|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ В РФ | | | | | | | | | |
|  | | | |  | | |  | | |
|  | | ФГБОУ ВО «Пермский государственный национальный исследовательский университет» | | | | | |  | |
|  | | | |  | | |  | | |
|  | | Отчёт  по дисциплине «Формальные грамматики и методы трансляции» | | | | | |  | |
|  | | |  | | | |  | | |
|  | Работу выполнил  студент гр. ПМИ-4-18  Колесников А.С. \_\_\_\_\_\_\_\_\_  (подпись)  «\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2022 | | | |  | Проверил  ассистент кафедры МОВС  Пономарев Ф.А.\_\_\_\_\_\_\_\_\_  (подпись)  «\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2022 | | |  |
|  | | | | | | | | | |
| Пермь 2022 | | | | | | | | | |

Оглавление

[1 Структура компилятора 3](#_Toc94468823)

[2 Модуль ввода-вывода 4](#_Toc94468824)

[3 Лексический анализатор 5](#_Toc94468825)

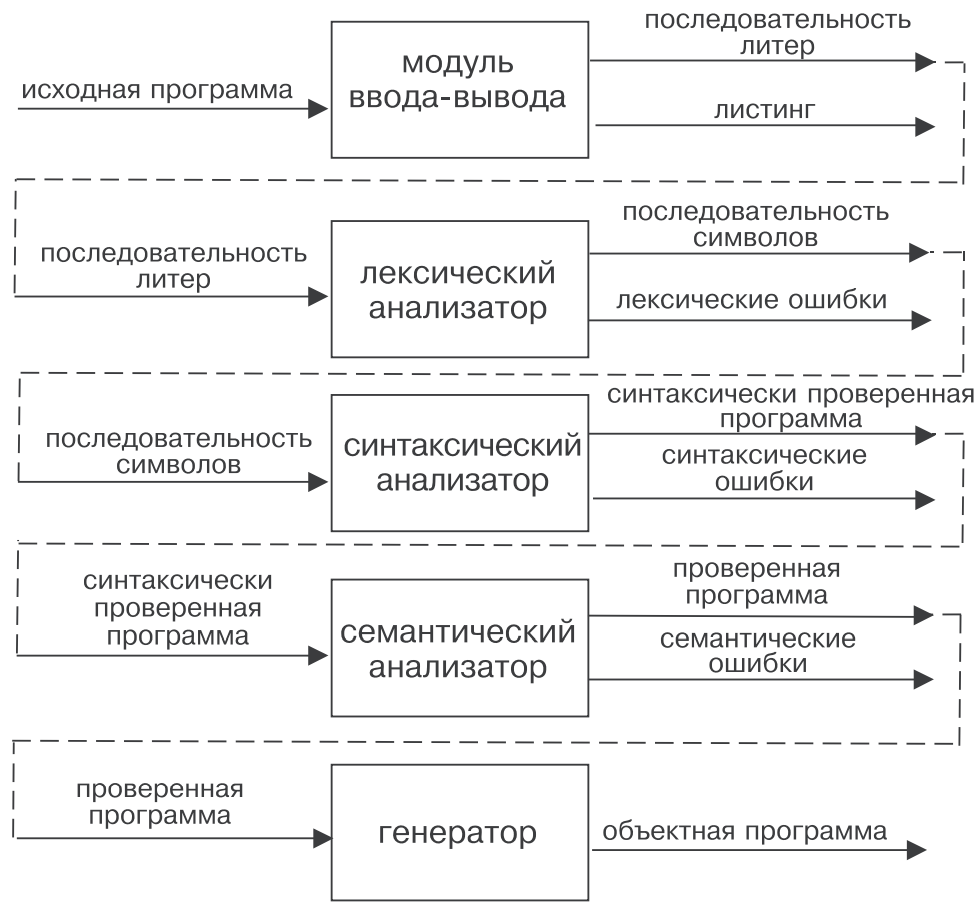
[4 Синтаксический анализатор 6](#_Toc94468826)

[5 Семантический анализатор 7](#_Toc94468827)

1. Структура компилятора

Компилятор – это программа которая переводит программу на языке высокого уровня в эквивалентную программу на другом языке. Обычно компилятор также выдает листинг, содержащий текст исходной программы и сообщения обо всех обнаруженных ошибках.

Разработка программного обеспечения подразумевает модульность и хорошую структурированность программ. Учитывая это, представим компилятор как совокупность логически взаимосвязанных модулей (рис. 1).



Рисунок

1 – Структура компилятора

Модуль ввода вывода отвечает за все действия, связанные с чтением и записью данных. Для каждого типа устройства может существовать свой собственный модуль ввода-вывода.

Лексический анализатор осуществляет простой лексический анализ исходной программы. На вход он получает последовательность литер, а отдает последовательность лексем.

Синтаксический анализатор проверяет, удовлетворяет ли программа формальным правилам.

Семантический анализатор проверяет не нарушены ли неформальные правила описания языка.

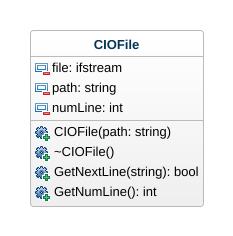
Генератор отображает независимое промежуточное представление исходной программы на реальную ЭВМ, поэтому должен переписываться для каждой новой машины.

1. Модуль ввода-вывода

Модуль ввода-вывода считывает последовательность литер исходной программы с внешнего устройства и передает их анализатору. Анализатор проверяет, удовлетворят ли эта последовательность литер правилам описания языка, и формирует в случае необходимости сообщения об ошибках.

Будем считать, что в результате очередного обращения к модулю ввода-вывода анализатор получает текущую строку.

Для печати сообщения об ошибке, анализатор должен передать модулю ввода-вывода причину и местоположение ошибки, следовательно, анализатору необходимо знать координаты всех литер во входном потоке.



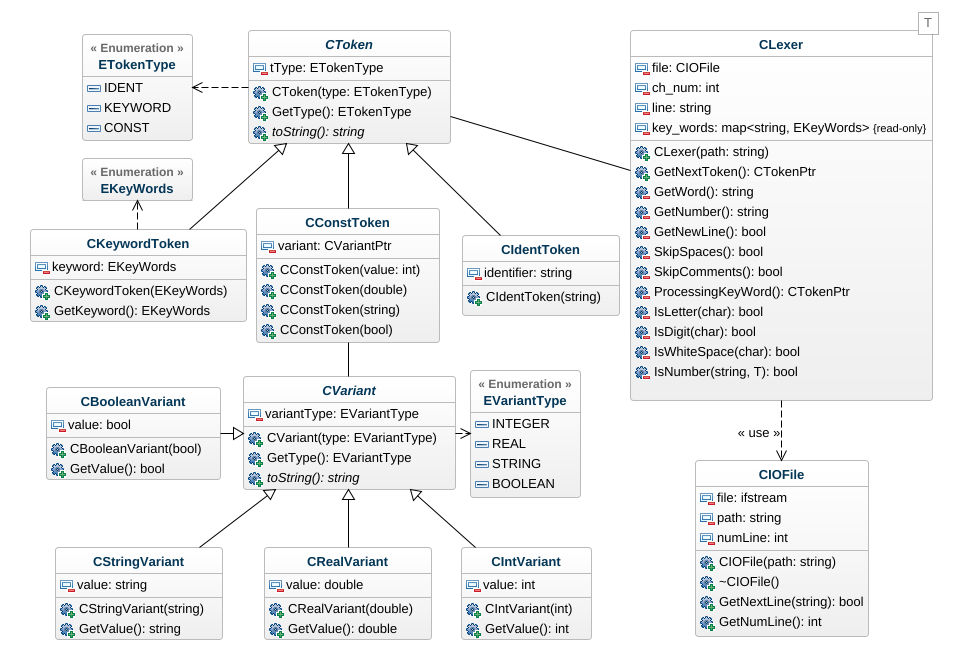
Рисунок

– Структура класса модуля ввода-вывода

1. Лексический анализатор

Лексический анализатор формирует лексемы исходной программы и строит их внутреннее представление.

На рисунке 3 представлена диаграмма классов, которые использует лексический анализатор для построения внутренней модели.



Рисунок

– Диаграмма классов лексического анализатора

Класс CToken базовый абстрактный класс для всех токенов.

Перечисление ETokenType содержит в себе все типы возвращаемых токенов классом CToken.

Класс CKeywordToken производный от CToken и отвечает за ключевые слова. Содержит тип ключевого слова, который представлен перечислением EKeyWords.

Класс CIdentToken производный от CToken и отвечает за хранения пользовательских идентификаторов.

Класс CConstToken производный от CToken и отвечает за хранения констант. Содержит указатель на CVariant.

Класс CVariant базовый абстрактный класс для типов данных. Хранит тип данных, представленный в виде перечисления EVariantType.

Класс CBooleanVariant отвечает за хранение логических значений типа boolean.

Класс CStringVariant отвечает за хранение строковых значений типа string.

Класс CRealVariant отвечает за хранение чисел с плавающей точкой типа real.

Класс CIntVariant отвечает за хранение целочисленных значений типа integer.

Класс CLexer отвечает за обработку кода и выдачи токенов по средством функции GetNextToken, которая возвращает указатель на класс CToken. Для считывания файла с кодом данный класс использует модуль ввода-вывода CIOFile.

1. Синтаксический анализатор

Синтаксический анализатор по мере необходимости запрашивает у модуля ввода вывода следующую.

1. Семантический анализатор