пример вывода своего лексера.

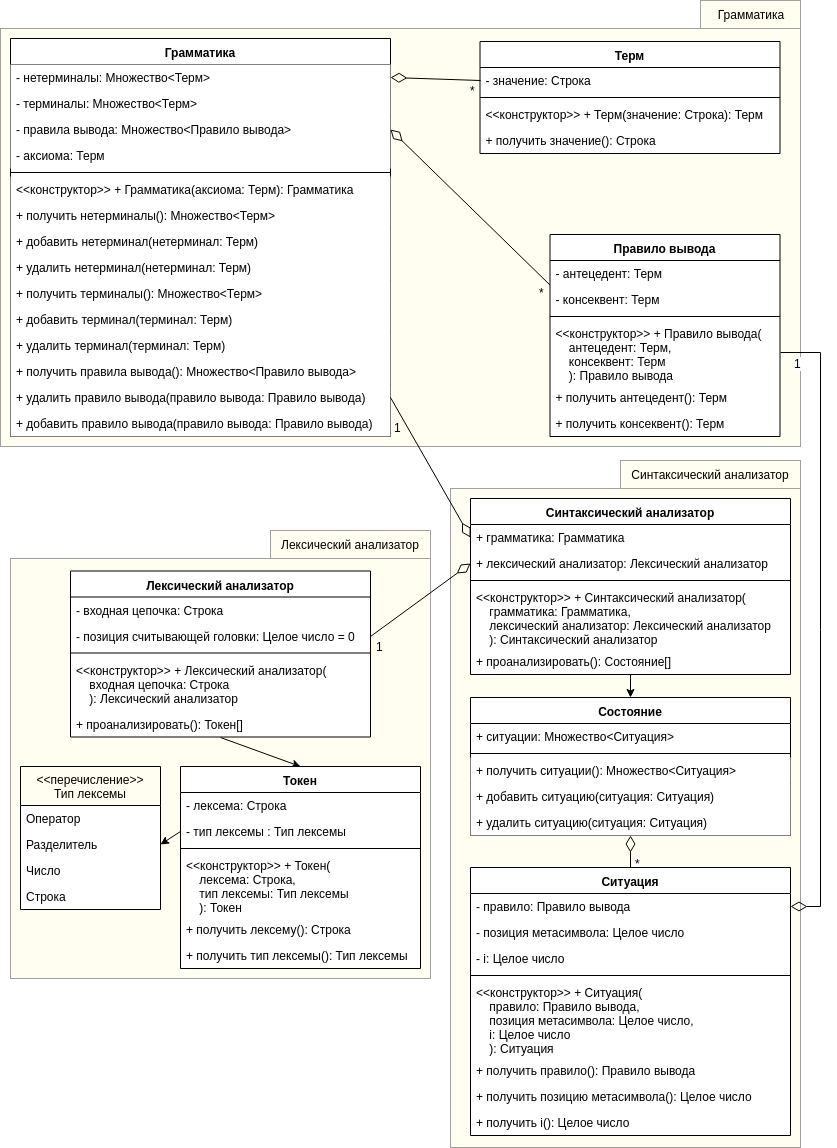


# 7 Проект семантического анализатора

Что зачем. Какой алгоритм синтаксического анализа выбран

## 7.1 Модель данных

Средствами чего реализовывали, какой язык для этого выбрали (для написания синтаксического анализатора), привести диаграммы моделей данных. Показать как моделируется дерево разбора. Ниже пример диаграммы.



## 7.2 Описание алгоритма X семантического анализа

Тут понятно. Просто распишите как в лекциях

## 7.3 Пример дерева разбора

Тут постройте по алгоритму дерево разбора – можете вставить вывод вашего парсера (семантического анализатора)

## 7.4 Типы ошибок

Аналогично разделу 6

# 8 Проект семантического анализатора

Аналогично 6

## 8.1 Модель данных (если есть)

Аналогично 6

## 8.2 Методы проверки нарушений контекстных условия

Как проверяются ошибки типов и тд (контекстных условий)

## 8.3 Типы ошибок

Аналогично 6

# 9 Проект генератора кода

## 9.1 Модель данных (если есть)

## 9.2 Метод кодогенерации

## 9.3 Примеры кодогенерации

## 9.4 Оптимизация и методы оптимизации

Описать сущность оптимизации при трансляции. Привести методы оптимизации, которые были использованы. Например, избавление от неиспользуемых строк кода.

# 10 Тестирование

## 10.1 Лексический анализатор

## 10.2 Синтаксический анализатор

## 10.3 Семантический анализатор

## 10.4 Генератор кода

Также привести примеры тестовых ситуаций, которые наглядно показывают проявления методов оптимизации.

# Заключение

Даже эмбрион знает, как писать заключение :)