МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПЕУБЛИКИ БЕЛАРУСЬ

Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий   
Кафедра программной инженерии

Утверждаю

Заведующая кафедрой

\_\_\_\_\_\_\_\_\_\_\_\_ Н.В. Пацей

подпись инициалы и фамилия

“\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2018г.

**ЗАДАНИЕ**

**к курсовому проектированию**

**по дисциплине** "Языки программирования"

Специальность: ПОИТ Группа:

Студент: Астахова Анастасия Алексеевна

(фамилия, имя, отчество)

**1. Тема проекта** Разработка компилятора ААА-2018

утверждена приказом по университету от «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2018 г. №

**2. Срок сдачи студентом законченного проекта:** \_\_\_ декабря 2018 г.

**3. Исходные данные к проекту:**

Разработка программы осуществляется на языке C++ (стандартизации International Standard ISO/IEC 14882:2014(E) Programming Language C++ 14) в среде разработки Visual Studio 2017. Операционная система под которой происходит разработка Windows 10 PRO(64-bit). Типы данных: integer и string. Функции библиотеки: min(integer, integer), max(integer, integer), fact(integer), strlen(string). Арифметические операции: +, -, \*,/. Оператор вывода в стандартный поток: out.

**4. Содержание расчетно-пояснительной записки (перечень подлежащих разработке вопросов):**

Введение

1) Спецификация языка программирования

2) Структура транслятора

3) Разработка лексического анализатора

4) Разработка синтаксического анализатора

5) Разработка семантического анализатора

6) Вычисление выражений

7) Генерация кода

8) Тестирование транслятора (Разработка и тестирование интерпретатора)

Приложения

Литература

**5. Перечень графического материала (с точным указанием обязательных чертежей)**

1) Граф

**6. Консультанты по проекту с указанием относящихся к ним разделов проекта**

|  |  |
| --- | --- |
| Раздел | Консультант |
| Разработка синтаксического и семантического анализатора. | Наркевич А. С. |
| Генерация кода. Разработка тестовых примеров. | Наркевич А. С. |
| Оформление пояснительной записки к курсовому проект. | Наркевич А. С. |
|  |  |

**7. Календарный план**

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Наименование этапов курсового проекта | Срок выполнения этапов проекта | Примечание |
| 1 | Спецификация специализированного языка ААА-2018 |  |  |
| 2 | Разработка лексического анализатора |  |  |
| 3 | Разработка синтаксического анализатора |  |  |
| 4 | Разработка семантического анализатора |  |  |
| 5 | Генерация кода |  |  |
| 6 | Тестирование компилятора |  |  |
| 7 | Оформление пояснительной записки к курсовому проект |  |  |
| 8 | Сдача проекта |  |  |

**8. Дата выдачи задания 15.09.2018**

Руководитель Наркевич А. С.

(подпись)

Задание принял к исполнению Астахова А.А.

(дата и подпись студента)

**Содержание**

[**Введение** 3](#_Toc527903738)

# Введение

Целью курсового проекта является разработка компилятора (т.е. программы, задачей которой является перевод программы, написанной на одном языке программирования, на другой язык) для своего собственного языка программирования – ААА-2018. Данный язык программирования, ААА-2018, предназначен для выполнения простейших арифметических операций над числами, и примитивных задач.

Под компиляцией подразумевается несколько этапов обработки исходного кода, это: а) препроцессирование – на этом шаге происходит обработка текста исходного кода – замена макросов на их значения и объединение в один файл содержимого заголовочных файлов с пользовательским кодом; б) трансляция (compile) – перевод полученного в процессе препроцессирования исходного кода в ассемблерный код; в) ассемблирование – перевод ассемблерного кода в двоичное представление; создание объектного файла, компоновка (связывание, линкование). Объединение полученного объектного файла с другими библиотечными файлами (содержащие вызываемые определения функций, структур, классов) и код запуска в один исполняемый файл.

Компиляция состоит из двух частей: анализа и синтеза. Анализ – это разбиение исходной программы на составные части и создание ее промежуточного представления, а синтез, в свою очередь, – конструирование требуемой целевой программы из промежуточного представления. В данном курсовом проекте исходный код транслируется на язык \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

Компилятор для языка ААА-2018 состоит из следующих частей:

– лексический и семантический анализаторы;

– синтаксический анализатор;

– семантический анализатор;

– генератор исходного кода.

Исходя из цели курсового проекта, я определила следующие задачи:

– разработка спецификации языка программирования;

– разработка транслятора, его структуры;

– разработка лексического анализатора;

– разработка синтаксического анализатора;

– разработка семантического анализатора;

– преобразование выражений;

– генерация кода;

– тестирование транслятора.

# **Глава 1. Спецификация языка программирования**

## **Характеристика языка программирования**

Язык ААА-2018 ­– это процедурный, универсальный, строго типизированный, компилируемый язык. Не является объектно-ориентированным.

## **Преобразование типов данных**

В языке программирования ААА-2018 преобразование типов данных не поддерживается, т.е. язык является строго типизированным.

## **Алфавит языка**

Алфавит языка ААА-2018 основан на кодировке Windows-1251, представленной на рисунке 1.1.



Рисунок 1.1 – Алфавит входных символов

Исходный код ААА-2018 может содержать символы латинского алфавита, цифры десятичной системы счисления от 0 до 9, русские символы разрешены только в строковых литералах.

## **Применяемые кодировки**

Для написания исходного кода на языке программирования ААА-2018 используется кодировка Windows-1251.

## **Символы сепараторы**

Символы, которые являются сепараторами представлены в таблице 1.1.

Таблица 1.1 – Сепараторы

|  |  |
| --- | --- |
| Сепаратор | Назначение |
| ;  :  (space)  ,  =  +  -  \*  <  >  ! | разделение инструкций |
| {  } | программный блок |
| [  ] | Блок условий в конструкциях ветвления и цикла |
| (  ) | параметры/приоритетность операций (в выражениях) |

## **Типы данных**

В языке AAA-2018 есть 2 типа данных: целочисленный и строковый. Описание типов данных, предусмотренных в данным языке представлено в таблице 1.2.

Таблица 1.2 – Типы данных языка BED-2018

|  |  |
| --- | --- |
| Тип данных | Описание типа данных |
| integer | Фундаментальный тип данных. Предусмотрен для объявления целочисленных положительных данных (4 байта).  Автоматически инициализируется нулевым значением.  Возможные операции:  *арифметические*  + – бинарный, суммирование;  - – бинарный, вычитание;  \* – бинарный, умножение;  = – присваивание значения; |
| string | Фундаментальный тип данных. Предусмотрен для объявления строк. (1 символ – 1 байт). Максимальное количество символов в строке – 255. Автоматическая инициализация строкой нулевой длины. |

## **Идентификаторы**

В имени идентификатора допускаются символы латинского алфавита нижнего регистра. Максимальная длина имени - 15 символов.

## **Литералы**

В языке существует 2 типа литералов: целого и символьного типов. Краткое описание литералов представлено в таблице 1.3.

Таблица 1.3 – Описание литералов

|  |  |
| --- | --- |
| Тип литерала | Описание |
| Литералы целого типа | Целочисленные неотрицательные литералы, инициализируются 0. Литералы только rvalue. |
| Строковые литералы | Символы, заключённые в ‘ ‘ (одинарные кавычки), инициализируются пустой строкой, строковые переменные. Только rvalue. |

## **Область видимости идентификаторов**

Область видимости «сверху вниз» (по принципу С++). В языке AAA-2018 требуется обязательное объявление переменной перед её использованием. Все переменные должны находиться внутри программного блока языка. Есть возможность объявления одинаковых переменных в разных блоках. Каждая переменная получает префикс – название функции, в которой она объявлена.

## **Инициализация данных**

Таблица 1.4 – Способы инициализации переменных

|  |  |
| --- | --- |
| Вид инициализации | Примечание |
| var <идентификатор>:<тип данных>; | Автоматическая инициализация: переменные типа integer инициализируются нулём, переменные типа string – пустой строкой. |
| <идентификатор> = <значение>; | Присваивание переменной значения. |

## **Инструкции языка**

Все возможные инструкции языка программирования AAA-2018 представлены в общем виде в таблице 1.5.

Таблица 1.5 – Инструкции языка программирования AAA-2018

|  |  |
| --- | --- |
| Инструкция | Запись на языке AAA-2018 |
| Объявление переменной | var <идентификатор> : <тип данных>; |
| Присваивание | <идентификатор> = <значение>/<идентификатор>; |
| Объявление внешней функции | use function<идентификатор> (<идентификатор>: <тип данных>,…):<типданных> {…} |
| Блок инструкций | main  {  …  } |
| Возврат из подпрограммы | return<идентификатор> / <литерал>; |
| Вывод данных | out (<идентификатор> / <литерал>); |
| Ветвление | if [<условие>] {  <инструкции>  };  if [<условие>] {  <инструкции>  }else{  <инструкции>  }; |
| Цикл | while [<условие>] {  <инструкции>  }; |

## **Операции языка**

Язык программирования ААА-2018 может выполнять арифметические операции, представленные в таблице 1.6.

Таблица 1.6 – Приоритетности операций языка программирования AAA‑2018

|  |  |
| --- | --- |
| Операция | Приоритетность операции |
| (  ) | 0 или 4 |
| , | 1 |
| \*  / | 2 |
| +  - | 3 |

Максимальным значением приоритетности является “0”, минимальным “4” соответственно.

Также языком поддерживается операция сравнения операндов. Данная операция не имеет приоритета.

## **Выражения и их вычисления**

Круглые скобки в выражении используются для изменения приоритета операций. Также не допускается запись двух подряд идущих арифметических операций. Выражение может содержать вызов функции.

## **Программные конструкции языка**

Ключевые программные конструкции языка программирования AAA-2018 представлены в таблице 1.7.

Таблица 1.7 – Программные конструкции языка AAA-2018

|  |  |
| --- | --- |
| Главная функция (точка входа в приложение) | main  {  } |
| Функция | function <идентификатор> (<идентификатор> : <тип>, …) : <тип>  {  …  return <выражение>;  } |

## **Область видимости**

В языке AAA-2018 переменные обязаны находится внутри программного блока функций (по принципу С++). Объявление глобальных переменных не предусмотрено. Объявление пользовательских областей видимости не предусмотрено.

## **Семантические проверки**

Таблица с перечнем семантических проверок, предусмотренных языком, приведена в таблице 1.8.

Таблица 1.8 – Семантические проверки

|  |  |
| --- | --- |
| Номер | Правило |
| 1 | Каждый идентификатор должен быть описан только один раз, с учетом блочной структуры описаний |
| 2 | Идентификатор должен быть объявлен до его использования. |
| 3 | Тип данных передаваемых значений в функцию должен совпадать с типом параметров при её объявлении |
| 4 | Тип данных передаваемых значений в функцию стандартной библиотеки должен соответствовать заявленному. |
| 5 | Соответствие между типами функции и возвращаемым значением |
| 6 | Все операнды в выражениях и операциях должны иметь типы, допустимые для данного выражения или операции |
| 7 | Если ошибка возникает на этапе лексического анализа, синтаксический  анализ не выполняется |

## **Распределение оперативной памяти на этапе выполнения**

Все переменные размещаются в куче.

## **Стандартная библиотека и её состав**

Функции стандартной библиотеки с описанием представлены в таблице 1.9. Стандартная библиотека написана на языке программирования C++.

Таблица 1.9 – Состав стандартной библиотеки

|  |  |  |  |
| --- | --- | --- | --- |
| Имя функции | Возвращаемое значение | Принимаемые параметры | Описание |
| max | integer | integer x-число  integer y-число | Функция вычисляет максимальное значение |
| min | integer | integer x-число  integer y-число | Функция вычисляет минимальное значение |
| fact | integer | integer x-число | Функция вычисляет факториал x |
| strlen | integer | integer x-число | Функция вычисляет длину строки |
| out | void | string x – строка  integer x-число | Функция выводит на консоль x |

## **Ввод и вывод данных**

Ввод данных не поддерживается языком программирования AAA-2018.

out (<идентификатор или литерал>); – вывод в стандартный поток вывода.

## **Точка входа**

Точкой входа является функция main.

## **Препроцессор**

Препроцессор в языке программирования AAA-2018 не предусмотрен.

## **Соглашения о вызовах**

В языке вызов функций происходит по соглашению о вызовах stdcall. Особенности stdcall:

– все параметры функции передаются через стек;

– память высвобождает вызываемый код;

– занесение в стек параметров идёт справа налево.

## **Объектный код**

AAA-2018 транслируется в язык

## **Классификация сообщений транслятора**

В случае возникновения ошибки в коде программы на языке AAA-2018 и выявления её транслятором в текущий файл протокола выводится сообщение. Их классификация сообщений приведена в таблице 1.10.

Таблица 1.10. Классификация сообщений транслятора

|  |  |
| --- | --- |
| Интервал | Описание ошибок |
| 0-19 | Системные ошибки |
| 100-109 | Ошибки параметров |
| 110-119 | Ошибки открытия и чтения файлов |
| 200-299 | Ошибки лексического анализа |
| 600-699 | Ошибки синтаксического анализа |
| 700-799 | Ошибки семантического анализа |