МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных Технологий

Кафедра Программной инженерии

Специальность 1-40 01 01 Программное обеспечение информационных технологий

Специализация Программирование интернет-приложений

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К КУРСОВОМУ ПРОЕКТУ НА ТЕМУ:**

«Разработка компилятора MAD-2021»

Выполнил студент Мозолевский Александр Дмитриевич

(Ф.И.О.)

Руководитель проекта ст. пр. Наркевич Аделина Сергеевна

(учен. степень, звание, должность, подпись, Ф.И.О.)

Заведующий кафедрой к. т. н., доц. Пацей Н.В.

(учен. степень, звание, должность, подпись, Ф.И.О.)

Консультанты ст. пр. Наркевич Аделина Сергеевна

(учен. степень, звание, должность, подпись, Ф.И.О.)

(учен. степень, звание, должность, подпись, Ф.И.О.)

Нормоконтролер ст. пр. Наркевич Аделина Сергеевна

(учен. степень, звание, должность, подпись, Ф.И.О.)

Курсовой проект защищен с оценкой

**Введение**

Задачей данного курсового проекта была поставлена разработка транслятора своего языка программирования MAD-2021. Он предназначен для выполнения простейших операций и арифметических действий над строками и числами.

Главной задачей транслятора заключается в том, чтобы сделать исходный код на данном языке программирования понятной компьютеру. Для решения этой задачи был выбран способ трансляции исходного кода моего языка программирования в исходный код на языке ассемблера. Язык ассемблера – это машинно-ориентированный язык, представляющий формат записи машинных команд, которые понятны для восприятия человеком.

Исходя из цели курсового проекта, были определены следующие задачи:

– разработка спецификации языка программирования;

– разработка структуры транслятора;

– разработка лексического анализатора;

– разработка синтаксического анализатора;

– разработка семантического анализатора;

– обработка выражений с помощью польской инверсии;

– генерация кода на язык ассемблера;

– тестирование транслятора;

Способы решения каждой задачи будут описаны в соответствующих главах курсового проекта.

В первой главе работы определена спецификация языка программирования.

Во второй главе представлена структура транслятора. В ней перечислены компоненты транслятора, их назначения и принципы взаимодействия.

В третьей главе описана разработка лексического анализатора, который создаёт таблицы лексем и идентификаторов.

В четвертой главе описана разработка синтаксического анализатора, который выполняет разбор исходного кода в соответствии с правилами языка программирования.

В пятой главе …

**Глава 1. Спецификация языка программирования**

**1.1 Характеристика языка программирования**

Язык программирования MAD-2021 – это строго типизируемый, процедурный язык высокого уровня, который транслируется в язык ассемблера в 2 этапа:

– Изначально исходный код транслируется в байт-код (промежуточное представление);

– Затем происходит перевод байт-кода в язык ассемблера.

**1.2 Алфавит языка**

Алфавит языка программирования – набор символов, которые могут использоваться при написании исходного кода.

Язык программирования MAD-2021 включает в себя кириллицу и символы латинского алфавита верхнего и нижнего регистров, арабские цифры, знаки препинания, знаки арифметических и логических операций.

**1.3 Символы сепараторы**

Сепараторы необходимы для разделения операций языка. Сепараторы, используемые в языке программирования MAD-2021, приведены в таблице 1.1.

Таблица 1.1 – Сепараторы

|  |  |
| --- | --- |
| Сепаратор | Назначение |
| ;  :  « » (пробел)  “  , | Разделение конструкций |
| =  +  -  \*  % | Арифметические операции |
| {} | Программный блок инструкций |
| () | Параметры функций, изменение приоритетов в выражениях |
| [] | Оператор доступа (индексатор) |

**1.4 Применяемые кодировки**

Для написания исходного кода на языке программирования MAD-2021 используется кодировка Windows-1251 – набор символов и кодировка, являющаяся стандартной 8-битной кодировкой для русских версий Microsoft Windows до 10-й версии – представленная на рисунке 1.1.

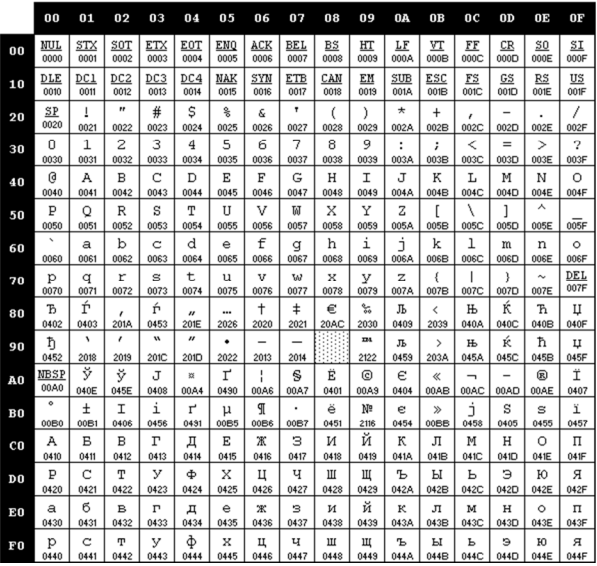


Рисунок 1.1 – Кодировка Windows-1251

**1.5 Типы данных**

В языке программирования MAD-2021 используются три основных типа данных, которые описываются в таблице 1.2. Пользовательские типы данных не поддерживаются.

Таблица 1.2 – Типы данных

|  |  |
| --- | --- |
| Тип данных | Описание типа данных |
| \_int | Фундаментальный тип данных, используемый для объявления целочисленных данных. Без явно указанной инициализации переменной, присваивается нулевое значение. |
| \_string | Фундаментальный тип данных, используемый для объявления строк. Без явно указанной инициализации переменной, присваивается нулевое значение (пустая строка). |
| \_bool | Фундаментальный тип данных, используемый для объявления логической переменной, которая принимает одно из двух значений: true или false. Без явно указанной инициализации переменной, присваивается нулевое значение (false). |

**1.6 Преобразование типов данных**

В языке программирования MAD-2021 преобразование типов данных не поддерживается.

**1.7 Идентификаторы**

Идентификаторы применяются для наименования переменных, функция и параметров. Идентификаторы, объявленные внутри функционального блока, получают префикс, который отображается в таблице идентификаторов. Зарезервированные идентификаторы не поддерживаются. Предусмотрены несколько правил составления идентификатора:

– состоит из символов латинского алфавита любого регистра и цифр;

– могут начинаться с символа нижнего подчеркивания «\_»;

– максимальная длина идентификатора равна 10. При превышении длины идентификатора она будет урезаться;

– идентификатор не может совпадать с ключевыми словами языка программирования.

– регулярное выражение для разбора идентификатора выглядит следующим образом: [\_]\*[a-zA-Z0-9]+

**1.8 Литералы**

В языке программирования MAD-2021 существует только 2 типа литералов: целые и символьные. Их краткое описание представлено в таблице 1.3.

Таблица 1.3 – Литералы

|  |  |
| --- | --- |
| Тип литерала | Описание |
| Литералы целого типа | Целочисленные литералы, только rvalue. |
| Строковые литералы | Символы, заключенные в двойные кавычки (“”), только rvalue. Максимальное число символов в литерале – 255. |

**1.9 Область видимости идентификаторов**

Область видимости в языке программирования MAD-2021 работает по принципу C++ («сверху вниз»). Перед использованием переменной необходимо её объявление. Допускается использование переменной только внутри её области видимости. Допускается объявление переменных с одинаковыми именами в разных программных блоках.

**1.10 Инициализация данных**

Способы инициализации переменных языка программирования MAD-2021 представлены в таблице 1.4.

Таблица 1.4 – Способы инициализации переменных

|  |  |
| --- | --- |
| Вид инициализации | Примечание |
| let <тип данных> <идентификатор>; | Автоматическая инициализация переменной. \_int – инициализируется нулем, \_str – пустой строкой, \_bool – false. |
| let <тип данных> <идентификатор> = <значение>; | Инициализация переменной с присваиванием значения. |

**1.11 Инструкции языка**

Все возможные инструкции языка программирования MAD-2021 представлены в общем виде в таблице 1.5.

Таблица 1.5 – Инструкции языка программирования MAD-2021

|  |  |
| --- | --- |
| Инструкция | Запись |
| Объявление переменной | let <тип данных> <идентификатор>;  let <тип данных> <идентификатор> = <выражение>; |
| Присваивание | <идентификатор> = <выражение >; |
| Объявление функции | func <тип данных> <идентификатор> ([<тип данных> <идентификатор>][, <тип данных> <идентификатор>]\*) {…} |
| Блок инструкций | {  …  } |
| Возврат из подпрограммы | ret <выражение>; |
| Вывод данных | write(<выражение>); |
| Оператор цикла | repeat (<условие>)  {…} |

**1.12 Операции языка**

Язык программирования MAD-2021 может выполнять арифметические операции, представленные в таблице 1.6.

Таблица 1.6 – Операции и их приоритеты

|  |  |
| --- | --- |
| Операция | Приоритет операции |
| (  ) | 0 |
| , | 1 |
| +  - | 2 |
| \*  / (деление с остатком)  % (деление по модулю) | 3 |

**1.13 Выражения и их вычисления**

В выражении должны участвовать операторы и операнды одного типа, а также функции, возвращающие значения того же типа. Круглые скобки в выражении используются для изменения приоритетов операций. Не допускается запись двух подряд арифметических операций. Также круглые скобки могут использоваться для передачи параметров функций. Фигурные скобки содержат блоки кода функций и циклов.

**1.14 Программные конструкции языка**

Ключевые программные конструкции языка программирования MAD-2021 представлены в таблице 1.7

Таблица 1.7 – программные конструкции

|  |  |
| --- | --- |
| Главная функция (точка входа в приложение) | main  {…} |
| Функция | func <тип данных> <идентификатор> ([<тип данных> <идентификатор>][, <тип данных> <идентификатор>]\*)  {  …  ret <выражение>;  } |

**1.15 Область видимости**

В языке программирования MAD-2021 переменные обязаны находиться внутри программного блока функций. Внутри разных областей видимости разрешено объявление переменных с одинаковыми именами. Все переменные, параметры или функции внутри области видимости получают префикс, который отображается в таблице идентификаторов. Объявление глобальных переменных не предусмотрено. Объявление пользовательских областей видимости не предусмотрено.

**1.16 Семантические проверки**

**…**

**1.17 Распределение оперативной памяти на этапе выполнения**

Все переменные размещаются в стеке. Таблицы лексем и идентификаторов размещены в структуры с выделенной под них оперативной памятью, которая очищается по окончанию работы транслятора.

**1.18 Стандартная библиотека и её состав**

**…**

**1.19 Ввод и вывод данных**

В языке программирования MAD-2021 ввод данных не поддерживается. Вывод данных происходит с помощью функции write(<выражение>);

**1.20 Точка входа**

Точкой входа в языке программирования MAD-2021 является функция main.

**1.21 Препроцессор**

В языке программирования MAD-2021 препроцессор не предусматривается.

**1.22 Соглашения о вызовах**

**…**

**1.23 Объектный код**

Язык программирования MAD-2021 транслируется в язык ассемблера.

**1.24 Классификация сообщений транслятора**

В случае возникновения ошибки в исходном коде программы на языке программирования MAD-2021 и выявлении её транслятором в файл протокола выводится сообщение. Классификация обрабатываемых ошибок приведена в таблице 1.10.

Таблица 1.?? – Классификация сообщений транслятора

|  |  |
| --- | --- |
| Интервал | Описание ошибок |
| 0-110 | Системные ошибки |
| 200-220 | Ошибки лексического анализа |
| 600-620 | Ошибки синтаксического анализа |
| 300-400 | Ошибки семантического анализа |

**1.25 Контрольный пример**

Контрольные примеры представлены в приложении А.

**Глава 2. Структура транслятора**

**2.1 Компоненты транслятора, их назначение и принципы взаимодействия**

Транслятор преобразует программу, написанную на языке программирования MAD-2021 в программу на языке ассемблера. Компонентами транслятора являются лексический, синтаксический и семантический анализаторы, а также генератор кода на языке ассемблера. Принцип взаимодействия представлен на рисунке 2.1.



Рисунок 2.1 – Структура транслятора

Лексический анализ – первая фаза трансляции. На вход лексический анализатор получает исходный код на языке программирования MAD-2021, в котором сепараторами были разделены слова. Задачей лексического анализатора является нахождение лексических ошибок и формирование таблиц лексем и идентификаторов.

Синтаксический анализ – это основная часть транслятора, предназначенная для распознавания синтаксических конструкций. Входным параметров для синтаксического анализа является таблица лексем. Синтаксический анализ распознаёт синтаксические конструкции, выявляет синтаксические ошибки при их наличии и формирует дерево разбора.

Семантический анализ, в свою очередь, является проверкой исходного кода программы на семантическую согласованность с определением конструкций языка, то есть проверяет правильность текста исходной программы с точки зрения семантики.

Генератор кода – этап транслятора, выполняющий генерацию ассемблерного кода на основе полученных данных на предыдущих этапах трансляции. Генератор кода принимает на вход таблицы идентификаторов и лексем и транслирует кода на языке программирования MAD-2021 в код на языке Ассемблера.

**2.2 Перечень входных параметров**

Входные параметры представлены в таблице 2.1.

Таблица 2.1 Входные параметры транслятора языка MAD-2021

|  |  |  |
| --- | --- | --- |
| Входной параметр | Описание | Значение по умолчанию |
| -in:<имя файла> | Входной файл с расширением .txt, в котором содержится исходный код на языке MAD-2021 | Не предусмотрено |
| -log:<имя файла> | Файл для записи протокола работы транслятора | <имя файла>.log |
| -out:<имя файла> | Файл для записи результата работы транслятора | <имя файла>.out.asm |
| -tokens | Ключ для вывода промежуточного представления кода | Отсутствует |
| -lex | Ключ для вывода таблицы лексем в консоль | Отсутствует |
| -id | Ключ для вывода трассировки синтаксического анализа в файл | Отсутствует |

**2.3 Перечень протоколов, формируемых транслятором и их содержимое**

Перечень протоколов, формируемых транслятором языка программирования MAD-2021 и их назначением представлен в таблице 2.2.

Таблица 2.2 – Протоколы, формируемые транслятором языка MAD-2021

|  |  |
| --- | --- |
| Формируемый протокол | Описание протокола |
| Файл журнала с параметром <log> | Содержит информацию о времени выполнения приложения; входных параметрах в приложение; код на языке программирования MAD-2021 с сепараторами и без избыточных пробелов, табуляций и переходов на новую строку; таблицы лексем и идентификаторов; промежуточное представление кода; трассировку синтаксического анализа; дерево разбора и время выполнения разбора; промежуточное представление кода после приведения его к польской нотации. |
| Выходной файл с параметром <out> | Содержит сгенерированный код на языке Ассемблера |