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 SES-2023»

Выполнил студент Сафонов Евгений Сергеевич

(Ф.И.О.)

Руководитель проекта ст. преподаватель Наркевич Аделина Сергеевна

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

Заведующий кафедрой к.т.н., доц. Смелов В.В

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

Консультанты Фамилия Имя Отчество

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

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

Нормоконтролер Фамилия Имя Отчество

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

Курсовой проект защищен с оценкой

Минск 2023

**Введение**

Данный курсовой проект представляет из себя создание собственного языка программирования SES-2023. Этот язык программирования представляет из себя простой учебный язык, который позволяет выполнять простейшие операции, арифметические действия над числами, создавать процедуры и функции и т.д.

Для создания собственного языка программирования требуется разработать собственный компилятор. В задачи компилятора входит: лексический анализ, семантический анализ и синтаксический анализ.

Исходя из раннее определённой цели курсового проекта, были определены следующие задачи:

* разработка спецификации языка программирования;
* разработка структуры транслятора;
* разработка лексического анализатора;
* разработка синтаксического анализатора;
* преобразование выражений;
* генерация кода на язык ассемблер;
* тестирование транслятора.

Далее пояснительная записка содержит решения выше поставленных задач.

****1 Спецификация языка программирования****

****1.1 Характеристика языка программирования****

Язык программирования SES-2023 – процедурный, компилируемый, строго типизированный язык.

****1.2 Определение алфавита языка программирования****

**Язык программирования SES-2023 использует стандартную таблицу символов Windows-1251, в которой определены латинские и русские символы, символы-разделители, символы математических операций и специальные символы.**

****1.3 Применяемые сепараторы****

**В языке SES-2023 определён набор символов сепараторов для разделения лексем друг от друга. Все сепараторы приведены ниже в таблице 1.1.**

|  |  |
| --- | --- |
| **Сепаратор** | **Назначение сепаратором** |
| **;** | **Разделитель инструкций** |
| **{}** | **Программный блок** |
| **()** | **Параметры функции, приоритет операций** |
| **,** | **Разделитель параметров** |
| **Пробел** | **Разделитель конструкций и лексем** |

Таблица 1.1 - Применяемые сепараторы

****1.4 Применяемые кодировки****

**Для написания исходного кода программы на языке SES-2023 используются латинские символы и символы разделители кодировки Windows-1251. Русские символы допускается использовать в строковых литералах.**

****1.5 Типы данных****

**В языке программирования предусмотрены следующие типы данных: целочисленный int (4 байта), целочисленный byte (1 байт), string строковый.**

|  |  |
| --- | --- |
| **int** | Фундаментальный тип данных. Автоматическая инициализация 0;  Возможные операции:   * арифметические(+, -, \*,/); * операции сравнения(<, >,>=,<=,==)   Возможные значения : -2 147 483 648 до 2 147 483 647 включительно. |
| **byte** | Фундаментальный тип данных. Автоматическая инициализация 0;  Возможные операции:   * арифметические(+, -,\*,/); * операции сравнения(<, >,>=,<=,==);   Возможные значения : -128 до 127 включительно. |
| string | Фундаментальный тип данных. Предусмотрен для объявления строк. (1 символ – 1 байт). Автоматическая инициализация строкой нулевой длины. Применяется кодировка Windows-1251. |
| bool | Фундаментальный тип данных. Занимает 1 байт, значение 0 или false или любое отличное от 0 или true, автоматическая инициализация 0 . |

Таблица 1.2 - Фундаментальные типы данных

****1.6 Преобразование типов данных****

**В языке SES-2023 доступно преобразование из byte в int, но не наоборот.** Все остальные типы данных определенны однозначно и не могут быть преобразованы в другие.

****1.7**** Идентификаторы

В языке SES-2023 идентификаторы являются именами для различных переменных, функций и параметров для этих функций. Не предусмотрено дублирование идентификаторов в одной области видимости. Так же имя идентификатора не может совпадать с каким-либо ключевым словом языка. Может состоять только латинских букв, цифр и знака нижнего подчёркивания. Первой в имени идентификатора не может быть цифра.

Примеры правильного идентификаторов: idenf, idenf123, iden12f, idenf\_123

**Пример неправильных идентификаторов: 12idenf, 123**

****1****.8 Литералы

Литерал в языке программирования — это константное значение, напрямую вписанное в исходный код программы.

Литералы могут быть представлены в исходном коде программы различными способами. Например:

* целочисленные литералы: 42, 0xFF;
* литералы чисел с плавающей точкой: 3.14, 1.0e-5f;
* символьные литералы: 'A', 'b', '\n';
* строковые литералы: "Hello, world!", "123”;
* булевые литералы: true, false;

Численные литералы записываются как набор цифр для десятичного представления или как 0xff для записи в шестнадцатеричном представлении.

****1.9 Объявление данных****

**Для использования переменной она должна быть объявлена до этого момента. Переменная может быть объявлена в любом программном блоке, если она уже не была объявлена в данной области видимости. Также переменная может иметь глобальную область видимости, если будет объявлена вне любого программного блока.**

**Правила объявления переменной:**

<тип данных> <идентификатор>;

1.10 Инициализация данных

При объявлении переменной допускается инициализация данных. Объектами-инициализаторами могут быть идентификаторы, литералы, выражения и вызов функции. Предусмотрены значения по умолчанию, если переменные не инициализированы: 0 – для целочисленных типов данных, пустая строка (строка размером 0) – для строкового типа данных, 0 (false) – для логического типа данных.

Переменные инициализируются с помощью операции присваивания. Тип данных должен соответствовать типу объявленной переменной, которую нужно инициализировать.

Правила инициализации представлены ниже:

<тип данных> <идентификатор>=<данные>;

1.11 Инструкции языка

Все инструкции существующие в языка SES-2023 представлены на таблице 1.3 ниже.

Таблица 1.3 — Инструкции языка

|  |  |
| --- | --- |
| Инструкция языка | Синтаксис |
| Вызов функций | <идентификатор  функции>(<идентификатор  /  литерал>,…) |
| Возврат из функции | return <идентификатор> | <литерал> | <выражение>; |
| Объявление переменной | <тип данных> <идентификатор> ; |
| Присваивание | <идентификатор>=<литерал>|<идентификатор>; |
| Вывод данных | write <идентификатор> | <литерал>;  writeline <идентификатор> | <литерал>; |
| Объявление внешней функции | <тип данных> function <идентификатор> (<тип данных> <идентификатор>, …) {<тело\_функции>}  <тело\_функции>::=<инструкция><набор\_инструкций>|<инструкция> |

****1.12 Операции языка****

**Язык SES-2023 может выполнять различные арифметические операции и операции сравнения. Все операции представлены в таблице 1.4.**

**Таблица 1.4— Операции языка**

|  |  |  |  |
| --- | --- | --- | --- |
| Операция | Арифметическое назначение | Приоритетность операции | Свойства |
| () | Приоритетность операций | 0 | Ассоциативность, коммутативность, дистрибутивность |
| + | Сложение | 1 | Коммутативность, ассоциативность |
| - | Вычитание | 1 | Антикоммутативность,  ассоциативность |
| \* | Умножение | 2 | Коммутативность,  ассоциативность |
| / | Деление | 2 | Антикоммутативность, ассоциативность |
| >,>= | Сравнение чисел | 3 | Антикоммутативность |
| <,<= | Сравнение чисел | 3 | Антикоммутативность |
| == | Сравнение чисел | 3 | Коммутативность |

**Порядок выполнения операций SES-2023 одинаковым приоритетом следует левоассоциативности.**

****1.13 Выражения и их вычисления****

Предусмотрены следующие правила составления выражений:

* выражения записываются до ввода сепаратора ‘;’;
* выражение может содержать вызов функций;
* рассматриваются слева направо;
* для изменения приоритета операция используются круглые скобки ();

1.14 Конструкции языка

Ключевые программные конструкции языка программирования SES-2023 представлены в таблице 1.5.

**Таблица 1.5 — Конструкции языка**

|  |  |
| --- | --- |
| Конструкция | Представление в языке |
| Главная функция | main  {  …  }; |
| Функция | <тип данных> function <идентификатор> (<идентификатор>, …)  {  …  return <идентификатор\литерал>;  }; |
| Условный оператор | if (<условие>) {<инструкции языка>}else {<инструкции языка>} |
| Цикл | while(<условие>){<инструкции языка>}; |

****1.15 Область видимости идентификаторов****

**Все идентификаторы, объявленные внутри блока, являются локальными. Идентификаторы переменные внутри какой-либо функции, доступны только лишь в ней. Это же распространяется и для параметров.** Объявление пользовательских областей видимости не предусмотрено.

****1.16 Семантические проверки****

Перечень семантических проверок, предусмотренных языком, приведен в таблице 1.7.

Таблица 1.7 - Перечень семантических проверок

|  |  |
| --- | --- |
| Номер | Правило |
| 1 | Идентификаторы функций не должны повторяться |
| 2 | Операнды в операторах ветвления и выхода из функции должны быть целочисленного типа |
| 3 | Тип данных передаваемых значений в функцию должен совпадать с типом параметров при её объявлении |
| 4 | Идентификатор должен быть объявлен до его использования. |
| 5 | Операнды в арифметическом выражении не могут быть различных типов |
| 6 | Тип возвращаемого функцией значения должен совпадать с типом функции |

1.17 Распределение оперативной памяти на этапе выполнения

Транслированный код использует две области памяти. В сегмент констант заносятся все литералы. В сегмент данных заносятся переменные, параметры функций. Локальная область видимости в исходном коде определяется за счет использования переменной, хранящей имя родительского блока, что и обуславливает их локальность на уровне ассемблерного кода, у глобальных переменных имя родительского блока отсутствует.

1.18 Стандартная библиотеки и её состав

В языке SES-2023 присутствует стандартная библиотека, которая автоматически подключается при трансляции исходного кода. Содержимое библиотеки и описание функций представлено в таблице 1.8.

Таблица 1.8 - Стандартная библиотека языка SES-2023

|  |  |
| --- | --- |
| Функция | Описание |
| int factor(int x) | Математическая функция. Принимает в качестве параметра целое число. Возвращает произведение всех чисел от 1 до заданного числа. |
| Int power(int base, int exponent) | Математическая функция. Принимает в качестве параметров два целочисленных значения. Возвращает целочисленное число, значение, которого является параметр base возведённый в степень exponent. |

****1.19 Ввод и вывод данных****

Ввод данных в языке SES-2023 не предусмотрен. Вывод данных осуществляется с помощью ключевого слова “write” или “writeline” для вывода строка с переводом каретки. Есть возможность выводить значение переменных и строк, которые будут заключены в двойные кавычки.

****1.20 Точка входа****

Точкой входа в программе является ключевое слово “main”. Точка входа не может отсутствовать или быть переопределена.

1.21 Препроцессор

Препроцессор в языке программирования SES-2023 не предусмотрен.

1.22 Соглашения о вызовах

Используется соглашение stdcall, все параметры передаются в стек справа налево, память освобождается вызываемым кодом.

1.23 Объектный код

Язык ses-2023 транслируется в исходный код для языка ассемблера.

1.24 Классификация сообщений транслятора

Классификация ошибок представлена в таблице 1.9.

Таблица 1.9 - Классификация ошибок

|  |  |
| --- | --- |
| Номера ошибок | Характеристика |
| 0 – 99 | Системные ошибки |
| 100 – 104 | Ошибки входных параметров |
| 105 – 109 | Ошибки при открытии файла |
| 110 – 119 | Ошибки при чтении файла |
| 120 – 140 | Ошибки лексического анализа |
| 600 – 610 | Ошибки синтаксического анализа |
| 700 – 710 | Ошибки семантического анализа |

Обрабатываются ошибки на всех этапах обработки исходного кода, то есть во время прохождения различных этапов анализа.

1.25 Контрольный пример

|  |
| --- |
| function example(int a)  {  int b=5;  int b=b+a;  return b;  } main()  {  string str=”hello world”;  write str;  writeline ”Hello world!”;  byte t;  t=fact(3);  t=power(t,3);  if(t>15)  {  writeline(t);  } } |