Санкт-Петербургский государственный политехнический университет

Институт компьютерных наук и технологий

**Кафедра «Компьютерные системы и программные технологии»**

**Отчёт по курсовой работе**

**Дисциплина**: Прикладное программирование

**Тема**: Интерпретатор Tiny BASIC

Выполнил

студент гр.23531/3 \_\_\_\_\_\_\_\_\_ Ю.С. Макарова

Руководитель

\_\_\_\_\_\_\_\_\_ Т.Л. Сидорина

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2018 г

Санкт-Петербург

2018

Санкт-Петербургский государственный политехнический университет

**ЗАДАНИЕ**

**НА ВЫПолнение курсовой работы**

студенту группы 23531/3 Макаровой Юлии Сергеевне

*(номер группы) (фамилия, имя, отчество)*

***1. Тема работы:*** Интерпретатор Tiny BASIC

***2. Срок сдачи законченного проекта (работы)*** 13 сентября 2018 г.

***3. Исходные данные к проекту (работе)***: *описание необходимого функционала программы, документация языка Tiny BASIC, техническое задание.*

***4. Содержание пояснительной записки*** (перечень подлежащих разработке вопросов): введение, основная часть (*метод решения, методика испытаний)* заключение, список использованных источников, приложение.

***Дата получения задания***: «21» марта 2018 г.

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Т.Л. Сидорина

*(подпись) (инициалы, фамилия)*

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ю.С. Макарова

*(подпись студента) (инициалы, фамилия)*

\_\_\_\_\_\_\_\_\_\_\_\_

*(дата)*

**Содержание**

[**Введение** 4](#_Toc524598924)

[1. Метод решения 5](#_Toc524598925)

[2. Методика испытаний 7](#_Toc524598926)

[**Заключение** 8](#_Toc524598927)

[**Список использованных источников** 8](#_Toc524598928)

[**Приложение** 9](#_Toc524598929)

## **Введение**

В данном курсовом проекте будет реализован интерпретатор Tiny BASIC на языке С. Результат работы интерпретатора обеспечивается техническим заданием, согласованным с преподавателем, а также документацией интерпретируемого языка.

# Метод решения

Алгоритм работы простого интерпретатора:

1. прочитать инструкцию;
2. проанализировать инструкцию и определить соответствующие действия;
3. выполнить соответствующие действия;
4. если не достигнуто условие завершения программы, прочитать следующую инструкцию и перейти к пункту 2.

Для работы с выражениями будет использоваться синтактический анализатор выражений. Выражения разбиваются на составные части (лексемы). Функция, выполняющая разбор выражений, будет решать следующие действия:

- игнорировать пробелы и символы табуляции;

- извлекать каждую лексему из исходного текста;

- преобразовывать лексему во внутренний формат;

- определять тип лексемы.

Каждая лексема будет иметь два формата: внешний и внутренний. Каждому элементу выражения в результате разбора присваивается тип лексемы.

Исходный код программы на языке TINY BASIC пишется в текстовом редакторе. Вывод результатов происходит в выходной файл.

Пример входного файла:

10 I = 5

20 PRINT "SIMPLE TEST", I

30 I = I + 1

40 IF I < 15 THEN GOTO 20

Запуск программы производится с помощью командной строки. В функции main происходит выделение памяти для программы и обработка текста входного файла. В зависимости от типа лексемы происходит либо присваивание значения переменной, либо вызов функций реализации команд.

Также в программе реализован полный простой синтаксический анализатор рекурсивного спуска для целых числовых выражений.

Функция get\_token( ) возвращает следующую лексему в выражении. Она сканирует текст программы, анализируя по одному символу.

Функция look\_up() сравнивает внешний формат лексемы с таблицей лексем, определенной при разработке анализатора.

Функция putback() возвращает лексему во входной поток.

Анализатор поддерживает следующие операторы: "+", "-", "\*", "/". Все эти операции могут использоваться в сочетании с круглыми скобками.

# Методика испытаний

Для проверки правильности работы интерпретатора напишем текстовый файл следующего вида:

10 input i

20 if i < 3 then goto 50

30 print 100

35 gosub 100

40 print 34 - 6 \* 2

50 a = 100 / 3

60 if a > 5 then print a

120 end

100 print "Это подпрограмма"

110 return

При запуске программы напечатается слово «input:»

Выходной файл, если было введено число < 3:

33

Выходной файл, если было введено число >= 3:

100

Это подпрограмма

22

33

## **Заключение**

В ходе выполнения курсового проекта был реализован интерпретатор Tiny BASIC, работа которого обеспечивается документацией языка и согласованным с преподавателем техническим заданием. Правильность работы была проверена с помощью нескольких входных текстовых файлов. Код написан на основе книги Герберта Шилдта «C для профессиональных программистов», где подробно описаны детали разработки интерпретатора с примерами.

## **Список использованных источников**

1. Герберт Шилдт «C для профессиональных программистов» С [Электронный ресурс] // URL: http://aiut.tugab.bg/Library/Software/Doc/C/c\_za\_progr.pdf
2. Теория языков программирования и методы трансляции: Методические указания к выполнению практических занятий. – СПБ.: СЗТУ, 2003 – 36 с.
3. Справочник функций языка С [Электронный ресурс] // URL: <https://prog-cpp.ru/>

## **Приложение**

Код программы: https://github.com/JulyMak/tynybasic