МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное

образовательное учреждение высшего образования

ЧЕРЕПОВЕЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Институт информационных технологий

Математического и программного обеспечения ЭВМ

Модуль «Информатика»

УТВЕРЖДАЮ

Зав. кафедрой МПО ЭВМ

д. т.н. \_\_\_\_\_\_\_\_\_\_ Ершов Е.В.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.

Разработка алгоритмического обеспечения и построение

лексического анализатора компилятора

Техническое задание на курсовую работу

Листов 8

Руководители: доцент Ганичева О. Г., Пышницкий К.М.

Исполнитель: студент гр. 1ПИб-02-2оп-22

Зернов В.А

2024 год

Введение

Курсовая работа ставит целью разработку алгоритмического обеспечения и построения лексического анализатора, которое позволит пользователю наглядно увидеть лексемы в коде и понять часть принципа работы компилятора.

Данный проект предназначен для формирования знаний и навыков по дисциплине Модуль «Информатика».

1. Основания для разработки:

Основанием для разработки является задание на курсовую работу по дисциплине Модуль «Информатика», выданное на кафедре МПО ЭВМ ИИТ ЧГУ.

Дата утверждения: 13 февраля 2024 года.

Наименование темы разработки: «Разработка алгоритмического обеспечения и построение лексического анализатора компилятора».

1. Назначение разработки

Учебная программа должна моделировать работу лексического анализатора, которая решает следующие задачи: выделить все лексемы языка C++, знаки операций, построить КС-грамматику, удалять лишние пробелы, отступы, переходы на следующие строки, однострочные и многострочные комментарии, поддерживать строки, подсчитывать количество строк в программе и выводить ошибки при наличии некорректных лексем.

1. Требования к программе

3.1. Требования к функциональным характеристикам

3.1.1. Модель лексического анализатора должна обрабатывать текст из выбранных пользователем файлов в графическом интерфейсе, подготовить текст к последующей обработке путем удаления лишних пробелов, отступов, однострочных и многострочных комментариев, излишних символов новой строки, подсчитывать строки кода.

3.1.2. Программа должна находить ключевые слова и знаки отношений, записывать их в таблицу без повторений.

3.1.3. Программа должна формировать таблицы с классами лексем, найденными во входном коде.

3.1.4. Программа должна реализовывать построение дескрипторного кода и псевдокода.

3.1.5. Программа должна обнаруживать ошибки, указывать номер строки и идентифицировать ошибку. Также программа должна построить КС-грамматику для заданного в курсовой работе варианта задания.

3.2 Требования к надежности

Для правильного функционирования программы необходимо:

1. Входной файл должен иметь текстовый формат.
2. Кодировка символов исходного файла должна быть ASCII или совместимая с ней и содержать только ASCII символы.
3. Вводимые лексемы должны удовлетворять возможным лексемам, которые определены с помощью состояний конечного автомата.
   1. Условия эксплуатации

Программа должна быть совместима с операционной системой пользователя. Условия эксплуатации программы зависят от условий эксплуатации персонального компьютера.

3.4 Требования к составу и параметрам технических средств

* Тактовая частота процессора более 1 GHz;
* Количество оперативной памяти 1Гб или более;
* Свободное место на системном накопителе от 512 Мб и более;
* Операционная система Windows 7 и выше;
* Процессорная архитектура x86 или x64;
* Монитор;
* Клавиатура;
* Мышь.

3.5 Требования к информационной и программной совместимости

Проект написан на языке C++ с использованием среды разработки Visual Studio 2022. Для редактирования или сборки программы необходимы инструменты версии v142 и пакет SDK для Windows 10.0. Если требуется поддержка более ранних версий Windows, необходимо настроить проект и установить соответствующий SDK для нужной версии Windows.

3.6 Требования к маркировке и упаковке

Распространяется передачей исходных файлов проекта как на физических носителях, так и через другие средства передачи. Также может передаваться уже собранная версия программы третьим лицам. Авторское право сохраняется за автором исходного кода до явной передачи его третьим лицам.

3.7 Требования к транспортировке и хранению

Транспортировку и хранение рекомендуется осуществлять на оптическом носителе DVD-диске. При соблюдении условий хранения выбранного носителя срок хранения ограничивается сроком хранения хотя бы одной копии программного обеспечения на каком-либо носителе.

3.8 Специальные требования

Наличие минимальных познаний о языке программирования C++, знания некоторых теоретических сведений о работе и устройстве компиляторов, лексических анализаторов.

4. Требования к программной документации

4.1 Программная документация должна содержать расчетно-пояснительную записку с содержанием:

* Титульный лист
* Оглавление
* Введение
  + - * + Теоретическая часть
        + Практическая часть
* Заключение
* Список литературы
* Приложения.

4.2. Требования к оформлению:

* Документ;
* Печать на отдельных листах формата А4 (210х297 мм); оборотная сторона не заполняется; листы нумеруются. Печать возможна ч/б;
* Файлы предъявляются на компакт-диске: РПЗ с ТЗ; программный код;
* Листы и диск в конверте вложены в пластиковую папку скоросшивателя;
* Страницы;
* Ориентация – книжная; отдельные страницы, при необходимости, альбомная;
* Поля: верхнее, нижнее – по 2 см, левое – 3 см, правое – 1 см;
* Абзацы;
* Межстрочный интервал – 1,5, перед и после абзаца – 0;
* Шрифты;
* Кегль – 14. В таблицах шрифт 12. Шрифт листинга – 10;
* Рисунки;
* Подписывается под ним по центру: Рис. Х. Название;
* В приложениях: Рис.П1.3. Название;
* Таблицы;
* Подписывается: над таблицей, выравнивание по правому: «Таблица Х»;
* В следующей строке по центру Название;
* Надписи в шапке (имена столбцов, полей) – по центру;
* В теле таблицы (записи) текстовые значения – выровнены по левому, числа, даты – по правому.

5. Стадии и этапы разработки

В данном пункте представлены этапы разработки, которые будут выполняться при разработке программы, в соответствии с табл.1.

Таблица 1

Этапы разработки

| Наименование этапа разработки | Сроки разработки | Результат выполнения | Отметка о выполнении |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Сбор необходимой информации для выполнения данной курсовой работы | 12.03.2024 | Овладение необходимыми знаниями для написания кода программы |  |
| Анализ задачи и требований к проекту | 06.04.2024 | Выявление и составление четкого алгоритма программы |  |
| Реализация алгоритма по обработке кода | 15.03.2024 | Получение технически правильного кода программы |  |
| Написание функции для нахождения ключевых слов и знаков отношений | 08.03.2024 | Получение технически правильного кода программы |  |
| Написание функции для нахождения ключевых слов и всех знаков операций | 10.03.2024 | Получение технически правильного кода программы |  |
| Реализация построения дескрипторного кода и псевдокода | 13.03.2024 | Получение технически правильного кода программы |  |
| Написание функции по обнаружению ошибок и построению КС-грамматики | 18.03.2024 | Получение технически правильного кода программы |  |

Продолжение таблица 1

| Тестирование | 25.05.2024 | Исправление ошибок и сравнение результатов работы программы и самостоятельных подсчетов |  |
| --- | --- | --- | --- |
| Разработка РПЗ | 18.03.2024-25.05.2024 | Разработанное РПЗ |  |

1. Порядок контроля и приемки

Контроль выполнения работы осуществляется преподавателем. В случае обнаружения недостатков, исполнитель работы обязан исправить их в срок, согласованный с преподавателем. Контроль разработки продукта осуществляется поэтапно, в соответствии с графиком, представленным в табл. 2.

Таблица 2

Порядок контроля и приемки

| Наименование контрольного этапа выполнения курсовой работы | Сроки контроля | Результат выполнения | Отметка о приемке результата контрольного этапа |
| --- | --- | --- | --- |
| Сдача технического задания | 10.04.2024 | Документ “Техническое задание” проверен |  |
| Демонстрация работы программы | 27.05.2024 | Работа программы проверена |  |
| Оформление РПЗ | 29.05.2024 | Оформление РПЗ принято |  |
| Сдача РПЗ |  | Оценка |  |