

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ИМЕНИ В.Ф. УТКИНА»

КАФЕДРА

«ВЫЧИСЛИТЕЛЬНАЯ И ПРИКЛАДНАЯ МАТЕМАТИКА»

**МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ
Б1.В.ДВ.06 «РАЗРАБОТКА КОМПИЛЯТОРОВ»**

Направление подготовки – 09.03.04 «Программная инженерия»

Направленность (профиль) подготовки
«Программная инженерия»

ОПОП академического бакалавриата

Квалификация выпускника – бакалавр

Формы обучения – очная

Рязань 2022 г

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Рекомендуется следующим образом организовать время, необходимое для изучения дисциплины:

Для освоения лекционного материала следует: изучить конспект лекции в тот же день, после лекции: 10 – 15 минут, повторно прочитать конспект лекции за день перед следующей лекцией: 10 – 15 минут. Также следует изучить теоретический лекционный материал по рекомендуемому учебнику/ учебному пособию: 1 час в неделю.

Следует максимально использовать лекционное время для изучения дисциплины, понимания лекционного материала и написания конспекта лекций. В процессе лекционного занятия студент должен уметь выделять важные моменты и основные положения. При написании конспекта лекций следует придерживаться следующих правил и рекомендаций.

1. При ведении конспекта рекомендуется структурировать материал по разделам, главам, темам. Вести нумерацию формул. Выделять по каждой теме постановку задачи, основные положения, выводы. Кратко записывать те пояснения лектора, которые показались особенно важными. Это позволит при подготовке к сдаче зачёта и экзамена не запутаться в структуре лекционного материала.
2. Лекционный материал следует записывать в конспект лишь после того, как излагаемый лектором тезис будет вами дослушан до конца и понят.
3. При конспектировании следует отмечать непонятные, на данном этапе, положения, доказательства и пр.
4. Рекомендуется по каждой теме выразить свое мнение, комментарий, вывод.

Подготовка к практическим занятиям:

Практические занятия по дисциплине существенно дополняют лекции. В процессе анализа теоретических положений и решения практических задач студенты расширяют и углубляют свои знания, полученные из лекционного курса и учебников, приобретают умение применять общие закономерности к конкретным случаям. В процессе решения задач развивается логическое мышление и вырабатываются навыки вычислений, работы со справочной литературой. Практические занятия способствуют закреплению знаний и практических навыков, формированию конструктивного стиля мышления, расширению кругозора.

При подготовке к практическому занятию необходимо внимательно ознакомиться с соответствующим теоретическим материалом по конспекту

лекций и рекомендованному учебнику, затем изучить конспект или материалы предыдущего практического занятия и выполнить заданное расчетное задание: 1 – 2 часа в неделю.

Следует максимально использовать аудиторное время практических занятий. В процессе занятия студент должен активно участвовать в дискуссиях, обсуждениях и решениях практических задач и вести конспект практических занятий отдельно от конспекта лекций.

Дополнительно в часы самостоятельной работы студенты могут повторно решить задачи, с которыми они плохо освоились во время аудиторных занятий, и обязательно те задачи, которые не получились дома при предыдущей подготовке к практическим занятиям.

Подготовка к лабораторным работам.

Перед началом проведения лабораторной работы необходимо ознакомиться с методическими указаниями к данной лабораторной работе, внимательно ознакомиться с заданием и желательно заранее выполнить подготовку проекта в используемой инструментальной среде, чтобы время лабораторного занятия использовать для исправления ошибок, модификации проекта и защиты данной работы.

Выполнение каждой из запланированных работ заканчивается предоставлением отчета. Требования к форме и содержанию отчета приведены в методических указаниях к лабораторным работам или определяются преподавателем на первом занятии. Отчет по лабораторной работе студент должен начать оформлять еще на этапе подготовки к ее выполнению. Допускаясь к лабораторной работе, каждый студент должен представить преподавателю «заготовку» отчета, содержащую: оформленный титульный лист или название и номер работы при ведении общего конспекта, цель работы, задание, проект решения, полученные результаты, выводы.

Изучение методических указаний к лабораторной работе – 2 часа перед выполнением лабораторной работы и в ходе разработки проекта и 2 часа для оформления отчета, отладки проекта и подготовки к сдаче работы.

После выполнения лабораторной работы необходимо согласовать полученные результаты с преподавателем. Важным этапом является защита лабораторной работы. В процессе защиты студент отвечает на вопросы преподавателя, касающиеся теоретического материала, относящегося к данной работе, и проекта, реализующего его задание, комментирует полученные в ходе работы результаты. При подготовке к защите лабораторной работы рекомендуется ознакомиться со списком вопросов по изучаемой теме и попытаться самостоятельно на них ответить, используя конспект лекций и рекомендованную

литературу. Кроме чтения учебной литературы рекомендуется активно использовать информационные ресурсы сети Интернет по изучаемой теме.

Подготовка к сдаче экзамена или зачета:

Экзамен/зачет – форма промежуточной проверки знаний, умений, навыков, степени освоения дисциплины. Главная задача экзамена/зачета состоит в том, чтобы у студента по окончании изучения данной дисциплины сформировались определенное представление об общем содержании дисциплины, определенные теоретические знания и практические навыки, определенный кругозор. Готовясь к экзамену/зачету, студент приводит в систему знания, полученные на лекциях, на практических и лабораторных занятиях, разбирается в том, что осталось непонятным, и тогда изучаемая им дисциплина может быть воспринята в полном объеме с присущей ей строгостью и логичностью, ее практической направленностью.

Экзамены/зачеты дают возможность преподавателю определить теоретические знания студента и его практические навыки при решении определенных прикладных задач. Оцениваются: понимание и степень усвоения теоретического материала; степень знакомства с основной и дополнительно литературой, а также с современными публикациями; умение применить теорию к практике, решать определенные практические задачи данной предметной области, правильно проводить расчеты и т. д.; знакомство с историей данной науки; логика, структура и стиль ответа, умение защищать выдвигаемые положения.

Значение экзаменов/зачетов не ограничивается проверкой знаний, являясь естественным завершением обучения студента по данной дисциплине, они способствуют обобщению и закреплению знаний и умений, приведению их в стройную систему, а также устранению возникших в процессе обучения пробелов.

Подготовка к экзамену/зачету – это тщательное изучение и систематизация учебного материала, осмысление и запоминание теоретических положений, формулировок, формул, установление и осмысление внутривидовых связей между различными темами и разделами дисциплины, закрепление теоретических знаний путем решения определенных задач.

Перед экзаменом назначается консультация, ее цель – дать ответы на вопросы, возникшие в ходе самостоятельной подготовки студента, студент имеет возможность получить ответ на все неясные ему вопросы, кроме того, преподаватель будет отвечать на вопросы других студентов, что будет способствовать повторению и закреплению знаний всех присутствующих. Преподаватель на консультации, как правило, обращает внимание на те разделы, по которым на предыдущих экзаменах ответы были

неудовлетворительными, а также фиксирует внимание на наиболее трудных разделах курса.

На непосредственную подготовку к экзамену обычно дается 3 – 5 дней. Этого времени достаточно для углубления, расширения и систематизации знаний, полученных в ходе обучения, на устранение пробелов в знании отдельных вопросов, для определения объема ответов на каждый из вопросов рабочей программы дисциплины.

Планируйте подготовку к зачету/экзамену, учитывая сразу несколько факторов: неоднородность в сложности учебного материала и степени его проработки в ходе обучения, свои индивидуальные способности. Рекомендуется делать перерывы в занятиях через каждые 50-60 минут на 10 минут. После 3-4 часов занятий следует сделать часовой перерыв. Чрезмерное утомление приведет к снижению тонуса интеллектуальной деятельности. Целесообразно разделять весь рабочий день на три рабочих периода – с утра до обеда, с обеда до ужина и с ужина до сна. Каждый рабочий период дня должен заканчиваться отдыхом не менее 1 часа. Работая в сессионном режиме, студент имеет возможность увеличить время занятий с 10 (как требовалось в семестре) до 12 часов в сутки.

Подготовку к экзаменам или зачетам следует начинать с общего планирования своей деятельности. С определения объема материала, подлежащего проработке, необходимо внимательно сверить свои конспекты с программой дисциплины, чтобы убедиться, все ли разделы отражены в лекциях, отсутствующие темы изучить по учебнику. Второй этап предусматривает системное изучение материала по данному предмету с обязательной записью всех выкладок, выводов, формул. На третьем этапе – этапе закрепления – полезно чередовать углубленное повторение особенно сложных вопросов с беглым повторением всего материала.

Рекомендации по работе с литературой:

Теоретический материал курса становится более понятным, когда дополнительно к прослушиванию лекции и изучению конспекта изучаются и книги по данному предмету. Литературу по дисциплине рекомендуется читать как в бумажном, так и в электронном виде (если отсутствует бумажный аналог). Полезно использовать несколько учебников и пособий по дисциплине. Рекомендуется после изучения очередного параграфа ответить на несколько вопросов по данной теме. Кроме того, полезно мысленно задать себе следующие вопросы (и попробовать ответить на них): «о чем этот параграф?», «какие новые понятия введены, каков их смысл?», «зачем мне это нужно по специальности?».

Рекомендуется самостоятельно изучать материал, который еще не прочитан на лекции и не применялся на лабораторном или практическом занятии, тогда занятия будут гораздо понятнее. В течение недели рекомендуется выбрать время (1 час) для работы с литературой.

Лабораторная работа №1. Создание системной утилиты, управляемой параметрами запуска.

Цель работы:

Создание консольной утилиты для генерации арифметических выражений и их трансляции в словесное представление.

Задание:

Необходимо сделать консольную утилиту, которая имеет два режима работы: генерация и трансляция арифметических выражений.

1) Режим генерации. Утилита принимает в качестве параметра аргумент «G», имя файла, количество строк, минимальное количество операндов, максимальное количество операндов.

Пример: lab1.exe G output.txt 2 4 8

После этого утилита записывает в указанный файл output.txt указанное количество строк арифметических выражений вида:

$$1 + 2 * 7 - 8 + 5$$

$$4 - 6 : 4 + 3 * 2 + 5 * 6$$

Для генерации в качестве операндов используются цифры 1...9 и четыре арифметические операции + - * :

Количество операндов в каждом выражении задаётся случайным образом, исходя из двух последних аргументов вызова (т.е. в нашем примере это от 4 до 8 операндов).

2) Режим трансляции. Утилита принимает в качестве параметра аргумент «Т», имя транслируемого файла, имя выходного файла.

Пример: lab1.exe T output.txt output_trans.txt

После этого утилита считывает входной файл output.txt и транслирует его в output_trans.txt заменяя цифры и операции на их словесное представление. Для предыдущего примера трансляция будет выглядеть следующим образом:

один плюс два умножить на семь минус восемь плюс пять

четыре минус шесть делить на четыре плюс три умножить на два плюс пять умножить на шесть

Утилита должна правильно обрабатывать ошибочный ввод, например, передача неправильного количества параметров, неправильные значение параметров, передача для трансляции несуществующего файла и т.д. и выдавать описание ошибки в случае её возникновения.

Лабораторная работа №2. Создание системной утилиты, для лексического анализа.

Цель работы:

Создание консольной утилиты для лексического анализа арифметических выражений, создание последовательности токенов и таблицы символов.

Задание:

Необходимо создать консольную утилиту, которая на вход принимает текстовый файл с последовательностью символов, представляющих произвольное арифметическое выражение, а на выходе создаёт файл с последовательностью токенов и информацией по ним и файл, содержащий таблицу символов.

Пример работы утилиты: lab2.exe inputExpr.txt

Утилита принимает в качестве параметра имя файла с выражением inputExpr.txt:

$var1 + (9.5 - 5 * (var2 - 0.6)) / var3$

В выражении могут быть четыре арифметические операции +, -, *, /; скобки любого уровня вложенности, вещественные и целочисленные константы, идентификаторы переменных с традиционными правилами именования (состоят из букв английского алфавита, цифр, знака нижнее подчёркивание , должен начинаться с буквы или знака нижнего подчёркивания).

Далее, произведя лексический анализ, утилита создаёт выходной файл с последовательностью токенов tokens.txt следующего содержания:

- <id,1> - идентификатор с именем var1
- <+> - операция сложения
- <(> - открывающая скобка
- <9.5> - константа вещественного типа
- <5> - константа целого типа
- <*> - операция умножения

<(> - открывающая скобка
<id,2> - идентификатор с именем var2
<-> - операция вычитания
<0.6> - константа вещественного типа
<)> - закрывающая скобка
<)> - закрывающая скобка
</> - операция деления
<id,3> - идентификатор с именем var3

также создаётся файл с таблицей символов symbols.txt:

1 – var1
2 – var2
3 – var3

В процессе работы лексического анализатора могут возникать ошибки анализа из-за недопустимых символов в потоке. Подобные ошибки должны выявляться и правильно обрабатываться, выдавая в консоль сообщение об возникшей ошибке и позиции в потоке символов, на котором она возникла, например:

Лексическая ошибка! Недопустимый символ “#” на позиции 4
Лексическая ошибка! Идентификатор «1 var» не может начинаться с цифры на позиции 15
Лексическая ошибка! Неправильно задана константа «0.5.4555» на позиции 9

Обратите внимание, синтаксические ошибки лексическим анализатором не обрабатываются! Эта задача синтаксического анализатора. Таким образом ошибки вида «не закрыта открывающая скобка» или «два знака операции следуют подряд» или «у оператора не хватает операнда» не определяются на этапе лексического анализа.

Лабораторная работа №3. Создание системной утилиты, для синтаксического анализа.

Цель работы:

Модификация консольной утилиты, добавление синтаксического анализа, проверка синтаксических ошибок, генерация дерева разбора.

Задание:

Необходимо дополнить функционал консольной утилиты, созданной на предыдущей лабораторной работе, синтаксическим анализом и генератором синтаксического дерева.

Во-первых, после формирования последовательности токенов, утилита должна проверить синтаксис выражения, и сообщить об ошибке в случае их обнаружения.

Например:

Синтаксическая ошибка! У операции $\langle * \rangle$ на позиции 4 отсутствует операнд
Синтаксическая ошибка! У открывающей скобки $\langle (\rangle$ на позиции 15 отсутствует закрывающая скобка.

Синтаксическая ошибка! После идентификатора $\langle id,2 \rangle$ на позиции 26 отсутствует операция.

Во-вторых, после проверки синтаксиса, в случае если ошибок не обнаружено, утилита должна построить синтаксическое дерево заданного выражения вида:



Все токены скобок приоритета операций при этом отбрасываются, тем самым приоритет операций задаётся порядком обхода дерева.

В-третьих, утилита должна после формирования синтаксического дерева, визуализировать его в текстовый файл `syntax_tree.txt` с помощью `ascii` символов любым способом.

Например так:

```

<=>
|---<id,1>
|---<+>
  |---<id,2>
  |---<*>
    |---<id,3>
    |---<60>

```

Или так (более сложный вариант):

```

          <=>
        /---|---\
<id,1>      <+>
          /---|---\
        <id,2>      <*>
              /---|---\
                <id,3>      <60>

```

Лабораторная работа №4. Создание системной утилиты, для семантического анализа.

Цель работы:

Модификация консольной утилиты, добавление семантического анализа, проверка семантических ошибок, построение постфиксной записи арифметического выражения.

Задание:

После получения синтаксического дерева на предыдущей лабораторной работе, необходимо дополнить утилиту функцией трансляции выражения в постфиксный вид, используя обход синтаксического дерева в глубину. В итоге должно получиться постфиксное выражение:

Входное выражение: $A + B * 9$

Выходное выражение: $A B 9 * +$

После того как была сгенерирована постфиксная запись выражения, необходимо спросить у пользователя значения переменных, присутствующих в выражении. После их ввода пользователем, программа по шагам вычисляет выражение в постфиксном виде:

A B 9 * +
Введите A?
7.3
Введите B?
-4

Вычисление выражения по постфиксной записи:
7.3 -4 9 * +
7.3 -36 +
-28.7

Таким образом был получен результат введённого выражения. Весь вывод и расчёт постфиксной записи, а также значения переменных необходимо продублировать в файл postfix.txt

Практическое занятие по теме № 1

Вопросы:

1. Язык. Задачи обработки текстов. Языки программирования. Алфавит. Синтаксис. Семантика.
2. Архитектура фон Неймана.
3. Язык программирования. Языки низкого уровня. Языки высокого уровня.
4. Поколения языков программирования.
5. Парадигмы языков программирования. Императивные языки.
6. Парадигмы языков программирования. Функциональные языки.
7. Парадигмы языков программирования. Декларативные языки.
8. Парадигмы языков программирования. Объектно-ориентированные языки.
9. Трансляторы. Проблема трансляции языка программирования.

10. Виртуальная трехадресная машина. Переход к мнемокоду. Примеры.
11. Требования, предъявляемые к транслятору.
12. Компиляция и интерпретация. Примеры.
13. Компилятор. Целевая программа. Схема.
14. Интерпретатор. Схема.
15. Система обработки языка. Схема. Препроцессор. Компилятор.

Практическое занятие по теме № 2

Вопросы:

1. Система обработки языка. Схема. Ассемблер. Компоновщик. Загрузчик.
2. Структура компилятора. Анализ.
3. Структура компилятора. Синтез.
4. Фазы компиляции. Схема.
5. Лексический анализ. Токены и атрибуты. Пример.
6. Синтаксический анализ. Грамматическая структура. Пример.
7. Семантический анализ. Пример.
8. Приведения (coercion). Пример.
9. Генерация промежуточного кода. Пример.
10. Оптимизация кода. Пример.
11. Генерация кода. Пример.
12. Таблица символов. Пример. Проходы компилятора.
13. Инфиксная и постфиксная форма выражения. Назначение. Примеры.
14. Модель начальной стадии компилятора. Схема. Лексический анализатор. Токены.
15. Фаза генерации промежуточного кода. Виды промежуточного кода.

Практическое занятие по теме № 3

Вопросы:

1. Абстрактное синтаксическое дерево. Пример.
2. Контекстно-свободная грамматика. Пример.
3. Определение грамматик. Компоненты грамматик. Терминалы и нетерминалы.
4. Лексемы. Токены. Терминалы.
5. Пример составления грамматики для выражений, состоящих из цифр и знаков плюс и минус.
6. Выведение (Порождение). Пример для грамматики для выражений, состоящих из цифр и знаков плюс и минус.
7. Пример составления грамматики для функций и их параметров.

8. Деревья разбора. Пример. Свойства.
9. Деревья разбора. Пример. Узлы. Метки. Корень. Родитель. Дочерний
10. Деревья разбора. Пример. Родственный узел. Лист. Внутренний узел. Потомок. Предок.
11. Пример построения дерева для выражения, состоящих из цифр и знаков плюс и минус.
12. Неоднозначности грамматики. Пример.
13. Ассоциативность операторов. Пример грамматики и дерева разбора.
14. Приоритет операторов. Пример построения грамматики.
15. Синтаксически управляемая трансляция. Пример.

Практическое занятие по теме № 4

Вопросы:

1. Синтаксически управляемая трансляция. Атрибуты.
2. Синтаксически управляемая трансляция. Схемы трансляции.
3. Постфиксная запись. Правило трансляции. Пример трансляции.
4. Постфиксная запись. Арность. Правило вычисления. Пример вычисления.
5. Синтезированные атрибуты. Синтаксически управляемое определение. Вычисление атрибутов.
6. Аннотированное дерево разбора. Пример.
7. Синтаксически управляемые определения для трансляции инфиксных выражений в постфиксные.
8. Обходы дерева. Обход в глубину. Схема.
9. Обходы дерева. Прямой и обратный порядок.
10. Схемы трансляции. Семантические действия. Пример продукции и дерева разбора с семантическим действием.
11. Семантические действия. Пример действий трансляции выражения в постфиксную запись, дерево разбора.
12. Разбор. Классы методов разбора.
13. Нисходящий анализ. Построение дерева разбора.
14. Предиктивный анализ. Использование пустых продукций.
15. Разработка предиктивного анализатора. Левая и правая рекурсия.

Библиографический список

1. Ахо А., Сети Р., Ульман Дж. Компиляторы: принципы, технологии и инструменты. – М.: Издательский дом "Вильямс", 2001.
2. Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции. Т. 1. – М.: Мир, 1978.
3. Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции. Т. 2. – М.: Мир, 1978.
4. Братчиков И. Л. Синтаксис языков программирования. – М.: Наука, 1975.
5. Вайнгартен Ф. Трансляция языков программирования. – М.: Мир, 1977.
6. Гинзбург С. Математическая теория контекстно-свободных языков. – М.: Мир, 1977.
7. Гладкий А. В. Формальные грамматики и языки. – М.: Наука, 1973.
8. Грис Д. Конструирование компиляторов для цифровых вычислительных машин. – М.: Мир, 1975.
9. Гросс М., Лантен А. Теория формальных грамматик. – М.: Мир, 1971.
10. Кнут Д. О переводе (трансляции) языков слева направо. / В сб. "Языки и автоматы". – М.: Мир.
11. Кнут Д. Семантика контекстно-свободных языков. / В сб. "Семантика языков программирования". – М.: Мир, 1980.
12. Лебедев В. Н. Введение в системы программирования. – М.: Статистика, 1975.
13. Льюис Ф., Розенкранц Д., Стирнз Р. Теоретические основы построения трансляторов. – М.: Мир, 1979.
14. Льюис Ф., Розенкранц Д., Стирнз Р. Атрибутные трансляции. / В сб. "Семантика языков программирования". – М.: Мир, 1980.
15. Оалева Э. А., Самойленко В. П., Семенова О. Н. Формальные методы описания перевода. Учеб. пособие. – СПб.: СПбГЭТУ, 2000.
16. Оалева Э. А., Самойленко В. П. Формальные грамматики и распознающие автоматы. Учеб. пособие. – Л.: ЛЭТИ, 1991.

17. Опалева Э. А., Самойленко В. П., Семенова О. Н. Разработка языковых процессоров. Методические указания к курсовой работе. – СПб.: СПбГЭТУ, 2002.
18. Пратт, Зелковиц М. Языки программирования: реализация и разработка. – СПб.: Питер, 2001.
19. Рейуорд-Смит В. Дж. Теория формальных языков. Вводный курс. – М.: Радио и связь, 1988.
20. Себеста Р. У. Основные концепции языков программирования. / 5-е изд; Пер. с англ. – М.: Издательский дом "Вильямс", 2001.
21. Хопкрофт Дж., Мотвани Р., Ульман Дж. Введение в теорию автоматов языков и вычислений. / 2-е изд. Пер. с англ. – М.: Издательский дом Вильямс, 2002.
22. Пентус А.Е. Математическая теория формальных языков. - М.: Изд-во "Интернет университет информационных технологий - ИНТУИТ.ру", 2006. - 248 с.: ил.
23. Хопкрофт Дж., Мотвани Р., Ульман Дж. Введение в теорию автоматов, языков и вычислений. 2-е издание. - М.: Издательский дом "Вильямс", 2002. - 528 с.
24. Хантер Р. Основные концепции компиляторов. - М.: Издательский дом "Вильямс", 1984. - 256 с.
25. Соколов А.П. Системы программирования: теория, методы, алгоритмы. Учебное пособие. - М.: Финансы и статистика, 2004. - 320 с.
26. Свердлов С. Языки программирования и методы трансляции: Учебное пособие
27. Пратт Т., Зелковиц М. Языки программирования: разработка и реализация. 4-е изд. - СПб.: Питер, 2002. - 688 с.: ил.
27. Керниган Б., Ритчи Д. Язык программирования Си. — 2-е изд. — М.: Вильямс, 2007. — С. 304. — ISBN 0-13-110362-8.
28. Гукин Д. Язык программирования Си. — М.: Диалектика, 2006. — С. 352. — ISBN 0-7645-7068-4.
29. Подбельский В. В., Фомин С. С. Курс программирования на языке Си: учебник. — М.: ДМК Пресс, 2012. — 318 с. — ISBN 978-5-94074-449-8.
30. Прата С. Язык программирования С: Лекции и упражнения. — М.: Вильямс, 2006. — С. 960. — ISBN 5-8459-0986-4.
31. Прата С. Язык программирования С (С11). Лекции и упражнения, 6-е издание. — М.: Вильямс, 2015. — 928 с. — ISBN 978-5-8459-1950-2.

32.Шилдт Г. С: полное руководство, классическое издание. — М.:
Вильямс, 2010. — С. 704. — ISBN 978-5-8459-1709-6.

Составил: к.т.н., доцент кафедры
«Вычислительная и прикладная
математика»

Антипов О.В.

Заведующий кафедрой вычислительной и
прикладной математики, д-р техн. наук,
профессор

Овечкин Г.В.