# Выбор языка программирования

Для реализации компилятора языка Паскаль был выбран язык программирования C#.

# Модуль ввода-вывода

## Проектирование

Модуль ввода-вывода должен считывать исходный текст программы и позволять лексическому анализатору получать литеры по одной, при этом сохраняя информацию о позиции литеры. Также модуль ввода-вывода должен принимать ошибки для дальнейшего составления листинга ошибок. Каждая ошибка содержит позицию ошибки и код, по которому можно получить её описание.

## Реализация

Для реализации модуля был создан класс IOModule, который хранит номер текущей строки и позицию символа в этой строке.

Основной метод класса bool NextChar() считывает файл построчно и двигается по текущей строке, «передвигая» текущий символ на 1 позицию. Если метод встречает конец файла, он возвращает значение false, сигнализируя, что далее символов нет.

public *bool* NextChar()

    {

        if (current\_line == null)

            return false;

        if (charnumber == current\_line.Length)

        {

            current\_line = reader.ReadLine();

            while (current\_line == "")

                current\_line = reader.ReadLine();

            linenumber++;

            charnumber = 0;

        }

        if (current\_line == null)

            return false;

        current\_char = current\_line[(*int*)charnumber];

        charnumber++;

        return true;

    }

Также в экземпляре класса хранится список ошибок, который можно пополнять с помощью метода void AddError() и печатать с помощью метода PrintErrors, который принимает на вход функцию для печати.

    public *void* AddError(*uint* *error\_code*)

    {

        errors.Add(new *Error*(linenumber, charnumber, *error\_code*));

    }

    public *void* PrintErrors(*Action*<*string*> *Print*)

    {

        foreach (var error in errors)

*Print*(error.ToString());

    }

## Тестирование

Тестирование проводилось на файлах emptyfile.txt и pascal example.txt. Ошибки добавлялись «вручную».

IOModule iomodule = new IOModule("pascal example.txt");

while (iomodule.NextChar())

{

    Console.WriteLine(iomodule.Current\_char);

    iomodule.AddError(13);

}

iomodule.PrintErrors(Console.WriteLine);

Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст, экран, снимок экрана

Автоматически созданное описание

Изображение выглядит как текст, электроника

Автоматически созданное описание