

**Техническое задание на выполнение курсовой работы
по курсу «Программирование» (2 семестр) на тему:
«Разработка прототипа программной системы для проблемной области**

1. Цели выполнения работы: знать и уметь использовать

1. Метод разработки программного обеспечения для решения задач программирования.
2. Структуру простых С-программ.
3. Простые типы данных `int`, `double` и `char`.
4. Функции ввода/вывода `printf` и `scanf`.
5. Директивы препроцессора `#include` и `#define`.
6. Входные и выходные форматирующие строки с заполнителями для символьных и числовых данных.
7. Комментарии в качестве средства документации программ.
8. Присваивание с основными арифметическими операторами и приведением типов, написание простых арифметических выражений.
9. Процесс написания простых программ, которые выполняют ввод, вычисление результатов и вывод.
10. Перенаправление ввода/вывода и управляемые программой файлы, отличие интерактивной от пакетной обработки данных.
11. Описание программ, иллюстрирующих нисходящее решение задачи, стратегию «разделяй и властвуй» и процесс пошагового уточнения.
12. Структурные схемы в качестве средства отображения иерархических отношений между подзадачами задачи.
13. Понятие структурированной программы как комбинации управляющих структур с одним входом и одной точкой выхода.
14. Функции `void` и функции, возвращающие единственный результат, в качестве средства создания модульных программ.
15. Входные параметры для передачи данных в функции.
16. Синтаксис и семантику определяемых пользователем функций, которые не возвращают никакого результата и функций, которые возвращают единственный результат.
17. Понятие многократного использования кода, вызывая несколько стандартных библиотечных функций, используемых в вычислениях.
18. Синтаксис и семантику одно и двухвариантных предложений `if`.
19. Операторы сравнения, равенства и логические операторы, в т.ч. в шагах принятия решений.
20. Синтаксис и семантику составных предложений.
21. Конструкцию вложенных и многовариантных предложений `if` для реализации сложной логики условий и таблиц решений.
22. Некоторые символы построения блок-схем, в т.ч. для схематического изображения решений в алгоритмах.
23. Информацию о потоках данных на структурных схемах.
24. Методику ручной трассировки алгоритма.

25. Синтаксис и семантику оператора switch, в т.ч. в многовариантной структуре принятия решений.
26. Понятие повторения в алгоритмах.
27. Синтаксис и семантику C-предложений while, for и do-while.
28. Циклы для вычисления суммы и произведения списков чисел.
29. Предложения while и for для построения счетных циклов, условных циклов, циклов, управляемых меткой, и циклов, управляемых концом файла.
30. Составное предложение в качестве тела цикла.
31. Циклы для отображения таблиц с информацией.
32. Вложенность в управляющих структурах.
33. Переменные типа int в качестве флажков программы.
34. Некоторые C-операторы, имеющие побочные эффекты над своими операндами: инкремент, декремент и составные арифметические операторы присваивания.
35. Синтаксис и семантику указателей, используемых в качестве выходных и входных/выходных параметров.
36. C-правила обзора данных, управляющие видимостью макросов констант, имен функций, формальных параметров и локальных переменных.
37. Взаимосвязь между потоком данных, показанным на структурной схеме и процессом проектирования программы с помощью пошагового уточнения.
38. Комментарий к заголовку функции и прототип функции в качестве документации всех аспектов функции, необходимых вызывающей ее функции.
39. Внутреннее представление значений типа int, double и char.
40. Проблему числовых погрешностей в вычислениях, использующих значения типа double.
41. Представление об автоматическом и явном преобразовании типов данных.
42. Синтаксис и семантику перечислимых типов.
43. Одномерные и многомерные массивы с элементами типа int, char или double.
44. Индексированные переменные для управления элементами массива.
45. Передачу единичных элементов массива и целых массивов в качестве фактических параметров функций.
46. Индексированный цикл for для последовательного доступа к элементам массива.
47. Сравнение последовательного и произвольного доступа к элементам массива.
48. Представление массива как указателя.
49. Массивы в качестве входных и выходных параметров функций.
50. Алгоритмы для поиска и сортировки массивов.

51. Строковый тип данных и C-реализацию строк переменной длины как одномерных символьных массивов со специальным символом, обозначающим конец каждой строки.
52. Основные операции со строками, такие как ввод/вывод строки, строковое присваивание, извлечение подстроки, конкатенация, длина строки, сравнение строк и преобразование типов между строками и числами.
53. Строки как параметры функции, при этом сопоставляя с параметрами одномерного массива.
54. Указатель на строку или на любой массив и как он может быть возвращен в качестве результата функции.
55. Библиотечные C-функции для символьных: ввода/вывода, классификации и преобразования.
56. C-представление массивов в качестве указателей и как оно позволяет поддерживать несколько упорядоченных списка строк, только единожды сохраняя эти строки.
57. Библиотечные строковые C-функции, предоставляя им пространство для результатов и делая их чувствительными к размеру этого пространства.
58. Понятие и обычную форму рекурсивного алгоритма.
59. Процесс написания рекурсивных функций.
60. Рекурсивные алгоритмы для математических функций, для построения строк и для обработки массивов.
61. Рекурсивные алгоритмы для решения задачи Ханойских башен и для реализации операций над множествами.

2. Задачи работы

1. Сформулировать постановку задачи (1-2 стр.), т.е. содержательное описание проблемной области и ограничений.
2. Проанализировать задачу, т.е. описать требования к данным (константы, входные/выходные данные, переменные, формулы, файлы с данными).
3. Разработать исходный алгоритм решения задачи и уточнить его (минимум 10 шагов с подшагами).
4. Нарисовать структурную схему (минимум двухуровневую) с потоками данных и именами функций.
5. Проанализировать требования к данным и разработать алгоритм каждой функции.
6. Реализовать конечный алгоритм решения задачи.
7. Протестировать все варианты использования системы.

3. Исходные данные

1. Материалы лекций.
2. Рекомендуемая литература.
3. *IDE Dev-C++* (<http://www.bloodshed.net/>).
4. Заданная проблемная область.

4. Результаты работы

Прототип программной системы для заданной проблемной области в электронном виде; отчет о выполнении работы.

5. Содержание отчета

1. Титульный лист.
2. Содержание.
3. Техническое задание на выполнение курсовой работы.
4. Постановка задачи.
5. Требования к данным задачи.
6. Алгоритм решения задачи с уточнениями.
7. Структурная схема.
8. Требования к данным и алгоритмы каждой функции.
9. Листинг программной системы.
- 10.Экранные формы вариантов использования системы.

Задание выдал:

доцент каф. ВТ
Мороз Ю.В.

Задание принял:

студент гр. ИВБ-1-11
