

ПРОГРАММИРОВАНИЕ I

Программа курса «Современные методы и понятия программирования» на 2013-1014 гг

составил доцент, к.ф.-м.н. Бульонков М.А.

ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью курса является освоение студентами базовых понятий и методов программирования. Вводная часть курса даёт представление о различных сторонах программирования - как инженерной деятельности, как методологии и как научной дисциплины – и их взаимосвязи. Основная часть курса построена вокруг рассмотрения языков программирования, того, как используются и реализуются различные языковые конструкции. Несмотря на то, что основным языком, рассматриваемым в курсе, является язык C, детальное освоение этого языка не является ключевой целью. Практика современного программирования такова, что даже в рамках одной системы разработчик использует несколько языков, и способность осознанного выбора подходящего языка и инструментария является одним из основных профессиональных качеств. Достаточно низкий уровень большинства рассматриваемых языков также не случаен: он позволяет проследить не только назначение языковых конструкций, но и насколько эффективную реализацию они допускают. Предполагается, что на основе такого знания студент сможет более осознанно изучать и использовать на практике современные языки более высокого уровня рассматриваемые в последующих курсах.

Таким образом, у студента должен быть сформирован кругозор и понимание того, что и сами языки программирования являются «живыми» объектами со своей логикой развития. Курс представляет базовые средства – БНФ грамматики и синтаксические диаграммы - для описания языков программирования. Большая часть курса посвящена рассмотрению наиболее распространённых концепций, таких как типы данных, вычисления, управление на уровне выражений и операторов, модульная структура программы и т.п. При этом стимулируется альтернативное рассмотрение различных конструкций, позволяющих реализовать одну и ту же задачу.

Заканчивается курс краткое рассмотрение вопросов анализа сложности программ, некоторых методов преобразования и оптимизации программ. Студенты, слушающие курс, должны усвоить, что собственно кодирование составляет весьма незначительную часть деятельности программиста, а существенно более важные вопросы: понимаемость, сопровождаемость и эффективность программного обеспечения.

СОДЕРЖАНИЕ ЛЕКЦИЙ

Лекция 1. Области программирования как научно-технической дисциплины: пользовательское программирование, системное программирование, технология программирования, теоретическое программирование. Логическая модель ЭВМ. Дискретная память: байты, биты, слова, адресация. Вычислитель: виды команд управление и вычисление. Внешние и внутренние функции

операционной системы: управление устройствами и ресурсами, запуск процессов, файловая система, пользовательский интерфейс. Классификация языков программирования: машинный язык, языки макроассемблера, языки высокого и сверхвысокого уровня. Императивные, логические, функциональные языки программирования.

Лекция 2. Подходы к реализации языков программирования: интерпретаторы, трансляторы. T-диаграммы. Многофазная трансляция, многоуровневая интерпретация, кросс-компиляция, раскрутка. Понятие системы программирования: транслятор, исходный код, объектный код, редактор связей, загрузчик, готовая программа. Дополнительные средства разработки ПО: справочная система, отладчик, средства тестирования и профилирования, средства документирования, анализаторы исходного кода, средства сопровождения, версионирования и рефакторинга.

Лекция 3. Понятие языка программирования как знаковой системы. Понятие лексики, синтаксиса и семантики. Формальное описание лексики и синтаксиса. Форма Бэкуса-Наура, регуляризованная БНФ. Проблема национальных версий языков программирования. Контекстно-свободный и контекстно-зависимый синтаксис. Понятие синтаксического вывода и дерева разбора. Проблема неоднозначности. Синтаксические диаграммы. Устойчивость синтаксиса к ошибкам.

Лекция 4. Контекстно-зависимый анализ, анализ типов. Общее понятие об операционной, функциональной и аксиоматической семантиках. Понятие стиля в программировании: структурирование текста, мнемоничность имён, комментирование исходного текста. Понятие прагматики языковых конструкций. Преемственность языков программирования, обратная совместимость.

Лекция 5. Препробессор: назначение, язык препроцессора. Директивы препроцессора на примере языка C: включение файлов, определение и использование макропеременных, предопределённые макропеременные, условная трансляция, конструирование лексем. Проблемы использования препроцессоров: дублирование кода, невозможность рекурсии, несоответствие директив структуре программы,

Лекции 6. Объекты и типы данных. Области видимости, блочная структура, правила поиска. Присоединяющие операторы и квалификация имён. Способы импорта библиотечных имён. Анонимные объекты. Понятие типа данных: моделируемая категория, синтаксис, множество литеральных значений, набор операций, реализация доступа к памяти. Статический и динамический анализ типов, их достоинства и недостатки. Строгая типизация. Полиморфизм. Классификация типов данных: предопределённые и определяемые, простые и структурированные, неупорядоченные, упорядоченные и перечислимые, арифметические.

Лекция 7. Описание базовых типов на примере языков C и Pascal. Символы как целочисленные коды, необходимость многобайтных кодировок. Различные представления целых и вещественных чисел. Числа с фиксированной и плавающей точкой, проблема потери точности. Множества и битовые шкалы, битовая арифметика. Логические значения и логические связки. Понятие приведения типов.

Лекция 8. Указатели на примере языка C. Адресная арифметика, тип void*. Реализация массивов как указателей. Виды массивы в других языках программирования: многомерные, динамические, подвижные, непрямоугольные, массивы-дескриторы. Массовые операции над массивами на примере языков Альфа и APL. Строки на примере языков C и Pascal.

Лекция 9. Структуры и объединения, понятие выравнивания, псевдооперация sizeof. Проблема «дыр» в контроле типов, связанных с типом объединения и пример её решения в Algol-68. Присваивание, как элементарное действия изменения состояния памяти. Побочные эффекты. Операции, совмещённые с присваиванием, инкремент и декремент.

Лекция 10. Управление в программе. Управление на уровне выражений, приоритет операций, последовательные и условные выражения, связки Маккарти. Управление на уровне операторов: оператор goto и его недостатки. Условные операторы: логические, арифметические, переключатели, переходы по вычисляемой метке, на примере Pascal, C, Fortran. Оператора цикла: for, while, repeat, циклы по множеству, переменные и границы цикла. Операторы выхода и продолжения цикла на примерах языков C, Pascal, Альфа, SETL, Java.

Лекция 11. Распределение памяти: глобальная, автоматическая, динамическая. Динамическое создание и удаление объектов и массивов. Накладные расходы и типичные ошибки. Автоматическая сборка мусора. Примеры представления динамических структур данных: деревья, односвязные и двусвязные списки, стеки, очереди, упорядоченные списки, вагонная пмять.

Лекция 12. Процедуры и функции: описание и вызов на примере C, формальные и фактические параметры. Выполнение функции: автоматическая память, связывание параметров, выполнение тела. Оператор возврата. Граф вызовов и дерево вызовов, поколения переменных. Рекурсия: достоинства и недостатки. Вложенные процедуры на примере Pascal, динамический контекст. Переменное число параметров, необязательные параметры, позиционные и именованные параметры.

Лекция 13. Виды подстановки параметров. Подстановка параметров по ссылке, процедуры с несколькими результатами. Проблема синонимов. Подстановка параметров по значению-результату. Строгое и нестрогое вычисление параметров. Подстановка параметров по имени. Подстановка параметров по необходимости. Процедурные параметры: функции обратного вызова (callback).

Лекция 14. Реализация функций. Преобразование функций в процедуры. Преобразование в функции с одним параметром. Использование стека фреймов. Реализация возврата из процедуры, вычисляемые метки, устранение процедур. Перемещение кода. Обработка исключительных ситуаций: коды ошибок, выход из глубокой рекурсии, нелокальные переходы setjmp/longjmp. Модульная структура программы, классы памяти.

Лекция 15. Ввод-вывод: логические и физические файлы, связь с объектами операционной системы. Виды языковых конструкций ввода-вывода: специальные конструкции, псевдопроцедуры, типизированные процедуры, форматный вывод (на пример C, Fortran, Modula-2). Низкоуровневый ввод-вывод: достоинства и недостатки. Буферизованный ввод-вывод:

типичные ошибки использования файлов. Стандартные файлы ввода-вывода. Форматный вывод: язык форматирования данных. Ввод-вывод указательных структур данных.

Лекция 16. Оценка сложности программ: временная и ёмкостная сложность, асимптотическое поведение. Сравнительный анализ алгоритмов. Примеры оптимизации и pessимизации программ.

ЛИТЕРАТУРА

Языки программирования

1. Болски М.И. Язык программирования Си. М.: Радио и связь. 1988.
2. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++, Второе издание: Пер. с англ. – М: Бином, 1999. 560 с.
3. Вирт Н., Иенсен К. Паскаль: руководство для пользователей и описание языка. М.: Финансы и статистика, 1982.
4. Джосьютис Н. C++ Стандартная библиотека. Для профессионалов. Пер. с англ. – СПб: Питер, 2004. 730с.
5. Калинин А.Г., Мацкевич И.В. Универсальные языки программирования. Семантический подход. М. Радио и связь 1991г. 400с.
6. Касьянов В.Н. Курс программирования на Паскале в заданиях и упражнениях. Новосибирск: НГУ, 2001.
7. Керниган Б., Ритчи Д., Фбьюэр А. Язык программирования Си. Задачи по языку Си. М.: Финансы и статистика, 1985
8. Прата С. Язык программирования C++. Лекции и упражнения. ДиаСофт, 2000.
9. Савитч У. Язык Java. Курс программирования. 2-е издание. М.: Издательский дом “Вильямс”. 2002.
10. Страуструп Б. Дизайн и эволюция C++. Пер. с англ. – М: ДМК Пресс; СПб: Питер, 2006. 448 с.
11. Страуструп Б. Язык программирования C++. Третье издание: Пер. с англ. – М: М: ДМК Пресс; СПб: Питер, 1999. 991 с.

Теория

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. - М.: Мир, 1978. - т. 1-2.
3. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. - М.: Мир, 1979.
4. Ахо А., Хопкрофт Дж., Ульман Дж. Структуры данных и алгоритмы. 384 стр., с ил.; 2000, 4 кв.; Вильямс
5. Вирт Н. Алгоритмы + структуры данных = программы. М.: Мир, 1985.
6. Грис Д. Наука программирования. М.: Мир, 1984.
7. Део Н., Нивергельт Ю., Рейнгольд Э. Комбинаторные алгоритмы: теория и практика. - М.: Мир, 1980.
8. Кармайкл Э., Хейвуд Д. Быстрая и качественная разработка программного обеспечения, 400 стр., с ил.; 2003, 1 кв.; Вильямс
9. Кнут Д. Искусство программирования на ЭВМ. – М.: Издательский дом “Вильямс”, т. I–III, 2000.

10. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. Серия Классические учебники: Computer Science. - М.: МЦНМО, 2002.
11. Кук Д., Бейз Г. Компьютерная математика. - М.: Наука, 1990.
12. Лавров С.С. Программирование. Математические основы, средства, теория. СПб: БХВ-Петербург, 2001.
13. Себеста Р.В. Основные концепции языков программирования, 5-е издание, 672 стр., с ил.; 2001, 3 кв.; Вильямс
14. Тамре Л. Введение в тестирование программного обеспечения, 368 стр., с ил.; 2003, 1 кв.; Вильямс
15. Фатрелл Р.Т., Шафер Д.Ф., Шафер Л.И. Управление программными проектами: достижение оптимального качества при минимуме затрат, 1136 стр., с ил.; 2003, 1 кв.; Вильямс
16. Хантер Р. Основные концепции компиляторов, 256 стр., с ил.; 2002, 4 кв.; Вильямс.
17. Хьюз Дж., Мичтом Дж. Структурный подход к программированию. М.: Мир, 1985.
18. Шень А. Программирование: теоремы и задачи. М.: МЦНМО, 1995.
19. Эндрюс Г.Р. Основы многопоточного, параллельного и распределенного программирования. - М.: Издательский дом "Вильямс". 2003.

ПРИЛОЖЕНИЕ. Экзаменационные задачи на удовлетворительную оценку

Тексты

1. Простой способ шифровки текста, состоящего из строчных латинских букв и знаков препинания, состоит в замене каждой буквы на букву с заданным циклическим сдвигом n (если сдвиг 1, то "А" заменяется на "В", "В" на "С", "Z" на "А"; если сдвиг 2, то "А" заменяется на "С", "У" на "А", "Z" на "В"). Написать процедуры зашифровки и расшифровки текста. Исходный текст берется из файла, результаты помещаются в другой файл.
2. Задан текстовый файл, содержащий последовательность идентификаторов, после каждого из которых стоит ровно один разделитель - точка. Нужно в каждом нечетном идентификаторе заменить первую букву на симметричную ('a' - 'z', 'v' - 'y', ... 'z' - 'a').
3. Задан текстовый файл, содержащий строчные латинских букв; между соседними словами пробел, в конце - точка. Написать программу, которая печатает в алфавитном порядке все буквы, которые входят только в одно единственное слово.
4. Составить программу для выдачи распределения слов в заданном текстовый файле по их длине (сколько слов из одной буквы, сколько из двух и т.д.). Слова состоят из строчных латинских букв и отделяются пробелами.
5. Задан текстовый файл, напечатать те буквы из 'А'...'М', которые встречаются в нём ровно два раза.
6. Задан текстовый файл, содержащий только строчные латинские буквы; между соседними словами пробел. Найти количество слов, содержащих больше трех букв "а".

7. Задан текстовый файл, символы которого перенумерованы, начиная с 0, найти номера первой и последней литеры самого длинного отрезка исходной последовательности, состоящего из букв латинского алфавита.

8. Подсчитать количество вхождений букв латинского алфавита в текстовом файле input.txt.

Результат в виде

a 5

b 4

...

z 0

выдать в файл output.txt.

9. Подсчитать и вывести на стандартный вывод количество строк, слов и символов в текстовом файле.

10. В строке записана последовательность натуральных чисел, разделённых нецифровыми символами: буквами, знаками препинания, пробелами и т.д. Найти сумму чисел последовательности.

Пример

текста:

В 12 часов 35 минут термометр показывал 23 градуса ниже нуля.

Ответ: 70.

11. Проверить, содержит ли данная строка все символы русского алфавита.

12. Проверить, сколько в файле строк являются палиндромами, т.е. читаются одинаково слева направо и справа налево (без учёта пробелов и знаков препинания). Не использовать дополнительную строку.

Пример

строк,

являющихся

палиндромами:

Он дивен, палиндром, и ни морд, ни лап не видно

А роза упала на лапу Азора

Числовые файлы (очередное число считывается с помощью fscanf)

13. В файле вещественных чисел найти первую пару стоящих рядом элементов, которые имеют разные знаки, а сумма их отрицательна. Если таких элементов нет - выдать соответствующее сообщение.

14. Задан непустой файл положительных целых чисел. Построить файл, не содержащий чисел, делящих максимальное число в исходном файле.

15. Найти длину максимальной неубывающей подпоследовательности в последовательности целых чисел.

16. Дан файл вещественных чисел, сформировать файл, содержащий только максимальные элементы из каждой максимальной неубывающей подпоследовательности подряд идущих элементов.
17. В файле целых чисел подсчитать количество элементов равных минимальному (файл просмотреть один раз).
18. Слить вместе два упорядоченных по возрастанию файла, сохранив упорядоченность и не дублируя совпадающие элементы.
19. В файле вещественных чисел найти первую пару стоящих рядом элементов, которые имеют разные знаки, а сумма их положительна. Если таких элементов нет - выдать соответствующее сообщение.
20. Найти три максимальных элемента в данном файле целых чисел.

Списки

21. Односвязный список задан указателем на первый элемент. Написать процедуру, переставляющие элементы списка в обратном порядке.
22. Элементы односвязного списка содержат поле *key* типа *int*. Список задан указателем на первый элемент. Упорядочить элементы списка по неубыванию значения поля *key*.
23. Написать процедуру, которая объединяет два упорядоченных по невозрастанию списка в один упорядоченный по невозрастанию, строя новый список. Элементы списков - целые числа.
24. Задан текстовый файл, содержащий последовательность вещественных чисел. Написать процедуру, которая строит по этой последовательности два односвязных списка, первый из которых содержит все неповторяющиеся положительные числа последовательности, а второй - все неповторяющиеся отрицательные

Матрицы, вектора

25. Написать процедуру, которая обнуляет в матрице $N \times N$ минимальный по абсолютной величине элемент, не лежащий на главной диагонали.
26. Задана матрица целых чисел $N \times N$, найти все номера столбцов, элементы которых упорядочены по возрастанию.
27. Дана вещественная матрица размера $M \times N$. Рассматривая ее как вектор строк, упорядочить строки по неубыванию суммы элементов строки.

28. Написать процедуру, которая печатает все числа, встречающиеся в вещественной матрице размера $M \times N$ более одного раза.
29. Написать процедуру, которая печатает все числа, встречающиеся в матрице размера $M \times N$ только один раз.
30. Для данной целочисленной матрицы $N \times N$, найти номер строки, в которой содержится наибольшее количество перемен знака (переменной знака в последовательности чисел называется ситуация, когда непосредственно или после нулевых значений за отрицательным числом идет положительное или за положительным - отрицательное).
31. Дан массив A целых чисел размера $M \times N$. По заданным значениям массива A построить вектор B длины M , k -ый элемент которого равен 1, если k -ая строка массива A симметрична, и 0 - в противном случае.

32. Задана матрица A размером $N \times M$. Построить матрицу B , такую что

$$B_{ij} = \sum_{k \geq i, l \geq j} A_{kl}$$

33. Написать функцию, которая для заданного массив целых чисел длины N , возвращает 1, если положительных чисел в нем больше, чем отрицательных, -1, если меньше, и 0, если их одинаковое количество.

Графы, деревья

34. Задано двоичное дерево, элементами которого являются целые числа. Написать рекурсивную функцию для нахождения наибольшего элемента дерева.
35. Написать функцию вычисления высоты двоичного дерева, то есть длины самого длинного пути от корня к листу.
36. В каждой вершине неупорядоченного двоичного дерева хранится вещественное число. Написать функцию, которая находит разницу между максимальным и минимальным числом в дереве (0 - для пустого дерева)
37. Написать процедуру, которая имеет параметром файл целых чисел, содержащий последовательность чисел, и которая выдает указатель на корень вновь построенное дерево двоичного поиска, имеющего данную последовательность в качестве результата префиксного обхода.

Разное

38. Многочлен от двух переменных задается в виде матрицы вещественных коэффициентов, в позиции (i,j) стоит коэффициент при $x^i y^j$. Найти произведение двух многочленов, заданных таким образом.
39. Простое число называется числом Мерсена, если оно может быть представлено в виде $2^p - 1$, где p - тоже простое число. Найти все числа Мерсена, меньшие данного n .
40. Время суток представлено в виде записи, содержащей информацию о часе, минутах и секундах. Написать процедуру, которая увеличивает значение времени на n секунд (после 23:59:59 идет 00:00:00).
41. Даны три натуральных числа A , B и N . Найти все натуральные числа не превосходящие N , которые можно представить в виде $Ar + Bq$, где p, q - натуральные, $p + q > 1$.
42. Гамма-функция $\Gamma(x)$ обладает свойством: $\Gamma(x+1) = x\Gamma(x)$. Пусть задана таблица приближенных значений функции на отрезке от $x=1.00$ до $x=2.00$ с шагом 0.01 . Описать рекурсивную функцию, приближенно вычисляющую $\Gamma(x)$ для $x > 1$ с помощью этой таблицы.
43. Назовем натуральное число палиндромом, если его десятичная запись читается одинаково с начала и с конца (например, 2112, 545). Найти все меньшие 100 натуральных числа, которые при возведении в квадрат дают палиндром.
44. Два натуральных числа представлены в k -ичной системе счисления как массивы "цифр" - натуральных чисел из интервала $[0..k-1]$. Найти разность заданных чисел в той же системе счисления..
45. Даны целые числа a_1, \dots, a_{30} . Пусть M - наибольшее, а m - наименьшее из a_1, \dots, a_{30} . Получить в порядке возрастания все целые числа из интервала (m, M) , которые не входят в последовательность a_1, \dots, a_{30} .
46. Написать процедуру, которая по паре целых чисел, представляющих числитель и знаменатель рационального числа, выдает два целых числа, представляющих числитель и знаменатель сокращенной дроби.
47. Написать функцию occurs, возвращающую количество вхождений строки target в строку source. Например, для source = "aabababa", target = "aba" результатом должно быть 3.
48. Написать процедуру отыскание корня уравнения для непрерывной на заданном интервале знакопеременной функции. Предполагать, что в заданном интервале корень один.
49. В программе задана непрерывная на отрезке $[a, b]$ функция $f: \mathbb{R} \rightarrow \mathbb{R}$. Вычислить приближенное значение интеграла.

50. Подсчитать количество единиц в двоичном представлении числа n .
51. Вычислить первые N членов ряда Фибоначчи: $f_0 = 0$, $f_1 = 1$, $f_i = f_{i-1} + f_{i-2}$.

«Базы данных»

52. Информация о преподавателе содержит следующие сведения: предмет: математика, физика, биология, химия ; штатный (1) или совместитель (0); фамилия (текст не более 20 символов); нагрузка в каждый из месяцев года (массив целых чисел). Месяцы года задавать в виде перечислимого типа. Дан файл, содержащий сведения о всех преподавателях. Напечатать фамилии всех преподавателей-совместителей, имеющих в третьей четверти нагрузку более 300 часов и указать название их предмета.
53. Анкета студента содержит: фамилию, номер группы, набор (не более 10) слушаемых курсов лекций. Составить программу, производящую ввод массива анкет и печатающую списки фамилий студентов, слушающих каждый курс лекций.
54. Анкета студента содержит: фамилию, номер группы, набор (не более 10) слушаемых курсов лекций. Составить программу, отыскивающую такие пары студентов (фамилии), которые слушают хотя бы один курс вместе.
55. Дано два файла `sklad.txt` и `order.txt`. В файле `sklad.txt` перечислен набор пар: название_товара (пробел) количество. Каждая пара на новой строке, причем название_товара не содержит пробелов. Каждый товар упоминается в файлах не более одного раза. Файл `order.txt` (заказ) содержит аналогичную информацию. Написать процедуру, которая преобразует файлы к состоянию после выполнения заказа. В файле `order.txt` должны остаться строки, соответствующие товарам, для которых нет достаточного количества на складе с указанием нехватки.