МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования

**"Южно-Уральский государственный университет"**

**(национальный исследовательский университет)**

**Факультет Вычислительной математики и информатики**

**Кафедра системного программирования**

**ОТЧЕТ   
по учебной практике**

бакалавра направления 010300.62 "Фундаментальная информатика и информационные технологии"

|  |  |
| --- | --- |
|  | Выполнил: студент группы ВМИ-101  Ишмурзин А.Р.  Проверил:  cтарший преподаватель  Никольская К.Ю. |

Челябинск-2016

Министерство образования и науки Российской Федерации

Южно-Уральский государственный университет

Кафедра системного программирования

УТВЕРЖДАЮ

Зав. кафедрой

системного программирования

\_\_\_\_\_\_\_\_\_\_\_\_\_ Л.Б. Соколинский

**ЗАДАНИЕ**

**по** **учебной практике**

1. **Цель работы**

Необходимо разработать распознаватель заданной символьной цепочки. Символьная цепочка задается с помощью формул Бэкуса-Наура.

1. **Исходные данные к работе**
2. *Йенсен К., Вирт Н.* Паскаль. Руководство пользователя и описание языка. М.: Компьютер, 1995.
3. *Льюис Ф., Розенкранц Д., Стирнз Р.* Теоретические основы проектирования компиляторов. М.: Мир, 1979.
4. **Перечень подлежащих разработке вопросов**
5. Выполнить анализ требований и разработать спецификации программы.
6. Провести проектирование программы.
7. Реализовать спроектированные модули.
8. Провести тестирование и отладку реализованных модулей.
9. **Сроки**

Дата выдачи задания: "27" Июня 2016 г.

Срок сдачи законченной работы: "24" Июля 2016 г.

**Руководитель:**

cтарший преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_ Никольская К.Ю.

подпись

**Задание принял к исполнению:**

\_\_\_\_\_\_\_\_\_\_\_\_\_ Ишмурзин А.Р.

подпись

**ОГЛАВЛЕНИЕ**

[1. Спецификация 5](#_Toc327919175)

[2. Проектирование 5](#_Toc327919176)

[2.1. Модульная структура 5](#_Toc327919177)

[2.2. Интерфейсы модулей 5](#_Toc327919178)

[3. Кодирование 5](#_Toc327919179)

[3.1. Структура текста программы 5](#_Toc327919180)

[3.2. Алгоритмы реализации модулей 5](#_Toc327919181)

[3.2.1. Блок транслитерации 5](#_Toc327919182)

[3.2.2. Лексический блок 5](#_Toc327919183)

[3.2.3. Синтаксический блок 6](#_Toc327919184)

[3.2.4. Блок идентификации ключевых слов 6](#_Toc327919185)

[3.3. Размер текста программы (в строках) 6](#_Toc327919186)

[4. Тестирование 6](#_Toc327919187)

[4.1. Автономное тестирование 6](#_Toc327919188)

[4.2. Комплексное тестирование 6](#_Toc327919189)

[Заключение 6](#_Toc327919190)

[Литература 7](#_Toc327919191)

# Спецификация

Необходимо разработать распознаватель заданной символьной цепочки. Символьная цепочка задается с помощью формул Бэкуса-Наура:

<цепочка>::=<оператор repeat-until>

<условие>::=<вызов подпрограммы>

<вызов подпрограммы>::=<идентификатор>**(** <список параметров> **)**;

<список параметров>::=<список идентификаторов>

<список идентификаторов>::=<идентификатор> |

<идентификатор> , <список идентификаторов>

<идентификатор>::=<буква>|

<идентификатор><буква>|

<идентификатор><цифра>

<буква>::=**A | B | C | D | E | F | ... | Z**

<цифра>::=**0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9**

<список параметров>::=<список идентификаторов>

<оператор>::=<вызов подпрограммы>

<список параметров>::=<элемент массива>

<элемент массива>::=<идентификатор>**[** <индексное выражение> **]**

<индексное выражение>::=<целая константа>

<целая константа>::=<целое со знаком> | <целое без знака>

<целое со знаком>::=<знак><целое без знака>

<знак>::= **+ | -**

<целое без знака>::=<цифра> | <целое без знака>

<список параметров>::=<целая константа>

<целая константа>::=<целое со знаком> | <целое без знака>

Помимо этого на цепочку накладывается следующее семантическое ограничение:

идентификатор, входящий в цепочку, не должен совпадать с ключевыми словами языка Pascal .

**Описание входных данных**

Цепочка записана в текстовом файле INPUT.TXT, который состоит из одной стро-ки. Длина цепочки не превышает 80 символов.

**Описание выходных данных**

Результат распознавания необходимо записать в текстовый файл OUTPUT.TXT в

одно из следующих сообщений: ACCEPT, если цепочка допустима, и REJECT, если цепочка недопустима.

**Примеры входных и выходных данных**

|  |  |
| --- | --- |
| **INPUT.TXT** | **OUTPUT.TXT** |
| repeat myproc(1) until myfunc(a); | ACCEPT |
| repeat myproc(+16) until myfunc(param1, param2); | ACCEPT |
| repeat myproc(-123) until myfunc(p1, p2, p3, p4); | ACCEPT |
| repeat begin(-123) until myfunc(p1, p2, p3, p4); | REJECT |
| repeat myproc(-123) until myfunc(p1, p2, p3, p4) | REJECT |
| repeat myproc(-123 par) until myfunc(p1, p2, p3, p4); | REJECT |
| repeat myproc(-123) until: myfunc(p1, p2, p3, p4); | REJECT |

# Проектирование

Данный раздел следует разбить на следующие пункты:

## Модульная структура

В данном пункте следует поместить *рисунок со схемой модульной структуры* с кратким описанием назначения входящих в нее модулей.

## Интерфейсы модулей

В данном пункте следует поместить *заголовки и спецификации модулей*, упомянутых в пункте 2.1. Заголовок модуля должен содержать имя модуля, имена и типы формальных параметров. Спецификация модуля должна содержать назначение модуля, описание формальных параметров модуля и их семантики (для модулей-функций также описание возвращаемого результата и его семантики).

В данном пункте также следует поместить *определение используемых типов данных* с комментариями, поясняющими семантику этих типов.

# Кодирование

Данный раздел следует разбить на следующие пункты:

## Структура текста программы

В данном пункте следует поместить *рисунок со схемой структуры текста программы* с кратким описанием назначения входящих в нее модулей unit и экспортируемых данными модулями unit подпрограмм.

## Алгоритмы реализации модулей

Данный пункт следует разбить на следующие подпункты:

### Блок транслитерации

В данном пункте необходимо поместить *таблицу транслитерации* символьной цепочки.

### Лексический блок

В данном пункте необходимо описать построение *обрабатывающего автомата лексического блока*: построение и редукция конечного распознавателя лексического блока, затем примитивные процедуры и преобразование распознавателя в обрабатывающий автомат.

При использовании стандартных методов поиска недостижимых и эквивалентных состояний (например, методом разбиения) следует поместить соответствующую библиографическую ссылку.

### Синтаксический блок

Данный пункт должен иметь содержание, аналогичное пункту 3.2.1 – применительно к *конечному автомату синтаксического блока*.

### Блок идентификации ключевых слов

В данном пункте необходимо поместить *описание используемого метода идентификации ключевых слов* и поместить соответствующую библиографическую ссылку.

## Размер текста программы (в строках)

Данный пункт отчета должен присутствовать, если выполнены работы по кодированию программной системы. В данном пункте следует поместить округленный до сотен *общий размер созданных исходных текстов в строках* (включая пустые строки, комментарии, спецификации программных файлов и др.).

# Тестирование

Данный раздел следует разбить на следующие пункты:

## Автономное тестирование

Данный пункт следует разбить на подпункты в соответствии с модульной структурой программы и в каждом пункте поместить *протокол тестирования* *соответствующего* *модуля*.

## Комплексное тестирование

В данном пункте следует поместить *протокол тестирования головного модуля* программы.

# Заключение

В данном разделе следует поместить *краткую сводку всех полученных результатов*: перечислить пройденные этапы технологического цикла разработки и соответствующие выполненные работы и указать не пройденные этапы (не выполненные работы) и работы, выполненные лишь частично.

# Литература

1. *Цымблер М.Л.* Учебные курсы прошлых лет. Основы программирования: [http://mzym.susu.ru/courses/pb/index.html] 10.07.2014
2. *Цымблер М.Л.* Учебные курсы прошлых лет. Языки программирования: [http://mzym.susu.ru/courses/pl/index.html] 10.07.2014
3. *Карпов Ю.Г.* Теория и технология программирования. Основы построения трансляторов. – СПб.: БХВ-Петербург, 2012. – 272 с.: ил.
4. *Цымблер М.Л*. Учебная практика студентов направления 010300.62 «Фундаментальная информатика и информационные технологии». Методические указания. [http://sp.susu.ru/student/courses/StudyPractice/index.html]