Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра программной инженерии

ЯЗЫКИ ПРОГРАММИРОВАНИЯ

**Методические указания к выполнению курсового проекта
для студентов специальности 1-40 01 01
«Программное обеспечение информационных технологий»**

Минск 2018

УДК 004.43(075.8)
ББК 32.973-01я75
Я41

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*.

Языки программирования :
Я41 методические указания к выполнению курсового проекта для студентов специальности 1-40 01 01 «Программное обеспечение информационных технологий» /сост. : А. С. Наркевич, В. В. Смелов, – Минск : БГТУ, 2018. – 29 с.

Пособие предназначено для студентов, выполняющих курсовой проект по дисциплине «Языки программирования» и содержит план работы над проектом, требования к проекту, описание структуры пояснительной записи. Результатом курсового проекта является разработанная студентом система программирования, включающая спецификацию языка программирования и программную реализацию транслятора с этого языка.

УДК 004.43(075.8)
ББК 32.973-01я75

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Целью выполнения курсового проекта по дисциплине «Языки программирования» является приобретение навыков разработки системы программирования (трансляторов, интерпретаторов).

В процессе выполнения курсового проекта студент:

- получит навыки проектирования систем программирования;
- изучит основы теории формальных грамматик и основы общей теории компиляторов;
- приобретет навыки разработки программного обеспечения систем программирования.

Для успешной защиты курсового проекта студент должен:

- разработать спецификацию языка программирования;
- разработать программную реализацию лексического анализатора;
- разработать программную реализацию синтаксического анализатора;
- разработать программную реализацию семантического анализатора;
- разработать программную реализацию генератора кода;
- выполнить тестирование, разработанного программного обеспечения;
- 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).

1 ТРЕБОВАНИЯ К КУРСОВОМУ ПРОЕКТУ

Курсовой проект включает в себя следующие обязательные части: пояснительная записка с графическим материалом и электронный носитель, содержащий программный проект, исходные данные для контрольного примера и текст пояснительной записки.

Наименование разрабатываемого языка программирования должно соответствовать формату: FIO-YYYY, где FIO – инициалы фамилии, имени, отчества студента на английском языке; YYYY – год, например: студент Сидоров Иван Владимирович, название языка SIV-2018.

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:

- количество типов данных: 2;
- наличие нескольких видов программных блоков (функций, процедур и пр.);
- наличие стандартной библиотеки (не менее 2-х функций);
- поддержка выражений с вызовом функций;
- наличие оператора вывода (print или подобного).

Требования к контрольному примеру:

- контрольный пример должен отражать все конструкции языка.

Тестирование:

- тесты должны обеспечивать проверку вывода всех диагностических сообщений, генерируемых транслятором.

1.3 Дополнительные (повышающие бал) требования:

- использование более 2-х типов данных, массивов;
- использование управляющих (условных операторов, операторов цикла и пр.) конструкций языка;
- генерация кода в код на языке ассемблера.

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 записки. 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 приложении Б.

2.4 Содержание пояснительной записки

В содержании указываются все разделы и подразделы пояснительной записки и номера их начальных страниц.

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 пояснительной записки со ссылкой на её разделы.

2.6 Основная часть пояснительной записки

Основная часть пояснительной записки должна включать следующие разделы и подразделы.

1. Спецификация языка программирования

Спецификация языка программирования – это точное формализованное описание набора правил, определяющих синтаксис и семантику языка. В разделе описывается спецификация языка программирования FIO-YYYY.

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 нестрого типизированный язык) и т.п.

1.2. Определение алфавита языка программирования

Определить множество символов, используемых для записи конструкций языка программирования, а также 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 текста на разрабатываемом языке.

1.5. Типы данных

Описать поддерживаемые языком типы данных, их размеры в байтах и допустимые диапазоны значений, принцип размещения в па-

мяти, инициализацию по умолчанию (если предусмотрено спецификацией), 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 ограничение его длины (если оно предусмотрено), дать формальное описание идентификатора. Привести примеры правильных и неправильных идентификаторов.

1.8. Литералы

Дать определение литерала языка программирования, указать типы (если предусмотрено спецификацией), допустимые диапазоны значений и правила записи. Привести примеры правильных и неправильных литералов.

1.9. Объявление данных

Определить правила, способы и место объявления и/или инициализации переменных в тексте программы, пояснить область их видимости.

1.10. Инициализация данных

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

1.11. Инструкции языка

Перечислить инструкции языка, описать их синтаксис и принцип применения. Привести примеры написания.

1.12. Операции языка

Перечислить операции, поддерживаемые языком, указать их приоритетность и основные свойства (ассоциативность, коммутативность и дистрибутивность), количество и допустимые типы используемых операндов и результата, порядок выполнения операций с одинаковым приоритетом, способ явного указания очередности выполнения операций.

1.13. Выражения и их вычисление

Дать определение выражений, допустимых в языке. Перечислить типы выражений, правила составления выражений, операции, используемые 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, используемые в языке: область видимости на уровне блоков кода, область видимости 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, динамическая область).

1.18. Стандартная библиотека и ее состав

Определить состав функций, входящих в стандартную библиотеку языка программирования. Определить принцип применения стандартной библиотеки.

1.19. Ввод и вывод данных

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

1.20. Точка входа

Точка входа – это поименованный адрес первой инструкции программы. Определить синтаксическое правило, описывающее точку входа.

1.21. Препроцессор

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

1.22. Соглашения о вызовах

Соглашение о вызовах – это правила передачи управления от вызывающего к вызываемому коду, определяющие способы передачи параметров и результата вычислений, возврат в точку вызова. Определить применяемые в языке соглашения о вызовах (стандартные, собственные), описать способ и порядок передачи.

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 указанием их кодов и поясняющих текстов.

1.25. Контрольный пример

Разработать программу на языке FIO-YYYY, наиболее полно демонстрирующую 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. Получив на вход исходный текст, транслятор проверяет его принадлежность заданному языку и определяет набор грамматических правил языка. Процесс трансляции состоит из фаз: лексический анализ, синтаксический анализ, семантический анализ и генерация кода. На всех фазах трансляции применяется таблица идентификаторов, которая пополняется дополнительной информацией в ходе трансляции.

2.1. Компоненты транслятора их назначение и принципы взаимодействия

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

2.2. Перечень входных параметров транслятора

Перечислить в табличном виде входные параметры транслятора, описать их назначение и значения по умолчанию.

2.3 Протоколы, формируемые транслятором

Перечислить в табличном виде протоколы, формируемые транслятором, и описать их назначение.

3. Разработка лексического анализатора

Первая фаза работы компилятора называется лексическим анализом, а программа, её реализующая, – лексическим анализатором. Лексический анализатор преобразует исходный текст, заменяя лексические единицы языка их внутренним представлением – лексемами. Для описания лексики языка программирования применяются регулярные грамматики, относящиеся к типу 3 иерархии Хомского. Язык, заданный регулярной грамматикой, называется регулярным языком (типа 3 иерархии Хомского). Регулярный язык однозначно задается регулярным выражением, а распознавателями для регулярных языков являются конечные автоматы.

3.1. Структура лексического анализатора

Дать определение лексического анализатора и привести его структурную схему. Описать входные данные, результаты работы анализатора, перечислить его параметры.

3.2. Контроль входных символов

Привести таблицу разрешенных символов, используемую для контроля, описать принцип ее применения.

3.3. Удаление избыточных символов

Определить понятие «избыточный символ», описать алгоритм удаления избыточных символов.

3.4. Перечень ключевых слов

Перечислить в табличной форме все ключевые слова языка, сепараторы, символы операций, соответствующие им лексемы и регулярные выражения.

Построить графы переходов конечных автоматов соответствующие регулярным выражениям и привести реализующие их фрагменты кода на языке C++.

3.5. Основные структуры данных

Перечислить основные структуры данных лексического анализатора (таблицу лексем и таблицу идентификаторов), описать назначение всех полей структур, привести их реализацию на языке C++.

3.6. Структура и перечень сообщений лексического анализатора

Описать формат сообщений лексического анализатора, включающий префикс, номер ошибки, пояснительный текст сообщения об ошибке, номер строки и номер позиции исходного кода и т.п. Привести перечень сообщений лексического анализатора в табличной форме.

3.7. Принцип обработки ошибок

Описать действие лексического анализатора при обнаружении ошибки в исходном коде программы. Задать действующий 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 лексического анализатора, привести реализующий его программный код на языке C++.

3.10. Контрольный пример

Представить результаты работы лексического анализатора (таблицы лексем и идентификаторов), полученные при выполнении контрольного примера.

4. Разработка синтаксического анализатора

Вторая фаза работы компилятора называется синтаксическим анализом, назначением которой является распознавание синтаксических конструкций языка и формирование промежуточного кода. Правила языка программирования описываются с помощью контекстно-свободных грамматик (тип 2 иерархии Хомского). Программа, выполняющая синтаксический анализ, называется синтаксическим анализатором. Исходными данными синтаксического анализатора являются таблицы лексем и идентификаторов. Лексемы являются для синтаксического анализатора терминальными символами контекстно-свободной грамматики. Результат работы синтаксического анализатора – дерево разбора (промежуточное представление кода). Распозна-

вателями 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 для выполнения синтаксического разбора, описать порядок приведения грамматики (исключение недостижимых символов, лямбда-правил и цепных правил) и преобразование грамматики 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 цепочки языка из контрольного примера.

4.4. Основные структуры данных

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

4.5. Описание алгоритма синтаксического разбора

Описать алгоритм синтаксического разбора, построить обобщенную блок-схему алгоритма синтаксического анализа.

4.6. Структура и перечень сообщений синтаксического анализатора

Представить перечень и формат сообщений синтаксического анализатора.

4.7. Параметры синтаксического анализатора и режимы его работы

Перечислить входные параметры транслятора, используемые синтаксическим анализатором в своей работе (для создания протокола

разбора, сохранения дерева разбора и т.п.), описать режимы работы синтаксического анализатора с заданными параметрами.

4.8. Принцип обработки ошибок

Описать действие синтаксического анализатора, при обнаружении ошибки в исходном коде, определить порядок вывода сообщений об ошибках, указать установленное ограничение на количество ошибок.

4.9. Контрольный пример

Привести распечатку протокола синтаксического разбора конструкций языка 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 фазу или совмещаться с фазами лексического и синтаксического анализа. Семантический анализатор использует синтаксическое дерево и информацию из таблицы идентификаторов для проверки исходного текста на соответствие семантическим правилам языка. Основные действия семантического анализатора:

- проверка семантических правил исходного языка;
- дополнение внутреннего представления программы операторами и действиями, неявно предусмотренными семантикой исходного языка.

5.1. Структура семантического анализатора

Описать назначение семантического анализа, его входные и выходные данные. Привести схему семантического анализатора, поясняющую его взаимодействие с другими компонентами транслятора.

5.2. Функции семантического анализатора

Перечислить семантические проверки с указанием фаз их выполнения (лексического, синтаксического анализаторов или отдельно) и функций, реализующих проверку, привести краткое описание этих функций.

5.3. Структура и перечень сообщений семантического анализатора

Представить в табличном виде перечень, структуру и текст сообщений семантического анализатора.

5.4. Принцип обработки ошибок

Описать действие семантического анализатора, при обнаружении ошибки в исходном коде 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. Включить в пояснительную 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 размещается между операндами, называется инфиксной. Обратная польская нотация (называют также польской инверсной записью, ПОЛИЗ) — это форма записи математических выражений, в которой операторы расположены после своих операндов. Выражения, представленные в обратной польской нотации, легко вычисляются, время вычисления — линейное. Выражение в обратной польской нотации читается слева направо: операция выполняется над двумя операндами, непосредственно стоящими перед знаком этой операции. Результат операции заменяет в выражении последовательность её операндов и символ операции. Результатом 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, используемые 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 контрольного примера в обратную польскую нотацию.

6.3. Программная реализация обработки выражений

Привести фрагмент кода транслятора на языке C++, реализующего преобразование выражений в обратный польский формат, дать краткие пояснения к коду, указать параметр транслятора, позволяющий отобразить в протоколе результаты преобразования выражений в обратный польский формат.

6.4. Контрольный пример

Привести часть протокола контрольного примера, отображающую результаты преобразования выражений в обратную польскую запись (формат: выражение – обратная польская запись для выражения).

7. Генерация кода

В разделе описывается процесс генерации кода. Генерация кода – четвертая последняя фаза работы транслятора. Исходными данными для генератора кода является промежуточное представление исходной программы. Одной из основных задач на этапе генерации кода является планирование памяти для переменных, литералов. Решение о распределении памяти принимается либо в процессе генерации промежуточного кода, либо при генерации целевого кода.

Самый простой подход к выполнению курсового проекта – это генерация кода в JavaScript. Интерпретатором JavaScript является js-движок браузера. Входными данными для генерации являются таблицы лексем и идентификаторов и дерево разбора.

Другой подход: можно выполнить генерацию в код на языке ассемблера. Затем вызвать транслятор ассемблера для получения объектного кода.

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

7.1. Структура генератора кода

Определить целевой язык, в который выполняется трансляция исходного кода с разрабатываемого языка программирования, описать процесс генерации кода, привести структуру генератора кода.

7.2. Представление типов данных в оперативной памяти

Подробно описать построение модели памяти, определить соответствие типов данных в исходном языке программирования типам данных целевого языка, дать подробные пояснения.

7.3. Статическая библиотека

Привести состав статической библиотеки, назначение функций, входящих в состав библиотеки, способ подключения библиотеки, этап, на котором производится подключение статической библиотеки.

7.4. Особенности алгоритма генерации кода

Построить обобщенную блок-схему алгоритма генерации кода и привести ее описание, указать особенности алгоритма.

7.5. Входные параметры генератора кода

Описать входные параметры генератора кода, указать их назначение.

7.6. Контрольный пример

Привести результат генерации кода на основе контрольного примера.

8. Тестирование транслятора

Подготовить контрольный пример, демонстрирующий правильную работу компилятора. Выполнить подбор тестов, которые должны быть включены в тестовый комплект. Тесты должны полностью покрывать список ошибок, обнаруживаемых тестируемым компилятором на разных фазах трансляции.

8.1. Общие положения

Описать общие принципы, лежащие в основе тестов, описать действия 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 виде таблицы, содержащей фрагмент исходного кода с ошибкой и соответствующее диагностическое сообщение (код ошибки, этап, текст сообщения, место ошибки с указанием строки и позиции в исходном коде).

9. Разработка и тестирование интерпретатора

Описать процесс разработки и тестирования интерпретатора.

9.1. Структура и перечень сообщений времени выполнения

Представить в табличном виде перечень, структуру и текст сообщений 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.

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 использованных 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. д.

3 ОФОРМЛЕНИЕ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ

3.1 Общие требования

При оформлении пояснительной записки следует руководствоваться требованиями СТП-001-2010 БГТУ. Дипломные проекты (работы).

Пояснительная записка к курсовой работе оформляется в соответствии 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 методическом 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 препинания обязательно ставится пробел. Нельзя сокращать слова (кроме сокращений, установленных правилами орфографии). Графический материал оформляется на листах формата А3 с рамкой.

Текст следует печатать, соблюдая поля: правое – 10±1 мм; верхнее – 20±1 мм; левое – 23±1 мм; нижнее – 15±1 мм. При наличии на листе рамки и основной надписи по форме 2 расстояние между верхней 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», «Заключение», «Список использованных источников», «Графический материал», а также каждый из основных разделов и каждое из приложений следует начинать с нового листа.

Заголовки элементов текста «Содержание», «Введение», «Заключение», «Список использованных источников», «Графический материал» следует записывать в начале соответствующих страниц строчными буквами, первая буква – прописная; шрифт – полужирный. Расположение – 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 цифрами, точка в конце не ставится; подразделы нумеруются в пределах 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 заголовок 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 титульном листе, первой странице пояснительной записки, номер не указывается. Номер проставляется арабской цифрой без точки в правом верхнем углу страницы.

Графический материал, размещенный на листе формата А3, учитывается как одна страница.

3.4 Перечисления

В тексте пояснительной записки могут быть использованы перечисления. Пункты перечисления записывают после двоеточия в виде списка, каждый с абзацного отступа. Перед каждым пунктом ненумерованного списка перечисления следует ставить тире. При необходимости ссылки в тексте на один или несколько пунктов перечисления, перечисление оформляют в виде маркированного списка, каждый пункт начинают со строчной буквы русского алфавита (за исключением 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 перечисления выполняют с абзацными отступами по отношению к основному.

Пример выполнения простого перечисления.

Основные действия семантического анализатора:

- проверка соблюдения в исходной программе семантических правил входного языка;
- дополнение внутреннего представления программы в компиляторе операторами и действиями, неявно предусмотренными семантикой входного языка.

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 термины;
- применять сокращения слов, кроме установленных правилами 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 сокращения 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 записываются на уровне формулы в круглых скобках справа в конце строки. Пояснения символов и числовых коэффициентов, входящих в формулу, должны быть приведены непосредственно под формулой. Пояснения должны начинаться со слова «где» и далее следует описание каждого символа в той последовательности, в которой они приведены в формуле по одному в строке.

Размер шрифта символов в формулах и уравнениях должен соответствовать 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 иллюстраций.

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

цы или графического материала, к которым они относятся. Если примечание 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 размером 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. Рисунки нумеруются арабскими цифрами сквозной нумерацией внутри раздела и располагаются с абзацного отступа. Рисунки каждого приложения состоят из обозначения приложения, точки и сквозной нумерации, выполненной арабскими цифрами. Вначале располагается сам рисунок, затем подрисуночный текст в виде: Рисунок, номер, наименование 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 оформляют в виде: Таблица, номер, наименование таблицы. Заголовок таблицы 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 указанием номера таблицы. Шапку таблицы при переносе части таблицы повторяют. 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 часовой стрелке.

Таблицы слева, справа и снизу ограничивают линиями.

3.10 Ссылки

Ссылки на разделы, подразделы, перечисления, таблицы, иллюстрации, формулы и приложения пояснительной записки следует выполнять 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 приложении Д…».

Ссылку на литературный источник выполняют с указанием порядкового номера источника, под которым он внесен в «Список использованных источников» пояснительной записки. Номер ссылки проставляется арабскими цифрами в квадратных скобках, например: [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 прописной 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 обозначается «Приложение А».

Приложения должны иметь заголовки, которые записывают симметрично 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., использованные при выполнении курсовой работы. Источники в списке располагаются в порядке ссылок в тексте записки или по алфавиту,

нумеруются арабскими цифрами без точки и печатаются с абзацного отступа, при этом дается библиографическое описание каждого источника в соответствии с ГОСТ 7.1, ГОСТ 7.12. Общий шаблон описания книги, у которой не более трех авторов: ФИО_автора, название книги, точка, тире, город, двоеточие, издательство, запятая, год издания, точка, тире, количество страниц, буква «с», точка. Название города 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 руководство: 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. Биллинг. – М.: Бином, 2012 – 488 с. 2

Приложение А (обязательное)

Образец оформления титульного листа

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕР-СИТЕТ»

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 XXX-201X»

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 201_

ПРИЛОЖЕНИЕ Б (ОБЯЗАТЕЛЬНОЕ)

Образец оформления листа задания

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕР-СИТЕТ»
Факультет информационных технологий
Кафедра программной инженерии

Утверждаю
Заведующая кафедрой
_____ Н.В. Пацей
подпись инициалы и фамилия

“___”_____201_г.

ЗАДАНИЕ
к курсовому проектированию по дисциплине
"Языки программирования"

Специальность: _____Группа: ____

Студент: _____Фамилия Имя Отчество_____
(фамилия, имя, отчество)

1. Тема проекта _____Разработка компилятора XXX-201X__
2. Срок сдачи студентом проекта: ___ декабря 201_ 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, под которой происходит разработка, Windows 7 SP1 (64-bit). Типы данных: integer и string. Функции стандартной библиотеки: integer strlen(string) – определение длины строки, string substr (string, string) - извлечение подстроки. Арифметические операции: +, -, *. Оператор вывода в стандартный поток: output.

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

- Введение
- Спецификация языка программирования
- Структура транслятора
- 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 обязательных чертежей)

- Граф дерева разбора

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

Раздел	Консультант
Разработка синтаксического и семантического анализатора.	ФИО руководителя
Генерация кода. Разработка тестовых примеров.	ФИО руководителя
Оформление пояснительной записки к курсовому проект.	ФИО руководителя

7. Календарный план

№ п/п	Наименование этапов курсового проекта	Срок выполнения этапов проекта	Примечание
1	Спецификация специализированного языка XXX-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 15.09.201_

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

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

ЯЗЫКИ ПРОГРАММИРОВАНИЯ

Методические указания к выполнению курсового проекта

Составители: Наркевич Аделина Сергеевна, Смелов Владимир Владиславович

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.