МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО «Российский химико-технологический

университет им. Д.И.Менделеева»

Новомосковский институт (филиал)

**Программирование и основы алгоритмизации**

Программа , контрольные задания и методические указания по курсу

2-е издание, переработанное

Новомосковск

2015

**УДК 681.326**

**ББК 22.18**

**Б447**

Рецензенты:

кандидат технических наук, доцент *И.Д. Моисеева*

(НИ (филиал) ФГБОУ ВО РХТУ им. Д.И. Менделеева)

кандидат технических наук, доцент *А.Г.Лопатин*

(НИ (филиал) ФГБОУ ВО РХТУ им. Д.И. Менделеева)

Составители: Предместьин В.Р. Мелешкина В.А.

Б447  **Программирование и основы алгоритмизации:** Программа и контрольные задания по курсу /ФГБОУ ВО РХТУ им. Д.И.Менделеева, Новомосковский институт (филал); Новомосковск, 2015. – 17 с.

2-е издание, переработанное

Программа , контрольные задания и методические указания содержат перечень основных понятий и определений по программе курса, перечень лабораторных работ, а так же контрольные задания с рекомендациями и указаниями к решению.

Предназначены для студентов направления 15.03.04 «Автоматизация технологических процессов и производств» заочной формы обучения.

Библиогр.:7 назв.

УДК 681.326

ББК 22.18

©Новомосковский институт (филиал)

ФГБОУ ВО «Российский химико-технологический

университет им. Д.И. Менделеева», 2015

### Оглавление

Введение 4

1.Цели и задачи дисциплины 5

2.Программа курса 5

3.Лабораторные работы 6

4.Перечень методических указаний и методических

материалов к программе курса 6

5.Контрольные задания 6

6.Тест для защиты контрольного задания 11

Библиографический список 16

**Введение**

Курс «Программирование и основы алгоритмизации» изучается в течение восьмого семестра по направлению 15.03.04 „Автоматизация технологических процессов и производств”. В соответствии с учебным планом студенты заочного отделения выполняют контрольные задания по курсу , позволяющие усвоить синтаксис и семантику алгоритмического языка программирования , методы построения алгоритмов и их реализации посредством современных средств программирования.

**1.ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ**

1.1Цель преподавания дисциплины

Цель преподавания дисциплины заключается в формировании у студентов знаний и умений в области алгоритмизации и программирования.

1.2 Задачи изучения дисциплины

В результате изучения дисциплины „ Программирование и алгоритмизация ” студенты

**должны знать**  **:**

- основные принципы и методы построения алгоритмов;

- основные теоретические положения в области структурного программирования и структур данных;

- принципы программирования на языке „Си ++” ;

**должны уметь :**

- составлять программы для решения прикладных задач;

- доводить прикладные программы до реализации на ЭВМ.

**должны иметь навыки :**

- составления алгоритмов прикладных задач;

-программирование на языке„Си ++”.

**2.ПРОГРАММА КУРСА**

2.1. Наименование тем, основные понятия дисциплины

Введение в разработку алгоритмов.

Полное построение алгоритма , основные виды , этапы проектирования алгоритмов. Синтаксис и семантика алгоритмического языка программирования. Документирование программных средств.

Некоторые основные приемы и алгоритмы.

Структурное и модульное программирование, типизация и структуризация программных данных, статические и динамические данные. Сложные структуры данных (связные списки, стеки, деревья), потоки ввода-вывода, файлы.

Методы разработки алгоритмов.

Класс алгоритмов. Методы частных целей, подъёма, отрабатывания назад. Эвристики. Программирование с отходом назад. Рекурсия. Итерация.

Алгоритмы машинной математики.

Алгоритмы сортировки простыми включениями, параллельной и быстрой сортировки. Алгоритм двоичного поиска.

Методы и средства объектно-ориентированного программирования.

Характеристики объектов. Абстракция. Инкапсуляция. Модульность. Иерархия.

Понятие классов и конструкторов.

**3. ЛАБОРАТОРНЫЕ РАБОТЫ (перечень)**

Лабораторная работа №1. Создание элементарных программ на Си ++. Их компиляция и выполнение.

Лабораторная работа №2. Основные элементы программирования ,вывод; типы данных; операций ; условные операторы и циклические конструкции.

Лабораторная работа №3. Описание и определение функций в программах.

Лабораторная работа № 4. Использование в программах структур данных (массивы, файлы).

**4.ПЕРЕЧЕНЬ МЕТОДИЧЕСКИХ УКАЗАНИЙ И МЕТОДИЧЕСКИХ МАТЕРИАЛОВ К ПРОГРАММЕ КУРСА**

4.1 Основная литература

1. Технология программирования на языке С++: Учебно-методическое пособие, Предместьин В. Р., Моисеева И. Д., Семенова М. Э., Гербер Ю. В. - Новомосковск 2012.-106с.

**5. КОНТРОЛЬНЫЕ ЗАДАНИЯ**

В процессе изучения курса „Программирование и основы алгоритмизации” студенты должны выполнить контрольную работу, заключающуюся в следующем .

**1**.написание реферата в соответствии с перечнем тем.

Названия тем соответствуют названиям вопросов , выносимых на экзамен, сдаваемый студентами по дисциплине в восьмом семестре . Номер темы соответствует регистрационному номеру студента в журнале преподавателя.

ТЕМЫ РЕФЕРАТОВ

1).Алгоритмы. Соглашения , принятые для написания алгоритмов.

2).Основные этапы полного построения алгоритма.

3).Структурное программирование.

4).Структурированные и неструктурированные блок- схемы.

5).Массивы. Связанные списки.

6).Алгоритмы удаления и добавления элементов связанного списка.

7).Стековые списки и стеки.

8).Очереди.

9).Методы частных целей , подъема и отрабатывания назад.

10).Эвристические алгоритмы.

11).Программирование с отходом назад.

12).Рекурсия. Итерация.

13).Сортировка .Алгоритм сортировки методом прямого включения.

14).Алгоритм быстрой сортировки.

15).Поиск. Процедура двоичного поиска.

16).Семь основных элементов программирования.

Для написания реферата необходимо использовать литературу [1, 2] из библиографического списка.

**2.** разработка алгоритмов решения предложенных заданий и их реализация на языке программирования „Си++”.

**2.1**. Разработать алгоритм и написать программу для выполнения следующих действий:

**2.1.1**.Сформировать логическим путем двумерный массив размерностью 55 в памяти компьютера , состоящий из 0 и 1, и вывести его на экран дисплея. Расположение 0 и 1 выбирается в соответствии с номером варианта, который соответствует регистрационному номеру студента в журнале преподавателя.

**2.1.2.** Создать новый файл на жестком диске и записать в него из памяти компьютера сформированную последовательность „0” и „1”.

**2.2.** Разработать алгоритм и написать программу для выполнения следующих действий :

**2.2.1**. Открыть файл, сформированный в предыдущем задании, и считать из него в память компьютера записанные данные.

**2.2.2.** Вывести результат на экран дисплея. Картинка должна совпадать с исходными данными задания 2.1.1.

**2.2.3.** Найти количество „0”и „1 ” в файле . Для реализации этого блока сформировать отдельную функцию .

**Варианты функций**

**1**.10001 **2.** 10000 **3.** 00001 **4.** 11111 **5.** 00000 **6.** 10000 **7**.00001

01010 11000 00011 01010 00000 11000 00011

00100 10100 00101 00100 00100 10100 00101

01010 11000 00011 00000 01010 10010 01001

10001 10000 00001 00000 11111 11111 11111

**8.** 00001 **9.** 10000 **10**. 11100 **11.** 00111 **12.**  11111 **13.**10001

00011 11000 11000 00110 01010 11011

11111 11111 00100 00100 00100 10101

11000 00011 00110 01100 01010 11011

10000 00001 00111 11100 11111 10001

При выполнении задания целесообразно использовать учебно-методическое пособие [2]. Литература рекомендуемая для изучения языка С++ приводится библиографическом списке [3-7].

Указания к выполнению заданий

При написании первой программы (п..2.1.) в первую очередь необходимо выделить память и определить тип данных, которые будут

использоваться в программе (стр.15-17, 57-60 ) . Например , строка char а[5][5] выделяет память размерностью 25 байт. Затем , для того ,чтобы сформировать нужную последовательность 0и1, необходимо использовать циклические и условные операторы, операции сравнения и логические операции (стр.27-37). Например , для формирования следующей последовательности

00100

00100

11111

00100

00100

можно написать следующий блок операторов:

for(i=0;i<5;i ++)

for(j=0;j<5;j++)

{ a[i][j]=0;

if((i= =3) ¦¦ (j= =3) a[i][j]=1;

}.

Для вывода на экран сформированного массива используются функции вывода printf или cout (стр.8-14).

Создание файла и записи в него информации производится функциями creat и write (стр.65-68).

Запись в файл с именем “file” в примере происходит следующим образом:

handle = creat("file", S\_IREAD | S\_IWRITE);

write(handle, a, sizeof(a));

close(handle);

Здесь а – адрес по которому хранятся данные (в примере адресом является имя массива), а sizeof(a) равное 25 – это количество записываемых байт.

Один из вариантов такой программы выглядит следующим образом:

#include <fcntl.h>

#include <sys\stat.h>

#include <stdio.h>

#include <io.h>

void main()

{

char a[5][5];

int i, j, handle;

for (i = 0; i < 5; i ++)

{

for (j = 0; j < 5; j ++)

{

a[i][j] = 0;

if ((i == 2) || (j == 2)) a[i][j] = 1;

printf ("%3d", a[i][j]);

}

printf ("\n");

}

\_fmode = O\_BINARY;

handle = creat("file", S\_IREAD | S\_IWRITE);

write(handle, a, sizeof(a));

close(handle);

}

При компиляции написанной программы помимо синтаксических ошибок наиболее вероятна ошибка, свидетельствующая об отсутствии описания прототипов, используемых функций. Для определения названия файла прототипа необходимо воспользоваться опцией „Help” редактора языка, а для подключения к программе этого файла используется директива # inсlude < имя файла > , которая вставляется перед основным модулем программы.

При написании второй программы для считывания информации из файла необходимо использовать функции open и read (стр.65-68). Для реализации п.2.2.3. в виде отдельной функции необходимо изучить стр. 43-47 учебно-методического пособия .

В качестве одного из вариантов программы можно предложить следующий.

#include <fcntl.h>

#include <sys\stat.h>

#include <stdio.h>

#include <io.h>

int put\_ratio(char a[][5]);

void main()

{

char a[5][5];

int i, j, handle;

handle = open ("file", O\_RDWR);

read (handle, a, 25);

close (handle);

for (i = 0; i < 5; i ++)

{

for (j = 0; j < 5; j ++)

{

printf("%3d", a[i][j]);

}

printf ("\n");

}

printf ("%d", put\_ratio(a));

}

int put\_ratio(char a[][5])

{

int i, j, sum;

sum = 0;

for (i = 0; i < 5; i ++)

{

for (int j = 0; j < 5; j ++)

{

if (a[i][j] == 1) sum ++;

}

}

return sum;

}

В этом примере в основной программе информация из файла с именем “file” записывается по адресу a, и затем выводится на экран. Для расчета количества “1” используется функция put\_ratio().

Кроме того, для защиты работы необходимо ответить на тест из 15 вопросов , которые выбираются автоматически из следующего списка:

1.Какова основная особенность структурированных блок-схем

а) один вход и один выход

б) два входа и один выход

в) один вход и два выхода

2. Какую структуру данных необходимо использовать для внесения или удаления элементов в середине массива

а) связанный список

б) список

в) очередь

3. Какой алгоритм быстрее и проще реализовать

а) оптимальный

б) эвристический

4. Какая модель используется при разработке алгоритма, известного как программирование с отходом назад

а)ориентированная сеть

б) дерево

в) сеть со многими циклами

5. В каком виде записывается альтернатива как структура управления

а) do S1;S2 od

б) if B then S1 else S2 fi

в) while B do S1 od

6. Какой из этапов полного построения алгоритмов является последним

а) реализация алгоритма

б) проверка программы

в) составление документации

7. Какое слово из соглашения , принятого для описания алгоритмов , используется с оператором присваивания

а) and

б) do

в) set

8. С чего начинается работа алгоритма , основанного на отработке назад

а) с постановки задачи и движения к ее цели или решению

б) с цели или решения и движения к начальной постановке задачи

9. Какие решения дают алгоритмы , основанные на методе подъема

а) всегда оптимальные

б) всегда приближенные

в) как оптимальные , так и приближенные

10. Что такое рекурсия

а) метод определения функции посредством той же функции

б) метод определения функции посредством другой функции

в) метод определения переменной посредством той же переменной

11. Функция дважды рекурсивна ,если

а) функция определена через саму себя

б) функция и один из ее элементов определен через самих себя

в) функция и два ее аргумента определены через самих себя

12. Сколько списков необходимо для реализации алгоритма сортировки прямым включением

а) один

б) два

в) три

13. Как расположены ключи в исходном файле, к которому применяется алгоритм двоичного поиска

а) упорядочено

б) неупорядочено

14. Какой тип данных имеет наибольший размер (в битах)

а) char

б) int

в) long

15. Какой тип данных используется для организации символьного массива

а) char

б) int

в) short

16. Какой подход используется в Си++ для определения строки

а) использование символьного массива

б) использование указателя на символ

в) оба подхода

17. Какой массив используется для реализации очереди

а) одномерный

б) двумерный

в) трехмерный

18. Выражение s=a+b++ в Си++ означает

а) сложить а и b , присвоить результат s и увеличить b на единицу

б) увеличить b на единицу , сложить a и b и присвоить результат s

19. Какая функция считывает символ с клавиатуры, не отображая его на экране

а) scanf

б) gets

в) getch

20. В конструкции «If< значение> оператор 1» оператор 1 выполняется

а) когда «значение» отлично от нуля

б) когда «значение» равно нулю

21. Какой цикл является наиболее общим и может использоваться вместо двух других

а) while

б) for

в)do … while

22. Функция типа void возвращает

а) целые числа

б) вещественные числа

в) ничего не возвращает

23. Динамическое распределение памяти позволяет программе

а) запрашивать некоторое количество памяти (в байтах)

б) возвращать адреса , которые можно запомнить в указателе

в) запрашивать некоторое количество памяти и возвращать адреса, которые можно запомнить в указателе

24. Что делает программа , когда выполняется оператор continue

а) выходит из цикла

б) пропускает остаток цикла

в) пропускает следующий за continue оператор

25. Что содержат объектные файлы

а) исходный код программы

б) двоичные машинные инструкции, созданные компилятором

в) исполняемый код программы

26. Распределением какой памяти управляют функции new и delete

а) статической памятью

б) свободной памятью (кучей)

27. Синтаксис каких функций ввода-вывода проще и удобней

а) printf и scanf

б) cin и cout

28. Как задается начальное значение указателя

а) в этом нет необходимости

б) инициализируются автоматически

в) задаются программно

29. Что является базовым адресом массива pointer [index]

а) pointer

б) &pointer

в) pointer[0]

30. Режим O\_RDWR в функции «open» означает

а) файл открывается только для чтения

б) файл открывается только для записи

в) файл открывается и для чтения и для записи

31. Что делает функция strcat

а) копирует строку из источника в приемник

б) присоединяет строку из источника к концу строки приемника

в) присоединяет строку из приемника к концу строки источника

32. Что делает функция itoa

а) преобразует строку в целое число

б) преобразует целое число в строку

в) преобразует в строку вещественное число

33. Какое свойство общее у классов и структур

а) членами и тех, и других могут быть как данные ,так и функции

б) синтаксис объявления объекта

в) степень доступности по умолчанию

34. Какой подход характерен для объектно- ориентированного программирования

а) разделение больших задач на мелкие подзадачи

б) представление задачи в виде взаимодействий между абстракциями

35. Чему будет равно возвращаемое значение функции , если не возникло сбоев при выполнении функции write(handle,buf,4\*sizeof(int))

а) 4

б) 8

в) 16

36. Для чего используется директива typedef

а) определяет тип данных

б) определяет новый тип данных

в) связывает тип данных с некоторым именем

37. Что составляет 16 байт

а) слово

б) длинное слово

в) параграф

38. В чем заключается принцип работы стека

а) первый зашел, последний вышел

б) первый зашел, первый вышел

в) последний зашел, первый вышел

39. Какая побитовая операция означает исключающее ИЛИ

а) &

б) |

в) ^

40. Какой массив используется для организации связанного списка

а) одномерный

б) двумерный

в) трехмерный

**Библиографический список**

1.Трофимов В.В. Алгоритмитизация и програмирование. Учебник для академического бакалавриата

2. Киммел П. и др.Borland С++5, пер. с англ., - СПб: BHV-Санкт Петербург 2000г-976 с.

3. Дейтел Х. Как программировать на С++. –М.: Бином, 2000.-1024с.

4. Кегрниган Б., Ритчи Д. Язык программирования С/ Пер. с англ. – СПб.: Невский Диалект, 2000.-352 с.

*Учебное издание*

**Программирование и основы алгоритмизации**

Программа , контрольные задания и методические указания по курсу

2-е издание, переработанное

Составители:

Предместьин В.Р.

Мелешкина В.А.

Издано на средства авторов

Компьютерная верстка:\_\_\_

Редактор: \_\_\_\_

Подписано в печать \_\_\_\_\_\_ Формат 60х84 1/16.

Бумага «SvetoCopy». Отпечатано на ризографе.

Усл.печ.л. \_\_ Уч.-изд.л. \_\_

Тираж \_\_ экз. Заказ №

ФГБОУ ВО Российский химико-технологический университет им.Д.И.Менделеева

Новомосковский институт (филиал). Издательский центр

Адрес университета:125047, Москва, Миусская площадь пл.,9

Адрес института:301665 Новомосковск, Тульской обл., ул. Дружбы, 8