Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Кафедра программной инженерии**

**КОНСТРУИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

**Методические указания к выполнению курсового проекта для студентов специальности 1-40 01 01**

**«Программное обеспечение информационных технологий»**

Минск 2022

УДК 004.43(075.8) ББК 32.973-01я75 К41

Рассмотрены и рекомендованы к изданию редакционно-издательским советом университета

Рецензенты:

кандидат физико-математических наук, доцент, доцент кафедры управления информационными ресурсами Академии управления при

Президенте Республики Беларусь, *Н. И. Белодед*;

доцент кафедры информатики и веб-дизайна БГТУ, *А. А. Дятко*.

**Конструирование программного обеспечения:**

К41 методические указания к выполнению курсового проекта для студентов специальности 1-40 01 01 «Программное обеспечение ин-формационных технологий» /сост. : А. С. Наркевич – Минск : БГТУ, 2022. – 29 с.

Пособие предназначено для студентов, выполняющих курсовой проект по дисциплине «Конструирование программного обеспечения» и содержит краткие теоретические сведения, план работы над проектом, требования к проекту, описа-ние структуры пояснительной записи. Результатом курсового проекта является разработанная студентом система программирования, включающая специфика-цию языка программирования и программную реализацию транслятора с этого языка.

**УДК 004.43(075.8) ББК 32.973-01я75**

 УО «Белорусский государственный технологический университет», 2022

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ ...........................................................................................4](#_page_40_0)

[1ТРЕБОВАНИЯККУРСОВОМУПРОЕКТУ.............................................5](#_page_42_0)

[1.2 Минимальные требования к курсовому проекту............................5](#_page_42_0)

[1.3 Дополнительные (повышающие бал) требования:.........................5](#_page_42_0)

[2 ОСНОВНЫЕ РАЗДЕЛЫ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ.................6](#_page_46_0)

[2.1 Структура пояснительной записки ..................................................6](#_page_46_0)

[2.2 Титульный лист..................................................................................6](#_page_46_0)

[2.3 Задание на курсовой проект..............................................................6](#_page_46_0)

[2.4 Содержание пояснительной записки ...............................................6](#_page_46_0)

[2.5 Введение..............................................................................................7](#_page_52_0)

[2.6 Основная часть пояснительной записки..........................................7](#_page_52_0)

[2.7 Заключение.......................................................................................18](#_page_65_0)

[2.8 Графический материал....................................................................19](#_page_67_0)

[2.9 Список использованных источников.............................................19](#_page_67_0)

[2.10 Приложения....................................................................................19](#_page_67_0)

[3 ОФОРМЛЕНИЕ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ.............................20](#_page_71_0)

[3.1 Общие требования ...........................................................................20](#_page_71_0)

[3.2 Структурные элементы записки.....................................................20](#_page_71_0)

[3.3 Нумерация страниц..........................................................................21](#_page_75_0)

[3.4 Перечисления ...................................................................................21](#_page_75_0)

[3.5 Изложение текста.............................................................................22](#_page_78_0)

[3.6 Формулы ...........................................................................................22](#_page_78_0)

[3.7 Примечания ......................................................................................22](#_page_78_0)

[3.8 Рисунки .............................................................................................23](#_page_82_0)

[3.9 Таблицы.............................................................................................23](#_page_82_0)

[3.10 Ссылки.............................................................................................24](#_page_85_0)

[3.11 Приложения....................................................................................24](#_page_85_0)

[3.12 Список использованных источников...........................................24](#_page_85_0)

[Приложение А (обязательное)...........................................................26](#_page_90_0)

[Приложение Б (обязательное) ...........................................................27](#_page_92_0)

3

**ВВЕДЕНИЕ**

Целью выполнения курсового проекта по дисциплине «Конструи-рование программного обеспечения» является приобретение навыков разработки системы программирования (трансляторов, интерпретато-ров).

В процессе выполнения курсового проекта студент:

 получит навыки проектирования систем программирования;  изучит основы теории формальных грамматик и основы об-

щей теории компиляторов;

 приобретет навыки разработки программного обеспечения систем программирования.

Для успешной защиты курсового проекта студент должен разра-ботать:

 спецификацию языка программирования;

 программную реализацию лексического анализатора;

 программную реализацию синтаксического анализатора;  программную реализацию семантического анализатора;  программную реализацию генератора кода;

Необходимо выполнить тестирование, разработанного программ-ного обеспечения и подготовить пояснительную записку к курсовому проекту.

Предлагаемое пособие содержит требования к курсовому проекту (раздел 1), структуру и описание содержимого пояснительной записки (раздел 2), а также правила ее оформления (раздел 3).

4

**1ТРЕБОВАНИЯККУРСОВОМУПРОЕКТУ**

Курсовой проект включает в себя следующие обязательные части: пояснительная записка с графическим материалом. Материалы курсо-вого проекта, содержащие программный проект, исходные данные для контрольного примера и текст пояснительной записки, должны быть сохранены в файловом хранилище кафедры.

Наименование разрабатываемого языка программирования долж-но соответствовать формату: FIO-YYYY, где FIO – инициалы фами-лии, имени, отчества студента на английском языке; YYYY – год, например, студент Сидоров Иван Владимирович, название языка SIV-2022.

Общий объем пояснительной записки курсового проекта должен составлять примерно 50-60 страниц текста, включая приложения.

**1.2 Минимальные требования к курсовому проекту.**

Минимальные требования к языку программирования FIO-YYYY:  не менее двух типов данных, поддерживаемых языком;

 наличие нескольких видов программных блоков (функций, процедур и пр.);

 стандартная библиотека, содержащая не менее двух функ-ций;

 наличие одной управляющей конструкции языка;  поддержка выражений с вызовом функций;

 наличие оператора вывода данных.

Требования к контрольному примеру:

 контрольный пример должен отражать все конструкции языка.

Тестирование:

 тесты должны обеспечивать проверку вывода всех диагно-стических сообщений, генерируемых транслятором.

**1.3 Дополнительные (повышающие бал) требования:**

 использование более 2-х типов данных, массивов, пользова-тельских типов данных;

 использование нескольких управляющих (условных опера-торов, операторов цикла и пр.) конструкций языка;

 генерация кода в код на языке ассемблера;  генерация промежуточного кода в байт-код.

5

**2 ОСНОВНЫЕ РАЗДЕЛЫ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ**

**2.1 Структура пояснительной записки**

Пояснительная записка состоит из:  титульного листа;

 задания на курсовой проект;  содержания;

 введения;

 основной части пояснительной записки;  протокола тестирования;

 заключения;

 списка использованных источников;  графического материала;

 приложений.

**2.2 Титульный лист**

Титульный лист является первой страницей пояснительной запис-ки. Номер страницы на титульном листе не указывается. Пример оформления приведен в приложении А.

**2.3 Задание на курсовой проект**

Задание на курсовой проект формулируется преподавателем и включает:

 тему проекта;

 исходные данные к проекту;  срок сдачи проекта;

 содержание пояснительной записки;  перечень графического материала;

 календарный план выполнения проекта.

Пример задания на курсовое проектирование приведен в прило-жении Б.

**2.4 Содержание пояснительной записки**

В содержании указываются все разделы и подразделы поясни-тельной записки и номера их начальных страниц.

6

**2.5 Введение**

Введение – это небольшой обзор курсового проекта. Во введении следует указать цель выполнения курсового проекта, сформулировать задачи для достижения цели, кратко описать содержание пояснитель-ной записки со ссылкой на её разделы.

**2.6 Основная часть пояснительной записки**

Основная часть пояснительной записки должна включать разделы и подразделы, перечисленные ниже.

1. Спецификация языка программирования

Спецификация языка программирования – это точное формализо-ванное описание набора правил, определяющих синтаксис и семанти-ку языка.

В разделе описывается спецификация языка программирования FIO-YYYY. Рекомендуется использовать для описания языка форму Бэкуса-Наура.

1.1. Характеристика языка программирования

Описать область применения языка программирования и его свойства: компилируемый или интерпретируемый; уровень языка программирования, поддерживаемые парадигмы программирования; контроль типов (статическая, динамическая типизация, строго или не-строго типизированный язык) и т.п.

1.2. Определение алфавита языка программирования

Определить множество символов, используемых для записи кон-струкций языка программирования, а также символы времени испол-нения (символы, применяемые на этапе выполнения).

1.3. Применяемые сепараторы

Перечислить используемые в языке сепараторы (разделители, ограничители), описать назначение и правила их применения.

1.4. Применяемые кодировки

Указать кодировку символов, используемую для написания ис-ходного текста на разрабатываемом языке.

1.5. Типы данных

7

Описать поддерживаемые языком типы данных, их размеры в байтах и допустимые диапазоны значений, принцип размещения в па-мяти, инициализацию по умолчанию (если предусмотрено специфи-кацией), контроль типов, применяемые операции.

1.6. Преобразование типов данных

Определить допустимость преобразования типов данных: явные и неявные преобразования, операторы преобразования типов (если они допускаются языком).

1.7. Идентификаторы

Определить понятие идентификатора языка, применение и огра-ничение его длины (если оно предусмотрено), дать формальное опи-сание идентификатора. Привести примеры правильных и неправиль-ных идентификаторов.

1.8. Литералы

Дать определение литерала языка программирования, указать поддерживаемые языком типы литералов, допустимые диапазоны значений и правила записи. Привести примеры правильных и непра-вильных литералов.

1.9. Объявление данных

Определить правила, способы и место объявления и/или инициа-лизации переменных в тексте программы, пояснить область их види-мости.

1.10. Инициализация данных

Описать правила инициализации данных, вид инициализации (яв-ный, неявный), указать применяемую инициализацию по умолчанию, привести примеры.

1.11. Инструкции языка

Перечислить инструкции языка, описать их синтаксис и принцип применения. Дать формальное описание этих инструкций. Привести примеры написания.

1.12. Операции языка

Перечислить операции, поддерживаемые языком, указать их при-оритетность и основные свойства (ассоциативность, коммутативность и дистрибутивность), количество и допустимые типы используемых операндов и результата, порядок выполнения операций с одинаковым

8

приоритетом, способ явного указания очередности выполнения опе-раций.

1.13. Выражения и их вычисление

Дать определение выражений, допустимых в языке. Перечислить типы выражений, правила составления выражений, операции, исполь-зуемые в выражениях, допустимые типы операндов, а также порядок вычисления подвыражений.

1.14. Конструкции языка

Перечислить программные конструкции языка, используемые для управления процессом вычисления, дать их формальное описание и привести примеры, указать особенности работы с процедурами и/или функциями, программными блоками.

1.15. Область видимости идентификаторов

Определить принципы видимости идентификаторов, используе-мые в языке: область видимости на уровне блоков кода, область ви-димости на уровне функций, при объявлении параметров функции, при объявлении идентификаторов вне функций.

1.16. Семантические проверки

Перечислить и описать правила семантической проверки текста языка.

1.17. Распределение оперативной памяти на этапе выполнения. Описать принцип распределения памяти на этапе выполнения,

виды областей памяти (область кода, статическая область, стек, дина-мическая область).

1.18. Стандартная библиотека и ее состав

Определить состав функций, входящих в стандартную библиотеку языка программирования, назначение каждой функции, описать пара-метры, принимаемые функцией, и возвращаемое значение.

Определить способ подключения и принцип применения стан-дартной библиотеки.

1.19. Ввод и вывод данных

Определить предусмотренные языком операторы ввода и вывода данных, описать их применение, дать формальное описание этих опе-раторов.

9

1.20. Точка входа

Точка входа – это поименованный адрес первой инструкции про-граммы.

Определить синтаксическое правило, описывающее точку входа.

1.21. Препроцессор

Указать наличие препроцессора в языке программирования. Опи-сать применяемые директивы, назначение, формальный синтаксис и принципы их применения.

1.22. Соглашения о вызовах

Соглашение о вызовах – это правила передачи управления от вы-зывающего к вызываемому коду, определяющие способы передачи параметров и результата вычислений, возврат в точку вызова.

Определить применяемые в языке соглашения о вызовах (стан-дартные, собственные), описать способ и порядок передачи парамет-ров.

1.23. Объектный код

Определить целевой язык трансляции (ассемблер, собственный байт-код, JavaScript или другой).

1.24. Классификация сообщений транслятора Классифицировать сообщения об ошибках трансляции с указани-

ем их кодов и поясняющих текстов.

1.25. Контрольный пример

Разработать программу на языке FIO-YYYY, наиболее полно де-монстрирующую все возможности разрабатываемого языка. Привести исходный код этой программы.

2. Структура транслятора

В данном разделе описывается структура транслятора. Транслятор преобразует исходный текст программы в текст целевого языка. По-лучив на вход исходный текст, транслятор проверяет его принадлеж-ность заданному языку программирования и определяет набор грам-матических правил языка. Процесс трансляции состоит из фаз: лекси-ческий анализ, синтаксический анализ, семантический анализ и гене-рация кода. На всех фазах трансляции применяется таблица иденти-фикаторов, которая пополняется дополнительной информацией в ходе трансляции.

10

2.1. Компоненты транслятора их назначение и принципы взаимо-действия

Дать определение транслятора, привести схему транслятора, по-яснить принцип его работы. Описать компоненты транслятора, их назначение, входные и выходные данные.

2.2. Перечень входных параметров транслятора

Перечислить в табличном виде входные параметры, управляющие работой транслятора, описать их назначение и значения по умолча-нию.

2.3 Протоколы, формируемые транслятором

Перечислить в табличном виде протоколы, формируемые транс-лятором, и описать их назначение.

3. Разработка лексического анализатора

Первая фаза работы компилятора называется лексическим анали-зом, а программа, её реализующая, – лексическим анализатором.

Лексический анализатор преобразует исходный текст программы, заменяя лексические единицы языка их внутренним представлением – лексемами.

Для описания лексики языка программирования применяются ре-гулярные грамматики, относящиеся к типу 3 иерархии Хомского. Язык, заданный регулярной грамматикой, называется регулярным языком (типа 3 иерархии Хомского). Регулярный язык однозначно за-дается регулярным выражением, а распознавателями для регулярных языков являются конечные автоматы.

3.1. Структура лексического анализатора

Дать определение лексического анализатора и привести струк-турную схему лексического анализатора. Перечислить входные дан-ные, результаты работы анализатора и параметры, управляющие его работой.

3.2. Контроль входных символов

Привести таблицу разрешенных символов, используемую для контроля правильности написания исходного текста, описать принцип ее применения.

3.3. Удаление избыточных символов

Определить понятие «избыточный символ», описать алгоритм удаления избыточных символов (словесно или в виде блок-схемы).

11

3.4. Перечень ключевых слов

Перечислить в табличной форме все ключевые слова языка, сепа-раторы, символы операций, а также соответствующие им лексемы и регулярные выражения, описать их назначение.

Для ключевых слов языка построить графы переходов конечных автоматов соответствующие регулярным выражениям и привести реа-лизующие их фрагменты кода на языке С++.

3.5. Основные структуры данных

Перечислить основные структуры данных, используемые для реа-лизации лексического анализатора (таблицу лексем и таблицу иден-тификаторов). Объяснить назначение таблиц и полей соответствую-щих структур, привести их реализацию на языке С++.

3.6. Структура и перечень сообщений лексического анализатора Описать формат сообщений лексического анализатора, включаю-

щий префикс, номер ошибки, пояснительный текст сообщения об ошибке, номер строки и номер позиции исходного текста программы и т.п. Привести перечень сообщений лексического анализатора в таб-личной форме.

3.7. Принцип обработки ошибок

Описать действие лексического анализатора, выполняемое при обнаружении ошибки в исходном коде программы. Указать действу-ющий лимит на количество ошибок.

3.8. Параметры лексического анализатора

Описать входные параметры лексического анализатора, указать их назначение и принцип применения.

3.9. Алгоритм лексического анализа

Описать алгоритм работы лексического анализатора (словесно или в виде блок-схемы), привести реализующий его программный код на языке C++.

3.10. Контрольный пример

Представить результаты работы первой фазы трансляции – лекси-ческого анализатора (таблицы лексем и идентификаторов), получен-ные при выполнении контрольного примера, написанного на языка программирования FIO-YYYY.

12

4. Разработка синтаксического анализатора

Вторая фаза работы компилятора называется синтаксическим анализом, назначением которой является распознавание синтаксиче-ских конструкций языка и формирование промежуточного кода.

Правила языка программирования описываются с помощью кон-текстно-свободных грамматик (тип 2 иерархии Хомского). Програм-ма, выполняющая синтаксический анализ, называется синтаксическим анализатором. Исходными данными синтаксического анализатора яв-ляются таблицы лексем и идентификаторов, полученными на первой фазе трансляции. Лексемы являются для синтаксического анализатора терминальными символами контекстно-свободной грамматики. Ре-зультат работы синтаксического анализатора – дерево разбора (про-межуточное представление кода). Распознавателями для контекстно-свободных языков являются односторонние недетерминированные распознаватели с ограниченной магазинной памятью (МП-автоматы).

4.1. Структура синтаксического анализатора

Определить место и назначение синтаксического анализатора, описать входные и выходные данные и параметры, управляющие его работой.

4.2. Контекстно-свободная грамматика, описывающая синтаксис языка

Определить формальную грамматику языка FIO-YYYY для вы-полнения синтаксического разбора исходного текста программы. Описать порядок приведения грамматики (исключение недостижимых символов, лямбда-правил и цепных правил) и преобразование грамма-тики в нормальную форму Грейбах.

Привести перечень и описание терминальных, нетерминальных символов и правил грамматики языка.

4.3. Построение конечного магазинного автомата

Привести формальное описание конечного автомата с магазинной памятью, описать алгоритм работы МП-автомата, построить его схему и привести последовательность мгновенных состояний МП-автомата, демонстрирующую успешный разбор для цепочки языка из контроль-ного примера.

4.4. Основные структуры данных

Продемонстрировать программный код основных структур дан-ных на языке С++, описывающих контекстно-свободную грамматику.

13

4.5. Описание алгоритма синтаксического разбора

Описать алгоритм синтаксического разбора. Построить обобщен-ную блок-схему алгоритма синтаксического анализа.

4.6. Структура и перечень сообщений синтаксического анализа-тора

Представить полный перечень и формат сообщений синтаксиче-ского анализатора. Привести примеры.

4.7. Параметры синтаксического анализатора и режимы его работы

Перечислить управляющие входные параметры транслятора, ис-пользуемые синтаксическим анализатором в своей работе (для созда-ния протокола разбора, сохранения дерева разбора и т.п.), описать ре-жимы работы синтаксического анализатора с заданными параметрами.

4.8. Принцип обработки ошибок

Описать принцип работы синтаксического анализатора, при обна-ружении ошибки в исходном коде, определить порядок вывода сооб-щений об ошибках, указать установленное ограничение на количество ошибок.

4.9. Контрольный пример

Привести распечатку протокола синтаксического разбора кон-струкций языка FIO-YYYY из контрольного примера и полученное дерево разбора, дать подробные пояснения к протоколу.

5. Разработка семантического анализатора Семантический анализ – третья фаза работы транслятора.

Семантический анализ может быть явно выделен в отдельную фа-зу или совмещаться с фазами лексического и синтаксического анали-за. Семантический анализатор использует синтаксическое дерево и информацию из таблицы идентификаторов для смысловой проверки исходного текста на соответствие семантическим правилам языка. Основные действия семантического анализатора:

 проверка семантических правил исходного языка;

 дополнение внутреннего представления программы операто-рами и действиями, неявно предусмотренными семантикой исходного языка.

14

5.1. Структура семантического анализатора

Описать назначение семантического анализа, его входные и вы-ходные данные. Привести структурную схему семантического анали-затора, поясняющую его взаимодействие с другими компонентами транслятора.

5.2. Функции семантического анализатора

Перечислить семантические проверки с указанием фаз их выпол-нения (лексического, синтаксического анализаторов или выделенной в отдельную фазы семантического анализатора). Описать функции, реа-лизующие проверку, привести краткое описание этих функций, вход-ные и выходные данные.

5.3. Структура и перечень сообщений семантического анализатора

Представить в табличном виде перечень, структуру и текст сооб-щений семантического анализатора.

5.4. Принцип обработки ошибок

Описать действие семантического анализатора, при обнаружении ошибки в исходном тексте программы, определить порядок вывода сообщений об ошибках, указать лимит на количество ошибок.

5.5. Контрольный пример

Скорректировать контрольный пример для демонстрации трех ошибок, диагностируемых семантическим анализатором на разных этапах трансляции. Описать каждый случай и включить в пояснитель-ную записку отчет с распечаткой выданных сообщений.

6. Вычисление выражений

В разделе описываются выражения допускаемые языком, форма, принципы построения и вычисление выражений.

Обычная форма выражений, в которой знак операции размещает-ся между операндами, называется инфиксной. Обратная польская но-тация (называют также польской инверсной записью, ПОЛИЗ) – это форма записи математических выражений, в которой операторы рас-положены после своих операндов. Выражения, представленные в об-ратной польской нотации, легко вычисляются, время вычисления – линейное.

Выражение в обратной польской нотации читается слева направо. Операция выполняется над двумя операндами, непосредственно сто-ящими перед знаком этой операции. Результат операции заменяет в

15

выражении последовательность её операндов и символ операции. Ре-зультатом вычисления всего выражения является результат последней вычисленной операции.

ПОЛИЗ удобна как для вычисления выражений, так и в качестве промежуточной формы представления выражений в трансляторе. Принцип обратной польской записи может быть применен не только к выражениям, но и операторам языков программирования.

6.1. Выражения, допускаемые языком

Описать выражения допускаемые языком (типы данных, исполь-зуемые в выражениях, приоритетность операций, использование функций в выражениях), привести типичные примеры выражений из контрольного примера.

6.2. Польская запись и принцип ее построения

Определить назначение обратной польской записи, принципы ее построения, привести примеры преобразования выражений из кон-трольного примера в обратную польскую нотацию.

6.3. Программная реализация обработки выражений

Привести фрагмент кода транслятора на языке С++, реализующе-го преобразование выражений в обратный польский формат, дать краткие пояснения к коду.

Предусмотреть наличие и указать параметр транслятора, позво-ляющий отобразить в протоколе результаты преобразования выраже-ний в обратный польский формат.

6.4. Контрольный пример

Привести часть протокола контрольного примера, отображающую результаты преобразования выражений в обратную польскую запись в формате: выражение в первой строке вывода, обратная польская за-пись для данного выражения во второй строке вывода.

7. Генерация кода

В данном разделе описывается процесс генерации кода. Генерация кода – четвертая последняя фаза работы транслятора.

Исходными данными для генератора кода является промежуточное представление исходной программы на языке FIO-YYYY.

Одной из основных задач на этапе генерации кода является пла-нирование памяти для размещения переменных, литералов. Решение о распределении памяти принимается либо в процессе генерации про-межуточного кода, либо при генерации целевого кода.

16

Самый простой подход к выполнению курсового проекта – это генерация кода в JavaScript. Интерпретатором JavaScript является js-движок браузера. Входными данными для генерации являются табли-цы лексем и идентификаторов и дерево разбора, полученное на фазе синтаксического анализа.

Другой подход: можно выполнить генерацию в код на языке ас-семблера. Затем вызвать транслятор ассемблера для получения объ-ектного кода.

Еще один подход – разработка транслятора-интерпретатора (ис-ходный код транслируется в байт-код – последовательность команд для некоторой виртуальной машины, затем он сразу интерпретирует-ся).

7.1. Структура генератора кода

Определить целевой язык, в который выполняется трансляция ис-ходного кода с разрабатываемого языка FIO-YYYY, описать процесс генерации кода, привести структуру генератора кода.

7.2. Представление типов данных в оперативной памяти Подробно описать построение модели памяти. Определить соот-

ветствие типов данных в исходном языке программирования типам данных целевого языка (представить в табличном виде), дать подроб-ные пояснения.

7.3. Статическая библиотека

Привести состав статической библиотеки, назначение функций, входящих в состав библиотеки. Указать способ подключения библио-теки, этап, на котором производится подключение статической биб-лиотеки, перечислить инструкции языка, используемые для этих це-лей.

7.4. Особенности алгоритма генерации кода

Построить обобщенную блок-схему алгоритма генерации кода и привести ее описание, указать особенности алгоритма. Привести пе-речень и описание разработанных для реализации генерации кода функций.

7.5. Входные параметры, управляющие генерацией кода

Описать входные параметры генератора кода, указать их назначение.

7.6. Контрольный пример

Привести результат генерации кода на основе контрольного примера.

17

8. Тестирование транслятора

Подготовить контрольный пример, демонстрирующий правиль-ную работу компилятора.

Выполнить подбор тестов, которые должны быть включены в те-стовый комплект. Тесты должны полностью покрывать список оши-бок, обнаруживаемых тестируемым компилятором на разных фазах трансляции.

8.1. Общие положения

Описать общие принципы, лежащие в основе тестов, описать дей-ствия компилятора при обнаружении ошибки, указать протоколы, в которые будут выводиться результаты тестирования.

8.2. Результаты тестирования

Сгруппировать описание тестовых наборов, демонстрирующих проверки на разных этапах трансляции. Представить результаты в ви-де таблицы, содержащей фрагмент исходного кода с ошибкой и соот-ветствующее диагностическое сообщение (код ошибки, этап, текст сообщения, место ошибки с указанием строки и позиции в исходном коде).

9. Разработка и тестирование интерпретатора

Подробно описать процесс разработки и тестирования интерпре-татора.

9.1. Структура и перечень сообщений времени выполнения Представить в табличном виде перечень, структуру и текст сооб-

щений времени выполнения компилятора.

**2.7 Заключение**

В заключении формулируются краткие выводы по результатам выполненной работы. Приводятся количественные и качественные характеристики реализации транслятора: количество типов данных, инструкций языка, лексем, правил грамматики, наличие стандартной библиотеки и ее состав, примерное количество строк кода на языке реализации, время трансляции контрольного примера и т.п.

18

**2.8 Графический материал**

Графический материал выполняется на листе формата А3 по ГОСТ 2.301 и представляет граф дерева разбора, полученного на фазе синтаксического анализа.

**2.9 Список использованных источников**

Перечислить книги, статьи, электронные ресурсы, которые были использованы при выполнении работы. Информация о правилах оформления этого списка приведена в пункте 3.12. Список использо-ванных источников должен содержать не менее пяти наименований.

**2.10 Приложения**

Приложения содержат материалы вспомогательного характера: таблицы большого объема, блок-схемы алгоритмов, графы переходов, тексты программ, результаты тестирования, трассировки и т. п.

19

**3 ОФОРМЛЕНИЕ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ**

**3.1 Общие требования**

При оформлении пояснительной записки следует руководство-ваться требованиями СТАНДАРТ ОРГАНИЗАЦИИ СТП БГТУ 001-2019. ПРОЕКТЫ (РАБОТЫ) ДИПЛОМНЫЕ.

Пояснительная записка к курсовой работе оформляется в соответ-ствии с правилами оформления текстовых документов, изложенными в ГОСТ 2.105–95 Общие требования к текстовым документам, и с правилами оформления курсовых работ, изложенными в данном ме-тодическом указании.

Текст записки должен быть напечатан на одной стороне листа формата А4. Используемый шрифт – Times New Roman, размер шрифта – 14, межстрочный интервал – одинарный. Цвет шрифта – черный. Абзацный отступ – 1.25 см. В тексте после знаков препина-ния обязательно ставится пробел. Нельзя сокращать слова (кроме со-кращений, установленных правилами орфографии).

Графический материал оформляется на листах формата А3 с рам-кой.

Текст следует печатать, соблюдая поля: правое – 101 мм; верх-нее – 201 мм; левое – 231 мм; нижнее – 151 мм. При наличии на листе рамки и основной надписи по форме 2 расстояние между верх-ней границей основной надписи с последней строкой текста, если лист полностью заполняется текстом, должно составлять 10–15 мм.

**3.2 Структурные элементы записки**

Структурные элементы записки: «Содержание», «Введение», «За-ключение», «Список использованных источников», «Графический ма-териал», а также каждый из основных разделов и каждое из приложе-ний следует начинать с нового листа.

Заголовки элементов текста «Содержание», «Введение», «Заклю-чение», «Список использованных источников», «Графический мате-риал» следует записывать в начале соответствующих страниц строч-ными буквами, первая буква – прописная; шрифт – полужирный. Рас-положение – симметрично тексту и отделяются от него интервалом в 18 пт.

Текст основной части делят на разделы, разделы – на подразделы, подразделы – на пункты и т.д. Разделы нумеруются арабскими циф-рами, точка в конце не ставится; подразделы нумеруются в пределах

20

раздела. Номер раздела отделяется от номера подраздела точкой. В конце номера подраздела точка не ставится. За номером раздела или подраздела следует его название, записанное с прописной буквы без точки в конце. Переносы слов в заголовках не допускаются. Если за-головок состоит из двух предложений, их разделяют точкой. Номер и название раздела или подраздела записывается с абзацного отступа, шрифт – полужирный, отделяется от текста интервалом в 18 пт.

**3.3 Нумерация страниц**

Нумерация страниц пояснительной записки сквозная. На титуль-ном листе, первой странице пояснительной записки, номер не указы-вается. Номер проставляется арабской цифрой без точки в правом верхнем углу страницы.

Графический материал, размещенный на листе формата А3, учи-тывается как одна страница.

**3.4 Перечисления**

В тексте пояснительной записки могут быть использованы пере-числения. Пункты перечисления записывают после двоеточия в виде списка, каждый с абзацного отступа. Перед каждым пунктом ненуме-рованного списка перечисления следует ставить тире. При необходи-мости ссылки в тексте на один или несколько пунктов перечисления, перечисление оформляют в виде маркированного списка, каждый пункт начинают со строчной буквы русского алфавита (за исключени-ем ё, з, о, г, ь, й, ы, ъ) с проставленной после нее круглой скобкой. Для дальнейшей детализации перечислений (сложные перечисления) необходимо использовать арабские цифры с проставленными после них круглыми скобками. Запись подчиненных пунктов сложного пе-речисления выполняют с абзацными отступами по отношению к ос-новному.

Пример выполнения простого перечисления. Основные действия семантического анализатора:

 проверка соблюдения в исходной программе семантических правил входного языка;

 дополнение внутреннего представления программы в ком-пиляторе операторами и действиями, неявно предусмотрен-ными семантикой входного языка.

21

**3.5 Изложение текста**

Текст пояснительной записки должен быть кратким, четким и не допускать различных толкований. В тексте пояснительной записки не допускается:

 применять обороты разговорной речи;

 применять для одного и того же понятия различные терми-ны;

 применять сокращения слов, кроме установленных правила-ми орфографии русского языка.

Перечень допускаемых сокращений русских слов установлен в ГОСТ 2.316 и ГОСТ 7.12, белорусских – в СТБ 7.12.

Если в пояснительной записке принята особая система сокраще-ния слов или наименований, то в ней должен быть приведен перечень принятых сокращений.

**3.6 Формулы**

Формулы располагаются в тексте в отдельных строках, по центру строки. Формулы нумеруются внутри раздела (формат: номер раздела, порядковый номер формулы через точку). Номера формул записыва-ются на уровне формулы в круглых скобках справа в конце строки. Пояснения символов и числовых коэффициентов, входящих в форму-лу, должны быть приведены непосредственно под формулой. Поясне-ния должны начинаться со слова «где» и далее следует описание каж-дого символа в той последовательности, в которой они приведены в формуле по одному в строке.

Размер шрифта символов в формулах и уравнениях должен соот-ветствовать размеру основного шрифта текста. Размер индексов при основных символах в формулах и уравнениях – 9 п.

Ссылки на формулы, ранее приведенные в тексте записки, а также на формулы в приложениях необходимо выполнять с использованием их номера, например: «…по формуле (2.8)…».

**3.7 Примечания**

Примечания следует применить в пояснительной записке, если необходимы пояснения по содержанию текста, таблиц или иллюстра-ций.

Примечания необходимо помещать непосредственно после тек-стового материала (рекомендуется в конце пункта, подпункта), табли-

22

цы или графического материала, к которым они относятся. Если при-мечание одно, то после слова «Примечание» следует ставить тире, а за ним с прописной буквы – его текст. Одно примечание не нумеруется. Номер примечания от его текста точкой не отделяют. Примечание к таблице необходимо помещать в конце таблицы над обозначающей ее окончание чертой.

Текст примечаний рекомендуется печатать шрифтом разме-ром 12 п.

**3.8 Рисунки**

Все рисунки (чертежи, схемы, графики, структурные схемы и др. кроме таблиц) должны располагаться непосредственно после ссылки на них в тексте. Рисунок располагается так, чтобы его удобно было смотреть без поворота листа или с поворотом по часовой стрелке. Ри-сунки нумеруются арабскими цифрами сквозной нумерацией внутри раздела и располагаются с абзацного отступа. Рисунки каждого при-ложения состоят из обозначения приложения, точки и сквозной нуме-рации, выполненной арабскими цифрами. Вначале располагается сам рисунок, затем подрисуночный текст в виде: Рисунок, номер, наиме-нование рисунка. Например: Рисунок 2.3 Схема транслятора.

Рисунок отделяют от текста интервалом 14 п.

Не допускается отрыв (перенос со страницы на страницу) рисунка и подрисуночного текста.

**3.9 Таблицы**

Таблицы нумеруются внутри раздела. Заголовок таблицы оформ-ляют в виде: Таблица, номер, наименование таблицы. Заголовок таб-лицы выравнивают по левому краю таблицы. Если таблица выходит за размер листа, то ее делят на части. При переносе части таблицы на другой лист заголовок помещают только над первой частью, над остальными частями пишут слова «Продолжение таблицы» с указани-ем номера таблицы. Шапку таблицы при переносе части таблицы по-вторяют. Заголовки строк и столбцов пишут с прописной буквы.

Таблицу располагают в записке непосредственно после текста, в котором она упоминается.

Таблицу следует отделять от текста интервалом 12 п. Допускается выполнять таблицы, размещая их вдоль длинной стороны листа таким образом, чтобы таблица читалась при повороте листа на 90 по часо-вой стрелке.

23

Таблицы слева, справа и снизу ограничивают линиями.

**3.10 Ссылки**

Ссылки на разделы, подразделы, перечисления, таблицы, иллю-страции, формулы и приложения пояснительной записки следует вы-полнять по следующим примерам:

 «…структура транслятора, описанная в разделе 2…»;  «…по пункту б) перечисления…»;

 «…символы приведены в таблице 1.1…»;  «…изображено на рисунке 3.8…»;

 «…правило грамматики, проведенное в формуле (5.3)…»;

 «…результаты тестирования представлены в приложе-нии Д…».

Ссылку на литературный источник выполняют с указанием по-рядкового номера источника, под которым он внесен в «Список ис-пользованных источников» пояснительной записки. Номер ссылки проставляется арабскими цифрами в квадратных скобках, напри-мер: [1], [1,2,5].

Пример ссылки на источники:

 «…согласно п. 3.4 стандарта [7]…».

**3.11 Приложения**

Каждое приложение начинается с нового листа с указанием наверху посередине страницы слова «Приложение» с первой пропис-ной буквы и его обозначения.

Приложения по ГОСТ 2.105 обозначаются заглавными буквами русского алфавита, начиная с А, за исключением букв Ё, З, Й, О, Ч, Ь , Ы, Ъ. После слова «Приложение» пишется буква, идентифицирующая его последовательность. Если в документе одно приложение, оно обо-значается «Приложение А».

Приложения должны иметь заголовки, которые записывают сим-метрично тексту с прописной буквы в следующей строке.

Затем через одну пустую строчку следует текст приложения.

**3.12 Список использованных источников**

Источники – это книги, учебники, статьи из Интернета и т.д., ис-пользованные при выполнении курсовой работы. Источники в списке располагаются в порядке ссылок в тексте записки или по алфавиту,

24

нумеруются арабскими цифрами без точки и печатаются с абзацного отступа, при этом дается библиографическое описание каждого ис-точника в соответствии с ГОСТ 7.1, ГОСТ 7.12. Общий шаблон опи-сания книги, у которой не более трех авторов: ФИО\_автора, название книги, точка, тире, город, двоеточие, издательство, запятая, год изда-ния, точка, тире, количество страниц, буква «с», точка. Название го-рода дается целиком, допустимы только сокращения «М.» (Москва) и «СПб.» (Санкт-Петербург); название издательства – без кавычек. Если у книги один, два или три автора, то вначале указывается фамилия, потом – инициалы. Примеры приведены ниже.

Один, два или три автора:

Шуп, Т. Решение инженерных задач на ЭВМ: практическое руко-водство: пер. с англ. / Т. Шуп. – М.: Мир, 1982. – 238 с.

Стандарты:

ГОСТ 19.701-90. Схемы алгоритмов, программ данных и систем. – М.: Изд-во стандартов, 2004. – 26 с.

Список использованных источников

1 Биллинг, В. А. Основы программирования на C# / В. А. Бил-линг. – М.: Бином, 2012 – 488 с. 2

25

**Приложение А (обязательное)**

Образец оформления титульного листа

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных Технологий

Кафедра Программной инженерии

Специальность 1-40 01 01 Программное обеспечение информацион-

ных технологий

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К КУРСОВОМУ ПРОЕКТУ НА ТЕМУ:**

«Разработка компилятора ХХХ-201Х»

Выполнил студент Фамилия Имя Отчество

(Ф.И.О. студента)

Руководитель проекта Фамилия Имя Отчество

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

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

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

Консультанты Фамилия Имя Отчество

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

Курсовой проект защищен с оценкой

Минск 202\_

26

**ПРИЛОЖЕНИЕ Б (ОБЯЗАТЕЛЬНОЕ)**

Образец оформления листа задания

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет ФИТ Кафедра1 ПИ

Специальность 1-40 01 01 Программное обеспечение информацион-ных технологий

УТВЕРЖДАЮ Заведующий кафедрой \_\_\_\_\_\_\_\_\_\_\_\_ Н.В. Пацей

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2022 г.

**ЗАДАНИЕ**

**к курсовому проектированию по дисциплине** «Конструирование программного обеспечения»

Специальность: Группа:

Студент: Фамилия Имя Отчество

(фамилия, имя, отчество)

1. Тема проекта Разработка компилятора ХХХ-202Х

2. Срок сдачи студентом проекта: \_\_\_ декабря 202\_ 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 2015 update 2. Операционная система Windows 7 SP1 (64-bit). Типы данных: integer и string. Функ-ции стандартной библиотеки: integer strlen(string) – определение дли-ны строки, string substr (string, string) – извлечение подстроки. Ариф-метические операции: +, -, \*. Оператор вывода в стандартный поток: output. Условный оператор: if.

4. Содержание расчетно-пояснительной записки (перечень вопро-сов подлежащих разработке):

 Введение

 Спецификация языка программирования  Структура транслятора

27

 Разработка лексического анализатора

 Разработка синтаксического анализатора  Разработка семантического анализатора  Вычисление выражений

 Генерация кода

 Тестирование транслятора (и/или Разработка и тестирование интерпретатора)

 Заключение  Литература  Приложения

5. Перечень графического материала (с точным указанием обяза-тельных чертежей)

 Граф дерева разбора

6. Консультанты по проекту с указанием относящихся к ним раз-делов проекта

Раздел Консультант Разработка синтаксического и семантического ана- ФИО руководителя лизатора.

Генерация кода. Разработка тестовых примеров. ФИО руководителя

Оформление пояснительной записки к курсовому ФИО руководителя проекту.

7. Календарный план

№ Наименование этапов курсового проекта Срок выполнения Примечание

1 Спецификация специализированного языка ХХХ-202Х

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 15.09.202\_

Руководитель Фамилия И.О. (подпись, фамилия, имя, отчество)

Задание принял к исполнению Фамилия И.О. (подпись и фамилия, имя, отчество студента)

28

**КОНСТРУИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

Методические указания к выполнению курсового проекта

Составители: Наркевич Аделина Сергеевна

Редактор Компьютерная верстка Корректор

Издатель:

УО «Белорусский государственный технологический университет» Свидетельство о государственной регистрации издателя, Изготовителя, распространителя печатных изданий № 1/227 от 20.03.2014.

Ул. Свердлова, 13а, 220006, г. Минск.

29