Некоммерческое акционерное общество

АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Кафедра инженерной кибернетики

ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

Методические указания по выполнению расчетно-графических работ для студентов специальности 5В070200

Алматы 2014

СОСТАВИТЕЛИ: Н.В. Сябина, Л.Н.Рудакова. Технологии программирования. Методические указания по выполнению расчетно-графических работ для студентов специальности 5В070200 - Автоматизация и управление. – Алматы: АУЭС, 2014. – 23 с.

Для контроля знаний, полученных студентами в результате самостоятельной работы, в курсе «Технологии программирования» предусмотрены три расчетно-графические работы. Настоящие методические указания включают краткие теоретические сведения, варианты заданий, рекомендации к их выполнению и контрольные вопросы по темам:

1) Использование функций при работе с массивами.

2) Использование файлов и структур.

3) Использование графики.

В приложениях содержится весь необходимый справочный материал.

Методические указания предназначены для студентов всех форм обучения специальности 5В070200 - Автоматизация и управление.

Ил. 1, табл. 17, библиогр. – 8 назв.

Рецензент: канд. техн. наук, старший преподаватель Г.Д. Мусапирова

Печатается по плану издания некоммерческого акционерного общества «Алматинский университет энергетики и связи» на 2014 г.

1 Расчетно-графическая работа. Использование функций при работе с массивами

Цель: научиться описывать структурные алгоритмы с помощью различных нотаций и получить практические навыки использования функций при работе с массивами.

1.1 Задания к расчетно-графической работе

1.1.1 Выбрать вариант задания (таблицы 1.1 и 1.2). Разработать алгоритмы решения задач. Выполнить их графическое описание с помощью блок-схемы и одной из нотаций:

- для нечетных вариантов с помощью Flow-формы;
- для четных вариантов с помощью диаграммы Насси-Шнайдермана.

1.1.2 Выполнить программную реализацию задач из таблиц 1.1 и 1.2 с использованием функций. При решении задачи из таблицы 1.2 зашифровать и вывести на экран заданную пользователем фразу. Реализовать ключ шифровки, предложенный в варианте. Шифр защитить паролем, позволяющим получить исходный текст. При вводе неверного пароля выдать соответствующее сообщение.

1.1.3 Оформить отчет в соответствии с требованиями стандарта АУЭС.

Таблица 1.1 – Варианты заданий к задаче 1

№ вар	Задание
1	В массиве *K(n)* в порядке убывания представлены достоинства денежных знаков (купюр и монет) валютной системы некоторой страны. Реализовать выдачу в этой системе заданной суммы m минимальным числом денежных знаков.
2	Число делится на 11, если разность между суммой цифр, стоящих на нечетных местах, и суммой цифр, стоящих на четных местах, кратна 11. Проверить этот признак для всех натуральных чисел, не превосходящих заданного *m*, и вывести числа, кратные 11.
3	Дана целочисленная матрица m*n. Переставить строки матрицы по возрастанию максимальных элементов каждой строки.
4	Найти все натуральные числа, не превосходящие заданного n, десятичная запись которых есть строго возрастающая последовательность цифр.
5	Любая целочисленная денежная сумма s >7 руб. может быть выдана без сдачи «трешками» и «пятерками». Найти для заданной суммы s необходимое количество «трешек» и «пятерок».
6	Сформировать матрицу заданной пользователем размерности (не менее 10x10), которая поделена диагоналями на 4 треугольника, элементы левого треугольника равны 1, верхнего - 2, правого – 3, нижнего – 4.

7	Каждое из натуральных чисел заменить числом, получающимся при записи его десятичных цифр в обратном порядке.
8	Заданное натуральное число n, не превосходящее 1000, записать прописью, т.е. вывести соответствующее количественное числительное, например: 375 – «триста семьдесят пять».
9	Найти все натуральные числа, не превосходящие заданного n, десятичная запись которых есть строго возрастающая последовательность цифр.
10	Натуральное число называется *совершенным*, если оно равно сумме всех своих простых делителей (например, 6=1+2+3). Найти все совершенные числа, не превосходящие заданного n.
11	Имеется набор гирь весом 1, 2, 5, 10, 20, 50, 100… грамм. Как взвесить тело заданной массы m грамм на равноплечих весах, используя минимальное число гирь?
12	Имеется по одной гире весом 1, 2, 5, 10, 20, 50, 100… грамм. Разрешается класть гири на любую чашку весов. Как уравновесить тело заданной массы m грамм, используя минимальное число гирь? Например, масса 27 грамм уравновешивается: 27+1+2=10+20 либо 27+1+2+20=50.
13	Утверждается, что разность любого натурального числа и суммы его цифр кратна 9. Проверить этот факт для всех чисел, лежащих между заданными m и n.
14	Найти 20 первых троек *пифагоровых* чисел, то есть целых k, l, m таких, что $k^2+l^2=m^2$.
15	В массиве *K(n)* в порядке возрастания представлены достоинства денежных знаков (купюр и монет) валютной системы некоторой страны. Реализовать выдачу в этой системе заданной суммы m максимальным числом денежных знаков.
16	Найти все натуральные числа, не превосходящие заданного n, десятичная запись которых есть строго убывающая последовательность цифр.
17	Дана целочисленная матрица m*n. Переставить строки матрицы по возрастанию сумм элементов каждой строки. Если суммы равны, то приоритет имеет строка, в которой первый элемент больше.
18	Дана целочисленная матрица m*n. Переставить столбцы матрицы по убыванию максимальных элементов каждого столбца.
19	Сформировать массив заданной пользователем размерности (не менее 10x10), в котором все нулевые элементы размещены попарно в шахматном порядке (сначала 2 нулевых).
20	Транспонированием квадратной матрицы называется такое ее преобразование, при котором строки и столбцы меняются ролями: i-й столбец становится i-й строкой. Для матрицы заданной пользователем размерности получить транспонированную матрицу.

21	Шахматную доску будем представлять символьной матрицей размера 8*8. Даны натуральные p и q (1<=p<=8,1<=q<=8), местоположение ферзя (Ф) определяет пользователь, задавая его координаты. Выявить поля, находящиеся под угрозой ферзя и отметить их *, а остальные поля - символом 0. Вывести полученный массив на экран.
22	Для заданного *m* получить таблицу первых *m* простых чисел.
23	Перевести заданное целое число в систему римского счета.
24	Шахматную доску будем представлять символьной матрицей размера 8*8. Даны натуральные p и q (1<=p<=8,1<=q<=8), местоположение коня (К) определяет пользователь, задавая его координаты. Выявить поля, находящиеся под угрозой коня и отметить их *, а остальные поля - символом 0. Вывести полученный массив на экран.
25	Найти все натуральные числа от 1 до 1000, которые совпадают с последними разрядами своих квадратов, например: 25^2=625; 76^2=5676.

Т а б л и ц а 1.2 – Варианты заданий к задаче 2

№ вар	Принципы шифровки
1	Заменить все гласные буквы на согласные, а согласные буквы соответствующим ASCII-кодом.
2	Заменить все буквы символикой игральных карт, например, А - 6♦, В - 6♥ и т.д.
3	Заменить все буквы числом, соответствующим их позиции в кириллице.
4	Заменить все буквы символикой нотной грамоты, учитывая, что имеется шесть полных октав по 7 нот. Например, А – до$_1$, Б – ре$_1$ и т.д.
5	Каждой букве в тексте поставить в соответствие число, равное 2 в степени её позиции в исходной строке.
6	Заменить все согласные буквы гласными, а гласные буквы соответствующим ASCII-кодом.
7	Заменить все буквы по следующему принципу: A на Z, B на Y, C на X и т.д.
8	Заменить все согласные буквы на гласные, а гласные буквы соответствующим ASCII-кодом.
9	Используя шифр перестановки, закодировать фразу следующим образом: исходный текст записать в прямом порядке построчно в матрицу, а вывести в вектор по столбцам.
10	Закодировать фразу следующим образом: буквы, стоящие в строке в нечетных позициях заменить буквами в четных позициях и переписать полученный текст в обратном порядке.
11	Используя шифр перестановки, закодировать фразу следующим образом: исходный текст записать построчно в матрицу, а вывести текст в вектор.

12	Заменить все буквы символами азбуки Морзе.
13	Заменить все буквы соответствующим ASCII-кодом.
14	Закодировать фразу, используя символику псевдографики.
15	Заменить все буквы римскими цифрами разного цвета.
16	Заменить все согласные буквы римскими цифрами, а гласные – арабскими.
17	Используя шифр перестановки, закодировать фразу следующим образом: исходный текст записать по столбцам в матрицу, а вывести в вектор построчно.
18	Заменить все буквы различными комбинациями математических символов.
19	Каждой букве в тексте поставить в соответствие число, равное квадрату её позиции в латинице.
20	Заменить все гласные буквы римскими цифрами, а согласные – арабскими.
21	Используя шифр перестановки, закодировать фразу следующим образом: исходный текст записать в обратном порядке построчно в матрицу, а вывести в вектор по столбцам.
22	Заменить все буквы различными знаками препинания и цифрами.
23	Заменить все буквы «смайликами» разного цвета.
24	Каждой букве в тексте поставить в соответствие число, равное её позиции в латинице.
25	Каждой букве в тексте поставить в соответствие число, равное её позиции в исходной строке, умноженной на количество ее повторов.

1.2 Общие рекомендации к выполнению работы

При выполнении задач из таблицы 1.1 необходимо учитывать, что римские цифры обозначаются следующими латинскими буквами:

1	I	500	D
5	V	1000	M
10	X	5000	$\overline{V}$
50	L	10000	$\overline{X}$
100	C	…	…

При выполнении задач из таблицы 1.2 следует учитывать, что в шифре перестановки буквы открытого текста не замещаются на другие, а меняется сам порядок их следования. Например, в шифре простой колонной перестановки исходный открытый текст записывается построчно (число букв в строке фиксировано), а шифртекст получается считыванием букв по колонкам. Расшифровка производится аналогично: шифртекст записывается по вертикали, а открытый текст можно затем прочесть по горизонтали.

Необходимые справочные материалы приведены в приложениях А и Б.

1.3 Контрольные вопросы

1.3.1 Какой максимальной размерности может быть многомерный массив?

1.3.2 Каковы особенности работы с квадратными матрицами?

1.3.3 Как организовать обработку матрицы произвольной размерности по столбцам (по строкам)?

1.3.4 Как организовать передачу значений элементов массива с помощью функции?

1.3.5 В чем заключаются особенности работы с символьными массивами?

1.3.6 Какие стандартные функции используются для работы со строками?

1.3.7 Сколько значений может передать функция?

1.3.8 В чем разница между формальными и фактическими параметрами?

1.3.9 С какой целью используются прототипы функций в программах?

1.3.10 Что представляет собой перегрузка функций?

2 Расчетно-графическая работа. Использование файлов и структур

Цель: получить практические навыки использования файлов и структур при решении задач.

2.1 Задания к расчетно-графической работе

2.1.1 Выбрать в соответствии с таблицей 2.1 вариант задания и реализовать структуру.

2.1.2 Организовать ввод и вывод данных структуры (не менее 10 записей), используя файлы.

2.1.3 Используя поля созданной структуры, выполнить выборку или по возможности вычисления.

2.1.4 Оформить отчет в соответствии с требованиями стандарта АУЭС.

Таблица 2.1 – Варианты заданий

№ вар.	Тема	Поля структуры
1	Кулинария	Наименование продукции; Дата изготовления; Срок хранения; Вес; Цена.
2	Выставка собак	Порода; Пол; Кличка; Возраст; Владелец.
3	Компьютеры	Модель; Производитель; Комплектация; Стоимость; Количество.
4	Аренда автомобилей	Марка; Год выпуска; Пробег; Стоимость; Срок аренды.
5	Библиотека	Автор книги; Название; Издательство; Год выпуска; Количество.

6	Переводы	Язык перевода; Стоимость за стр; Объем заказа; Срок выполнения; Исполнитель.
7	Каталог сотовых телефонов	Наименование; Модель; Производитель; Стоимость; Особенности.
8	Коллекция пластинок	Название альбома; Исполнитель; Год выпуска; Кол-во записей; Стоимость.
9	Самолеты	Наименование; ФИО конструктора; Год выпуска; Количество; Грузоподъемность.
10	Компьютерные игры	Название; Разработчик; Версия; Жанр; Стоимость.
11	Стипендия	ФИО студента; Курс; Средний балл; Размер стипендии; Надбавки.
12	Расписание автобусов	Номер маршрута; Пункт отправления; Время отправления; Пункт назначения; Время прибытия.
13	Сбербанк	Номер счета; ФИО вкладчика; Наименование депозита; Сумма вклада; Процентная ставка; Срок.
14	Репертуар кинотеатров	Кинотеатр; Фильм; Жанр; Период показа; Время.
15	Прайс-лист	Вид услуги; Объем работ; Стоимость; Срок исполнения; Скидки.
16	Недвижимость	Наименование; Площадь; Этажность; Количество комнат; Стоимость.
17	Аукцион	Дата проведения; Наименование лота; Первоначальная стоимость; ФИО покупателя; Сумма приобретения.
18	Туристическое агентство	Страна; Предлагаемые маршруты; Условия проживания и поезда; Экскурсионное обслуживание; Стоимость путевки.
19	Автосалон	Марка; Год выпуска; Технические характеристики; Особенности; Техническое состояние; Цена.
20	Спортсмены	ФИО; Гражданство; Происхождение; Вид спорт; Клуб (команда); Личные достижения
21	Заселение в гостиницу	ФИО; Паспортные данные; Дата приезда; Номер; Дата выезда; Стоимость проживания; Стоимость дополнительных услуг.
22	Вакансии	Фирма; Должность; Условия труда; Оплата; Требования к кандидату; Контактный телефон.
23	Типография	Дата заказа; Наименование заказа; Объем; Цена за единицу; Срок исполнения; Сумма.

24	Ремонт	Наименование работ; Используемый материал; Объем работ; Цена за м^2; Сумма.
25	Музеи	Наименование коллекции; Оригинал или копия; Владелец; Дата приобретения.

2.2 Общие рекомендации к выполнению работы

Структура – это группа связанных элементов. Прежде чем создать объект структуры, должен быть определен ее формат, что делается посредством объявления структуры. Переменные, составляющие структуру, называются ее элементами (полями).

Имя структуры – это спецификатор типа. Ключевое слово struct указывает на начало объявления структуры.

К отдельным элементам структуры доступ осуществляется с помощью оператора «точка». Чтобы обратиться к элементу структуры, нужно перед его именем поставить имя структурной переменной и оператор «точка».

Общий формат доступа:

имя_структурной_переменной . имя_элемента

Структуры полезны, когда требуется объединить несколько переменных с разными типами под одним именем. Это делает программу более компактной и более гибкой для внесения изменений. Также структуры незаменимы, когда необходимо сгруппировать некоторые данные, например, запись из базы данных или контакт из книги адресов.

В расчетно-графической работе используются структуры, считываемые из файлов. Чтение и запись данных в файл может осуществляться, например, с помощью стандартных функций write (запись) и read (чтение) библиотеки fstream.h. В приложении В приводятся примеры функции write и read, в которых в качестве одного из параметров используют оператор приведения типов (char *), представляющий собой указатель на символьную строку.

2.3 Контрольные вопросы

2.3.1 Чем структура отличается от объединения?
2.3.2 Как размещаются в памяти элементы структуры?
2.3.3 С какой целью используются структуры?
2.3.4 Как определить размер структуры?
2.3.5 Как обрабатываются элементы структуры?
2.3.6 Что представляет собой структурный тэг?
2.3.7 Что представляет собой файл?
2.3.8 Какие существуют виды файлов?
2.3.9 Как осуществляется обработка файлов?
2.3.10 Какие основные функции используются при работе с файлами?

3 Расчетно-графическая работа. Использование графики

Цель: получение навыков работы в графическом режиме и применения встроенных графических процедур и функций.

3.1 Задания к расчетно-графической работе

3.1.1 Выбрать в соответствии с таблицей 3.1 вариант задания.

3.1.2 Используя известные алгоритмические конструкции и графические возможности языка С++, реализовать экранную заставку с предложенным видеоэффектом. Видеоэффект разрешается дополнить (видоизменить), согласовав внесенные изменения с преподавателем.

3.1.3 Оформить отчет в соответствии с требованиями стандарта АУЭС.

Таблица 3.1 – Варианты заданий

№	Варианты заданий
1	«Водный мир» - имитация движения объектов в глубинах океана.
2	«Спираль» - кривая, которая увеличивается из центра экрана, изменяя цвет.
3	«Падающая звезда» - по звездному небу перемещается, «падая» звезда.
4	«Вселенная» - звезды мерцают на небе, создавая эффект перемещения.
5	«Метеорит» - по звездному небу летит метеорит.
6	«Бегущая строка» - задаваемый текст, перемещающийся по случайной траектории на экране.
7	«Вечный двигатель» - кривая, которая формируется на экране и изменяет свою форму с течением времени.
8	«Теннисный мячик» — шарик, который летает по экрану и отражается от верхней и нижней границ экрана, изменяя цвет.
9	Экран постепенно заполняется случайным образом буквами, при нажатии клавиши буквы изменяют цвет.
10	«Летающие кубы» - 3 куба разного цвета, которые летают по случайной траектории на экране.
11	«Вращение» - задаваемый текст вращается вокруг своей оси, изменяя цвет.
12	«Жук-пожиратель» — фигурка, которая перемещается по экрану по случайной траектории и «съедает» буквы.
13	«Удав» — то же, что и «жук», но к тому же он увеличивается в размерах по мере «поедания» букв.
14	«Соты» - экран условно делится на квадраты, которые последовательно заполняются шестиугольниками.
15	«Удав» — фигурка, которая перемещается по экрану по случайной траектории и «съедает» только гласные и увеличивается в размерах по мере «поедания» букв.

16	«Звездолет» - объект перемещается по экрану, имитируя эффект полета.
17	«Вселенная-1» - планеты солнечной системы вращаются вокруг Солнца.
18	«Метаморфозы» - шар, перемещаясь по экрану, изменяет свою форму.
19	«Дискотека» - разноцветные лучи, зафиксированные с одного конца, высвечивают эллипсы на экране, которые меняют свое положение
20	«Вечный двигатель-1» - окружность, которая формируется на экране и изменяет свою форму с течением времени.
21	«Геометрический вальс» - окружность, которая формируется на экране и перемещается по случайной траектории.
22	«Исчезновение» - задаваемый текст, при нажатии клавиши начинает исчезать с экрана. Размер текста должен быть подобран так, чтобы он занимал по возможности весь экран.
23	«Летающие шары» - 4 шара разного цвета, которые летают по экрану.
24	Экран случайным образом постепенно заполняется кругами разного цвета до нажатия клавиши.
25	«Трубопровод» - цилиндр вытягивается и, изгибаясь, изменяет свой цвет.

3.2 Общие рекомендации к выполнению работы

В отличие от математической системы координат, графический экран выглядит так, как показано на рисунке 3.1:

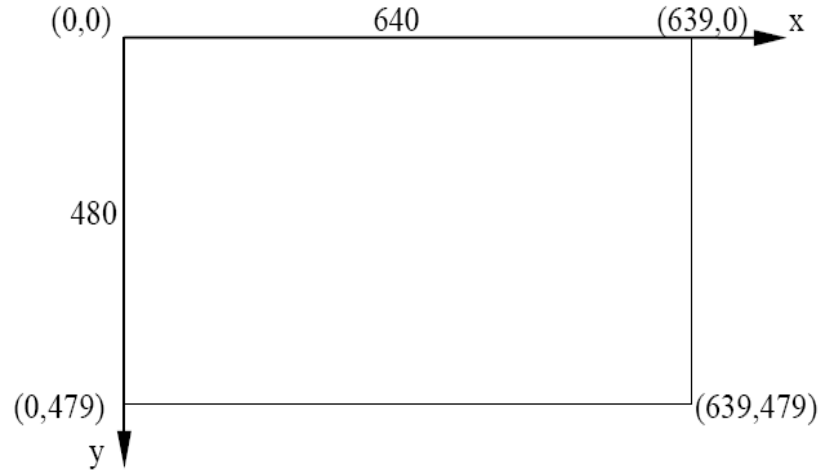

Рисунок 3.1 – Внешний вид графического экрана

Наиболее часто используется графический режим монитора, при котором поддерживается разрешение 640*480*16. Здесь 16 – это максимальное количество цветов, которое одновременно может присутствовать на изображении. В заголовочном файле graphics.h определены константы, соответствующие цветам стандартной палитры (таблица Г.1).

Изменение одного из цветов стандартной палитры производится функцией

void setpalette (int index, int color);

где intindex – номер из стандартной палитры, int color – цвет в диапазоне от 0 до 63 (палитра EGA).

Настройка палитры EGA осуществляется функцией

void setrgbpalette (int color, int red, int green, int blue);

где red, green и blue изменяются в диапазоне от 0 до 255, причем малым значениям соответствуют темные цвета, большим – более яркие. Если они имеют одинаковые значения, то формируется один из оттенков серого цвета.

Графический экран представляет собой массив пикселей. Каждый пиксель соответствует одной точке на экране и может иметь свой цвет. Функции определения цвета приведены в таблице Г.2.

Группа линий на плоскости образует контурную фигуру (отрезок прямой линии, дугу, окружность, прямоугольник, эллипс и т.д.). Кроме формы, фигуры могут отличаться цветом линии (контура), ее толщиной или типом. По умолчанию в графическом режиме существуют следующие настройки: текущий цвет контура – WHITE (белый), толщина – один пиксель, тип – сплошная линия. Прототипы функций изменения параметров контура фигур приведены в таблице Г.2.

Для отображения наиболее часто используемых фигур можно воспользоваться функциями стандартной графической библиотеки (таблица Г.2). Типы линий контура приведены в таблице Г.3.

Плоскостные фигуры - это фрагменты плоскости экрана, ограниченные замкнутым контуром. Их можно получить из контурных путем закрашивания области внутри или вне замкнутой сплошной линии, образующей контур. Прототипы функций, позволяющих получить некоторые плоскостные фигуры, приведены в таблице Г.2.

Вывод текста в графическом режиме можно осуществить с использованием прототипов функций из таблицы Г.4. Текстовая информация отображается на экране с учетом параметров: цвет, тип шрифта, размер шрифта и направление.

Размер символов (по вертикали и горизонтали) определяется как произведение стандартного размера (8*8 пикселей) на параметр charsize, то есть если значение charsize будет равно 3, то каждый символ, отображающийся на экране, будет вписан в квадрат 24*24 пикселя.

Параметр font, задающий стиль шрифта, подключает к программе файлы с расширением *.chr (нестандартные «шрифты»), поэтому необходимо сделать эти файлы доступными (проще всего скопировать их в текущую директорию). Все необходимые для работы со шрифтами функции приведены в таблицах Г.5 и Г.6.

Для вывода текста на экран в графическом режиме можно использовать и функции для текстового режима (например, printf()), однако они имеют ряд недостатков. Например, при использовании функции printf() для вывода

текста на каком-либо цветном фоне позади надписи появится ее «фон» (черный прямоугольник, равный длине выводимого текста). Также отсутствует возможность изменения внешнего вида выводимого текста (размера шрифта, стиля и т.д.).

3.3 Контрольные вопросы

3.3.1 В чем заключаются особенности использования графического и текстового режимов монитора?

3.3.2 Как выполняется инициализация графики?

3.3.3 На что влияет тип видеоадаптера?

3.3.4 В чем преимущество графического режима при работе с текстом?

3.3.5 С какой целью используется палитра?

3.3.6 Какие характеристики нужно определить для рисования фигур?

3.3.7 В чем отличие между контурными и плоскостными фигурами?

3.3.8 Как получить дополнительные шрифты?

3.3.9 Как регулируется размер символов?

3.3.10 Какие функции используются при работе с окнами в графическом режиме?

Приложение А

Описание структурных алгоритмов

Таблица А.1 – Соответствие различных способов описания алгоритмов

Структура	Псевдокоды	Flow-формы	Диаграммы Насси-Шнейдермана
Следование	<действие 1> <действие 2>	<действие 1> <действие 2> <действие 3>	<действие 1> <действие 2> <действие 3>
Ветвление	**Если** <условие> **то** <действие 1> **иначе** <действие 2> **Все-если**	**Если** <Условие> **то** <действие 1> **иначе** <действие 2>	<Условие> да нет <действие 1> <действие 2>
Цикл-пока	**Цикл-пока** <условие> <действие> **Все-цикл**	**Пока** <Условие> <действие>	**Пока** <Условие> <действие>
Выбор	**Выбор** <код> <код 1>: <действие 1> <код 2>: <действие 2> **иначе** <действие 3> **Все-выбор**	**Выбор** <Код> код 1 <действие 1> код 2 <действие 2> код 3 <действие 3>	<Код> 1 2 **иначе** <действ.1> <действ.2> <действ.3>
Цикл с парамет-ром	**Для** <индекс> = <n>,<m>,<h> <действие > **Все-цикл**	**Для** i=n, m, h <действие>	**Для** i=n, m, h <действие>
Цикл-до	**Выполнять** <действие> **До** <условие>	<действие> **До** <Условие>	<действие> **До** <Условие>

Приложение Б

Основные элементы языка С++

Таблица Б.1 - Специальные и управляющие символы

	Вид	Название	Вид	Название
Специальные символы	+	Плюс	|	Черта
	++	Приращение	||	Логическое ИЛИ
	-	Минус	!	Восклицательный знак
	--	Уменьшение	->	Стрелка
	*	Звездочка	=	Операция присваивания
	/	Наклонная черта	= =	Знак равенства
	\	Обратный слеш	!=	Не равно
	//	Двойной слеш	>	Больше
	.	Точка	<	Меньше
	,	Запятая	<=	Меньше или равно
	:	Двоеточие	<<	Сдвиг влево
	::	Разрешение	>>	Сдвиг вправо
	;	Точка с запятой	< >	Угловые скобки
	‘	Апостроф	( )	Круглые скобки
	“	Кавычки	[ ]	Квадратные скобки
	^	«Крышка»	{ }	Фигурные скобки
	%	Знак процента	/* */	Скобки комментария
	&	Амперсанд	#	Знак номера
	&&	Логическое И	~	Тильда
Управляющие символы	\a	Сигнал динамика	\t	Горизонтальная табуляция
	\b	BS, забой символа	\v	Вертикальная табуляция
	\f	Новая страница	\\	Обратная косая черта
	\n	Новая строка	\0	Нулевой символ (байт)
	\r	Возврат каретки	\000	Восьмеричная константа
	\”	Двойная кавычка	\xhhh	Шестнадцатеричная константа
	\’	Апостроф	\?	Знак вопроса

Таблица Б.2 - Зарезервированные слова в С++

and	char	false	int	private	switch	virtual
and_eq	class	float	long	protected	template	void
asm	compl	for	mutable	public	this	volatile
auto	const	else	namespace	register	throw	while
bitand	continue	enum	new	return	true	xor
bitor	default	explicit	not	short	try	xor_eq
bool	delete	friend	not_eq	signed	typedef	
break	do	goto	operator	sizeof	typename	
case	double	if	or	static	union	
catch	extern	inline	or_eq	struct	unsigned	

Таблица Б.3 – Типы данных с разными комбинациями модификаторов

Тип	Диапазон изменения		Размер в байтах (битах)
	от	до	
void	-	-	0
char (signed char)	-128	127	1 (8)
unsigned char	0	255	1 (8)
wchar_t	0	65535	2 (16)
bool	True (Истина)	False (Ложь)	1 (8)
int (signed int, short int, signed short int)	-32768	32767	2 (16)
unsigned int (unsigned short int)	0	65535	2 (16)
long int (signed long int)	-2147483648	2147483647	4 (32)
unsigned long int	0	4294967295	4 (32)
float	3.4E-38	3.4E+38	4 (32)
double	1.7E-308	1.7E+308	8 (64)
long double	3.4E-4932	3.4E+4932	10 (80)

Примечание – Размер в байтах и диапазон изменения могут варьироваться в зависимости от компилятора, процессора и операционной системы (среды).

Таблица Б.4 – Перечень операций C++, их приоритет и порядок выполнения

Уровень	Оператор	Порядок	Уровень	Оператор	Порядок
1	( ) . [ ] -> ::	⇒	9	&	⇒
2	* & ! ~ ++ -- + - sizeof new delete	⇐	10	^	⇒
3	. * -> *	⇒	11	|	⇒
4	* / %	⇒	12	&&	⇒
5	+ -	⇒	13	||	⇒
6	<< >>	⇒	14	?:	⇐
7	< <= > >=	⇒	15	= *= /= += -= %= <<= >>= &= ^= |=	⇐
8	== !=	⇒	16	,	⇒

Примечания

1 Наивысший приоритет имеют операторы 1 уровня, низший – 16 уровня.

2 Знак ⇒ обозначает выполнение операций слева направо, а знак ⇐ - выполнение операций справа налево.

3 Унарные операторы (+) и (-), находящиеся на уровне 2, обладают более высоким приоритетом, чем арифметические (+) и (-) с уровня 5. Символ & на уровне 2 - оператор обращения по адресу, а символ & на уровне 9 битовый оператор AND. Символ * на уровне 2 - оператор обращения к указателю, а символ * на уровне 4 – оператор умножения.

4 В отсутствие круглых скобок операторы, находящиеся на одном уровне, обрабатываются согласно их расположению слева направо.

Таблица Б.5 – Основные математические функции

Наименование функции	Функция	Тип		Заголовочный файл
		результата	аргумента	
Абсолютное значение	abs(x) cabs(x) fabs(x)	int double float	int double float	<stdlib.h> <math.h> <math.h>
Арккосинус	acos(x)	double	double	<math.h>
Арксинус	asin(x)	double	double	<math.h>
Арктангенс	atan(x)	double	double	<math.h>
Косинус	cos(x)	double	double	<math.h>
Синус	sin(x)	double	double	<math.h>
Экспонента e^x	exp(x)	double	double	<math.h>
Степенная функция x^y	pow(x,y)	double	double	<math.h>
Логарифм натуральный	log(x)	double	double	<math.h>
Логарифм десятичный	log10(x)	double	double	<math.h>
Корень квадратный	sqrt(x)	double	double	<math.h>
Тангенс	tan(x)	double	double	<math.h>

Таблица Б.6 – Символы преобразования в функциях ввода-вывода

Формат вывода	Значение	Формат ввода	Значение
%c	вывод символа (char)	%c	чтение символа (char)
%d	целое десятичное число (int)	%d	целое десятичное число (int)
%i	целое десятичное число (int)	%i	целое десятичное число (int)
%e (%E)	число (float/double) в виде x.xx e+xx (x.xx E+xx)	%e	чтение числа типа float/double
%f (%F)	число float/double с фиксиро-ванной запятой xx.xxxx	%h	чтение числа типа short int
%g (%G)	число в виде f (F) или e(E) в зависимости от значения	%o	чтение восьмеричного числа
%s	строка символов	%s	чтение строки символов
%o	целое число (int) в восьмеричном представлении	%x	чтение шестнадцатеричного числа (int)
%u	беззнаковое десятичное число (unsigned int)	%p	чтение указателя
%x (%X)	целое число (int) в шестнад-цатеричном представлении	%n	чтение указателя в увеличенном формате
%p (%n)	указатель		

Примечание – К форматам можно применять модификаторы l и h, например, %ld (long в десятичном виде), %lo (long в восьмеричном виде), %lu (unsigned long), %lx (long в шестнадцатеричном виде), %lf (long float с фиксированной точкой), %le (long float в экспоненциальной форме), %lg (long float в виде f или e в зависимости от значения).

Приложение В

Примеры использования файлов при работе со структурами

Пример В.1. *Запись структуры в файл:*

```
#include<fstream.h>
void   main(void)
{
struct   date
     {
          int   day; char   month[9]; int   year;
     }d={25,"ноября",1991};
ofstream   out_file("file3.dat") ;
out_file.write((char    *) &d, sizeof(date));
out_file.close (); }
```

Пример В.2. *Чтение структуры из файла и вывод ее элементов на экран:*

```
#include<iostream.h>
#include<fstream.h>
void   main(void)
{
     struct   date
{
     int   day; char   month[9]; int   year;
}d;
ofstream   in_file("file3.dat");
in_file.read((char   *) &d, sizeof(date));
cout<<d.day<<"   "<<d.month<<"   "<<d.year<<endl;
in_file.close () ; }
```

В результате последовательного выполнения этих двух программ на экран дисплея будет выведено: *25 ноября 1991*.

Приложение Г

Справочник графических функций

Таблица Г.1 - Именованные константы цветов

Цвет	Const	Значение
Черный	BLACK	0
Синий	BLUE	1
Зеленый	GREEN	2
Бирюзовый	CYAN	3
Красный	RED	4
Сиреневый	MAGENTA	5
Коричневый	BROWN	6
Светло-серый	LIGHTGRAY	7
Серый	DARKGRAY	8
Голубой	LIGHTBLUE	9
Светло-зеленый	LIGHTGREEN	10
Светло-бирюзовый	LIGHTCYAN	11
Светло-красный	LIGHTRED	12
Светло-сиреневый	LIGHTMAGENTA	13
Желтый	YELLOW	14
Белый	WHITE	15

Таблица Г.2 – Прототипы графических функций для работы с объектами

Функция	Описание
void arc (int x, int y, int stangle, int endangle, int radius);	дуга с центром в (x, y) и радиусом radius. Параметры stangle и endangle задают круговые координаты начальной и конечной точки. Угол измеряется в градусах и отсчитывается против часовой стрелки, где 0 градусов соответствует трем часам на циферблате. Если stangle равен 0, а endangle равен 360, функция arc() рисует полную окружность.
void bar (int left, int top, int right, int bottom);	рисует закрашенный прямоугольник. Прямоугольник закрашивается текущим цветом и с использованием текущего шаблона заполнения. Верхний левый и нижний правый углы прямоугольника заданы параметрами (left, top) и (right, bottom) соответственно.
void drawpoly (int numpoints, int *polypoints)	рисует многоугольник.
void circle (int x, int y, int radius);	окружность с центром в точке (x, y) и радиусом radius (единица измерения – пиксель).

void bar3d (int left, int top, int right, int bottom, int depth);	рисует “трёхмерный прямоугольник” (подобие параллелепипеда), начиная от верхнего левого угла до правого нижнего угла и с указанием глубины depth.
void ellipse (int x, int y, int stangle, int endangle, int xradius, int yradius);	рисует эллипс с координатами в точке (x, y), начальным и конечным углами stangle, endangle, и радиусами xradius, yradius по осям x и y.
void fillellipse (int x, int y, int rx, int ry);	рисует эллипс с центром в точке (x, y), горизонтальной и вертикальной осями rx и ry соответственно, и закрашивает его текущим цветом, используя текущий шаблон.
void fillepoly (int numpoints, int *polipoints);	рисует контур многоугольника, имеющий numpoints точек, а затем закрашивает его. Polipoints – указатель на последовательность из (numpoints*2) целых чисел. Каждая пара чисел (x, y) является координатами вершины многоугольника.
void line (int x1, int y1, int x2, int y2);	чертит на экране прямую линию от точки с координатами (x1, y1) до точки с координатами (x2, y2).
void pieslice (int x, int y, int stangle, int endangle, int radius);	рисует и закрашивает сектор круга с центром в точке (x, y) и радиусом radius. Сектор рисуется от угла stangle до угла endangle. Угол измеряется в градусах и отсчитывается против часовой стрелки, где 0 градусов соответствует трем часам на циферблате.
void rectangle (int left, int top, int right, int bottom);	чертит прямоугольник, расположенный на экране горизонтально (вертикально), координатой левого верхнего угла (left, top) и правого нижнего – (right, bottom).
void floodfill (int x, int y, int border);	область, ограниченная цветной границей, закрашивается установленным образцом и цветом.
void getbkcolor();	возвращает значение типа int текущего цвета фона.
unsigned getpixel (int x, int y);	обратная функция, которая определяет цвет точки с координатами (x, y).
void getcolor();	возвращает значение типа int установленного на данный момент цвета.
void cleardevice();	очищает экран в графическом режиме и возвращает текущую позицию указателя в точку с координатами (0, 0).

void getimage (int left, int top, int right, int bottom, void far *bitmap);	сохраняет картинку в специально отведённой области памяти, где left, top, right, bottom координаты картинки, *bitmap - указатель на область памяти.
void getpixel (int x, int y);	возвращает номер цвета точки, находящейся по координатам (x, y).
void clearviewport();	стирает область просмотра и перемещает текущую позицию в точку с координатами (0, 0) относительно области просмотра.
void linerel (int dx, int dy);	рисует линию от текущей позиции до точки, находящейся на относительном расстоянии от текущей позиции, затем передвигает текущую позицию.
void lineto (int x, int y);	рисует линию от текущей позиции до точки с координатами (x, y), затем переносит текущую позицию в (x, y).
void putpixel (int x, int y, int color);	рисует точку с координатами (x, y) и цветом color.
void sector (int x, int y, int stangle, int endangle, int xradius, int yradius);	рисует сектор круга установленным цветом, затем заполняет его, используя образец и цвет, установленные функциями setfillstyle или setfillpattern.
void setbkcolor (int color);	устанавливает цвет фона по его номеру, заданному переменной color.
void setcolor (int color);	устанавливает цвет по его номеру, заданному переменной color.
void setfillstyle (int pattern, int color);	устанавливает образец и цвет заливки.
void setlinestyle (int linestyle, unsigned upattern, int thickness);	устанавливает стиль рисуемых линий для функций рисования line, lineto, rectangle, drawpoly.

Таблица Г.3 – Типы линий

Название	Описание	Значение
SOLID_LINE	сплошная	0
DOTTED_LINE	пунктирная	1
CENTER_LINE	штрих-пунктирная	2
DASHED_LINE	штриховая	3
USERBIT_LINE	Тип, определяемый пользователем (параметр upattern)	4

Таблица Г.4 – Прототипы графических функций для работы с текстом

Функция	Описание
void moverel (int dx, int dy);	перемещает текущую позицию на относительное расстояние.
void moveto (int x, int y);	перемещает текущую позицию в точку с координатами (x, y).
void outtext (char far *textstring);	выводит строку в графическом режиме. Строка должна быть заключена в кавычки.
void outtextxy (int x, int y, char far *textstring);	выводит строку в графическом режиме, предварительно перейдя в точку с координатами (x, y).
void settextjustify (int horiz, int vert);	устанавливает выравнивание текста для графического режима. По умолчанию стоит выравнивание по верхнему левому углу.
void settextstyle (int font, int direction, int charsize);	устанавливает характеристики текста: стиль, положение (горизонтальное или вертикальное), размер шрифта.
void setusercharsize (int multx, int divx, int multy, int divy);	устанавливает определённый пользователем «коэффициент сжатия» шрифта. По умолчанию ширина шрифта определена как multx:divx, высота как multy:divy.
void textheight (char far *textstring);	возвращает высоту строки в количестве точек.
void textwidth (char far *textstring);	возвращает ширину строки в количестве точек.

Таблица Г.5 - Типы шрифтов

Шрифт	Константа	Значение
Точечный шрифт 8x8 пикселей	DefaultFont	0
Утроенный шрифт TRIP.CHR	TriplexFont	1
Уменьшенный шрифт LITT.CHR	SmallFont	2
Прямой шрифт SANS.CHR	SansSerifFont	3
Готический шрифт GOTH.CHR	GothicFont	4

Таблица Г.6 - Особые шрифты

Шрифт	Значение	Файл
«Рукописный» шрифт	5	scri.chr
Одноштриховый шрифт типа Courier	6	simp.chr
Красивый наклонный шрифт типа Times Italic	7	tscr.chr
Шрифт типа Times Roman	8	Icom.chr
Шрифт типа Courier увеличенного размера	9	euro.chr
Крупный двухштриховый шрифт	10	bold.chr

Список литературы

1 Страуструп Б. Язык программирования C++. – М., 2012.

2 Потопахин В. Искусство алгоритмизации. - М.: «ДМК Пресс», 2011.

3 Сэджвик Р. Алгоритмы на C++. – М., «Вильямс», 2011.

4 Давыдов В.Г. Технологии программирования C++. – СПб., 2005.

5 Павловская Т.А. C/C++. Структурное программирование. – СПб., 2010.

6 Немцова Т.И. Программирование на языке высокого уровня. Программирование на языке C++. М.: «Форум», 2012.

7 Ашарина И.В. Основы программирования на языках C и C++. - М., Горячая линия-Телеком, 2012.

8 Аляев Ю.А., Козлов О.А. Алгоритмизация и языки программирования Pascal, C++, VisualBasic: Учебно - справочное пособие. – М.: Финансы и статистика, 2004.

Содержание

Сводный план 2014 г., поз. 116

Наталья Валерьевна Сябина
Лариса Николаевна Рудакова

ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

Методические указания по выполнению расчетно-графических работ
для студентов специальности 5В070200

Редактор Л.Т. Сластихина
Специалист по стандартизации Н.К. Молдабекова

Подписано в печать __. __. __.
Тираж *100* экз.
Объем *1.5* уч.-изд. л.

Формат 60х84 1/16
Бумага типографская №1
Заказ _____. Цена 750 тг.

Копировально-множительное бюро
некоммерческого акционерного общества
«Алматинский университет энергетики и связи»
050013, Алматы, ул. Байтурсынова, 126