Во время работы с курсовым проектом был создан лексический анализатор. Лексический анализатор является частью компилятора, которая считывает литералы программы и строит из них лексемы. Лексический анализ обрабатывает исходный текст, полученный от пользователя, и распознает лексемы, а также классифицирует их.

Лексический анализатор выделяет из текста лексемы различных типов: идентификаторы, литералы (числовые и символьные константы), разделители. Выделение (сборка) лексемы сопровождается проверкои1 её правильности. Обнаруженные лексические ошибки фиксируются. Язык описания лексических единиц в большинстве случаев является регулярным, то есть может быть описан с помощью регулярных грамматик. Распознавателями регулярных языков являются конечные автоматы. Одним из способов описания конечного автомата является графическое его представление в виде маркированного однонаправленного графа, в котором узлы соответствуют состояниям конечного автомата, дуги отображают переходы из одного состояния в другое, а символы маркировки дуг соответствуют функции перехода конечного автомата.

### Руководство программиста

Входными данными для транслятора является текст программы, полученный от пользователя, с синтаксисом подмножества языка программирования C. Выходными данными транслятора являются таблица лексического разбора программы, список ключевых слов, разделителей, идентификаторов и литералов, используемых в программе после выполнения лексического анализа. В случае обнаружения ошибки в одном из анализаторов программа может вывести сообщение об обнаружении ошибки синтаксическим анализатором. Ниже представлены основные методы программы и их назначения:

Класс Recoznizer.cs включает в себя такие методы как: Begin, Programm, ListOfDescriptions, ListOfOperatoins, Descriptions, AddDescriptions, ListOfVariables, Type, U, Operator, Conditional, Assignment, Z, W, Option, ListOfOperations, S, X,Sign, Expression, Skip, Next. Все вышеупомянутые методы данного класса реализовывают работу нисходящего анализатора. Каждый метод представляет собой определенное грамматическое правило. Методы содержат в себе полученные лексемы из класса Token.cs.

Класс Validate.cs содержит следующие методы: IsLetters, IsNumbers, IsSeparator. Данный класс используется для проведения лексического анализа.

Класс Token.cs хранит два структуры данных, содержащих токены для классификации лексем (ключевые слова и разделители). С помощью этого класса проводится классификация лексем, для дальнейшей работы.

Класс Form1.cs отвечает за взаимодействие с пользователем, и запуском отдельных процессов транслятора. Button1.cs принимает на вход код от пользователя и выводит его на textbox2.cs Button2.cs принимает на вход список лексем, и проводит лексический анализ. Button3.cs, основываясь на данных, полученных от лексического анализа, проводит классификацию лексем