|  |  |
| --- | --- |
| МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ | |
| Федеральное государственное бюджетное образовательное учреждение высшего образования  «Пермский государственный национальный исследовательский университет» | |
| **ЧИСЛЕННЫЕ МЕТОДЫ** *Лабораторная работа №2*  **«Прямые методы решения систем линейных алгебраических уравнений»** | |
| Варианты № 1, 7 | |
|  | Работу выполнили студенты группы ПМИ-1,2:  Авраменко Дарья, Никитас Николай |
|  | Проверил:  профессор, доктор физико-математических наук  С. В. Русаков  “\_\_\_\_” 20\_\_ г. |
| Пермь 2018 | |

**1. Задание**

Разработать компилятор для программ на языке Pascal. Язык программирования, на котором должен быть написан компилятор, должен быть объектно-ориенитрованным.

**Проектирование**

Компиляция состоит из следующих фаз: модуль ввода-вывода, анализатор, генератор. Анализатор, в свою очередь, состоит из 3 модулей: лексический, синтаксический и семантический анализаторы. Весь компилятор принимает следующую структуру:

Модуль ввода-вывода получает исходный код программы и преобразует в последовательность литер

Лексический анализатор осуществляет лексический анализ, строит идентификаторы, ключевые слова, разделители, числа

Синтаксический анализатор проверяет, удовлетворяет ли программа формальным правилам

Генератор генерирует объектную программу

**Модуль ввода-вывода**

Модуль ввода-вывода содержит класс IO

public IO(string path).

В конструктор класса IO передается путь к файлу с текстом программы на языке Pascal, в строковое поле ProgramText

public string ProgramText { get; private set; }

считывается текст всей программы построчно. Из модуля ввода-вывода лексический анализатор берет очередной символ, для этого есть метод

public char Nextch()

Для проверки лексем с длиной более 1 необходимо брать следующий символ. В случае, если он не подошел к лексеме, нужно вернуться на символ обратно, для этого есть метод

public void Back()

**Лексический анализатор**

Я разделил лексемы на идентификаторы, ключевые константы, целые и вещественные числа, разделители, символы операции, спецификаторы. Конструктор анализатора получает экземпляр класса ввода вывода

public Lexical(IO Input\_Reader)

Лексема характеризуется типом, значением, номером строки и положением в строке

public struct Lexem

{

public int LinePosition;

public int LineNumber;

public object value;

public string type;

}

Получение следующей лексемы происходит в методе

public Lexem NextSym()

Идентификаторы и ключевые слова начинаются с латиницы, константы с цифры.

Вывод всех лексем в цикле в методе

public void PrintLexem()