

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Курганский государственный университет»

Кафедра программного обеспечения автоматизированных систем

ОСНОВЫ ПРОГРАММИРОВАНИЯ

Методические указания
к выполнению лабораторных и практических работ
для студентов направления подготовки 09.03.04
«Программная инженерия»,
10.05.03 «Информационная безопасность автоматизированных систем»
Часть 1

Курган 2016

Кафедра: «Программное обеспечение автоматизированных систем»

Направление подготовки: 09.03.04 «Программная инженерия»,
10.05.03. «Информационная безопасность автоматизированных систем».

Дисциплина: «Основы программирования».

Составил: канд. техн. наук, доцент А.М. Семахин.

Утверждены на заседании кафедры «30» сентября 2015 г.

Рекомендованы методическим советом университета «19» декабря 2014 г.

ВВЕДЕНИЕ

Дисциплина «Основы программирования» имеет целью дать студентам теоретические знания и практические навыки в разработке приложений на языке C++.

Предмет дисциплины – технология программирования на языке C++.

Задачи дисциплины – дать представление о методах программирования, основных этапах разработки структурных, объектно-ориентированных, визуальных приложений и применении языка программирования C++ в решении прикладных задач.

Методические указания состоят из двух частей. В части 1 рассматриваются основы теории программирования на языке C++: базовые конструкции языка, структурная и объектно-ориентированная технологии программирования. Методические указания содержат теоретическое обоснование и варианты заданий для выполнения лабораторных и практических работ.

Лабораторный практикум (34 часов).

Практические работы (16 часов).

Методические указания разработаны в соответствии с требованиями государственного образовательного стандарта по подготовке бакалавров по направлениям 09.03.04 «Программная инженерия» и 10.05.03. «Информационная безопасность автоматизированных систем».

1 Базовые конструкции языка C++

1.1 Типы данных

1.1.1 Базовые типы данных

Тип данных (data type) – множество допустимых значений данных с набором операций, которые применимы к этим данным [1, с. 35].

Тип данных определяет:

- внутреннее представление данных в памяти компьютера;
- множество значений, которые могут принимать величины этого типа;
- операции и функции, которые можно применять к величинам этого типа [2, с. 22-23].

Типы данных языка C++ подразделяются на базовые (fundamental) и производные (derived).

Базовый тип данных целый (int), символьный (char), расширенный символьный (wchar_t), логический (bool), вещественный (float), вещественный с двойной точностью (double). Спецификаторы типа – короткий (short), длинный (long), знаковый (signed), беззнаковый (unsigned).

Производный тип данных – массивы (arrays), перечисления (enumerations), структуры (structures), объединения (unions), классы (classes), функции

(functions), указатели (pointers), ссылки (references), указатели на члены классов (pointers to classmembers) [3, с. 35].

1.1.2 Лабораторная работа №1 «Расчет математического выражения»

Вариант 1. Создайте программу, в которой по известной начальной скорости V и времени полета тела T определяется угол α , под которым тело брошено по отношению к горизонту ($\alpha = \arcsin(\frac{g * T}{2 * V})$).

Вариант 2. Для тела, брошенного под углом α к горизонту с начальной скоростью V , определите дальность полета тела L (воспользуйтесь соотношением $L = \frac{V^2 * \sin(2 * \alpha)}{g}$).

Вариант 3. Разработайте программу, в которой по максимальной высоте подъема H и дальности полета L определяется начальная скорость тела V и угол α , под которым тело брошено к горизонту. Воспользуйтесь соотношениями $tg(\alpha) = 4 * H / L$ и $V = \sqrt{g * L / \sin(2 * \alpha)}$.

Вариант 4. Человек, находящийся на краю обрыва высотой H , бросает с начальной скоростью V камень под углом α к горизонту. Написать программу, в которой по введенному пользователем времени t определяется положение камня (высота от дна обрыва $x(t)$ и расстояние до края обрыва $y(t)$). Предусмотреть случай, когда камень упал на дно обрыва (использовать тернарный оператор). Уравнение движения камня вдоль горизонтальной оси записывается в виде $y(t) = H + V * t * \sin(\alpha) - g * t^2 / 2$. Координата отсчитывается от дна обрыва. Время полета камня T определяется по формуле $T = \frac{V * \sin(\alpha)}{g} * (1 + \sqrt{1 + \frac{2 * g * H}{V^2 * \sin^2(\alpha)}})$.

Вариант 5. Самолет летит на высоте H над землей со скоростью U . При подлете к наземному объекту он сбрасывает бомбу. Написать программу, в которой вычисляется подлетное расстояние S до объекта, над которым производится сбрасывание бомбы. Бомба до земли летит время $T = \sqrt{2 * H / g}$. Расстояние вычисляется по формуле $S = U * T$.

Вариант 6. Лодка плывет из пункта А в пункт Б и обратно. Скорость лодки V , расстояние между пунктами S и скорость течения реки U задаются пользователем. Написать программу, в которой вычисляется общее время движения лодки T . Учесть, что при движении по течению лодка плывет время $t_1 = S / (V + U)$, а при движении против течения на путь между пунктами уходит время $t_2 = S / (V - U)$. Общее время $T = t_1 + t_2$.

Вариант 7. Мотоциклист едет из пункта А в пункт Б и обратно. Написать программу, в которой по известной средней скорости движения V , расстоянию S между пунктами и времени движения t из пункта А в пункт Б вычисляется

средняя скорость V_1 мотоциклиста при движении из пункта А в пункт Б и скорость V_2 при движении из пункта Б в пункт А. Скорость V_1 мотоциклиста вычисляется по формуле $V_1 = S/t$. Скорость V_2 мотоциклиста вычисляется по формуле $V_2 = S/(2 * S/V - t)$.

Вариант 8. Поезд метро едет на протяжении времени t с постоянной скоростью V , а затем такой же промежуток времени тормозит до полной остановки. Разработать программу для вычисления пути S , пройденного поездом. Путь S рассчитывается по формуле $S = S_1 + S_2$. Пути S_1 и S_2 – пути, пройденные поездом при равномерном и равнозамедленном движении соответственно. Путь S_1 рассчитывается по формуле $S_1 = V * t$. Путь S_2 рассчитывается по формуле $S_2 = a * t^2 / 2$, где $a = V/t$ – ускорение при равнозамедленном движении. Для сравнения рассчитайте результат по соотношению $S = 3 * V * t / 2$.

Методические указания

- 1 Создайте проект «Консольное приложение Win 32».
- 2 Разработайте блок-схему алгоритма решения задачи.
- 3 Формализуйте алгоритм решения задачи на ПЭВМ.
- 4 Выведите на экран видеомонитора результаты работы программы.
- 5 Оформите отчет по лабораторной работе.

1.1.3 Практическая работа №1 «Линейный процесс»

Вариант 1. Разработайте программу, предназначенную для вычисления работы, совершаемой постоянным током на участке за определенное время по формуле $A = i * U * t$, где t – время прохождения тока, U – напряжение на участке, i – сила тока.

Вариант 2. Разработайте программу, которая вычисляет по формуле $S_n = \frac{2 * a_1 + d * (n - 1)}{2} * n$, сумму n членов арифметической прогрессии S_n .

Входные данные программы: a_1 – первый член прогрессии, d – разность, n – число членов прогрессии.

Вариант 3. Создайте программу расчета объема V , массы m и площади S основания металлического слитка. Известны плотность ρ , высота h и радиус основания R цилиндрического слитка, полученного в металлургической лаборатории. Расчетные формулы $S = 2 * \pi * R$, $V = \pi * R^2 * h$, $m = \rho * V$.

Вариант 4. Разработайте программу, которая рассчитывает гипотенузу и углы треугольника. Заданы два катета прямоугольного треугольника.

Вариант 5. Разработать программу, которая рассчитывает площадь S прямоугольного треугольника. Известны гипотенуза c и прилежащий угол α треугольника.

Вариант 6. Разработайте программу, которая вычисляет площадь S и периметр P квадрата. Задана диагональ квадрата d .

Вариант 7. Разработайте программу, которая вычисляет площадь S прямоугольника. Заданы диагональ прямоугольника d и угол α между диагональю и большей стороной.

Вариант 8. Разработайте программу, которая рассчитывает углы треугольника α, β, γ . Треугольник задан величинами своих сторон – a, b, c .

1.2 Выражения и операции

1.2.1 Переменные и операции

Переменная (variable) – именованная область памяти компьютера, значение которой изменяется при работе программы [1, с. 35].

Переменные подразделяются на локальные и глобальные.

Локальная переменная (local variable) – переменная, используемая в блоке программы или функции, где она объявлена. Глобальная переменная – переменная, областью действия которой является вся программа.

Выражение (expression) – конструкция языка для вычисления значения в соответствии со значениями операндов [1, с. 46].

Выражения состоят из операндов, знаков операций и скобок и применяются для вычисления значения.

Операция (operation) – действие, выполняемое над операндами [1, с. 46].

Операции подразделяются на унарные, бинарные и тернарные. В языке C++ применяются арифметические, логические, побитовые операции и операции отношения.

Операнд (operand) – часть команды, определяющая аргумент, над которым выполняется операция [1, с. 46].

Операндами являются данные и результаты операции. В качестве операндов используются переменные, константы, вызовы функций.

1.2.2 Приоритет и направление операций

Значение выражения вычисляется в порядке, соответствующем приоритету операций.

Приоритет операций (operation precedence) – очередность выполнения операций в выражении [1, с. 46].

Направление выполнения операций – свойство операции выполняться слева направо или наоборот.

Операции наивысшего приоритета – разрешение видимости ($::$), индекс массива ($[]$), вызов функции ($()$), выбор члена структуры или класса ($., \rightarrow$). Это операции 16 класса и выполняются слева направо.

Операция наименьшего приоритета – операция запятая ($,$). Это операция 1 класса и выполняется слева направо.

1.2.3 Лабораторная работа №2 «Математические функции»

Разработайте программу, рассчитывающую выходные переменные двух функций. Используйте математические функции, определенные в заголовочном файле math.h (cmath). Для расчета выходных переменных y_1 и y_2 создайте функции.

Вариант 1. $y_1 = 2 * \sin^2 * (3 * \pi - 2 * \alpha) * \cos^2 (5 * \pi + 2 * \alpha)$.

$$y_2 = \frac{1}{4} - \frac{1}{4} * \sin\left(\frac{5}{2} * \pi - 8 * \alpha\right).$$

Вариант 2. $y_1 = \cos \alpha + \sin \alpha + \cos 3 * \alpha + \sin 3 * \alpha$.

$$y_2 = 2 * \sqrt{2} * \cos \alpha * \sin\left(\frac{\pi}{4} + 2 * \alpha\right).$$

Вариант 3. $y_1 = \frac{\sin 2 * \alpha + \sin 5 * \alpha - \sin 3 * \alpha}{\cos \alpha + 1 - 2 * \sin^2 * 2 * \alpha}$.

$$y_2 = 2 * \sin \alpha.$$

Вариант 4. $y_1 = \frac{\sin 2 * \alpha + \sin 5 * \alpha - \sin 3 * \alpha}{\cos \alpha - \cos 3 * \alpha + \cos 5 * \alpha}$.

$$y_2 = \operatorname{tg} 3 * \alpha.$$

Вариант 5. $y_1 = 1 - \frac{1}{4} * \sin^2 * 2 * \alpha + \cos 2 * \alpha$.

$$y_2 = \cos^2 \alpha + \cos^4 \alpha.$$

Вариант 6. $y_1 = \cos \alpha + \cos 2 * \alpha + \cos 6 * \alpha + \cos 7 * \alpha$.

$$y_2 = 4 * \cos \frac{\alpha}{2} * \cos \frac{5}{2} * \alpha * \cos 4 * \alpha.$$

Вариант 7. $y_1 = \cos^2\left(\frac{3}{8} * \pi - \frac{\alpha}{4}\right) - \cos^2\left(\frac{11}{8} * \pi + \frac{\alpha}{4}\right)$.

$$y_2 = \frac{\sqrt{2}}{2} * \sin \frac{\alpha}{2}.$$

Вариант 8. $z_1 = \cos^4 x + \sin^2 y + \frac{1}{4} * \sin^2 2 * x - 1$.

$$z_2 = \sin(y + x) * \sin(y - x).$$

Методические указания

- 1 Создайте проект «Консольное приложение Win 32».
- 2 Разработайте блок-схему алгоритма решения задачи.
- 3 Изучите функции стандартной библиотеки [2, с. 409].
- 4 Продумайте структуру программы. Используйте четыре блока консольного приложения.

5 Формализуйте алгоритм решения задачи. Используйте приоритет и направление операций [4, с. 28].

6 Выведите на экран видеомонитора результаты работы программы.

7 Оформите отчет по лабораторной работе.

1.2.4 Лабораторная работа №3 «Табулирование функции»

Вычислите и выведите на экран в виде таблицы значения функции y в интервале от X_n до X_k с шагом h , где a, b, c – действительные числа.

Вариант 1

$$y = \begin{cases} a * x^2 + b, \text{ при } x + 2 < 0, b \neq 0 \\ \frac{x - a}{x - c}, \text{ при } x + 2 > 0 \text{ и } c \neq 0 \\ \frac{10 * x}{c - 4}, \text{ в остальных случаях} \end{cases}$$

Вариант 2

$$y = \begin{cases} \frac{1}{a * x} - b, \text{ при } x + 5 < 0, c = 0 \\ \frac{x - a}{x}, \text{ при } x + 5 > 0 \text{ и } c \neq 0 \\ \frac{10 * x}{c - 4}, \text{ в остальных случаях} \end{cases}$$

Вариант 3

$$y = \begin{cases} a * x^2 + b * x + c, \text{ при } a < 0, c \neq 0 \\ \frac{-a}{x - c}, \text{ при } a > 0 \text{ и } c = 0 \\ a * (x + c), \text{ в остальных случаях} \end{cases}$$

Вариант 4

$$y = \begin{cases} -a * x - c, \text{ при } c < 0, x \neq 0 \\ \frac{x - a}{-c}, \text{ при } c > 0 \text{ и } x = 0 \\ \frac{b * x}{c - a}, \text{ в остальных случаях} \end{cases}$$

Вариант 5

$$y = \begin{cases} a - \frac{x}{10+b}, & \text{при } x < 0, b \neq 0 \\ \frac{x-a}{x-c}, & \text{при } x > 0 \text{ и } b = 0 \\ 3 * x + \frac{2}{c}, & \text{в остальных случаях} \end{cases}$$

Вариант 6

$$y = \begin{cases} a * x^2 + b^2 * x, & \text{при } c < 0, b \neq 0 \\ \frac{x+a}{x+c}, & \text{при } c > 0 \text{ и } b = 0 \\ \frac{x}{c}, & \text{в остальных случаях} \end{cases}$$

Вариант 7

$$y = \begin{cases} -a * x^2 - b, & \text{при } x < 5, c \neq 0 \\ \frac{x-a}{x}, & \text{при } x > 5 \text{ и } c = 0 \\ \frac{-x}{c}, & \text{в остальных случаях} \end{cases}$$

Вариант 8

$$y = \begin{cases} -a * x^2, & \text{при } c < 0, a \neq 0 \\ \frac{a-x}{c * x}, & \text{при } c > 0 \text{ и } a = 0 \\ \frac{x}{c}, & \text{в остальных случаях} \end{cases}$$

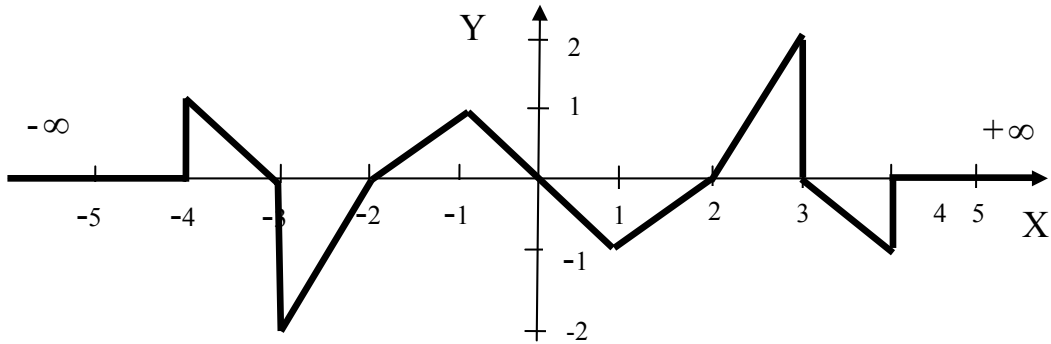
Методические указания

- 1 Создайте проект «Консольное приложение Win 32».
- 2 Разработайте блок-схему алгоритма решения задачи.
- 3 Продумайте структуру программы.
- 4 Формализуйте алгоритм решения задачи.
- 5 Используйте операторы ветвления и цикла.
- 6 Выведите на экран видеомонитора результаты работы программы.
- 7 Оформите отчет по лабораторной работе.

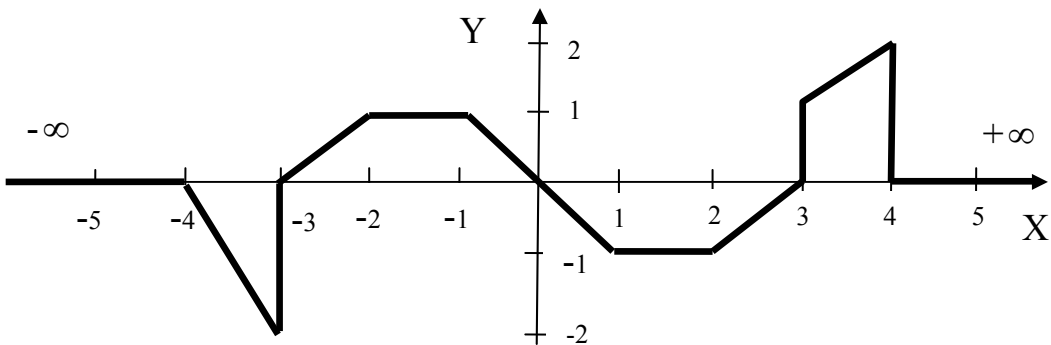
1.2.5 Практическая работа №2
«Вычисление значений функций, заданных графическим способом»

Разработайте программу на языке C++, формализующую расчет выходной переменной y функциональных зависимостей $y=f(x)$, заданных графическим способом.

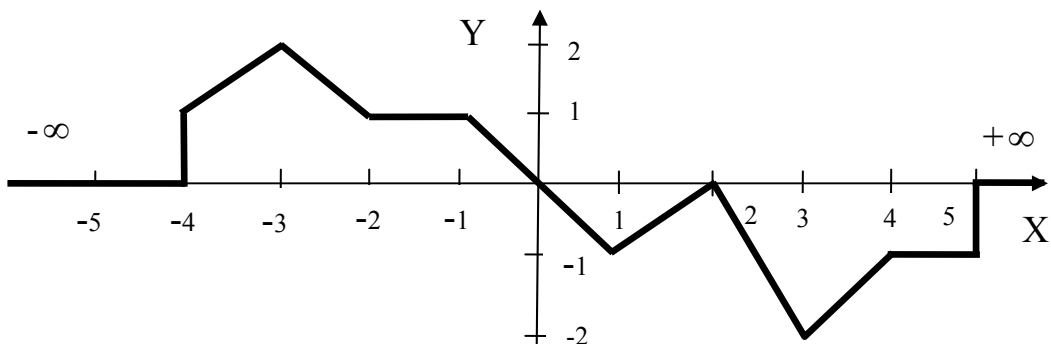
Вариант 1



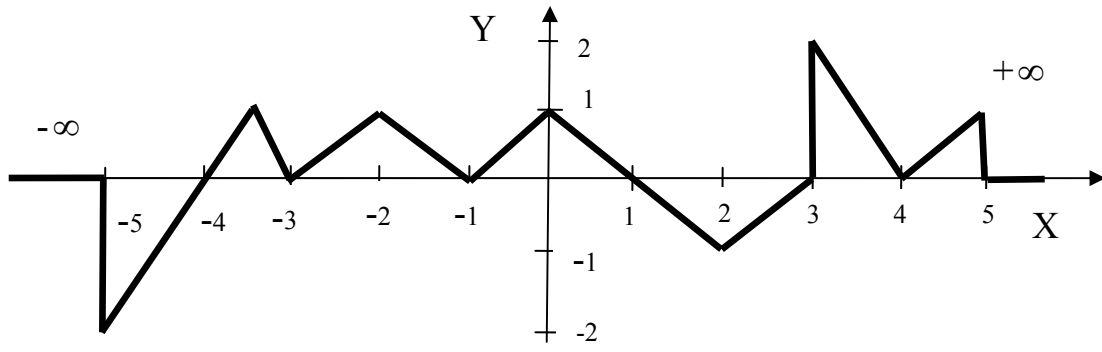
Вариант 2



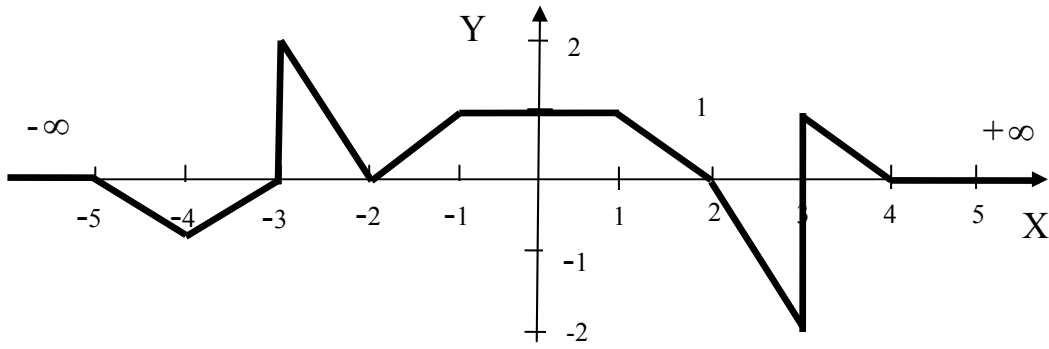
Вариант 3



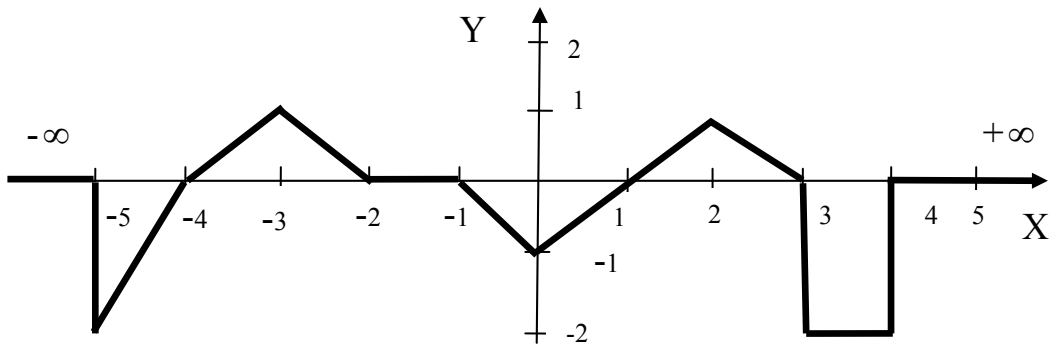
Вариант 4



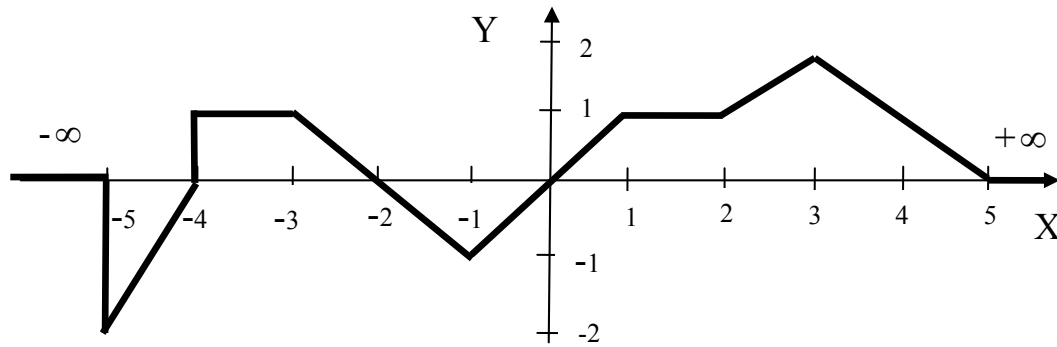
Вариант 5



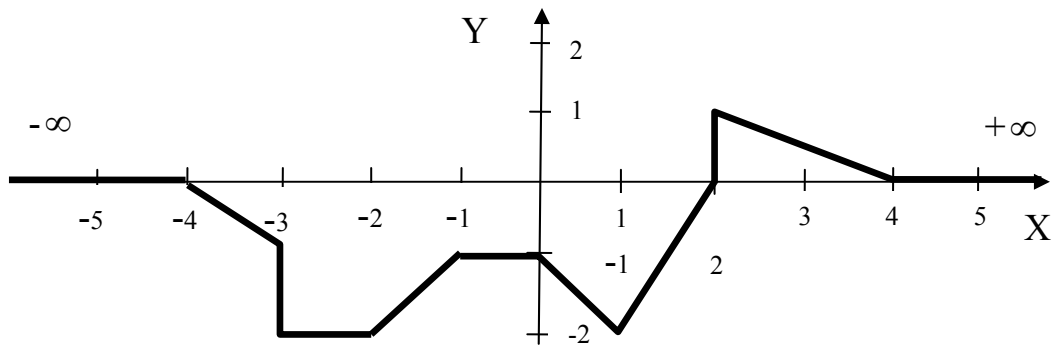
Вариант 6



Вариант 7



Вариант 8



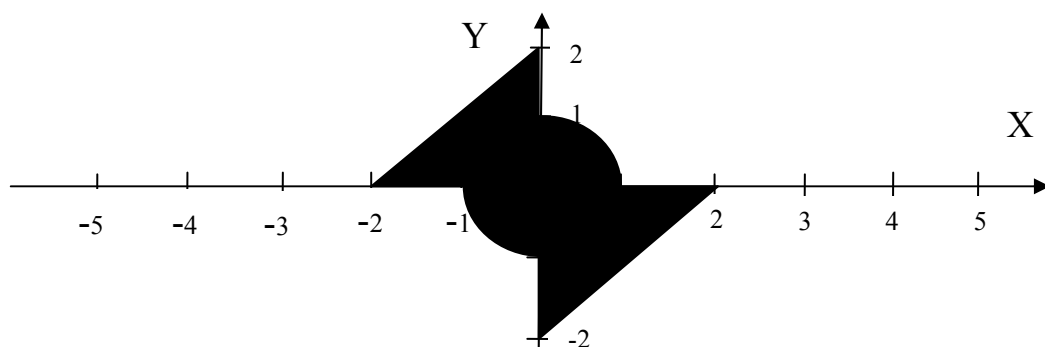
Методические указания

- 1 Преобразуйте графическую форму записи исходных данных в аналитическую форму записи.
- 2 Создайте проект «Консольное приложение Win 32».
- 3 Разработайте блок-схему алгоритма решения задачи.
- 4 Продумайте структуру программы.
- 5 Формализуйте алгоритм решения задачи.
- 6 Используйте операторы ветвления и цикла.
- 7 Выведите на экран видеомонитора результаты работы программы.
- 8 Оформите отчет по лабораторной работе.

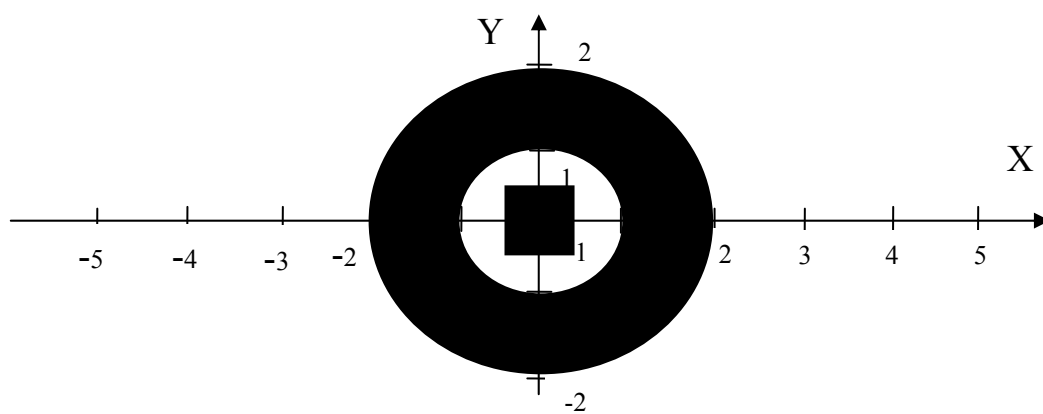
1.2.6 Практическая работа №3 «Попадание точки в плоскую фигуру»

Разработайте программу на языке C++, формализующую определение попадания точки, заданной координатами x и y , в плоскую фигуру.

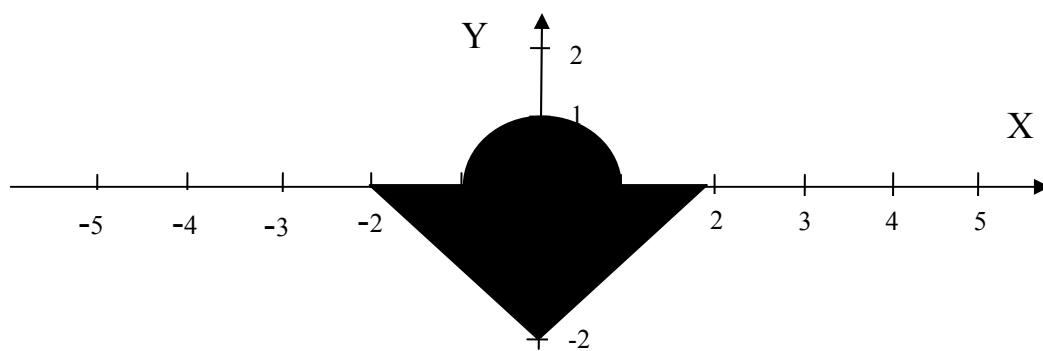
Вариант 1



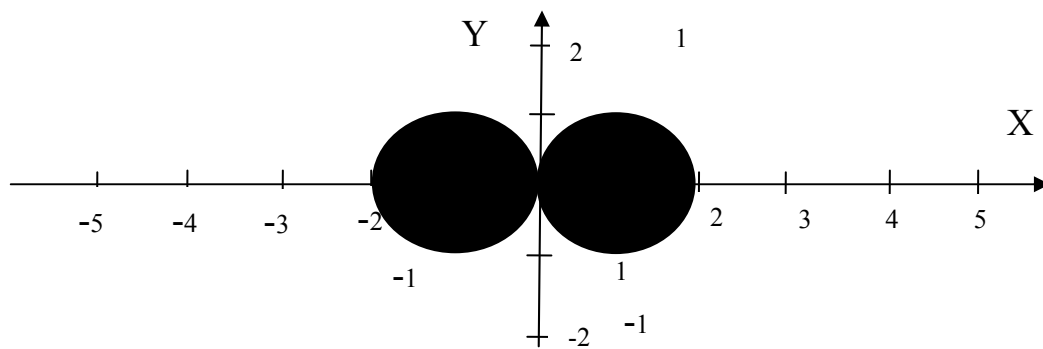
Вариант 2



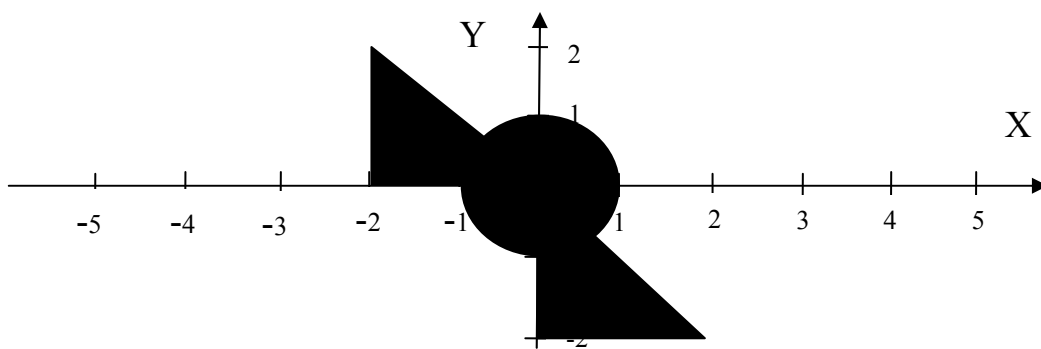
Вариант 3



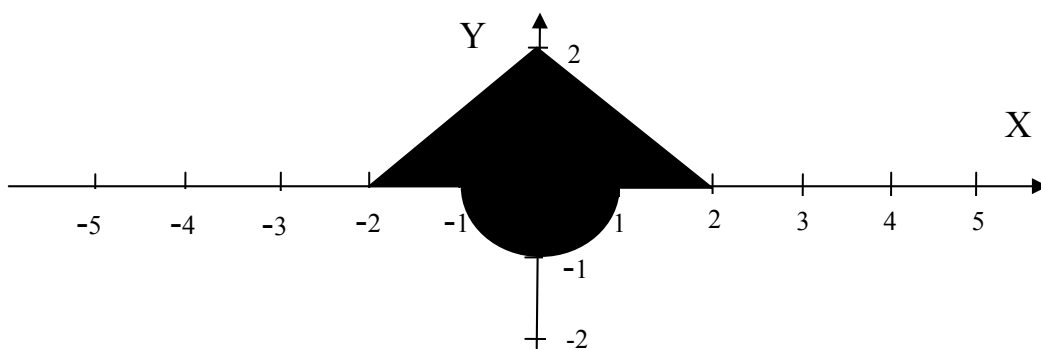
Вариант 4



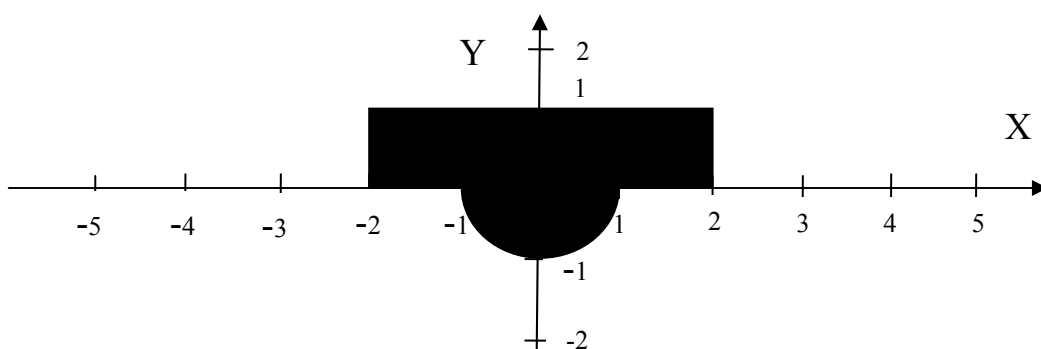
Вариант 5



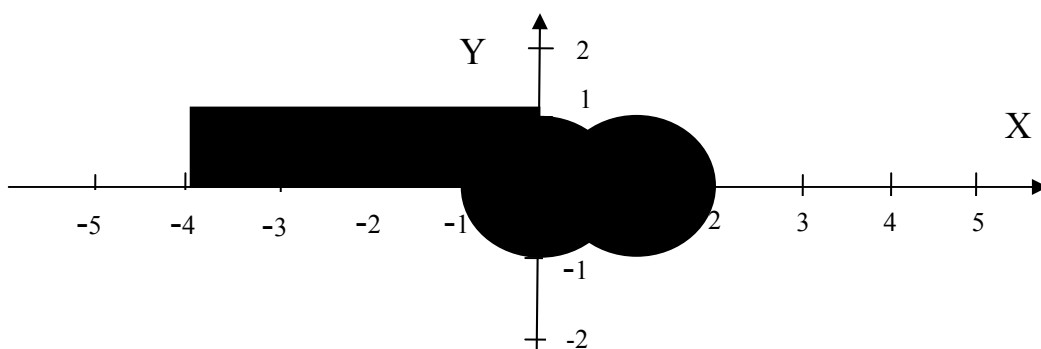
Вариант 6



Вариант 7



Вариант 8



Методические указания

- 1 Преобразуйте графическую форму записи исходных данных в аналитическую форму записи.
- 2 Создайте проект «Консольное приложение Win 32».
- 3 Разработайте блок-схему алгоритма решения задачи.
- 4 Продумайте структуру программы.
- 5 Формализуйте алгоритм решения задачи.
- 6 Используйте операторы ветвления и цикла.
- 7 Выведите на экран видеомонитора результаты работы программы.
- 8 Оформите отчет по лабораторной работе.

1.3 Управление ходом выполнения программы

1.3.1 Операторы

Оператор (statement) – установленное синтаксисом языка действие в программе [1, с. 53].

Операторы языка C++ подразделяются на группы:

- оператор выражение;
- операторы ветвления;
- операторы цикла;
- операторы передачи управления.

Выражение, заканчивающееся точкой с запятой, и пустой оператор относятся к первой группе операторов. Операторы ветвления включают условный оператор **if else** и оператор **switch** (переключатель). К операторам цикла относятся пошаговый оператор цикла **for**, цикл с предусловием **while**, цикл с постусловием **do while**. Операторы передачи управления – оператор безусловного перехода **goto**, оператор выхода из цикла **break**, оператор перехода к следующей итерации цикла **continue**, оператор возврата из функции **return**.

1.3.2 Лабораторная работа №4 «Операторы языка C++»

Вариант 1. Устаревшая денежная система Великобритании состояла из фунтов, шиллингов и пенсов. Один фунт был равен 20 шиллингам, а один шиллинг – 12 пенсам. Для записи использовалась система, состоящая из знака £ и трёх десятичных значений, разделённых точками. Например, запись £5.2.8 обозначала 5 фунтов, 2 шиллинга и 8 пенсов. Современная денежная система, принятая в 50-е годы XX века, состоит из фунтов и пенсов. Один фунт равен 100 пенсам. Такой фунт называется десятичным. В новой денежной системе указанная сумма обозначается как £5.13 (если быть точнее £5.1333333). Разработайте программу, которая преобразует сумму, записанную в старом

формате (фунты, шиллинги, пенсы), в сумму, записанную в новом формате (фунты, пенсы).

Вариант 2. Разработайте программу, изображающую на экране видеомонитора пирамиду из символов 'X'. Верхняя часть пирамиды должна выглядеть следующим образом:

```

      X
     XXX
    XXXXX
   XXXXXXX
  XXXXXXXXX

```

Пирамида должна быть высотой в 20 линий.

Вариант 3. Напишите программу, рассчитывающую сумму денег, которые вы получите при вложении начальной суммы с фиксированной процентной ставкой дохода через определённое количество лет. Пользователь вводит с клавиатуры начальный вклад, число лет и процентную ставку.

Вариант 4. Напишите программу – эквивалент калькулятора, выполняющего четыре основных арифметических операции. Программа должна запрашивать ввод пользователем первого операнда, знака операции и второго операнда. Для хранения операндов следует использовать переменные вещественного типа. Выбрать операцию можно при помощи оператора switch. В конце программа должна отображать результат на экране. Результат работы программы с пользователем имеет вид:

Введите первый операнд, операцию и второй операнд: 10/3

Результат равен 3,333333

Выполнить ещё одну операцию (y/n)?

Вариант 5. Напишите программу для вычисления положения тела, брошенного под углом к горизонту, в произвольный момент времени t . Учтёшь сопротивление воздуха: сила трения пропорциональна скорости тела. Начальная скорость тела равна V и направлена под углом α к горизонту. Для вычислений использовать дискретную модель. В векторном виде уравнение

имеет вид $m \cdot \ddot{\vec{r}} = m \cdot \vec{g} - \gamma \cdot \dot{\vec{r}}$, где через γ обозначен коэффициент пропорциональности, входящий в выражение для силы сопротивления воздуха

$\vec{F} = -\gamma \cdot \dot{\vec{r}}$. В проекциях на координатные оси получаем уравнения

$m \cdot \ddot{x} = -\gamma \cdot \dot{x}$ и $m \cdot \ddot{y} = -m \cdot g - \gamma \cdot \dot{y}$. В модели дискретного времени получаем уравнения $x_{n+1} = x_n + v_n \cdot \Delta t$, $y_{n+1} = y_n + u_n \cdot \Delta t$, $v_{n+1} = v_n - k \cdot v_n \cdot \Delta t$ и

$$u_{n+1} = u_n - g \cdot \Delta t - k \cdot u_n \cdot \Delta t,$$

где $k = \gamma / m$ – коэффициент;

x_n – горизонтальная координата в момент времени t_n ;

y_n – вертикальная координата в момент времени t_n ;

v_n – проекция скорости на горизонтальную ось в момент времени t_n ;

u_n – проекция скорости на вертикальную ось в момент времени t_n ;

Δt – шаг дискретности по времени.

Для сравнения приведём точное решение. Оно имеет вид

$$x(t) = \frac{V \cdot \cos(\alpha)}{k} * (1 - \exp(-k * t)), \quad y(t) = \frac{g/k + v * \sin(\alpha)}{k} * (1 - \exp(-k * t)) - \frac{g * t}{k}.$$

Вариант 6. Напишите программу для решения квадратного уравнения, которое имеет вид $a * x^2 + b * x + c = 0$. Параметры a , b и c вводятся пользователем. Использовать аналитические формулы для решений уравнения, при этом учесть различные варианты. Например, $a = 0$, отсутствие решений (комплексные решения).

Вариант 7. Дано натуральное число n ($n \geq 2$). Найти все меньшие n простые числа, используя решето Эратосфена. Решетом Эратосфена называют следующий способ. Выпишем все целые числа от 2 до n . Первое простое число 2. Подчеркнем его, а все большие числа, кратные 2, зачеркнем. Первое из оставшихся чисел 3. Подчеркнем его как простое, а все большие числа, кратные 3, зачеркнем. Первое число из оставшихся теперь 5, т.к. 4 уже зачеркнуто. Подчеркнем его как простое, а все большие числа, кратные 5, зачеркнем и т.д.

Вариант 8. Даны натуральные числа $A_i = \{a_1, a_2, \dots, a_n\}$, $i = \overline{1, n}$. Определить количество членов a_k последовательности a_1, a_2, \dots, a_n , являющихся

- 1) нечетными числами;
- 2) числами, кратными 3 и не кратными 5;
- 3) квадратами четных чисел;
- 4) числами, удовлетворяющими условию $a_k < \frac{a_{k-1} + a_{k+1}}{2}$;
- 5) числами, удовлетворяющими условию $2^k < a_k < k!$;
- 6) числами, имеющими четные порядковые номера и являющимися нечетными числами.

Методические указания

- 1 Создайте проект «Консольное приложение Win 32».
- 2 Разработайте блок-схему алгоритма решения задачи.
- 3 Продумайте структуру программы.
- 4 Формализуйте алгоритм решения задачи.
- 5 Используйте операторы языка C++.
- 6 Выведите на экран видеомонитора результаты работы программы.
- 7 Оформите отчет по лабораторной работе.

2 Пользовательский тип данных. Массивы и структуры

2.1 Массивы

2.1.1 Операции над элементами

Массив – конечная именованная последовательность однотипных величин [2, с. 58].

При объявлении массива применяют квадратные скобки, внутри которых указывается размерность. Размерность – количество элементов массива.

Массивы подразделяются на:

- одномерные;
- многомерные;
- статические;
- динамические.

Одномерный массив – массив, элементу которого соответствует один индекс. Многомерный массив – массив, элементу которого соответствует несколько индексов. Статический массив – массив, размерность которого нельзя изменить при выполнении программы. Динамический массив – массив, размерность которого можно изменить при выполнении программы.

Доступ к элементам массива осуществляет операция индексации. Обращение к элементу массива производится по номеру, расположенному в квадратных скобках после имени массива.

2.1.2 Лабораторная работа №5 «Одномерные массивы»

Вариант 1. В одномерном массиве, состоящем из n вещественных элементов, вычислите:

- 1) сумму отрицательных элементов массива;
- 2) произведение элементов массива, расположенных между максимальным и минимальным элементами.

Вариант 2. В одномерном массиве, состоящем из n вещественных элементов, вычислите:

- 1) сумму положительных элементов массива;
- 2) произведение элементов массива, расположенных между максимальным по модулю и минимальным по модулю элементами.

Вариант 3. В одномерном массиве, состоящем из n целых элементов, вычислите:

- 1) произведение элементов массива с четными номерами;
- 2) сумму элементов массива, расположенных между первым и последним нулевыми элементами.

Преобразуйте массив таким образом, чтобы сначала располагались все положительные элементы, а потом — все отрицательные (элементы, равные нулю, считать положительными).

Вариант 4. В одномерном массиве, состоящем из n вещественных элементов, вычислите:

- 1) сумму элементов массива с нечетными номерами;
- 2) сумму элементов массива, расположенных между первым и последним отрицательными элементами.

Преобразуйте массив, удалив из него все элементы, модуль которых не превышает числа A , введенного пользователем с клавиатуры. Освободившиеся в конце массива элементы заполните нулями.

Вариант 5. В одномерном массиве, состоящем из n вещественных элементов, вычислите:

- 1) максимальный элемент массива;
- 2) сумму элементов массива, расположенных до последнего положительного элемента.

Преобразуйте массив, удалив из него все элементы, модуль которых находится в интервале $[a,b]$. Освободившиеся в конце массива элементы заполните нулями.

Вариант 6. В одномерном массиве, состоящем из n вещественных элементов, вычислите:

- 1) минимальный элемент массива;
- 2) сумму элементов массива, расположенных между первым и последним положительными элементами.

Преобразуйте массив таким образом, чтобы сначала располагались все элементы, равные нулю, а потом все остальные.

Вариант 7. В одномерном массиве, состоящем из n целых элементов, вычислите:

- 1) номер максимального элемента массива;
- 2) произведение элементов массива, расположенных между первым и вторым нулевыми элементами.

Преобразуйте массив таким образом, чтобы в первой его половине располагались элементы, стоявшие в нечетных позициях, а во второй – элементы, стоявшие в четных позициях.

Вариант 8. В одномерном массиве, состоящем из n вещественных элементов, вычислите:

- 1) номер минимального элемента массива;
- 2) сумму элементов массива, расположенных между первым и вторым отрицательными элементами.

Преобразуйте массив таким образом, чтобы сначала располагались все элементы, модуль которых не превышает числа B , введенного пользователем с клавиатуры, а потом все остальные.

Методические указания

- 1 Создайте проект «Консольное приложение Win 32».
- 2 Разработайте блок-схему алгоритма решения задачи.

- 3 Создайте динамический одномерный массив.
- 4 Проинициализируйте элементы массива с помощью генератора случайных чисел.
- 5 Выведите исходный массив на экран видеомонитора.
- 6 Формализуйте алгоритм решения задачи.
- 7 Упорядочите исходный массив по возрастанию (убыванию) элементов методами обмена, выбора, вставки.
- 8 Выведите на экран видеомонитора результаты работы программы.

2.1.3 Практическая работа №4 **«Пользовательский тип данных. Одномерные массивы»**

Вариант 1. В одномерном массиве, состоящем из n вещественных элементов, вычислите:

- 1) номер минимального по модулю элемента массива;
- 2) сумму модулей элементов массива, расположенных после первого отрицательного элемента.

Преобразуйте массив, удалив из него все элементы, величина которых находится в интервале $[a,b]$. Освободившиеся в конце массива элементы заполните нулями.

Вариант 2. В одномерном массиве, состоящем из n вещественных элементов, вычислите:

- 1) номер максимального по модулю элемента массива;
- 2) сумму элементов массива, расположенных после первого положительного элемента.

Преобразуйте массив таким образом, чтобы сначала располагались все элементы, целая часть которых лежит в интервале $[a,b]$, а потом – все остальные.

Вариант 3. В одномерном массиве, состоящем из n вещественных элементов, вычислите:

- 1) количество элементов массива, лежащих в диапазоне от A до B ;
- 2) сумму элементов массива, расположенных после максимального элемента.

Вариант 4. В одномерном массиве, состоящем из n вещественных элементов, вычислите:

- 1) количество элементов массива, равных нулю;
- 2) сумму элементов массива, расположенных после минимального элемента.

Вариант 5. В одномерном массиве, состоящем из n вещественных элементов, вычислите:

- 1) количество элементов массива, больших C ;
- 2) произведение элементов массива, расположенных после максимального по модулю элемента.

Преобразуйте массив таким образом, чтобы сначала располагались все отрицательные элементы, а потом – все положительные (элементы, равные нулю, считать положительными).

Вариант 6. В одномерном массиве, состоящем из n вещественных элементов, вычислите:

- 1) количество отрицательных элементов массива;
- 2) сумму модулей элементов массива, расположенных после минимального по модулю элемента.

Замените все отрицательные элементы массива их квадратами и упорядочите элементы массива по возрастанию.

Вариант 7. В одномерном массиве, состоящем из n целых элементов, вычислите:

- 1) количество положительных элементов массива;
- 2) сумму элементов массива, расположенных после последнего элемента, равного нулю.

Преобразуйте массив таким образом, чтобы сначала располагались все элементы, целая часть которых не превышает числа A , введенного с клавиатуры, а потом – все остальные.

Вариант 8. В одномерном массиве, состоящем из n вещественных элементов, вычислите:

- 1) количество элементов массива, меньших C ;
- 2) сумму целых частей элементов массива, расположенных после последнего отрицательного элемента.

Преобразуйте массив таким образом, чтобы сначала располагались все элементы, отличающиеся от максимального не более чем на 20%, а потом – все остальные.

2.1.4 Лабораторная работа №6 «Двумерные массивы»

Вариант 1. Дана целочисленная прямоугольная матрица $[A_{8*10}]$.

Разработайте программу, определяющую величины:

- 1) количество строк, не содержащих ни одного нулевого элемента;
- 2) максимальное значение элементов, встречающихся в заданной матрице более одного раза;
- 3) суммы элементов строк и столбцов матрицы.

Суммы элементов строк и столбцов матрицы упорядочите по возрастанию методом обмена.

Вариант 2. Дана целочисленная прямоугольная матрица $[A_{10*14}]$.

Напишите программу, определяющую величины:

- 1) количество столбцов, не содержащих ни одного нулевого элемента;
- 2) максимальное и минимальное значение элементов матрицы;
- 3) среднее арифметическое значение элементов строк и столбцов матрицы.

Средние арифметические значения элементов строк и столбцов матрицы упорядочите по возрастанию методом выбора.

Вариант 3. Дана целочисленная прямоугольная матрица $[A_{11*13}]$.
Напишите программу, определяющую величины:

- 1) количество столбцов, содержащих хотя бы один нулевой элемент;
- 2) номер строки, в которой находится самая длинная серия одинаковых элементов;
- 3) среднее геометрическое значение элементов строк и столбцов матрицы.

Средние геометрические значения элементов строк и столбцов матрицы упорядочите по возрастанию методом вставки.

Вариант 4. Дана целочисленная квадратная матрица $[A_{14*14}]$.
Разработайте программу, определяющую величины:

- 1) сумму элементов в строках, которые не содержат отрицательных элементов;
- 2) максимум среди элементов выше главной диагонали матрицы;
- 3) суммы четных элементов строк и столбцов матрицы.

Суммы четных элементов строк и столбцов матрицы упорядочите по убыванию методом обмена.

Вариант 5. Дана целочисленная квадратная матрица $[A_{10*10}]$.
Разработайте программу, определяющую величины:

- 1) суммы элементов в столбцах, которые не содержат отрицательных элементов;
- 2) минимум среди модулей элементов ниже побочной диагонали матрицы;
- 3) суммы нечетных элементов строк и столбцов матрицы.

Суммы нечетных элементов строк и столбцов матрицы упорядочите по убыванию методом выбора.

Вариант 6. Дана целочисленная прямоугольная матрица $[A_{9*16}]$.
Напишите программу, определяющую величины:

- 1) суммы элементов в строках, которые содержат хотя бы один отрицательный элемент;
- 2) минимальный элемент матрицы;
- 3) среднее арифметическое значение четных элементов строк и столбцов матрицы.

Средние арифметические значения четных элементов строк и столбцов матрицы упорядочите по убыванию методом вставки.

Вариант 7. Дана целочисленная квадратная матрица $[A_{13*13}]$.
Разработайте программу, определяющую величины:

- 1) количество отрицательных элементов матрицы;
- 2) максимальный элемент матрицы;
- 3) суммы элементов матрицы, находящихся ниже главной диагонали.

Элементы матрицы, находящиеся выше главной диагонали, упорядочите по возрастанию методом обмена.

Вариант 8. Дана целочисленная квадратная матрица $[A_{11*11}]$. Напишите программу, определяющую величины:

- 1) количество положительных элементов матрицы;
- 2) максимальный и минимальный нечетные элементы строк матрицы;
- 3) суммы элементов матрицы, находящихся ниже побочной диагонали.

Элементы матрицы, находящиеся ниже побочной диагонали, упорядочите по возрастанию методом выбора.

Методические указания

1 Создайте проект «Консольное приложение Win 32».

2 Разработайте блок-схему алгоритма решения задачи.

3 Создайте динамический двумерный массив.

4 Проинициализируйте элементы двумерного массива с помощью генератора случайных чисел.

5 Выведите двумерный массив на экран видеомонитора в виде матрицы.

6 Формализуйте алгоритм решения задачи.

7 Выведите на экран видеомонитора результаты работы программы.

2.1.5 Практическая работа №5 «Обработка матриц»

Вариант 1. Дана целочисленная квадратная матрица $[A_{10*10}]$. Разработайте программу, определяющую величины:

- 1) количество ненулевых элементов матрицы;
- 2) максимальный и минимальный четные элементы столбцов матрицы;
- 3) суммы элементов матрицы, находящихся ниже побочной диагонали.

Элементы матрицы, находящиеся ниже побочной диагонали, упорядочите по убыванию простым методом обмена.

Вариант 2. Дана целочисленная квадратная матрица $[A_{12*12}]$. Разработайте программу, определяющую величины:

- 1) количество элементов матрицы, меньших величины В;
- 2) максимальный и минимальный нечетные элементы столбцов матрицы;
- 3) суммы элементов матрицы, находящихся выше побочной диагонали.

Элементы матрицы, находящиеся выше побочной диагонали, упорядочите по возрастанию простым методом вставки.

Вариант 3. Дана целочисленная квадратная матрица $[A_{11*11}]$. Разработайте программу, определяющую величины:

- 1) количество элементов матрицы, больших величины В;
- 2) минимальный по модулю элемент матрицы;
- 3) суммы элементов, расположенных по периметру матрицы.

Элементы, расположенные по периметру матрицы, упорядочите по убыванию простым методом выбора.

Вариант 4. Дана целочисленная квадратная матрица $[A_{14*14}]$.

Разработайте программу, определяющую величины:

- 1) количество элементов матрицы, находящихся в диапазоне от А до В;
- 2) максимальный по модулю элемент матрицы;
- 3) суммы элементов, расположенных на диагоналях матрицы.

Элементы, расположенные на диагоналях матрицы, упорядочите по убыванию простым методом выбора.

Вариант 5. Дана целочисленная квадратная матрица $[A_{15*15}]$.

Разработайте программу, определяющую величины:

- 1) количество элементов матрицы, содержащих простые числа;
- 2) минимум среди элементов главной и побочной диагоналей матрицы;
- 3) средние квадратичные значения столбцов и строк матрицы.

Средние квадратичные значения столбцов и строк матрицы упорядочите по убыванию простым методом вставки.

Вариант 6. Дана целочисленная квадратная матрица $[A_{12*12}]$.

Разработайте программу, определяющую величины:

- 1) количество элементов матрицы, содержащих простые числа;
- 2) максимум среди элементов главной и побочной диагоналей матрицы;
- 3) средние квадратичные значения четных столбцов и нечетных строк матрицы.

Средние квадратичные значения четных столбцов и нечетных строк матрицы упорядочите по возрастанию простым методом обмена.

Вариант 7. Дана целочисленная прямоугольная матрица $[A_{15*15}]$.

Разработайте программу, определяющую величины:

- 1) количество элементов матрицы, содержащих простые числа;
- 2) минимум элементов, расположенных ниже главной диагонали матрицы;
- 3) средние квадратичные значения нечетных столбцов и четных строк матрицы.

Средние квадратичные значения нечетных столбцов и четных строк матрицы упорядочите по возрастанию простым методом выбора.

Вариант 8. Дана целочисленная квадратная матрица $[A_{13*13}]$.

Разработайте программу, определяющую величины:

- 1) количество элементов матрицы, содержащих простые числа;
- 2) максимум элементов, расположенных выше главной диагонали матрицы;
- 3) средние арифметические значения столбцов и строк матрицы.

Средние арифметические значения столбцов и строк матрицы упорядочите по возрастанию простым методом вставки.

2.2 Структуры

2.2.1 Теоретическое введение

Структура (structure) – производный тип данных, объединяющий логически связанные элементы разных типов, объединенные общим именем [1, с. 121].

Формат объявления структуры имеет вид:

```
struct [имя_типа] {тип_1 элемент_1; тип_2 элемент_2; ... тип_n элемент_n;} [список_описателей];
```

Доступ к элементам структуры производится с помощью операций точка (.) и стрелка (→). Инициализация структуры осуществляется перечислением значений элементов в фигурных скобках. Операция присваивания определена для переменных одного структурного типа. Если элементом структуры является другая структура, то доступ к элементам выполняется через две операции выбора. Массив структуры обрабатывается аналогично массиву элементов базового типа. Структура может передаваться в функцию и возвращаться в качестве значения функции.

2.2.2 Лабораторная работа №7 «Структуры»

Вариант 1. Создайте структуру с именем STUDENT, содержащую поля: фамилия и инициалы, номер группы, курс, успеваемость (массив из четырех элементов).

Напишите программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из шести структур типа STUDENT;
- записи упорядочить по возрастанию номера группы;
- вывод на дисплей фамилий и номеров групп для всех студентов, включенных в массив, если средний балл студента больше 4.0;
- если таких студентов нет, вывод соответствующего сообщения.

Вариант 2. Создайте структуру с именем STUDENT, содержащую поля: фамилия и инициалы, номер группы, специальность, успеваемость (массив из четырех элементов).

Напишите программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из пяти структур типа STUDENT;
- записи упорядочить по возрастанию среднего балла;
- вывод на дисплей фамилий и номеров групп для всех студентов, имеющих оценки 4 и 5;
- если таких студентов нет, вывод соответствующего сообщения.

Вариант 3. Создайте структуру с именем STUDENT, содержащую следующие поля: фамилия и инициалы, номер группы, адрес проживания, успеваемость (массив из четырех элементов).

Напишите программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из пяти структур типа STUDENT;

- записи упорядочить в алфавитном порядке по адресу проживания;

- вывод на дисплей студентов, имеющих оценку 2;

- если таких студентов нет, вывод соответствующего сообщения.

Вариант 4. Создайте структуру с именем AEROFLOT, содержащую поля: пункт назначения, номер рейса, тип самолета, время полета.

Напишите программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из четырех элементов типа AEROFLOT;

- записи упорядочить по возрастанию номера рейса;

- вывод на экран номеров рейсов и типов самолетов, вылетающих в пункт назначения, название которого совпало с названием, введенным с клавиатуры;

- если таких рейсов нет, вывод на дисплей соответствующего сообщения.

Вариант 5. Создайте структуру с именем BUSABROAD, содержащую поля: пункт назначения, номер маршрута, марка автобуса, стоимость проезда.

Напишите программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из пяти элементов типа BUSABROAD;

- записи разместить в алфавитном порядке по названиям пунктов назначения;

- вывод на экран пунктов назначения и номеров маршрутов, обслуживаемых автобусом, марка которого введена с клавиатуры;

- если таких рейсов нет, вывод на дисплей соответствующего сообщения.

Вариант 6. Создайте структуру с именем WORKER, содержащую поля: фамилия и инициалы работника, название занимаемой должности, год поступления на работу, паспортные данные.

Напишите программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из четырех структур типа WORKER;

- записи разместить в алфавитном порядке по фамилии и инициалам;

- вывод на дисплей фамилий работников, чей стаж работы в организации превышает значение, введенное с клавиатуры;

- если таких работников нет, вывод на дисплей соответствующего сообщения.

Вариант 7. Создайте структуру с именем TRAIN, содержащую поля: пункт назначения, номер поезда, время отправления, время прибытия.

Напишите программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из пяти элементов типа TRAIN;

- записи разместить в алфавитном порядке по названиям пунктов назначения;

- вывод на экран информации о поездах, отправляющихся после введенного с клавиатуры времени;

- если таких поездов нет, вывод на дисплей соответствующего сообщения.

Вариант 8. Создайте структуру с именем TRAIN, содержащую поля: пункт назначения, номер поезда, время отправления, стоимость проезда.

Напишите программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из пяти элементов типа TRAIN;

- записи упорядочить по времени отправления поезда;

- вывод на экран информации о поездах, направляющихся в пункт, название которого введено с клавиатуры;

- если таких поездов нет, вывод на дисплей соответствующего сообщения.

Методические указания

1 Создайте проект «Консольное приложение Win 32».

2 Продумайте структуру программы.

3 Формализуйте алгоритм решения задачи.

4 Выполните русификацию консольного вывода.

5 Выведите на экран видеомонитора результаты работы программы.

6 Оформите отчет по лабораторной работе.

2.2.3 Практическая работа №6

«Пользовательский тип данных. Структуры»

Вариант 1. Создайте структуру с именем PULSAR, содержащую следующие поля: фамилия, год рождения, должность, пол.

Напишите программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из трех элементов типа PULSAR;

- записи упорядочить в алфавитном порядке по фамилии;

- вывод на экран информации о количестве инженеров-пенсионеров (женщины старше 55 лет, мужчины старше 60 лет);

- если таких сотрудников нет, вывод на дисплей соответствующего сообщения.

Вариант 2. Создайте структуру с именем STAR, содержащую следующие поля: фамилия, должность, зарплата, номер телефона.

Напишите программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из трех элементов типа STAR;

- записи упорядочить в алфавитном порядке по фамилии;

- вывод на экран информации о количестве сотрудников с минимальной и максимальной зарплатами;

- если таких сотрудников нет, вывод на дисплей соответствующего сообщения.

Вариант 3. Создайте структуру с именем ORION, содержащую следующие поля: фамилия, год рождения, пол; должность, стаж.

Напишите программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из трех элементов типа ORION;

- записи упорядочить в алфавитном порядке по фамилии;

- вывод на экран информации о количестве пенсионеров (мужчины старше 60 лет, женщины старше 55 лет) и количестве молодых специалистов (младше 25 лет);

- если таких сотрудников нет, вывод на дисплей соответствующего сообщения.

Вариант 4. Создайте структуру с именем UNIVER, содержащую следующие поля: фамилия, год рождения, пол, ученая степень.

Напишите программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из трех элементов типа UNIVER;

- записи упорядочить в алфавитном порядке по фамилии;

- вывод на экран информации о количестве молодых преподавателей, имеющих ученую степень. Удалить преподавателей-пенсионеров (мужчины старше 60 лет, женщины старше 55 лет), не имеющих ученой степени;

- если таких преподавателей нет, вывод на дисплей соответствующего сообщения.

Вариант 5. Создайте структуру с именем STUDENT, содержащую следующие поля: фамилия, номер группы, номер зачетной книжки, успеваемость (массив из четырех элементов).

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из трех структур типа STUDENT;

- записи упорядочить в алфавитном порядке по фамилии;

- вывод на дисплей фамилий и номеров групп для всех студентов, включенных в массив, если средний балл студента больше 3.5;

- если таких студентов нет, вывод соответствующего сообщения.

Вариант 6. Создайте структуру с именем STUDENT, содержащую следующие поля: фамилия, номер группы, паспортные данные, успеваемость (массив из четырех элементов).

Напишите программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из шести структур типа STUDENT;

- записи упорядочить в алфавитном порядке по фамилии;

- вывод на дисплей студентов, имеющих оценки 4 и 5;

- если таких студентов нет, вывод соответствующего сообщения.

Вариант 7. Создайте структуру с именем STUDENT, содержащую следующие поля: фамилия, номер зачетной книжки, номер группы, успеваемость (массив из четырех элементов).

Напишите программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из трех структур типа STUDENT;

- записи упорядочить в алфавитном порядке по фамилии;

- вывод на дисплей фамилий и номеров групп для всех студентов, имеющих хотя бы одну оценку 2;

- если таких студентов нет, вывод соответствующего сообщения.

Вариант 8. Создайте структуру с именем AEROFLOT, содержащую следующие поля: пункт отправления, пункт назначения, номер рейса, тип самолета.

Напишите программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из трех элементов типа AEROFLOT;

- записи упорядочить в алфавитном порядке по пункту назначения;

- вывод на экран номеров рейсов и типов самолетов, вылетающих в пункт назначения, название которого совпало с названием, введенным с клавиатуры;

- если таких рейсов нет, вывод на дисплей соответствующего сообщения.

3 Структурное программирование

3.1 Функции

3.1.1 Виды функций

Функция (function) – это именованная последовательность описаний и операторов, выполняющая законченное действие [2, с. 73].

Объявление функции (function declaration) – задание имени функции, типа возвращаемого значения, круглых скобок и списка параметров.

Определение функции (function definition) – задание заголовка и тела функции. Формат определения функции имеет вид:

[класс] тип имя ([список_параметров]) [throw (исключения)] {тело функции}

Вызов функции (function call) – выражение, значением которого является возвращаемый функцией результат вычисления.

Функция может принимать и возвращать значения.

Функции подразделяются на:

- встроенную;
- рекурсивную;
- перегруженную;
- шаблонную.

Встроенная функция (in-line function) – функция, тело которой встраивается в программу в месте её вызова.

Рекурсивная функция (recursive function) – функция, которая вызывает саму себя либо непосредственно, либо косвенно (через другую функцию).

Перегруженная функция (function overloading) – функция, объявленная два и более раз с одним именем, отличающаяся типом или количеством параметров или тем и другим одновременно.

Шаблонная функция (template function) – функция, которая автоматически перегружает сама себя.

3.1.2 Лабораторная работа №8 «Функции»

Вариант 1. В массиве вещественных чисел определите среднее арифметическое значение элементов, расположенных до наибольшего элемента. Создайте функцию для вычисления среднего значения элементов массива и определения его максимального элемента.

Вариант 2. Из двух целочисленных массивов различной размерности сформируйте общий массив. Упорядочите полученный массив в порядке возрастания его элементов. Найдите минимальные значения заданных массивов. Создайте функции для сортировки массива и определения его минимального значения.

Вариант 3. Найдите наибольший и наименьший элементы вещественного массива. Если таких элементов несколько, определить, сколько их. Создайте функции для определения максимального и минимального элементов в массиве.

Вариант 4. Удалите из массива вещественных чисел элемент, наименее отличающийся от среднего арифметического значения элементов массива. Создайте функции для вычисления среднего значения элементов массива и удаления элемента, наименее отличающегося от среднего арифметического значения.

Вариант 5. Замените минимальный элемент массива вещественных чисел на среднее арифметическое значение элементов массива. Создайте функции для вычисления среднего значения элементов массива и определения его минимального элемента.

Вариант 6. Создайте массив целых положительных чисел. Перепишите простые числа из заданного массива в другой массив. Первый массив упорядочите по возрастанию, второй по убыванию. Создайте функции для определения простого числа и сортировки элементов массива.

Вариант 7. В массиве целых положительных чисел определите количество совершенных чисел. Удалите из массива элементы, равные нулю. Создайте функции определения совершенного числа и удаления элементов массива, равных нулю.

Вариант 8. Сформируйте массив из двух вещественных массивов положительных чисел различной размерности. Найдите наибольший элемент в

сформированном массиве. Создайте функции для определения наибольшего элемента массива и удаления наибольшего элемента массива.

Методические указания

- 1 Создайте проект «Консольное приложение Win 32».
- 2 Продумайте структуру программы.
- 3 Формализуйте алгоритм решения задачи.
- 4 Примените структурный метод программирования.
- 5 Создайте функции и динамические массивы.
- 6 Выведите на экран видеомонитора результаты работы программы.
- 7 Оформите отчет по лабораторной работе.

3.1.3 Лабораторная работа №9 «Перегруженные функции»

Вариант 1. В одномерном массиве, состоящем из n элементов, вычислите:

- 1) максимальное значение элементов массива;
- 2) сумму модулей элементов массива, расположенных после минимального по модулю элемента.

Вариант 2. В одномерном массиве, состоящем из n элементов, вычислите:

- 1) максимальный по модулю элемент массива;
- 2) сумму элементов массива, расположенных между первым и вторым положительными элементами.

Вариант 3. В одномерном массиве, состоящем из n элементов, вычислите:

- 1) среднее арифметическое элементов, расположенных до первого и после последнего нулевых значений;
- 2) среднее квадратическое значение элементов массива.

Вариант 4. В одномерном массиве, состоящем из n элементов, вычислите:

- 1) сумму отрицательных элементов массива;
- 2) произведение элементов массива, расположенных между максимальным и минимальным элементами.

Вариант 5. В одномерном массиве, состоящем из n элементов, вычислите:

- 1) сумму положительных элементов массива;
- 2) произведение элементов массива, расположенных между максимальным по модулю и минимальным по модулю элементами.

Вариант 6. Разработайте программу с перегруженными функциями, выполняющими действия:

- вывод на экран линии в 50 символов '#';
- вывод заданного символа заданное число раз;

- печать 50 символов с возможностью задания символа.

Вариант 7. Разработайте программу, использующую перегруженные функции для вывода расстояния в виде числа футов и числа дюймов. Аргументом функции может являться как структурная переменная, так и переменная типа float.

Вариант 8. В одномерном массиве, состоящем из n элементов, вычислите:

- 1) сумму элементов массива с нечетными номерами;
- 2) сумму элементов массива, расположенных между первым и последним отрицательными элементами.

Методические указания

- 1 Создайте проект «Консольное приложение Win 32».
- 2 Продумайте структуру программы.
- 3 Формализуйте алгоритм решения задачи.
- 4 Примените структурный метод программирования.
- 5 Создайте перегруженные функции и динамические массивы.
- 6 Выведите на экран видеомонитора результаты работы программы.
- 7 Оформите отчет по лабораторной работе.

3.1.4 Лабораторная работа №10 «Шаблонные функции»

Разработайте программу с шаблонными функциями для типов int, float и double. Данные в функции передаются в качестве параметров. Использование глобальных переменных в функциях не допускается.

Вариант 1. Дана целочисленная прямоугольная матрица $[A_{11*13}]$.
Напишите программу, определяющую величины:

- 1) количество элементов матрицы, содержащих нечетные числа;
- 2) максимум элементов, расположенных по периметру прямоугольной матрицы;
- 3) суммы элементов, находящихся между минимальным и максимальным элементами столбцов.

Вариант 2. Дана целочисленная прямоугольная матрица $[A_{10*15}]$.
Напишите программу, определяющую величины:

- 1) количество столбцов, содержащих хотя бы один нулевой элемент;
- 2) номер строки, в которой находится самая длинная серия одинаковых элементов;
- 3) среднее геометрическое значение элементов строк и столбцов матрицы.

Вариант 3. Дана целочисленная прямоугольная матрица $[A_{12*15}]$.
Напишите программу, определяющую величины:

- 1) количество элементов матрицы, меньших величины B ;
- 2) максимальный и минимальный нечетные элементы столбцов матрицы;
- 3) суммы элементов матрицы, находящихся выше побочной диагонали.

Вариант 4. Дана целочисленная прямоугольная матрица $[A_{14*15}]$.

Напишите программу, определяющую величины:

- 1) количество нулевых элементов матрицы;
- 2) максимальный и минимальный нечетные элементы столбцов матрицы;
- 3) суммы элементов матрицы, находящихся выше побочной диагонали.

Вариант 5. Дана целочисленная квадратная матрица $[A_{13*13}]$. Напишите программу, определяющую величины:

- 1) количество отрицательных элементов матрицы;
- 2) максимальный элемент матрицы;
- 3) суммы элементов матрицы, находящихся выше главной диагонали.

Вариант 6. Дана целочисленная прямоугольная матрица $[A_{10*11}]$.

Напишите программу, определяющую величины:

- 1) количество элементов матрицы, больших величины B ;
- 2) минимальный по модулю элемент матрицы;
- 3) суммы элементов, расположенных по периметру матрицы.

Вариант 7. Дана целочисленная квадратная матрица $[A_{11*11}]$. Напишите программу, определяющую величины:

- 1) количество ненулевых элементов матрицы;
- 2) максимальный и минимальный четные элементы столбцов матрицы;
- 3) суммы элементов матрицы, находящихся ниже побочной диагонали.

Вариант 8. Дана целочисленная квадратная матрица $[A_{12*12}]$. Напишите программу, определяющую величины.

- 1) количество элементов матрицы, содержащих четные числа;
- 2) максимум элементов главной диагонали;
- 3) суммы элементов, находящихся между минимальным и максимальным элементами строк.

Методические указания

- 1 Создайте проект «Консольное приложение Win 32».
- 2 Продумайте структуру программы.
- 3 Формализуйте алгоритм решения задачи.
- 4 Примените структурный метод программирования.
- 5 Создайте шаблонные функции и динамические массивы.
- 6 Выведите на экран видеомонитора результаты работы программы.
- 7 Оформите отчет по лабораторной работе.

3.1.5 Практическая работа №7

«Структурное программирование. Шаблонные функции»

Вариант 1. Дана целочисленная квадратная матрица $[A_{12*12}]$. Напишите программу, определяющую величины:

- 1) количество элементов матрицы, содержащих простые числа;

- 2) максимум среди элементов, расположенных выше побочной диагонали;
- 3) средние квадратичные значения четных столбцов и нечетных строк матрицы.

Вариант 2. Дана целочисленная квадратная матрица $[A_{15*15}]$. Напишите программу, определяющую величины:

- 1) количество элементов матрицы, содержащих простые числа;
- 2) минимум среди элементов, расположенных выше главной диагонали матрицы;
- 3) средние квадратичные значения столбцов и строк матрицы.

Вариант 3. Дана целочисленная квадратная матрица $[A_{12*12}]$. Напишите программу, определяющую величины:

- 1) количество элементов матрицы, содержащих простые числа;
- 2) минимум среди элементов, расположенных ниже главной диагонали матрицы;
- 3) средние квадратичные значения нечетных столбцов и четных строк матрицы.

Вариант 4. Дана целочисленная квадратная матрица $[A_{11*11}]$. Напишите программу, определяющую величины:

- 1) количество элементов матрицы, содержащих простые числа;
- 2) максимум среди элементов, расположенных ниже главной диагонали матрицы;
- 3) средние арифметические значения столбцов и строк матрицы.

Вариант 5. Дана целочисленная квадратная матрица $[A_{15*15}]$. Напишите программу, определяющую величины:

- 1) количество элементов матрицы, содержащих простые числа;
- 2) минимум среди элементов, расположенных ниже побочной диагонали матрицы;
- 3) средние арифметические значения четных столбцов и нечетных строк матрицы.

Вариант 6. Дана целочисленная квадратная матрица $[A_{16*16}]$. Напишите программу, определяющую величины:

- 1) количество элементов матрицы, содержащих нулевые значения;
- 2) максимум среди элементов, расположенных выше побочной диагонали матрицы;
- 3) суммы положительных элементов столбцов и строк матрицы.

Вариант 7. Дана целочисленная прямоугольная матрица $[A_{8*10}]$. Напишите программу, определяющую величины:

- 1) количество строк, не содержащих ни одного нулевого элемента;
- 2) максимальное значение из чисел, встречающихся в заданной матрице более одного раза;
- 3) суммы элементов строк и столбцов матрицы.

Вариант 8. Дана целочисленная прямоугольная матрица $[A_{9*16}]$. Напишите программу, определяющую величины:

- 1) суммы элементов в строках, которые содержат хотя бы один отрицательный элемент;
- 2) минимальный элемент матрицы;
- 3) среднее арифметическое значение четных элементов строк и столбцов матрицы.

4 Объектно-ориентированное программирование

4.1 Классы

4.1.1 Теоретическое обоснование

Класс (class) – абстрактный тип данных, определяемый пользователем.

Формат описания класса имеет вид:

```
class <имя> {[private:] <описание закрытых элементов>
           public: <описание открытых элементов>};
```

Конструктор (constructor) – специальный метод класса, имеющий имя, совпадающее с именем класса, и предназначенный для инициализации полей класса. Конструктор имеет свойства:

- не возвращает значения;
- может иметь или не иметь параметров;
- не наследуется;
- не употребляется со служебными словами const, volatile, static и virtual.

Деструктор (destructor) – специальный метод класса, имеющий имя, совпадающее с именем класса, символ тильды перед именем класса и предназначенный для разрушения объекта. Деструктор имеет свойства:

- не возвращает значения;
- не имеет параметров;
- не наследуется;
- не употребляется со служебными словами const, volatile, static;
- может быть виртуальным.

Объект (object) – переменная типа имя класса.

4.1.2 Лабораторная работа №11 «Классы. Конструкторы и деструкторы»

Вариант 1. Создайте класс **Triangle** (треугольник) для представления треугольника. Поля данных должны включать углы и стороны. Требуется реализовать операции получения и изменения полей данных, вычисления площади, вычисления периметра, вычисления высот и определения вида треугольника (равносторонний, равнобедренный или прямоугольный).

Вариант 2. Создайте класс **Complex** (комплекс) для работы с комплексными числами. Комплексное число представляется парой чисел (a, b),

где a – действительная часть, b – мнимая часть. Требуется реализовать операции сложения **add**: $(a, b) + (c, d) = (a+b, c+d)$, вычитания **sub**: $(a, b) - (c, d) = (a-b, c-d)$, умножения **mul**: $(a, b) * (c, d) = (a*c-b*d, a*d+b*c)$, деления **div**: $(a, b) / (c, d) = (a*c+b*d, b*c-a*d) / (c*c+d*d)$, сравнение **equ**: $(a, b) = (c, d)$, если $(a=c)$ и $(b=d)$, сопряженное число **conj**: $\text{conj}(a, b) = (a, -b)$.

Вариант 3. Создайте класс **Account** (счет), представляющий банковский счет. В классе должны быть четыре поля: фамилия владельца, номер счета, процент начисления и сумма в рублях. Открытие нового счета выполняется операцией инициализации. Необходимо выполнить следующие операции:

- сменить владельца счета;
- снять некоторую сумму денег со счета;
- положить деньги на счет;
- вычислить проценты;
- перевести сумму в доллары;
- перевести сумму в евро;
- получить сумму прописью (преобразовать в числительное).

Вариант 4. Создайте класс **Goods** (товары). В классе должны быть представлены поля: наименование товара, дата оформления, цена товара, количество единиц товара, номер накладной, по которой товар поступил на склад. Реализовать методы изменения цены товара, изменения количества товара (увеличения и уменьшения), вычисления стоимости товара. Метод **toString()** должен выдавать в виде строки стоимость товара.

Вариант 5. Создайте класс **Payment** (зарплата). В классе должны быть представлены поля: фамилия, имя, отчество, оклад, год поступления на работу, процент надбавки, подоходный налог, количество отработанных дней в месяце, количество рабочих дней в месяце, начисленная и удержанная суммы. Реализовать методы:

- вычисления начисленной суммы;
- вычисления удержанной суммы;
- вычисления суммы, выдаваемой на руки;
- вычисления стажа.

Стаж вычисляется как полное количество лет, прошедших от года поступления на работу до текущего года. Начисления представляют собой сумму, начисленную за отработанные дни, и надбавки, то есть доли от первой суммы. Удержания представляют собой отчисления в пенсионный фонд (1% от начисленной суммы) и подоходный налог. Подоходный налог составляет 13% от начисленной суммы без отчислений в пенсионный фонд.

Вариант 6. Создайте класс **Bill** (счет), представляющий собой разовый платеж за телефонный разговор. Класс должен включать в себя поля: номер телефона, тариф за минуту разговора, скидка (в процентах), время разговора (в минутах) и сумма к оплате. Реализуйте метод вычисления суммы к оплате. В программе продемонстрировать создание, инициализацию и обработку массива объектов типа **Bill** с различными исходными данными для вычисления сумм к оплате. Вычислите общую сумму к оплате.

Вариант 7. Создайте класс **AvtoVlad**, хранящий информацию о владельце автомобиля: имя, номер автомобиля, номер техпаспорта, дату рождения, телефон, отделение регистрации ГИБДД. Доступ к данным класса организуйте посредством соответствующих методов. Разработайте программу, в которой создается массив объектов данного класса. В программе организуйте:

- ввод данных с клавиатуры;
- вывод данных массива на экран в виде таблицы;
- поиск и вывод информации о владельцах автомобилей, зарегистрированных в отделении ГИБДД;
- поиск и вывод информации об автовладельце по номеру автомобиля.

Вариант 8. Создайте класс **Abonent**, хранящий информацию о телефонном абоненте: фамилию, имя, дату рождения, адрес и телефонный номер. Доступ к полям класса организуйте посредством соответствующих методов. Разработайте программу, в которой создается массив объектов данного класса. В программе организуйте:

- ввод данных с клавиатуры;
- вывод данных массива на экран в виде таблицы;
- поиск и вывод информации о телефонных абонентах по номеру телефона.

Методические указания

- 1 Создайте проект «Консольное приложение Win 32».
- 2 Продумайте структуру программы.
- 3 Разработайте диаграмму классов.
- 4 Формализуйте алгоритм решения задачи.
- 5 Примените объектно-ориентированный метод программирования.
- 6 Создайте конструкторы и деструкторы.
- 7 Создайте экземпляр класса и массив объектов.
- 8 Выведите на экран видеомонитора результаты работы программы.
- 9 Оформите отчет по лабораторной работе.

4.1.3 Практическая работа №8 «Инкапсуляция. Конструкторы и деструкторы»

Вариант 1. Создайте класс типа дата с полями: день (1-31), месяц (1-12), год (целое число). Класс имеет конструктор; функции-члены установки дня, месяца и года; функции-члены получения дня, месяца и года, а также две функции-члены печати: печать по шаблону: «5 декабря 2015 года» и «05.12.2015». Функции-члены установки полей класса должны проверять корректность задаваемых параметров.

Вариант 2. Создайте класс типа время с полями: час (0-23), минуты (0-59), секунды (0-59). Класс имеет конструктор, функции-члены установки времени и функции-члены печати. Функции-члены установки времени

включают методы получения часа, минуты и секунды. Функции-члены печати включают методы печати по шаблонам «16 часов 18 минут 3 секунды» и «4 p.m. 18 минут 3 секунды». Функции-члены установки полей класса должны проверять корректность задаваемых параметров.

Вариант 3. Создайте класс типа прямоугольник. Поля – высота и ширина. Функции-члены вычисляют площадь, периметр, устанавливают поля и возвращают значения. Функции-члены установки полей класса должны проверять корректность задаваемых параметров. Создайте функцию печати.

Вариант 4. Разработайте класс, представляющий треугольник, который характеризуется длинами сторон. Для класса определите методы, вычисляющие периметр и площадь треугольника. Разработайте программу, демонстрирующую работу с данным классом.

Вариант 5. Создайте класс типа квадрат. Поле – сторона квадрата. Функции-члены вычисляют площадь, периметр квадрата, устанавливают поле и возвращают значения. Функция-член установки поля класса должна проверять корректность задаваемых параметров. Создайте функцию печати.

Вариант 6. Создайте класс типа окружность. Поле – радиус окружности. Функции-члены вычисляют площадь, длину окружности, устанавливают поле и возвращают значения. Функция-член установки поля класса должна проверять корректность задаваемых параметров. Создайте функцию печати.

Вариант 7. Создайте класс типа прямоугольник. Поля – высота и ширина прямоугольника. Функции-члены вычисляют площадь, периметр, устанавливают поля и возвращают значения. Функции-члены установки полей класса должны проверять корректность задаваемых параметров. Создайте функцию печати.

Вариант 8. Создайте класс типа круг. Поле – радиус круга. Функции-члены вычисляют площадь, длину круга, устанавливают поле и возвращают значения. Функция-член установки поля класса должна проверять корректность задаваемых параметров. Создайте функцию печати.

4.2 Наследование классов

4.2.1 Базовые и производные классы

Наследование (inheritance) – механизм языка C++, с помощью которого производный класс копирует поля и методы базового класса.

Базовый класс (base class) – класс, от которого наследуются поля и методы класса.

Производный класс (derived class) – класс, объекты которого наследуют данные и методы базового класса.

Определение производного класса имеет вид:

```
class имя: [private|protected|public] базовый класс {тело класса};
```

Доступ к членам базового класса из производного класса производится посредством ключей доступа: private, protected и public.

Одиночное (простое) наследование – наследование полей и методов производным классом от одного базового класса.

Множественное (сложное) наследование – иерархическая структура, в которой производный класс имеет два или более базовых класса.

Объявление производного класса при множественном наследовании имеет вид:

```
class      имя_производного_класса:      [public|protected|private]
имя_базового_класса1,  [public|protected|private]  имя_базового_класса2,...,
[public|protected|private]  имя_базового_классаN  {/* тело производного
класса*/};
```

4.2.2 Лабораторная работа №12 «Наследование»

Вариант 1. Издательская компания торгует книгами и аудиозаписями книг. Создайте класс **Publication**, в котором хранятся название (строка) и цена (тип float) книги. От этого класса наследуются два класса **Book**, который содержит информацию о количестве страниц в книге (тип int), и **Type**, который содержит время записи книги в минутах (тип float). В каждом из этих трех классов должен быть метод `getdata()`, через который можно получать данные от пользователя с клавиатуры, и `putdata()`, предназначенный для вывода этих данных.

Напишите функцию `main()` программы для проверки классов **Book** и **Type**. Создайте их объекты в программе и запросите пользователя ввести и вывести данные с использованием методов `getdata()` и `putdata()`.

Вариант 2. Создайте базовый класс **Car** (машина), характеризуемый торговой маркой (строка), числом цилиндров, мощностью. Определите методы переназначения и изменения мощности. Создайте производный класс **Lorry** (грузовик), характеризуемый также грузоподъемностью кузова. Определите функции переназначения марки и изменения грузоподъемности.

Вариант 3. Создайте базовый класс **Complex** (комплексное число) для реализации комплексных чисел в алгебраической форме и основных операций с ними: сложения, вычитания, умножения и деления. Создайте производный класс, в котором определите операции вычисления модуля комплексного числа и комплексно сопряженного.

Вариант 4. Создайте класс **Liquid** (жидкость), имеющий поля названия и плотности. Определите методы переназначения и изменения плотности. Создайте производный класс **Alcohol** (спирт), имеющий крепость. Определите методы переназначения и изменения крепости.

Вариант 5. Создайте базовый класс **Complex** (комплексное число) для реализации комплексных чисел в тригонометрической форме и основных операций с ними: сложения, вычитания, умножения и деления. Создайте производный класс, в котором определите операции вычисления действительной и мнимой частей комплексного числа, а также комплексно сопряженного.

Вариант 6. Создайте класс **Man** (человек) с полями: имя, возраст, пол и вес. Определите методы переназначения имени, изменения возраста и изменения веса. Создайте производный класс **Student**, имеющий поле года обучения. Определите методы переназначения и увеличения года обучения.

Вариант 7. Создайте базовый класс **Complex** (комплексное число) для реализации комплексных чисел в алгебраической форме и основных операций с ними: сложения, вычитания, умножения и деления (в том числе и на действительные числа). Создайте производный класс, в котором определите метод

$$\exp(z) = 1 + z + \frac{z^2}{2!} + \dots + \frac{z^n}{n!} + \dots$$

Вариант 8. Создайте класс **Triangle** (треугольник) с полями-сторонами. Определите методы изменения сторон, вычисления углов, вычисления периметра. Создайте производный класс **Equilateral** (равносторонний), имеющий поле площади. Определите метод вычисления площади.

Методические указания

- 1 Создайте проект «Консольное приложение Win 32».
- 2 Продумайте структуру программы.
- 3 Разработайте диаграмму классов.
- 4 Формализуйте алгоритм решения задачи.
- 5 Примените объектно-ориентированный метод программирования.
- 6 Создайте конструкторы и деструкторы.
- 7 Создайте базовый и производный классы.
- 8 Выведите на экран видеомонитора результаты работы программы.
- 9 Оформите отчет по лабораторной работе.

4.2.3 Практическая работа №9 «Одиночное наследование»

Вариант 1. Создайте класс студент, имеющий имя (указатель на строку), курс и идентификационный номер. Определите конструкторы, деструктор и функцию печати. Создайте public-производный класс студент-дипломник, имеющий тему диплома. Определите конструкторы по умолчанию и с разным числом параметров, деструкторы, функцию печати. Определите функции переназначения названия диплома и идентификационного номера.

Вариант 2. Создайте класс животное, имеющий классификацию (указатель на строку), число конечностей, число потомков. Определите конструкторы, деструктор и функцию печати. Создайте public-производный класс домашнее животное, имеющий кличку. Определите конструкторы по умолчанию и с разным числом параметров, деструкторы, функцию печати. Определите функции переназначения названия клички и числа потомков.

Вариант 3. Создайте класс машина, имеющий марку (указатель на строку), число цилиндров, мощность. Определите конструкторы, деструктор и

функцию печати. Создайте public-производный класс грузовик, имеющий грузоподъемность кузова. Определите конструкторы по умолчанию и с разным числом параметров, деструкторы, функцию печати. Определите функции переназначения марки и грузоподъемности.

Вариант 4. Создайте класс карта, имеющий ранг и масть. Карту можно перевернуть и открыть. Создайте класс колода карт, содержащий карты. Создайте два производных класса от колоды карт, в одном карты могут доставаться только по порядку, в другом – вытаскиваться произвольно.

Вариант 5. Создайте класс точка, который имеет координаты. Класс эллипсов и класс окружностей. Определите иерархию типов. Определите функции печати, конструкторы, деструкторы, вычисление площади.

Вариант 6. Создайте классы четырехугольник, квадрат и прямоугольник. Создайте из них иерархию. Определите функции печати, конструкторы и деструкторы, вычисления площади и периметра.

Вариант 7. Создайте класс транспортные средства, автомобиль, грузовик, пароход и самолет. Создайте из них иерархию. Определите функции печати, конструкторы и деструкторы.

Вариант 8. Создайте класс жидкость, имеющий название (указатель на строку), плотность. Определите конструкторы, деструктор и функцию печати. Создайте public-производный класс спиртные напитки, имеющий крепость. Определите конструкторы по умолчанию и с разным числом параметров, деструкторы, функцию печати. Определите функции переназначения плотности и крепости.

4.2.4 Практическая работа №10

«Множественное наследование. Виртуальные базовые классы»

Вариант 1. Задание 1. Создайте иерархию классов, описывающую студента, отца семейства и студента – отца семейства. Классы должны иметь конструкторы, включая конструктор копирования, виртуальные деструкторы, перегруженные функции вывода в поток и ввода в поток.

Задание 2. Создайте иерархию классов, описывающую человека, студента, отца семейства и студента – отца семейства. Используйте виртуальные базовые классы.

Вариант 2. Задание 1. Создайте иерархию классов, описывающую работника, отца-семейства и работника – отца семейства. Классы должны иметь конструкторы, включая конструктор копирования, виртуальные деструкторы, перегруженные функции вывода в поток и ввода в поток.

Задание 2. Создайте иерархию классов, описывающую человека, работника, отца семейства и работника – отца семейства. Используйте виртуальные базовые классы.

Вариант 3. Задание 1. Создайте иерархию классов – файл для чтения, файл для записи и файл для чтения и записи. Классы должны иметь

конструкторы, включая конструктор копирования, виртуальные деструкторы, перегруженные функции вывода в поток и ввода в поток.

Задание 2. Создайте иерархию классов, описывающую файл, файл для чтения, файл для записи и файл для чтения и записи. Используйте виртуальные базовые классы.

Вариант 4. Задание 1. Создайте иерархию классов, описывающую работника, женщину и работника – мать семейства. Классы должны иметь конструкторы, включая конструктор копирования, виртуальные деструкторы, перегруженные функции вывода в поток и ввода в поток.

Задание 2. Создайте иерархию классов, описывающую человека, работника, женщину и работника – мать семейства. Используйте виртуальные базовые классы.

Вариант 5. Задание 1. Создайте иерархию классов, описывающую операционную систему, прикладное программное обеспечение и Windows NT как операционную систему и прикладное программное обеспечение. Классы должны иметь конструкторы, включая конструктор копирования, виртуальные деструкторы, перегруженные функции вывода в поток и ввода в поток.

Задание 2. Создайте иерархию классов, описывающую программное обеспечение, операционную систему, прикладное программное обеспечение и Windows NT как операционную систему и прикладное программное обеспечение. Используйте виртуальные базовые классы.

Вариант 6. Задание 1. Создайте иерархию классов, описывающую данные-сигнал, данные-результат обработки и данные как результат обработки сигнала и как сигнал. Классы должны иметь конструкторы, включая конструктор копирования, виртуальные деструкторы, перегруженные функции вывода в поток и ввода в поток.

Задание 2. Создайте иерархию классов, описывающую данные-сигнал, данные-результат обработки и данные как результат обработки сигнала и как сигнал. Используйте виртуальные базовые классы.

Вариант 7. Задание 1. Создайте иерархию классов – море, залив и бухта. Классы должны иметь конструкторы, включая конструктор копирования, виртуальные деструкторы, перегруженные функции вывода в поток и ввода в поток.

Задание 2. Создайте иерархию классов – соленая вода, море, залив и бухта. Используйте виртуальные базовые классы.

Вариант 8. Задание 1. Создайте иерархию типов – корабль, пассажирский транспорт и пассажирский корабль. Классы должны иметь конструкторы, включая конструктор копирования, виртуальные деструкторы, перегруженные функции вывода в поток и ввода в поток.

Задание 2. Создайте иерархию классов – транспорт, корабль, пассажирский транспорт и пассажирский корабль. Используйте виртуальные базовые классы.

4.3 Перегрузка операторов

4.3.1 Переопределение операторов

Перегрузка операторов (operator overloading) – переопределение стандартного оператора языка C++ для работы с типами данных, созданных пользователем.

Запрещено перегружать операторы:

- оператор выбора члена класса (точка) .;
- оператор селектора члена класса (точка-звездочка).*;
- оператор разрешения области видимости ::;
- тернарный оператор ?::;
- статический оператор вычисления длины оператора в байтах sizeof;
- оператор превращения в строку #;
- оператор конкатенации ##.

Перегрузка операторов осуществляется при помощи операторной функции (operator function).

Формат операторной функции имеет вид:

тип функции имя_класса :: operator знак_оператора ()
{ /* Тело операторной функции */ }

Для перегрузки бинарных операторов используется формат:

тип функции имя_класса :: operator знак_оператора (описание параметра)
{ /* Тело операторной функции */ }

Операторная функция возвращает объект класса.

4.3.2 Лабораторная работа №13 «Перегрузка операций»

Вариант 1. Создайте класс **Complex** (комплексное число) в алгебраической форме $z = x + i * y$, включающий два поля класса: действительную часть (x) и мнимую часть (y) числа. Реализуйте методы вычисления и вывода корня комплексного числа. Перегрузите операции сложения, вычитания, деления и умножения комплексных чисел.

Вариант 2. Создайте класс **Fraction** (обыкновенная дробь), включающий поля: числитель и знаменатель. Реализуйте методы сокращения дроби и вывода дроби. Перегрузите операции сложения, вычитания, деления и умножения дробей.

Вариант 3. Создайте класс **Vector** (вектор), включающий поля: координаты вектора. Реализуйте методы вывода вектора и вычисления длины вектора. Перегрузите операции сложения (+), скалярного (%) и векторного (*) произведения векторов.

Вариант 4. Создайте класс **Matrix** (матрица), включающий поля: количество строк, количество столбцов, элементы матрицы. Реализуйте методы вывода матрицы и вычисления определителя. Перегрузите операции сложения, вычитания, деления, умножения матриц, умножения матрицы на число.

Вариант 5. Создайте класс **StraightLine** (прямая линия), включающий координаты двух точек (x_1, y_1) и (x_2, y_2) . Реализуйте методы вывода уравнения прямой $y = a * x + b$. Перегрузите операции проверки параллельности прямых (||) и определения угла между двумя прямыми (%).

Вариант 6. Создайте класс **Complex** (комплексное число) в алгебраической форме $z = x + i * y$, включающий два поля класса: действительную часть (x) и мнимую часть (y) числа. Реализуйте методы вычисления модуля комплексного числа, возведения комплексного числа в степень и вывода комплексного числа. Перегрузите операции сложения, вычитания, деления и умножения комплексных чисел.

Вариант 7. Создайте класс **Fraction** (обыкновенная дробь), включающий поля: числитель и знаменатель. Реализуйте методы определения обратной дроби и вывода дроби. Перегрузите операции сложения, вычитания и умножения дробей.

Вариант 8. Создайте класс **Vector** (вектор), включающий поля – координаты вектора. Реализуйте методы определения направляющих косинусов вектора и вывода всех характеристик вектора. Перегрузите операции сложения (+), скалярного (%) и векторного (*) произведения векторов.

Методические указания

- 1 Создайте проект «Консольное приложение Win 32».
- 2 Продумайте структуру программы.
- 3 Разработайте диаграмму классов.
- 4 Формализуйте алгоритм решения задачи.
- 5 Примените объектно-ориентированный метод программирования.
- 6 Создайте операторную функцию.
- 7 Выведите на экран видеомонитора результаты работы программы.
- 8 Оформите отчет по лабораторной работе.

4.3.3 Практическая работа №11 «Перегрузка операторов»

Вариант 1. Создайте класс целых чисел. Определите оператор ++ как функцию-член и -- как дружественную функцию.

Вариант 2. Создайте класс целых чисел. Определите оператор - как функцию-член и + как дружественную функцию.

Вариант 3. Создайте класс целых чисел. Определите оператор -- как функцию-член и ++ как дружественную функцию.

Вариант 4. Создайте класс целых чисел (long). Определите оператор ++ как функцию-член и -- как дружественную функцию.

Вариант 5. Создайте класс вещественных чисел. Определите оператор ++ как функцию-член и -- как дружественную функцию.

Вариант 6. Создайте класс целых чисел (`long`). Определите оператор `-` как функцию-член, `+` как дружественную функцию, оператор присваивания и операторы сравнений.

Вариант 7. Создайте класс координат. Определите оператор `+` как функцию-член и как дружественную функцию. Сложите и вычтите координаты друг с другом и с числом. Присвойте координаты, сравните координаты (`==`, `!=`).

Вариант 8. Создайте класс вещественных чисел (`double`). Определите оператор `++` как функцию-член и `--` как дружественную функцию.

4.4 Полиморфизм и виртуальные функции

4.4.1 Виртуальные функции

Виртуальная функция (`virtual function`) – функция-член, объявленная в базовом классе и переопределенная в производном классе.

Полиморфный класс – класс, содержащий хотя бы одну виртуальную функцию.

Для создания виртуальной функции применяется ключевое слово `virtual`.

Виртуальные функции обеспечивают механизм динамического связывания, который работает во время выполнения программы.

Чистая виртуальная функция (`pure function`) – виртуальная функция, не имеющая определения тела.

Абстрактный класс (`abstract class`) – класс, содержащий хотя бы одну чистую виртуальную функцию [5, с. 240].

Деструктор класса объявляется виртуальным, если доступ к динамическому объекту производного класса выполняется через указатель базового класса.

4.4.2 Лабораторная работа №14 «Абстрактные классы»

Вариант 1. Создайте абстрактный класс **Figure** (геометрическая фигура), в котором определены две чисто виртуальные функции: вычисление площади и периметра. Создайте иерархию классов, которые наследуются от данного класса. Иерархия классов должна включать следующие классы: **Triangle** (треугольник), **Circle** (окружность), **Parallelogram** (параллелограмм), **Rectangle** (прямоугольник), **Square** (квадрат). Разработайте программу, в которой множество геометрических фигур хранится в виде массива. Организуйте вывод на экран значений периметра и площади для каждой геометрической фигуры.

Вариант 2. Создайте абстрактный базовый класс **Pet** (домашний питомец), в котором определено поле, соответствующее имени питомца. Создайте производные классы **Cat** (кошка), **Dog** (собака). В программе

определите массив указателей на объекты базового класса, которые проинициализированы объектами производных классов.

Вариант 3. Создайте класс **Point** (точка на плоскости), характеризующийся двумя координатами. Создайте производный класс **ColoredPoint** (цветная точка), в котором добавьте поле, отвечающее за цвет точки. Разработайте программу, в которой продемонстрирован доступ к методам производного класса **ColoredPoint** с помощью указателя на базовый класс **Point**.

Вариант 4. Создайте абстрактный базовый класс **Number** (число) с виртуальными методами: арифметическими операциями сложения, вычитания, умножения, деления, возведения в степень, получения остатка от деления. Создайте производные классы **Integer** (целое) и **Real** (действительное).

Вариант 5. Создайте абстрактный базовый класс **Function** (функция) с виртуальными методами вычисления значения функции $y = f(x)$ в заданной точке x и вывода результата на экран. Определите производные классы **CubePolinom**, **Hyperbola** с собственными функциями вычисления y в зависимости от входного параметра x . Уравнение полинома имеет вид $y = a * x^3 + b * x^2 + c * x + d$. Уравнение гиперболы имеет вид $x^2/a^2 - y^2/b^2 = 1$.

Вариант 6. Создайте абстрактный базовый класс **Integer** (целое) с виртуальными арифметическими операциями и функцией вывода на экран. Определите производные классы **Decimal** (десятичное) и **Binary** (двоичное), реализующие собственные арифметические операции и функцию вывода на экран. Число представляется массивом, каждый элемент которого цифра.

Вариант 7. Издательская компания продает книги и аудиоверсии печатной продукции. Создайте класс **Publication** (публикация), хранящий название (строка) и цену (тип float) публикации. Создайте два порожденных класса: **Book** (книга), в котором происходит изменение счетчика страниц (тип int), и **Tape** (лента), в котором происходит изменение счетчика записанных на кассету минут. Каждый из классов должен иметь метод **getdata()**, запрашивающий информацию у пользователя, и **putdata()** для вывода данных на экран.

Вариант 8. Создайте абстрактный базовый класс **Series** (прогрессия) с виртуальными функциями вычисления j элемента прогрессии и суммы прогрессии. Определите производные классы: **Linear** (арифметическая) и **Exponential** (геометрическая). Арифметическая прогрессия задается математическим выражением $a_j = a_0 + j * d, j = 0, 1, 2, \dots$. Сумма арифметической прогрессии определяется по формуле $s_n = (n + 1) * (a_0 + a_n) / 2$. Геометрическая прогрессия описывается математическим выражением $a_j = a_0 * r^j, j = 0, 1, 2, \dots$. Сумма геометрической прогрессии рассчитывается по формуле $s_n = (n + 1) * (a_0 + a_n) / 2$.

Методические указания

- 1 Создайте проект «Консольное приложение Win 32».
- 2 Продумайте структуру программы.
- 3 Разработайте диаграмму классов.
- 4 Примените объектно-ориентированный метод программирования.
- 5 Создайте абстрактный класс и объявите чистые виртуальные функции.
- 6 Определите чистые виртуальные функции в производных классах.
- 7 Продемонстрируйте варианты вызова виртуальных функций с помощью указателей на базовый класс.
- 8 Создайте функцию вывода, получающую параметры базового класса по ссылке и демонстрирующую виртуальный вызов.
- 9 Формализуйте алгоритм решения задачи.
- 10 Выведите на экран видеомонитора результаты работы программы.
- 11 Оформите отчет по лабораторной работе.

4.4.3 Практическая работа №12 «Виртуальные функции»

Вариант 1. Создайте абстрактный базовый класс с виртуальной функцией площадь. Создайте производные классы: прямоугольник, круг, прямоугольный треугольник, трапеция – со своими функциями площади. Для проверки определите массив ссылок на абстрактный класс, которым присваиваются адреса различных объектов. Площадь трапеции определяется по формуле $S=(a+b)*h/2$.

Вариант 2. Создайте абстрактный класс с виртуальной функцией норма. Создайте производные классы: комплексные числа, вектор из 10 элементов, матрица (2x2). Определите функции нормы для комплексных чисел – модуль в квадрате, для вектора – корень квадратный из суммы элементов по модулю, для матрицы – максимальное значение по модулю.

Вариант 3. Создайте абстрактный класс (кривые) вычисления координаты y для некоторой x . Создайте производные классы: прямая, эллипс, гипербола – со своими функциями вычисления y в зависимости от входного параметра x .

Уравнение прямой имеет вид $y=ax+b$.

Уравнение эллипса имеет вид $x^2/a^2+y^2/b^2=1$.

Уравнение гиперболы имеет вид $x^2/a^2-y^2/b^2=1$.

Вариант 4. Создайте абстрактный базовый класс с виртуальной функцией сумма прогрессии. Создайте производные классы: арифметическая прогрессия и геометрическая прогрессия. Каждый класс имеет два поля типа `double`. Первое – первый член прогрессии, второе (`double`) – постоянная разность (для арифметической) и постоянное отношение (для геометрической). Определите функцию вычисления суммы, где параметром является количество элементов прогрессии.

Арифметическая прогрессия определяется математическим выражением $a_j = a_0 + j * d, j = 0, 1, 2, \dots$

Сумма арифметической прогрессии определяется математическим выражением $s_n = (n+1) * (a_0 + a_n) / 2$.

Геометрическая прогрессия определяется математическим выражением $a_j = a_0 * r^j, j = 0, 1, 2, \dots$

Сумма геометрической прогрессии определяется математическим выражением $s_n = (a_0 - a_n * r) / (1 - r)$.

Вариант 5. Создайте абстрактный класс млекопитающие. Определите производные классы – животные и люди. У животных определите производные классы – собака и корова. Определите виртуальные функции описания человека, собаки и коровы.

Вариант 6. Создайте базовый класс фигура, и производные классы круг, прямоугольник, трапеция. Определите виртуальные функции площадь, периметр и вывод на печать.

Вариант 7. Создайте базовый класс работник и производные классы: служащий с почасовой оплатой, служащий в штате и служащий с процентной ставкой. Определите функцию начисления зарплаты.

Вариант 8. Создайте абстрактный базовый класс с виртуальной функцией площадь поверхности. Создайте производные классы: параллелепипед, тетраэдр, шар – со своими функциями площади поверхности. Для проверки определите массив ссылок на абстрактный класс, которым присваиваются адреса различных объектов.

Площадь поверхности параллелепипеда определяется по формуле $S = 6 * x * y$. Площадь поверхности шара по формуле $S = 4 * \pi * R^2$. Площадь поверхности тетраэдра по формуле $S = a^2 * \sqrt{3}$, где а – ребро тетраэдра.

ЗАКЛЮЧЕНИЕ

Объектно-ориентированный язык программирования С++ создан для разработки высокопроизводительного программного обеспечения. Он обеспечивает концептуальный фундамент, на котором опираются другие языки программирования и современные средства обработки данных. Потомками языка С++ являются языки С# и Java.

Язык С++ предназначен для профессионального программирования. Сегодня быть профессиональным программистом высокого класса означает быть компетентным в С++.

Методические указания к выполнению лабораторных и практических работ позволяют студентам закрепить теоретический материал по дисциплине «Основы программирования» и приобрести практические навыки в разработке приложений на языке С++.

СПИСОК ЛИТЕРАТУРЫ

- 1 Довбуш Г. Ф., Хомоненко А. Д. Visual C++ на примерах. – СПб. : БХВ-Петербург, 2007. – 528 с.
- 2 Павловская Т. А. C/C++. Программирование на языке высокого уровня. – СПб. : Питер, 2006. – 461 с.
- 3 Эллис М., Строуструп Б. Справочное руководство по языку программирования C++ с комментариями. – М. : Мир, 1992. – 448 с.
- 4 Глушаков С. В., Дуравкина Т. В. Программирование на C++. – М. : АСТ, 2008. – 685 с.
- 5 Лаптев В. В. C++. Объектно-ориентированное программирование : учебное пособие. – СПб. : Питер, 2008. – 464 с.

Семахин Андрей Михайлович

ОСНОВЫ ПРОГРАММИРОВАНИЯ

Методические указания
к выполнению лабораторных и практических работ
для студентов направления подготовки 09.03.04
«Программная инженерия»,
10.05.03 «Информационная безопасность автоматизированных систем»
Часть 1

Редактор О. Г. Арефьева

Подписано в печать	Формат 60x84 1/16	Бумага 65 г/м ²
Печать цифровая	Усл. печ. л. 3,25	Уч.-изд. л. 3,25
Заказ	Тираж 25	Не для продажи

РИЦ Курганского государственного университета.
640000, г. Курган, ул. Советская, 63/4.
Курганский государственный университет.