МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Забайкальский государственный университет»

(ФГБОУ ВО «ЗабГУ»)

Факультет энергетический

Кафедра информатики, вычислительной техники и прикладной математики

**КУРСОВАЯ РАБОТА**

по Теории языков программирования и методам трансляции

на тему «Программная реализация транслятора»

Выполнили ст. гр. ИВТ-18

Гуков И.А.

Проверил доцент кафедры ИВТ и ПМ

Коган Е.С.

Чита

2021

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«Забайкальский государственный университет»

(ФГБОУ ВО «ЗабГУ»)

Факультет энергетический

Кафедра информатики, вычислительной техники и прикладной математики

**ЗАДАНИЕ**

на курсовую работу

По дисциплине: Теория языков программирования и методы трансляции

Студенту: Гукову Ивану Алексеевичу.

Специальности (направления подготовки): 09.03.01 Информатика и вычислительная техника

1 Тема курсовой работы: «Программная реализация транслятора»

2 Срок подачи студентом законченной работы: 28.12.2021

3 Исходные данные к работе:

1. Описание предметной области;

2. «Общие требования к построению и оформлению учебной текстовой документации» (МИ 01-02-2018)».

Дата выдачи задания: 15.09.2021

Руководитель курсовой работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Коган Е.С./

(подпись, расшифровка подписи)

Задание принял к исполнению

«15» сентября 2021 г.

Подпись студента\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Гуков И.А.





МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«Забайкальский государственный университет»

(ФГБОУ ВО «ЗабГУ»)

Факультет энергетический

Кафедра информатики, вычислительной техники и прикладной математики

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к курсовой работе

По дисциплине Теория языков программирования и методы трансляции

На тему «Программная реализация транслятора»

Выполнил студент группы ИВТ–18: Гуков Иван Алексеевич.

Руководитель работы: доцент кафедры ИВТ и ПМ Коган Е.С.

Чита

2021

РЕФЕРАТ

Пояснительная записка 19с., 6 рис., 0 табл., 7 источников, 0 прил.

КОНВЕРТОР, PASCAL, АЛГОРИТМ, АЛГОРИТМИЧЕСКИЙ ЯЗЫК, ПЕРЕМЕННАЯ, ЦИКЛ, УСЛОВИЕ.

Для достижения цели данной работы необходимо создать интерфейс взаимодействия пользователя и программы, а также реализовать программу конвертации из алгоритмического языка на язык Pascal.

Приложение позволяет конвертировать код, написанный на алгоритмическом языке, в код, написанный на языке Pascal.

Интерфейс был создан при помощи языка С#.

СОДЕРЖАНИЕ

# Введение……………………………...................................................................... 6

1. Теоретическая часть 7

1.1 Структура языка Pascal 7

1.2. Структура алгоритмического языка [9](#_4d34og8)

1.3. Перевод из языка Pascal в алгоритмический язык и в язык С++ 10

1.3.1. Функции обработки 10

1.3.2. Анализ ошибок 11

2. Практическая часть 12

2.1. Программная реализация 12

Заключение 18

Список использованных источников 19

ВВЕДЕНИЕ

В настоящее время существует огромное количество языков программирования. И для облегчения и ускорения перевода программы с одного языка на другой были придуманы специальные программы – трансляторы.

Транслятор — программа или техническое средство, выполняющее трансляцию программы — преобразование программы, представленной на одном из языков программирования, в программу на другом языке и, в определённом смысле, равносильную первой.

Целью данной курсовой работы является разработка конвертора с алгоритмического языка на язык Pascal. Для достижения поставленной цели было решено разработать программное средство, способное реализовать процесс конвертации.  
Интерфейс был создан при помощи языка разметки C#.

# **1. Теоретическая часть**

Для того, чтобы реализовать конвертор из алгоритмического языка в язык Pascal нам потребуется детально рассмотреть оба языка программирования, найти соответствие между ключевыми словами и конструкциями.

## **1.1 Структура языка Pascal**

Язык Pascal – это чисто процедурный язык программирования, часто использующийся для обучения структурному программированию. Особенностями языка являются строгая типизация и наличие средств структурного (процедурного) программирования. Язык предоставляет ряд встроенных структур данных: записи, массивы, файлы, множества и указатели.

Для того, чтобы описать алгоритм будущей программы используются следующие ключевые слова:

1. Program – это ключевое слово начало программы;

2. Var – это ключевое слово используется для описания переменных, которые будут использованы в программе;

3. Const – эта ключевое слово, после которого идет перечисление констант;

4. Begin – обозначение начала алгоритма;

5. End – обозначение конца алгоритма.

6. Array – ключевое слово для обозначения массива.

Для правильного описания переменных в алгоритме предусмотрены ключевые слова для обозначения их типов:

1. Integer – это целочисленный тип данных;

2. Real – эта лексема используется для обозначения вещественного типа данных;

3. String – это ключевое слово требуется для описания символьных

переменных;

4. Boolean – это ключевое слово говорит о том, что переменная логическая.

Циклические конструкции в алгоритмическом языке также используются, для их реализации используются такие ключевые слова:

1. Begin и end– используются для обозначения начала и конца цикла;

2. While – эта лексема используется для циклов с предусловием.

3. For – это ключевое слово используется для цикла со счётчиком.

4. Repeat, until - эта лексема используется для циклов с постусловием.

5. To, do, downto – ключевые слова являются параметрическими.

Для реализации условий в алгоритмическом языке используются ключевые слова:

1. If – используется перед условием, после которого будет происходить выполнение алгоритма;

2. Then – это ключевое слово, указывающее на действие, которое будет выполняться после невыполнения условия, указанное после «if»;

3. Else– – это ключевое слово, указывающее на действие, которое будет выполняться после невыполнения условия, указанное после «if»;

4. And, or, not, <, >, =, <=, > =, <> – логические знаки и выражений используются стандартные.

Ввод и вывод обозначается специальными словами Write, Writeln и Read, Readln соответственно.

## **1.2 Структура алгоритмического языка**

Алгоритмический язык программирования — формальный язык, используемый для записи, реализации и изучения алгоритмов. В отличие от большинства языков программирования, алгоритмический язык не привязан к архитектуре компьютера, не содержит деталей, связанных с устройством машины.

Основные служебные слова алгоритмического языка:

1. АЛГ – название алгоритма (программы)
2. ПЕР – объявление переменных
3. АРГ – аргумент
4. РЕЗ – результат
5. НАЧ – начало алгоритма
6. КОН – конец алгоритма
7. ДАНО – исходные данные в произвольной форме
8. НАДО – цель алгоритма

Для правильного описания переменных в алгоритме предусмотрены

следующие ключевые лексемы:

1. ЦЕЛ (целый)
2. ВЕЩ (вещественный)
3. СИМ (символьный)
4. СТР (литера) — строка
5. ЛОГ (логический)
6. ТАБ (таблица) — для обозначения массива

Для реализации условий и ветвления в алгоритмическом языке используются ключевые слова:

1. ЕСЛИ
2. ТО
3. ИНАЧЕ
4. ВСЕ
5. ВЫБОР
6. ПРИ
7. ЗНАЧ
8. И, ИЛИ, НЕ, ДА, НЕТ

Циклические конструкции в алгоритмическом языке также используются, для их реализации используются такие ключевые слова:

1. НЦ (начало цикла)
2. КЦ (конец цикла)
3. ПОКА
4. ДЛЯ
5. ОТ
6. ДО
7. ШАГ

Для обозначения ввода и вывода используются специальные слова

ВВОД и ВЫВОД

## **1.3. Перевод из языка алгоритмического языка в язык Pascal**

### **1.3.1. Функции обработки**

Лексема – это структурная единица языка, которая состоит из элементарных символов языка и не содержит в своём составе других структурных единиц языка. Лексемами языков программирования являются идентификаторы, константы, ключевые слова языка, знаки операций и т.п.

Функции обработки представляют собой анализ лексем, найденных в тексте исходной программы лексем. Этот перечень лексем можно представить в виде списка лексем.

Каждой лексеме в списке лексем соответствует некая уникальная лексема другого языка, зависящая от типа лексемы.

Результатом работы функций обработки является перечень всех найденных в тексте исходной программы лексем.

### **1.3.2. Анализ ошибок**

Анализ кода - один из самых надежных методов выявления дефектов. Он заключается в чтении программой исходного кода и высказывании рекомендаций по его улучшению. В процессе чтения кода выявляются ошибки или участки кода, которые являются ошибочными.

Функции обработки ошибок получают строку лексем от лексем исходного текста программы и проверяют, соответствует ли эта строка грамматике исходного языка.

В обработку ошибок входит генерация сообщений обо всех выявленных ошибках, причём достаточно внятных и полных.

1. **Практическая часть**

## **2.1. Программная реализация**

Конвертация основана на соответствии ключевых слов и конструкций из данных языков программирования. При считывании слов, введённых на Алгоритмическом языке, программа ищет им соответствие на языке Pascal.

### 

Рисунок 1 – Интерфейс программы

Левое окно позволяет вводить код на алгоритмическом языке. Также возможно считывание кода в поле из текстового файла при нажатии на соответствующую кнопку. Правое окно отвечает за результат конвертации в Pascal. Все результаты конвертации можно сохранить в файл с расширением .pas нажав соответствующие кнопки.

Для того, чтобы программа корректно выполнила конвертацию, при написании кода на алгоритмическом языке следует самостоятельно проверять логику работы программы. Все команды, кроме циклов и условий, должны прописываться по одному, это позволяет расширить функционал работы программы и производить конвертацию отдельных ключевых слов.

После завершения ввода требуется нажать кнопку «Перевести код», которая запустит программу конвертации.

В данном примере(рис.3) показаны возможности программы, такие как:

1. Инициализация переменных и массивов.
2. Реализация ввода/вывода.
3. Реализация цикла for.
4. Работа с массивами.

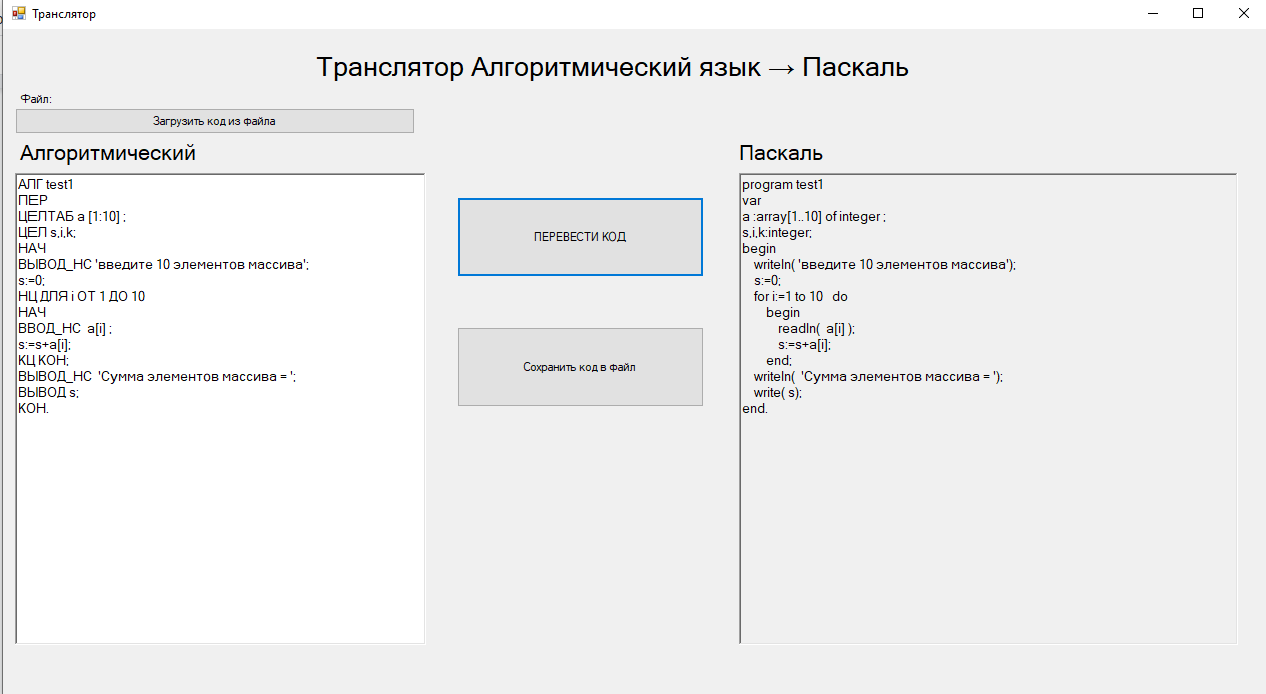


Рисунок 2 – Результат работы программы

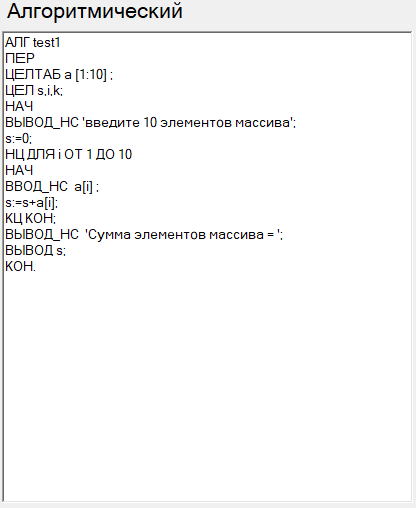


Рисунок 3 – Результат работы программы (изначальный код на алгоритмическом)

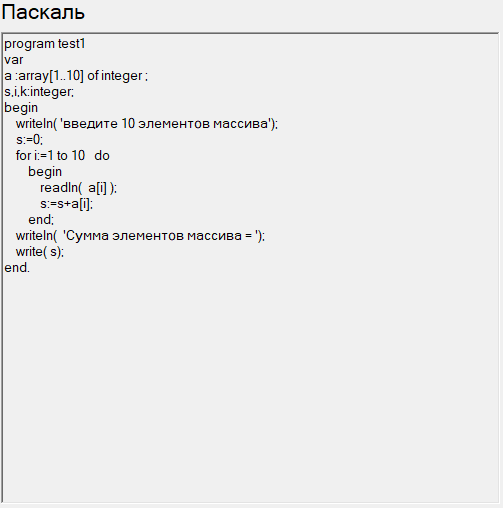


Рисунок 4 – Результат работы программы (конвертация в Pascal)

Также, на этапе анализа оригинального кода на алгоритмическом языке, предусмотрена проверка кода на лексические и синтаксические ошибки.

Если во время лексического анализа будет обнаружен код, не содержащий в себе ни одного известного ключа кода алгоритмического языка, появится сообщение об лексической ошибке, а участок кода, содержащий ошибку, будет выделен красным цветом.

При успешной проверке на лексические ошибки следует проверка на синтаксические ошибки. Например, если в коде присваивания переменной «s:=0;» убрать «s» или «0», то появится соответствующее предупреждение об синтаксической ошибки присваивания значения переменной, а участок кода, содержащий ошибку, будет выделен красным цветом.

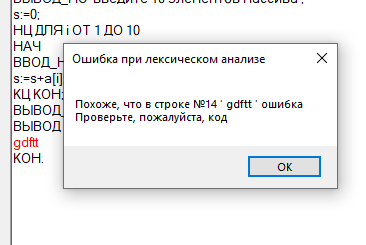


Рисунок 5 – Сообщение о лексической ошибке

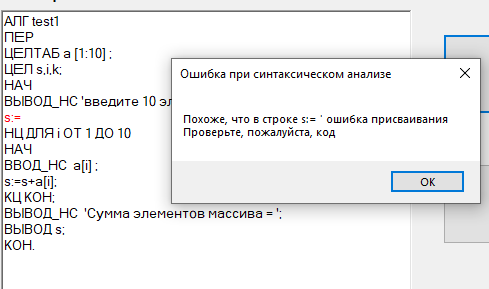


Рисунок 6 – Сообщение о синтаксической ошибке

Ниже приведены участки кода, показывающие операции конвертации:

Соответствие типов данных для переменных:

//типы переменных----------------------------

if (code.Contains("ЦЕЛ"))

{

NoKeys = false;

try

{

code = code.Replace("ЦЕЛ", "");

code = code.Replace(";", ":integer;");

AlgCodeLines[line] = code;

}

catch

{

AlgCodeLines.Add(code);

}

}

else

if (code.Contains("ВЕЩ"))

{

NoKeys = false;

try

{

code = code.Replace("ВЕЩ", "");

code = code.Replace(";", ":real;");

AlgCodeLines[line] = code;

}

catch

{

AlgCodeLines.Add(code);

}

}

else

if (code.Contains("СТР"))

{

NoKeys = false;

try

{

code = code.Replace("СТР", "");

code = code.Replace(";", ":string;");

AlgCodeLines[line] = code;

}

catch

{

AlgCodeLines.Add(code);

}

}

Таким образом была выстроена работа конвертора, основанная на соответствии языков программирования.

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы были получены навыки по проектированию программ, их отладке и документированию.

В результате работы была написана программа – транслятор из алгоритмического языка на язык Pascal, с использованием высокоуровневого объектно-ориентированного языка программирования C#.

# 

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Карпов, Ю.Г. Теория и технология программирования. Основы построения трансляторов: Учеб. пособие / Ю.Г Карпов Санкт-Петербург: БХВ-Петербург 2005. -272с.
2. Свердлов, С. З. Языки программирования и методы трансляции: учеб. Пособие / С.З Свердлов - Санкт-Петербург : Питер, 2007. - 638 с.
3. Яковлева Л. Л Информатика и программирование/Л. Л Яковлева Учеб. пособие. В 2 ч. Ч. 1. - Чита : ЗабГУ, 2014. - 213 с.
4. Черпаков И. В Основы программирования / И. В Черпаков Учебник и практикум - М. : Издательство Юрайт, 2017. – 219с.
5. Себеста, Роберт У Основные концепции языков программирования / пер. с англ. - 5-е изд. - Москва : Вильямс, 2001. - 672с.
6. Википедия – Свободная энциклопедия [Электронный ресурс] – URL: [https://ru.wikipedia.org/wiki/Паскаль\_(язык\_программирования)](https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D1%81%D0%BA%D0%B0%D0%BB%D1%8C_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)) (Дата обращения: 20.12.2021).
7. Википедия – Свободная энциклопедия [Электронный ресурс] – URL: [https://ru.wikipedia.org/wiki/Алгоритмический\_язык](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) (Дата обращения: 21.12.2021).