

Министерство образования и науки Российской Федерации

Амурский государственный университет

Т.А. Галаган

ЯЗЫК C++. СБОРНИК ЗАДАЧ И ТЕСТОВЫХ ЗАДАНИЙ

Учебное пособие

Благовещенск
Издательство АмГУ

2014

ББК 32.973-018.1 я73

Г15

Рекомендовано

учебно-методическим советом университета

Рецензент:

А.В. Рыженко, доцент кафедры математического анализа и моделирования АмГУ, канд. техн. наук

Галаган Т.А.

Г15 Язык C++. Сборник задач и тестовых заданий: Учебное пособие / Т.А. Галаган. – Благовещенск: Изд-во АмГУ, 2014. – 112 с.

Пособие предназначено не только для проведения занятий по дисциплине «Программирование», но и для обеспечения самостоятельной работы студентов по данной дисциплине. Оно посвящено изучению основ популярного языка программирования высокого уровня C++, формированию устойчивых навыков создания программ. Уделено внимание как структурному и модульному подходу программирования, так и объектно-ориентированному.

Для студентов направлений подготовки 230100.62 «Информатика и вычислительная техника», 230400.62 «Информационные системы и технологии», 230700.62 – «Прикладная информатика».

ББК 32.973-018.1 я73

©Амурский государственный университет, 2014

Пособие предназначено для изучения языка программирования C++ на практических занятиях, лабораторных работах, а также может служить обеспечением самостоятельной работы студентов по дисциплине «Программирование».

Несмотря на существование множества языков программирования, вот уже несколько десятилетий C++ не теряет своей актуальности и служит устойчивой первой ступеней в изучении основ программирования. Зная C++, студенту в дальнейшем достаточно просто самостоятельно изучить C#, JavaScript, PHP и др. На его основе легко усваиваются различные парадигмы программирования и алгоритмы разработки приложений.

Пособие состоит из двух частей. В первой из них даются задания, результатами выполнения которых служат программы на языке C++. Все задания разбиты на темы, последовательность которых позволяет постепенно входить в особенности языка C++, совершенствовать алгоритмы разрабатываемых программ.

Во второй части пособия содержатся авторские тестовые задания, позволяющие студентам самостоятельно проверять уровень освоения теоретического материала и практических навыков создания программ, заострять внимание на важных моментах, развивать внимательность и навыки понимания текстов программ, составленных другими программистами.

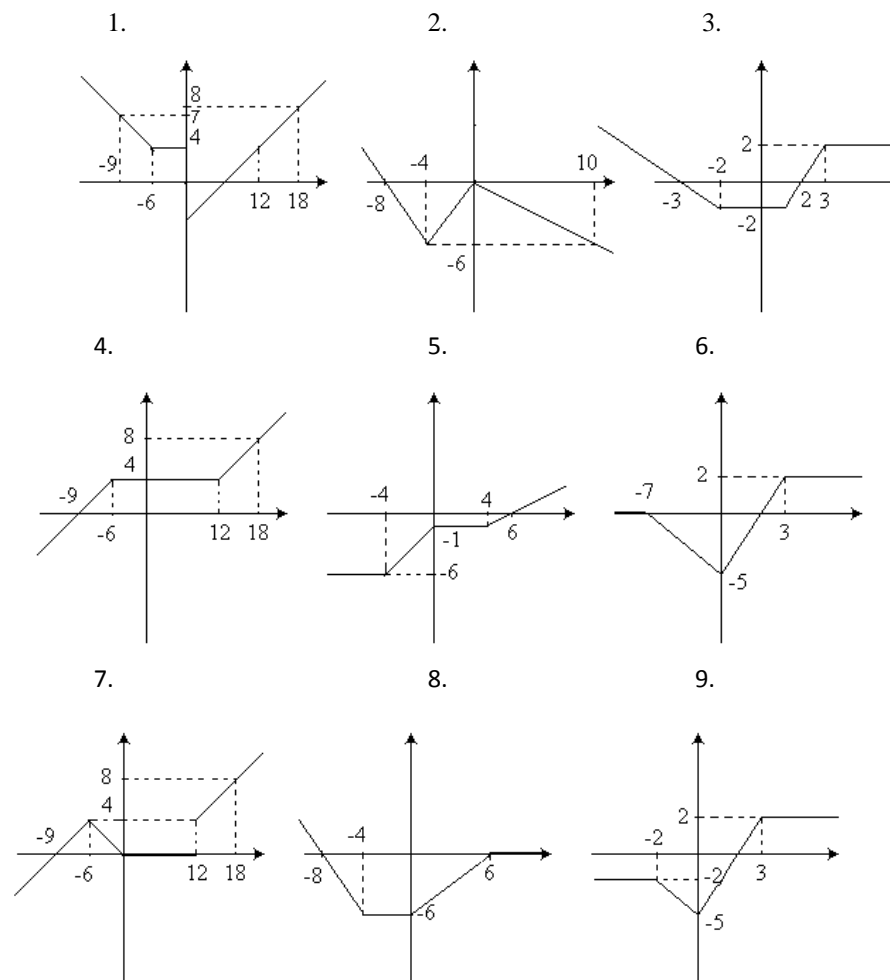
Программа обучения бакалавров указанных направлений подготовки по дисциплине «Программирование» в основном рассчитана на 2 семестра. Предполагается, что студенты имеют возможность пользоваться учебниками, поскольку теоретические сведения здесь не приводятся.

В пособии не рассматриваются вопросы программирования под Windows, речь идет о консольных приложениях.

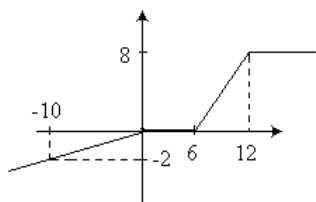
Фрагменты программ, представленные в тестах, протестированы в свободно распространяемом компиляторе Dev C++ v.4.9.9.

Задание 1.1. Для функций, заданных графически, определить значение y для заданного с клавиатуры значению x . Нарисовать блок-схему, написать программу на языке C++.

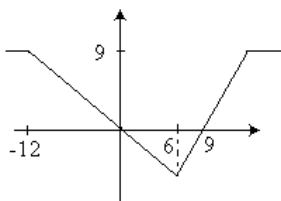
Варианты



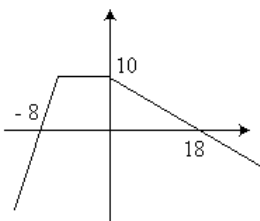
10.



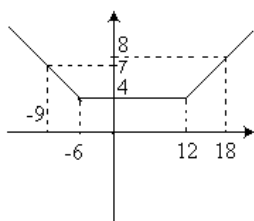
11.



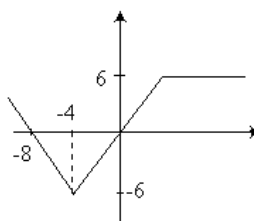
12.



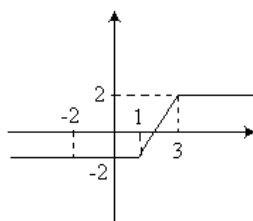
13.



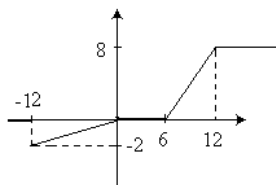
14.



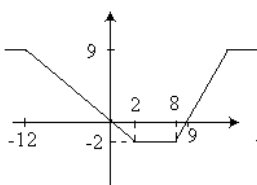
15.



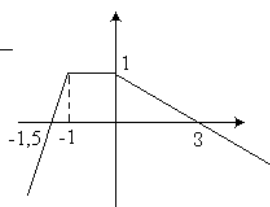
16.



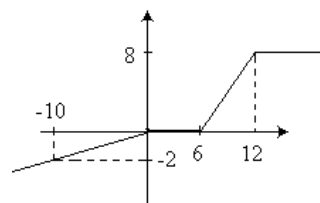
17.



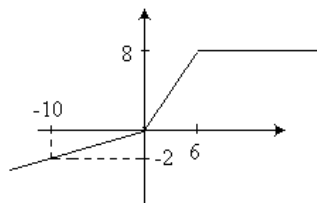
18.



19.



20.



Задание 1.2. Составить блок-схему и программу на языке C++. Ввести с клавиатуры трехзначное число и выполнить с ним действия согласно варианту.

Варианты

1. а) Какая из его цифр больше,
б) является ли сумма его цифр двузначным числом?
2. а) Какая из его цифр меньше,
б) является ли произведение его цифр трехзначным числом?
3. а) Какая из его цифр больше заданного числа a ,
б) является ли сумма его цифр кратной трем?
4. а) Что больше сумма его цифр или произведение двух последних,
б) верно ли, что последняя его цифра четная?
5. а) Верно ли, что все его цифры одинаковые,
б) верно ли, что его средняя цифра нечетная?
6. а) Содержит ли оно одинаковые цифры и какие,
б) является ли произведение его цифр двузначным числом?
7. а) Какая из его цифр больше,
б) является ли сумма его первой и последней цифр двузначным числом?
8. а) Верно ли, что оно начинается с четной цифры,
б) является ли сумма его цифр кратной пяти?
9. а) Какие из его цифр больше пяти,
б) является ли сумма его цифр четной?
10. а) Какая из его цифр больше, введенной с клавиатуры цифры,
б) является ли произведение его цифр нечетным числом?
11. а) Какие из его цифр меньше семи,
б) является ли сумма его цифр двузначным числом?
12. а) Сформируйте из числа новое, расположив цифры в порядке возрастания.
б) является ли полученное число четным?

13. а) Сформируйте из него новое число, расположив цифры в порядке убывания,

б) является ли полученное число нечетным?

14. а) Сформируйте из него новое число обменом первой и последней цифр,

б) является ли полученное число четным?

15. а) Входят ли в него числа 2 и 7,

б) является ли произведение его цифр кратным 6?

16. а) Входят ли в него цифры 0 и 5,

б) является ли произведение его цифр кратным 3?

17. а) Каких цифр больше четных или нечетных,

б) входят ли в состав числа цифры 0 и 1?

18. а) Что больше произведение цифр или сумма,

б) содержатся ли в числе цифры, отличные от 0 и 1?

19. а) Каких цифр (четных или нечетных) больше в составе числа?

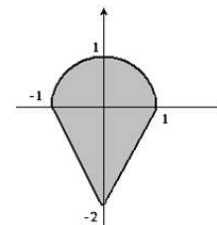
б) сформируйте из него новое число, расположив цифры в порядке возрастания.

20. а) Каких цифр (четных или нечетных) меньше в составе числа?

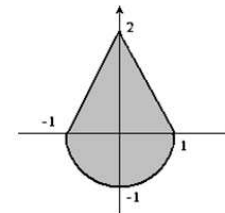
б) сформируйте из него новое число, расположив цифры в порядке возрастания.

Задание 1.3. Дана «мишень» в виде закрашенной области, изображенной на рисунке. Создать алгоритм для определения попадания точки с координатами (x, y) в мишень. Написать программу с использованием условного оператора `if`. Значения координат точки вводить с клавиатуры. Подобрать значения координат, протестировать программу для каждой из координатных четвертей.

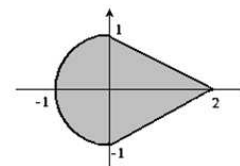
1.



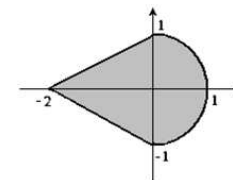
2.



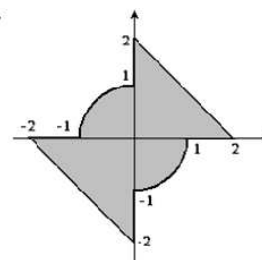
3.



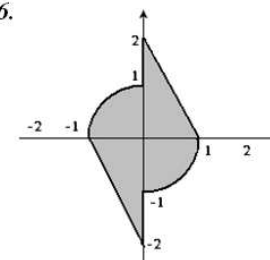
4.



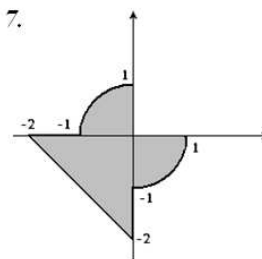
5.



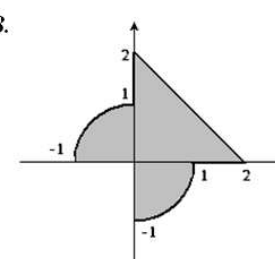
6.

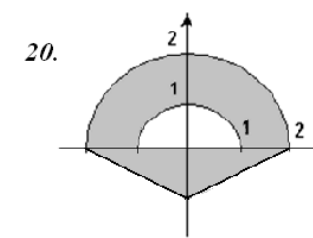
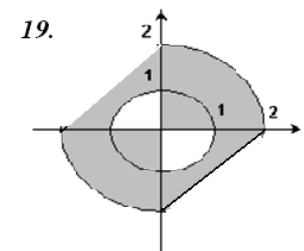
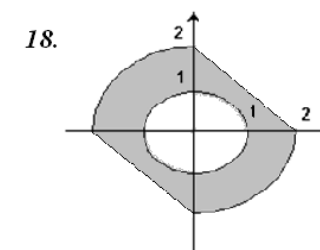
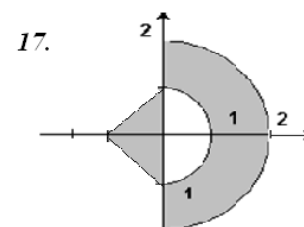
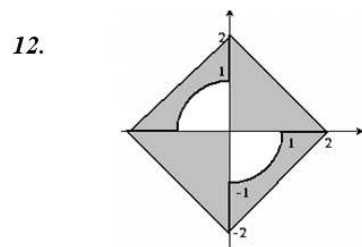
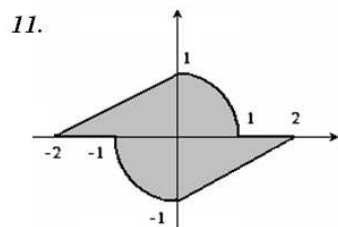
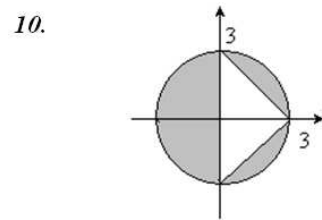
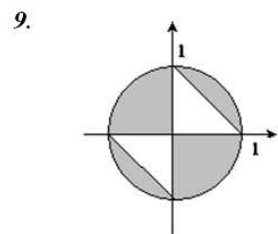


7.



8.





Тема 2. ЦИКЛИЧЕСКИЕ ПРОГРАММЫ

Задание 2.1. Вычислить и вывести на экран в виде таблицы все значения функции y на заданном интервале с шагом Δt . Составить блок-схему алгоритма. В программе использовать реализовать цикл с помощью оператора с предусловием `while`.

Варианты

- | | |
|--|--|
| 1. $y = \begin{cases} at^2 \ln t, & 1 \leq t \leq 2 \\ 1, & t < 1 \\ e^{at} \cos bt, & t > 2 \end{cases}$ | $a = -0.5, \quad b = 2, \quad t \in [0; 3], \quad \Delta t = 0.15$ |
| 2. $y = \begin{cases} \pi^2 - 7/t^2, & t < 1.3 \\ at^3 + 7\sqrt{t}, & t = 1 \\ \lg(t + 7\sqrt{t}), & t > 1.3 \end{cases}$ | $a = 1.5, \quad t \in [0.8; 2], \quad \Delta t = 0.1$ |
| 3. $y = \begin{cases} at^2 + bt + c, & t < 1.4 \\ a/t + \sqrt{t^2 + 1}, & t = 1.4 \\ (a + bt)/\sqrt{t^2 + 1}, & t > 1.4 \end{cases}$ | $a = 2.8, b = -0.2, c = 4 \quad t \in [1; 2], \quad \Delta t = 0.05$ |
| 4. $y = \begin{cases} \pi^2 - 7t^2, & t < 1.4 \\ at^3 + 7\sqrt{t}, & t = 1.4 \\ \ln(t + 7\sqrt{ t + a }), & t > 1.4 \end{cases}$ | $a = 1.65, \quad t \in [0.7; 2], \quad \Delta t = 0.1$ |

$$\begin{aligned}
5. \quad y &= \begin{cases} 1.5 \cos^2 t, & t < 2 \\ 1.8at, & t = 2 \\ (2-t) + 3tgt, & t > 2 \end{cases} & a=2.3, \quad t \in [0.2; 2.8], \quad \Delta t = 0.2 \\
6. \quad y &= \begin{cases} t\sqrt{t-a}, & t > a \\ t \sin at, & t = a \\ e^{-at} \cos at & t < a \end{cases} & a=2.5, \quad t \in [1; 5], \quad \Delta t = 0.5 \\
7. \quad y &= \begin{cases} bt - \lg bt, & bt < 1 \\ 1, & bt = 1 \\ bt + \lg bt, & bt > 1 \end{cases} & b=1.5, \quad t \in [0.1; 1], \quad \Delta t = 0.1 \\
8. \quad y &= \begin{cases} \sin at^2, & t > 3.5 \\ \lg t + \sqrt{t^2 + 1}, & t = 3.5 \\ \cos^2 t, & t < 3.5 \end{cases} & a=2.8, \quad t \in [2; 5], \quad \Delta t = 0.25 \\
9. \quad y &= \begin{cases} \lg(t+c), & t > 1 \\ a^3, & t = 1 \\ \sin^2 \sqrt{|at|}, & t < 1 \end{cases} & a=20.8, \quad c=4, \quad t \in [0.5; 2], \quad \Delta t = 0.1 \\
10. \quad y &= \begin{cases} (\ln^3 t + t^2) / \sqrt{t+c}, & t < 0.5 \\ 1/t + \sqrt{t+c}, & t = 0.5 \\ \cos t + t \sin^2 t, & t > 0.5 \end{cases} & c=2.2, \quad t \in [0.2; 2], \quad \Delta t = 0.2 \\
11. \quad y &= \begin{cases} (a+b)/(e^t + \cos t), & t < 2.8 \\ (a+t)/(t+1), & t = 2.8 \\ e^t + \sin t, & t > 2.8 \end{cases} & a=2.6, \quad b=-0.29, \quad t \in [0; 7], \quad \Delta t = 0.5 \\
12. \quad y &= \begin{cases} a \lg t + \sqrt{|t|}, & t > 1 \\ a^2, & t = 1 \\ 2a \cos t + 3t^2, & t < 1 \end{cases} & a=0.9, \quad t \in [0.8; 2], \quad \Delta t = 0.1 \\
13. \quad y &= \begin{cases} a/t + bt^2 + c, & t < 4 \\ t, & 4 \leq t \leq 6 \\ at + bt^2, & t > 6 \end{cases} & a=2.1, \quad b=1.8, \quad c=-20.4, \quad t \in [0; 12], \quad \Delta t = 1 \\
14. \quad y &= \begin{cases} \sqrt{at^2 + b \sin t + 1}, & t < 0.1 \\ at + b, & t = 0.1 \\ \sqrt{at^2 + b \cos t + 1}, & t > 0.1 \end{cases} & a=5, \quad b=0.2, \quad t \in [-2; 2], \quad \Delta t = 0.4 \\
15. \quad y &= \begin{cases} t\sqrt{t-a}, & t > a \\ \sin at, & t = a \\ e^{-at} \cos at & t < a \end{cases} & a=8, \quad t \in [5; 10], \quad \Delta t = 0.5
\end{aligned}$$

$$\begin{aligned}
16. \quad y &= \begin{cases} bt - \lg bt, & bt < 1 \\ 1, & bt = 1 \\ bt + \lg bt, & bt > 1 \end{cases} & b=5, \quad t \in [5; 10], \quad \Delta t = 0.5 \\
17. \quad y &= \begin{cases} \sin at^2, & t > 3.5 \\ \lg t + \sqrt{t^2 + 1}, & t = 3.5 \\ \cos^2 t, & t < 3.5 \end{cases} & a=8, \quad t \in [1; 7], \quad \Delta t = 0.25 \\
18. \quad y &= \begin{cases} \lg(t+c), & t > 4 \\ a^3, & t = 4 \\ \sin^2 \sqrt{|at|}, & t < 4 \end{cases} & a=10, \quad c=5, \quad t \in [1; 8], \quad \Delta t = 0.5 \\
19. \quad y &= \begin{cases} (t^3 + c)/4, & t > 4 \\ a^3 - 2, & t = 4 \\ \sqrt{|a-ct|}, & t < 4 \end{cases} & a=10, \quad c=5, \quad t \in [1; 8], \quad \Delta t = 0.5 \\
20. \quad y &= \begin{cases} \lg(t) - \ln(c), & t > 5 \\ a^3 + c, & t = 5 \\ e^{-2t} \sqrt{ac}, & t < 5 \end{cases} & a=10, \quad c=5, \quad t \in [1; 8], \quad \Delta t = 0.5
\end{aligned}$$

Задание 2.2. Составить блок-схемы, написать программы на языке C++ для вычисления значений $f(x)$. Значение аргумента x вводить с клавиатуры. В задание цикл с постусловием.

Варианты

$$\begin{aligned}
1. \quad f(x) &= \sum_{n=10}^{28} \frac{(3+x)^4}{n} & 2. \quad f(x) &= \sum_{n=3}^{15} \frac{x^5}{n+1} & 3. \quad f(x) &= \sum_{n=1}^7 \frac{(-x)^{n-1}}{(2n+1)(x+1)} \\
4. \quad f(x) &= \sum_{m=0}^{15} \frac{(\cos x - 1)}{(2m+1)x} & 5. \quad f(x) &= \sum_{n=20}^{30} \frac{nx^2}{(2+n)} & 6. \quad f(x) &= \sum_{n=3}^8 \frac{e^n x^2}{(2n+1)} \\
7. \quad f(x) &= \sum_{m=10}^{23} \frac{(\sin x - 1)}{m+1} & 8. \quad f(x) &= \sum_{n=1}^{23} \frac{(x/3)^2}{(2n+1)} & 9. \quad f(x) &= \sum_{n=1}^{12} \frac{(x/6)^{n-6}}{(2n)} \\
10. \quad f(x) &= \sum_{m=8}^{28} \frac{(x-1)^2}{(2m+1)(x+1)^2} & 11. \quad f(x) &= \sum_{m=0}^{38} \frac{(mx-1)}{(m+1)} & 12. \quad f(x) &= \sum_{m=0}^{10} \frac{(x-1)}{(m+1)(x+1)} \\
13. \quad f(x) &= \sum_{m=10}^{25} \frac{x^4}{2m+1} & 14. \quad f(x) &= \sum_{m=0}^{20} \frac{-mx}{(m+1)} & 15. \quad f(x) &= \sum_{m=10}^{14} \frac{1+x}{(2m+1)x^5} \\
16. \quad f(x) &= \sum_{n=1}^{18} (-x)^3 \frac{3}{(2n+10)} & 17. \quad f(x) &= \sum_{n=0}^5 \frac{x^{n-2}}{2n+1} & 18. \quad f(x) &= \sum_{n=3}^{23} \frac{(x+n)}{(2n)^2} \\
19. \quad f(x) &= \sum_{n=12}^{26} \frac{x^3 - n}{(2n+1)} & 20. \quad f(x) &= \sum_{n=12}^{26} \frac{x^3 - xn}{(2n+x)}
\end{aligned}$$

Задание 2.3. Изменить программу задания 1.3 следующим образом: посчитать количество очков, начисленных после десяти «выстрелов» по мишени. За каждое попадание в мишень начисляется 10 очков. Координаты выстрела задавать счетчиком случайных чисел.

Тема 3. ОБРАБОТКА МАССИВОВ

Задание 3.1. Составить блок-схему алгоритма решения задачи. Написать программу на языке C++. Элементы массива вводить с клавиатуры.

Варианты

1. В массиве A, содержащем 15 элементов, определить сумму положительных элементов, расположенных после минимального элемента данного массива.

2. В массиве B, содержащем 12 элементов, определить среднее арифметическое отрицательных элементов, расположенных после минимального элемента данного массива.

3. В массиве C, содержащем 14 элементов, определить произведение положительных элементов, расположенных до максимального элемента этого массива.

4. В массиве A, содержащем 14 элементов, определить сумму положительных элементов, расположенных между минимальным и максимальным элементами данного массива.

5. В массиве B, содержащем 16 элементов, определить среднее геометрическое отрицательных элементов, расположенных между минимальным и максимальными элементами данного массива.

6. В массиве C, содержащем 13 элементов, определить сумму минимального и максимального элементов данного массива.

7. В массиве H, содержащем 14 элементов, определить сумму положительных элементов, расположенных после минимального элемента данного массива.

8. В массиве X, содержащем 17 элементов, определить минимальный элемент среди положительных элементов данного массива.

9. В массиве X, содержащем 14 элементов, определить максимальный элемент среди отрицательных элементов данного массива.

10. В массиве A, содержащем 15 элементов, определить произведение положительных элементов, расположенных после минимального элемента данного массива.

11. В массиве B, содержащем 16 элементов, определить произведение отрицательных элементов, расположенных после максимального элемента данного массива.

12. В массиве D, содержащем 14 элементов, определить среднее арифметическое элементов, расположенных после минимального элемента данного массива.

13. В массиве D, содержащем 14 элементов, определить разность между минимальным и максимальным элементами данного массива.

14. В массиве Y, содержащем 14 элементов, заменить минимальный элемент нулем, а максимальный увеличить в десять раз.

15. В одномерном массиве A из 14 элементов определить номер первого положительного элемента из расположенных после минимального.

16. В массиве C из 16 элементов определить номер последнего отрицательного элемента из элементов, расположенных между максимальным и минимальными элементами данного массива.

17. В массиве K из 15 элементов найти сумму элементов, расположенных после минимального и стоящих на нечетных местах

18. В массиве из 12 элементов определить максимальный среди отрицательных и минимальный среди положительных элементов.

19. В массиве C, содержащем 14 элементов, определить произведение элементов, удовлетворяющих условию $a < c[i] < b$ и расположенных до максимального элемента этого массива. Значения переменных a и b вводить с клавиатуры.

20. В массиве B, содержащем 13 элементов, определить произведение отрицательных нечетных элементов, расположенных после максимального элемента данного массива.

Задание 3.2. Элементы массива задать с помощью счетчика случайных чисел. Сортировку элементов массива провести двумя методами: «пузырька» и методом прямого выбора.

Варианты

1. Элементы массива $A(20)$ отсортировать следующим образом: сначала расположить все положительные в порядке убывания, затем отрицательные элементы в порядке возрастания, в конце – все элементы, равные нулю.
2. Элементы массива $B(28)$ отсортировать следующим образом: сначала расположить все отрицательные в порядке убывания, затем положительные в порядке возрастания, а следом – равные нулю.
4. Первые десять элементов массива $M(30)$ отсортировать в порядке возрастания, а остальные в порядке убывания.
5. Отсортировать по убыванию элементы массива $P(32)$, расположенные до максимального элемента данного массива.
6. Отсортировать по возрастанию элементы массива $X(29)$, расположенные за минимальным элементом данного массива.
7. Отсортировать по возрастанию элементы массива $O(25)$, расположенные между минимальным и максимальным элементами данного массива.
8. Если положительных элементов массива $K(24)$ больше чем отрицательных, то отсортированные положительные элементы расположить в начале массива. Иначе в начале массива расположить отсортированные отрицательные элементы.
8. Вывести на экран сначала нулевые элементы массива $C(39)$, а за ними отсортированные по убыванию ненулевые элементы данного массива.
9. Первые пять элементов массива $T(23)$ оставить в неизменном порядке, следующие 10 отсортировать в порядке возрастания, за ними три элемента оставить без изменения, а оставшиеся отсортировать в порядке возрастания.
10. Если номер минимального элемента массива $H(28)$ меньше 18, отсортировать по возрастанию элементы, расположенные после него, иначе отсортировать элементы с 8 по 18 номер.
11. Если номер максимального элемента массива $K(25)$ больше 15, то отсортировать по убыванию элементы массива, расположенные до максимального эле-

мента, иначе отсортировать элементы данного массива, расположенные за максимальным элементом.

12. Расположить элементы массива $P(20)$ следующим образом: сначала минимальный элемент, затем максимальный, а далее оставшиеся элементы массива, отсортированные по возрастанию.

13. Вывести на экран сначала отсортированные по возрастанию элементы массива $Y(25)$, стоящие на четных местах, а затем отсортированные элементы, стоящие на нечетных местах.

14. Дан вектор $U(20)$. Вывести на экран элементы данного массива следующим образом: 1-й, 5-й, 15-й и 20-й элементы оставить без изменения, а элементы, расположенные между ними, отсортировать по убыванию.

15. Отсортировать в порядке возрастания положительные элементы массива $A(20)$, расположенные до минимального из элементов.

16. Отсортировать в порядке убывания элементы массива $E(46)$, расположенные до первого отрицательного его элемента.

17. Первые пять элементов массива $X(30)$ оставить без изменения, следующие десять отсортировать в порядке возрастания, а оставшиеся элементы в порядке убывания.

18. Последние десять и первые пять элементов массива $H(27)$ оставить без изменения, остальные отсортировать в порядке убывания.

19. Отсортировать по убыванию элементы массива $A(23)$, начиная с номера, введенного с клавиатуры.

20. Три элемента, расположенные в центре массива $H(25)$ оставить без изменения, остальные отсортировать по возрастанию. Сортировка должна быть выполнена по всем элементам, исключая центральные, а не отдельно по частям.

Задание 3.3. В соответствии с вариантом выполните обработку символьной строки, воспользуйтесь функциями библиотеки `string.h`

Варианты

1. Одну строку инициализировать в программе, другую – ввести с клавиатуры. Сравнить данные строки по длине. Если они не равны, присоединить к меньшей строке - большую. Определить количество слов в полученной строке и наименьшее из них вывести на экран.
2. Ввести две строки с клавиатуры. Посчитать в каждой из них количество гласных букв. В строке, содержащей большее число гласных, удалить все согласные буквы.
3. Одну строку инициализировать в программе, другую – ввести с клавиатуры. Соединить их содержимое. Определить длину полученной строки. Вывести на экран первую половину полученной строки, удаляя из нее гласные буквы.
4. Одну строку инициализировать в программе, другую – ввести с клавиатуры. Если их содержимое одинаково, оставить строки без изменения, иначе соединить содержимое строк. При этом первыми должны быть символы той строки, в которой первый символ по алфавиту раньше. Определить длину полученной строки.
5. Одну строку инициализировать в программе, другую – ввести с клавиатуры. В каждой из строк переместить первый символ в конец строки, после чего соединить строки, расположив вначале наибольшую из них по алфавиту. Определить в новой строке количество гласных букв.
6. В строке символов найти самое длинное и самое короткое слово. Сформировать новую строку, расположив в ее начале самое короткое слово, затем самое длинное, а потом все остальные слова в алфавитном порядке.
7. Определить в строке количество предложений. Если предложений более одного, копировать второе предложение в отдельную строку и вывести ее содержимое на экран. В новой строке отсортировать слова по алфавиту.
8. Одну строку инициализировать в программе, другую – ввести с клавиатуры. Сформировать из них новую строку, чередуя в ней слова из заданных строк. Определить длину строки, количество в ней гласных.

9. Две строки инициализировать в программе, третью – ввести с клавиатуры. Сформировать из них новую строку по следующему алгоритму: сначала соединить строки в порядке возрастания их длины, затем исключить из нее первое и среднее слова. Посчитать количество слов в итоговой строке, начинающихся с гласной буквы.
10. Одну строку инициализировать в программе, другую – ввести с клавиатуры. Сравнить строки, и если они не равны, соединить их. Отредактировать вновь полученную строку, исключив множественные пробелы.
11. Одну строку инициализировать в программе, другую – ввести с клавиатуры. Если строки не тождественны, соединить их, добавив к большей из них по длине меньшую. Отредактировать вновь полученную строку, удалив из нее самое короткое слово.
12. Одну строку инициализировать в программе, другую – ввести с клавиатуры. Присоединить к меньшей по длине строке – большую. Отредактировать вновь полученную строку, исключив из нее слова, содержащие меньше двух символов.
13. Одну строку инициализировать в программе, другую – ввести с клавиатуры. В строке меньшей длины добавить символы 'а', чтобы строки сравнялись по длине. В строке большей длины исключить все символы 'о'. В каждой из строк определить количество гласных и согласных букв.
14. Одну строку инициализировать в программе, другую – ввести с клавиатуры. Сравнить строки на тождественность и если это не так, соединить их содержимое. Во вновь полученной строке определить слово наименьшей длины.
15. Две строки инициализировать в программе. Составить из них третью, дважды чередуя содержимое заданных строк. Определить во вновь полученной строке букву, встречающуюся большее число раз.
16. Две строки инициализировать в программе, третью ввести с клавиатуры. Сформировать строку, расположив в ней содержимое трех строк в порядке увеличения их длин. Определить во вновь полученной строке количество вхождений буквы, введенной с клавиатуры.

17. Одну строку инициализировать в программе, другую ввести с клавиатуры. Присоединить к введенной строке инициализированную строку. В полученной строке отсортировать слова по алфавиту.

18. Ввести с клавиатуры строку. Определить во введенных символах количество гласных и согласных букв. Если гласных больше, вторую половину строки заполнить новыми символами, введенными с клавиатуры. Определить в новой строке количество каждой из букв алфавита.

19. Одну строку инициализировать в программе, другую – ввести с клавиатуры. Если строки не тождественны, сформировать из них третью строку, добавив к большей строке (по длине) меньшую и отделив их между собой пятью восклицательными знаками. Отредактировать вновь полученную строку, удалив из нее самое длинное слово.

20. Одну строку инициализировать в программе, другую – ввести с клавиатуры. Сформировать из них новую строку, добавив к большей по длине меньшую, а затем вновь записав большую. Отредактировать вновь полученную строку, удалив из нее гласные буквы.

Задание 3.4. Составить блок-схему алгоритма решения задачи. Написать программу на языке C++. Значения элементов массива вводить по желанию пользователя либо с клавиатуры, либо случайным образом. Осуществить вывод элементов первоначальной и измененной матрицы в виде таблицы.

Размерности матрицы определить константами, что значительно упростит отладку программы. Для проверки правильности работы вывести на экран все промежуточные значения переменных. При форматировании вывода элементов матрицы использовать манипуляторы ввода-вывода. Для реализации возможности ввода элементов матрицы различными способами использовать оператор switch.

Варианты

1. В матрице размерности 8 на 6 определить номер первого из столбцов, содержащих хотя бы один нулевой элемент. В каждой второй строке матрицы заменить максимальный элемент нулем. Найти сумму положительных элементов матрицы.

2. В матрице размерности 5 на 10 определить количество столбцов, содержащих хотя бы один нулевой элемент, вывести на экран их номера. В каждой строке матрицы поменять местами максимальный и минимальный элементы. Заменить положительные элементы последней строки нулями.

3. В матрице размерности 8 на 8 определить сумму элементов в столбцах, не содержащих отрицательные элементы. Найти минимум среди сумм модулей элементов диагоналей, параллельных главной диагонали матрицы.

4. В матрице размером 8 на 8 найти такие k , что k -ая строка матрицы совпадает с k -ым столбцом. Увеличить минимальный элемент матрицы в десять раз. Найти сумму элементов в тех строках, которые содержат хотя бы один отрицательный элемент.

5. В матрице размером 6 на 9 определить сумму модулей его отрицательных нечетных элементов каждой строки. Найти сумму элементов в тех столбцах, которые содержат хотя бы один нулевой элемент. Заменить положительные элементы второй и третьей строк единицами.

6. В матрице размером 10 на 10 найти сумму элементов в тех столбцах, которые содержат хотя бы один нулевой элемент. Заменить отрицательные элементы матрицы их модулями. В измененной матрице найти произведение элементов, расположенных ниже главной диагонали.

7. В матрице размером 8 на 8 определить номера минимальных элементов в каждой второй строке. Посчитать произведение отрицательных элементов, расположенных выше главной диагонали. Элементы побочной диагонали заменить нулями.

8. В матрице размерности 12 на 8 найти номер первого из столбцов, не содержащих ни одного из отрицательных элементов. В каждой третьей строке матрицы заменить минимальный элемент нулем, а максимальный сотней. Посчитать произведение элементов матрицы, по модулю не превосходящих 5.

9. В матрице размерности 8 на 9 определить произведение элементов в каждом втором столбце. Найти минимум среди сумм элементов диагоналей, парал-

лельных главной диагонали матрицы. В последнем столбце матрицы заменить положительные элементы единицами, а отрицательные – нулями.

10. В матрице размером 9 на 9 определить количество строк, содержащих хотя бы один положительный элемент. Определить номер столбца, в котором содержится минимальный элемент матрицы. На главной диагонали матрицы заменить положительные элементы нулями.

11. В матрице размерности 8 на 8 определить сумму минимального и максимального из ее элементов, расположенных под главной диагональю. В каждой второй строке заменить положительные элементы нулями. Во вновь полученной матрице определить среднее арифметическое каждого столбца.

12. В матрице размерности 10 на 10 поменять местами минимальный и максимальный элементы. Определить произведение положительных элементов пятого и десятого столбцов (отдельно). В каждой строке подсчитать сумму элементов.

13. В матрице размерности 7 на 7 посчитать произведение сумм положительных элементов строк. В каждом втором столбце заменить минимальный элемент нулем. Во вновь полученной матрице найти сумму элементов главной диагонали.

14. В матрице размерности 8 на 8 посчитать произведение элементов главной диагонали. Найденным значением заменить минимальный элемент матрицы. Посчитать сумму отрицательных элементов каждого столбца.

15. В матрице размерности 5 на 10 найти среднее арифметическое произведений элементов столбцов. Найденным значением заменить максимальный элемент матрицы. Определить количество отрицательных элементов каждого столбца.

16. В матрице размерности 6 на 9 найти среднее арифметическое положительных элементов каждого столбца. Определить минимальный элемент в каждой третьей строке. Максимальный элемент матрицы утроить.

17. В матрице размерности 8 на 8 определить среднее арифметическое отрицательных элементов. Заменить максимальный элемент каждого столбца найденным значением. В измененной матрице определить номер строки с наибольшим количеством отрицательных элементов.

18. В матрице размерности 7 на 7 определить является ли матрица симметричной относительно главной диагонали. Определить среднее арифметическое положительных элементов каждого столбца. Найти номера максимальных элементов каждой строки.

19. В матрице размерности 9 на 9 найти номера максимальных элементов в каждом столбце. В каждой третьей строке определить сумму элементов, расположенных до элемента равного нулю. Определить количество отрицательных элементов матрицы.

20. В матрице размерности 8 на 8 определить месторасположение максимального элемента из элементов, расположенных под главной диагональю. В каждом втором столбце определить среднее арифметическое отрицательных элементов. Заменить элементы первой строки соответствующими элементами последней строки.

Тема 4. СОЗДАНИЕ ПОЛЬЗОВАТЕЛЬСКИХ ФУНКЦИЙ

Задание 4.1. В программе создать функцию, грамотно определив ее параметры и тип, продемонстрировать ее вызов. Обязательно использовать прототип функции.

Варианты

1. Определить периметр треугольника, заданного координатами вершин. Длину стороны треугольника вычислять в функции. Посчитать среднее арифметическое периметров трех треугольников, координаты которых ввести с клавиатуры.

2. Определить функцию нахождения объема конуса по известному радиусу основания и высоте. С помощью данной функции найти объемы трех усеченных конусов. Среди них определить наибольший объем.

3. Вычислить значение $z = (\text{sign } x + \text{sign } y) / \text{sign } (x + y)$.

$$\text{где } \operatorname{sign} a = \begin{cases} -1, & \text{при } a < 0, \\ 0, & \text{при } a = 0, \\ 1, & \text{при } a > 0. \end{cases}$$

При вычислении значения проверять область допустимых значений функции.

4. Составить программу вычисления значений:

$$a = \frac{\sqrt{z^3 + 4z^2 + 7z + 1}}{2 + e^{2z^3 + z - 3}}, \quad b = \frac{t^2 + 13t + 16}{7t^3 + t^2 - 4}.$$

Для вычисления значений использовать функцию, находящую значение многочлена. Значения величин z, t вводить с клавиатуры.

5. Создать функцию нахождения скалярного произведения двух векторов в трехмерном пространстве. Использовать ее для нахождения среднего арифметического трех скалярных произведений произвольных векторов.

$$6. \text{ Вычислить значение } f = \frac{\max(a, b, c) * \min(a, c, d) - \max(b, c, d)}{\min(a, b, c)},$$

где a, b, c, d – некоторые значения, введенные с клавиатуры. При вычислении значения учитывать область допустимых значений функции. Для нахождения максимального и минимального значений использовать функции.

7. Создать функцию вычисления площади треугольника по трем заданным сторонам. Функция должна содержать проверку возможности построения треугольника. Продемонстрировать работу функции.

8. Создать функцию, определяющую принадлежит ли точка с заданными координатами (x_i, y_i) уравнению прямой $y = kx + b$. Использовать функцию для определения из некоторого набора точек, лежащих на одной прямой.

9. Создать функцию нахождения длины медианы треугольника по известным длинам сторон. Задать три треугольника, определить наименьшую гипотенузу.

10. Создать функцию нахождения факториала произвольного числа. С помощью ее посчитать $5! + (x+8)!/y!$. Значения переменных x и y ввести с клавиатуры.

11. Создать функцию вычисления определителя матрицы размерности 3 на 3.

12. По введенным с клавиатуры вещественным значениям переменных x, y, z вычислить $f(x, y, z) - f(1, 0, 3) + 2f(2x, y-1, yz)$, где $f(a, b, c) = \frac{\ln a - bc + c^3}{\sin a + 2\cos b + c^4}$. Учитывать область допустимых значений функции.

13. Создать функцию определения длины гипотенузы по известным длинам катетов прямоугольного треугольника. Задать случайным образом пять треугольников. Определить длину наибольшей гипотенузы

14. По известным длинам сторон треугольника вывести на экран длины его высот в порядке убывания. В программе определить функцию нахождения длины одной высоты.

$$15. \text{ Вычислить значение } f = \frac{\max(a, b, c) - 2\min(a, c, d) + 5\max(b, c, d)}{10\min(a, b, c)}, \text{ где } a, b, c, d$$

– некоторые значения, введенные с клавиатуры. Для нахождения максимального и минимального значений использовать функции.

16. Создать функцию, определяющую по известным координатам трех вершин площадь ромба. Вывести на экран в порядке убывания площади трех, заданных случайным образом ромбов.

17. Создать функцию, находящую площадь треугольника по известным длинам двух сторон и углу между ними. Случайным образом задать четыре треугольника. Вывести на экран их площади в порядке возрастания.

18. Определить функцию для определения объема тетраэдра по заданной длине ребра. Определить во сколько раз изменится эта площадь, если радиус увеличить в два раза, в 10 раз.

19. Создать функцию, находящую площадь поверхности цилиндра по известным радиусу основания и высоте. Определить во сколько раз изменится эта площадь, если радиус увеличить в два раза, в 50 раз.

$$20. \text{ Создать функцию для определения числа сочетаний: } C_N^K = \frac{k!}{N!(n-k)!}.$$

Задание 4.2. В соответствии с вариантом создать функцию, параметром которой является одномерный массив и его размерность. Тело функции реализовать

двумя способами: через обращение к элементам массива через указатели и обычным способом. Продемонстрировать возможность применения созданной функции к строкам двумерного массива. Элементы матрицы задать случайным образом, не забыв вывести их на экран.

Варианты

1. Поменять местами минимальный и максимальный элементы массива. Использовать функцию для каждой строки матрицы В (5, 4).
2. Вычислить среднее арифметическое отрицательных элементов массива. Использовать функцией для каждой строки матрицы А (4, 8).
3. Вычислить сумму отрицательных элементов массива в случае, если их более 3, и вернуть нуль иначе. Использовать данной функцией для каждой строки матрицы Х (4, 7).
4. Вычислить сумму элементов массива, расположенных между его максимумом и минимумом. Использовать функцию для каждой строки матрицы О (3, 8).
5. Подсчитать количество отрицательных элементов массива, порядковый номер которых меньше номера минимального элемента. Использовать функцию для каждой строки с четным номером матрицы А (8, 4).
6. Заменить положительные элементы массива нулями, а отрицательные их абсолютными величинами, увеличенными в 10 раз. Использовать функцию для каждой нечетной строки матрицы С (5, 6).
7. Подсчитать количество положительных элементов одномерного массива, порядковый номер которых больше номера максимального элемента. Использовать функцию для строк матрицы А (7, 6), с номером кратным 3.
8. Поменять местами первый элемент массива с последним, второй – с предпоследним, и т.д. Использовать функцию для каждой строки матрицы Н (4, 8).
9. Подсчитать среднее арифметическое элементов массива, стоящих на нечетных местах. Использовать функцию для каждой второй строки матрицы М (4, 6).

10. Перезаписать элементы массива, лежащие между 3-м и 8-м элементом в обратном порядке. Использовать функцию для каждой третьей строки матрицы Т (9, 10).

11. Найти среднее арифметическое значение из максимального и минимального элементов массива. Использовать данную функцию для каждой четвертой строки матрицы Е (8, 6).

12. Вычислить разность между количеством отрицательных и количеством положительных элементов массива. Использовать функцию для каждой строки матрицы Р (4, 14).

13. Найти произведение максимального и минимального элементов массива. Использовать функцию для каждой третьей строки матрицы К (12, 8).

14. Вычислить сумму произведений первого элемента одномерного массива с последним, второго – с предпоследним и т.д. Использовать функцию для каждой строки матрицы Т (5, 6).

15. Найти произведение номеров максимального и минимального элементов массива. Использовать функцию для каждой строки матрицы Е (6, 6).

16. Вычислить произведение количества отрицательных и количества положительных элементов одномерного массива. Использовать функцию для каждой строки матрицы Р (3, 10).

17. Найти произведение отрицательных элементов массива. Использовать функцию для каждой строки матрицы К (4, 8).

18. Найти сумму максимального и минимального отрицательных элементов массива. Воспользоваться функцией для каждой строки матрицы М (5, 7).

19. Найти среднее арифметическое максимального и минимального положительных элементов одномерного массива. Использовать функцией для каждой строки матрицы К (4, 8).

20. Вычислить среднее арифметическое неотрицательных элементов одномерного массива. Использовать функцию для каждой второй строки матрицы Р (12, 14).

Задание 4.3. В алгоритме решения задачи выделить необходимые функции (как правило, это повторяющаяся последовательность действий). Функция должна возвращать несколько значений, для этого использовать ссылки. В программе продемонстрировать вызов функции.

1. Расположить в порядке возрастания корни квадратных уравнений $x^2 + bx - 4 = 0$ и $ax^2 - 8\delta + c = 0$ (где a, b, c – значения, вводимые с клавиатуры). В случае мнимых корней считать их равными нулю.

2. Если квадратные уравнения $2x^2 + bx - 11 = 0$ и $ax^2 - \delta + c = 0$ (где a, b, c – значения, вводимые с клавиатуры) имеют только вещественные корни – определить их сумму, в случае всех мнимых корней сложить их вещественные части.

3. Даны два одномерных массива вещественных чисел A(12) и B(10). Вычислить значение $z = \frac{S_A^+ * S_B^-}{S_B^+ + S_A^-}$, где S_A^+ – сумма положительных элементов вектора A.

Необходимые суммы для одного вектора находить в одной функции.

4. Даны вещественные массивы A(14) и C(8). Вычислить значение $z = \frac{S_A^+ * S_C^+}{K_C^+ + K_A^+}$, где S_A^+, K_A^+ – сумма и количество положительных элементов вектора A.

Необходимые величины для одного вектора находить в одной функции.

5. Создать функцию, подсчитывающую по заданным сторонам треугольника величины его углов. Выведите значения полученных углов из главной функции.

6. Создать функцию, получающую некоторый вес – аргументы функции количество килограмм и грамм в исходном весе. Аргументом функции также является величина изменения заданного веса, выражаемая в граммах. Функция должна возвращать новый вес, с учетом его изменения, выраженный в граммах и килограммах.

7. Создать функцию, получающую в качестве аргументов длины сторон треугольника и подсчитывающую его периметр и площадь. Продемонстрируйте работу данной функции на примере трех треугольников. Расположите найденные значения площадей в порядке возрастания.

8. Создать функцию, возвращающую индексы и значение максимального элемента матрицы. Продемонстрировать работу данной функции для матриц A(3, 5), B(2, 5), C(5, 5).

9. Создать функцию нахождения площади поверхности и объема цилиндра по заданным радиусу основания и высоте. Продемонстрировать ее работу для двух цилиндров, вывести результаты работы функции из основной программы.

10. Создать функцию, возвращающую индексы и значение минимального элемента матрицы. Продемонстрировать работу данной функции для матриц A(2, 4), B(2, 4), C(4, 4).

11. Создать функцию, возвращающую индексы первого неотрицательного элемента матрицы. Продемонстрировать работу данной функции для матриц A(3, 6), B(2, 6), C(5, 6).

12. Создать функцию, вычисляющую максимальный и минимальный элементы вектора. Продемонстрируйте ее работу на примере векторов X(10), H(15).

13. Вычислить значение $T = \frac{\max(A) * \min(A + B) - \max(B)}{\max(A + B) + \min(A) * \min(B)}$, где A, B, C – массивы размерности 10. Значение максимума и минимума одного вектора вычислять в одной функции.

14. Определить функцию нахождения векторного произведения векторов трехмерного пространства (без использования массивов). Параметрами функции должны быть координаты. Продемонстрируйте работу функции.

15. Определить функции нахождения суммы векторов на плоскости и умножения вектора на скаляр. Параметры – координаты векторов. Задать случайным образом координаты векторов $\vec{a}, \vec{b}, \vec{c}, \vec{d}$ и посчитать $2\vec{a} + 3\vec{c} + \vec{d} + \vec{b}$.

16. По заданным сторонам треугольника определить длины биссектрисы, высоты и медианы всех его сторон.

17. Создать функцию нахождения первого элемента, равного нулю, и его местоположение в матрице. Использовать функцию для обмена минимальными значениями матриц A(4, 3) и B(7, 3).

18. Создать функцию нахождения площади поверхности и объема правильной треугольной пирамиды по заданным сторонам основания и высоте. Проде-

монстрировать ее работу для трех случаев, вывести на печать результаты работы функции из основной программы.

19. Создать функцию, возвращающую значения синуса, косинуса, тангенса и котангенса по введенной величине угла (в градусах). Использовать функцию для нахождения значения сложного тригонометрического выражения.

20. Создать функцию определения количества гласных и согласных букв строки, указанной в качестве параметра. Вывести результаты работы функции из основной программы.

Тема 5. СТРУКТУРЫ

Задание 5.1. В каждом из вариантов задания для организации хранения и обработки данных использовать массив структур. Структурный шаблон определять в соответствии с вариантом. При вводе данных применять методы динамического распределения памяти, обеспечить проверку правильности ввода данных. Организовать интерфейс пользователя на основе меню, обязательными пунктами которого будут: ввод данных, вывод всей информации, выход из программы. Остальные пункты меню согласно заданию варианта.

Варианты

1. Составить программу обработки итогов сессии. Структурный шаблон должен содержать фамилию, имя студента; номер группы; пять оценок, полученных на экзаменах с указанием названия предметов. По запросу пользователя предусмотреть вывод следующих сведений: средний балл определенной группы студентов; список неуспевающих студентов (в алфавитном порядке) с указанием предметов, по которым получены двойки; список отличников и их процент от общего числа студентов.

2. Составить программу назначения стипендии по итогам сессии. Структурный шаблон должен содержать фамилию, имя студента; номер группы; пять оценок, полученных на экзаменах с указанием названия предметов, стипендия в руб. Стипендия назначается программно. Студентам, получившим хотя бы одну тройку или двойку, стипендия не назначается. Студентам, получившим все пятер-

ки, назначить стипендию в 3000 руб., пятерок больше, чем четверок – 2700 руб., все четверки – 2500 руб. По запросу пользователя предусмотреть вывод следующих сведений: стипендиальная ведомость для группы, номер которой вводится с клавиатуры; список всех отличником; количество студентов, получающих стипендию, количество студентов, не получающих стипендию.

3. Составить программу, которая реализует компьютерный вариант кассового аппарата. В структурном шаблоне должны содержаться сведения одного чека: дата, список покупок с указанием их цены, платежная сумма, вносимая покупателем и сумма сдачи, которая должна высчитываться программно. В программе по запросу пользователя предусмотреть вывод на экран всех чеков и процедуру закрытия кассы (т.е. подсчет общей выручки за один день).

4. Создать компьютерный вариант записной книжки. В структурном шаблоне должны содержаться следующие сведения: фамилия, имя, дата рождения, номер телефона. По требованию пользователя предусмотреть вывод списка фамилий в алфавитном порядке с указанием номеров телефонов; списка лиц, которых необходимо поздравить с Днем рождения в заданном месяце (с указанием числа), фамилию ближайшего именинника выделить цветом.

5. Составить программу справочной службы аэропорта. Структурный шаблон должен содержать пункт назначения, номер рейса, дату и время вылета, время в полете, стоимость билета, наличие билетов в кассе. Время прибытия в пункт назначения высчитывается программно.

По запросу пользователя предусмотреть вывод информации о рейсах до заданного пункта назначения с указанием времени вылета, времени прибытия, номера рейса, наличия билетов и стоимости билета. Список отсортировать по времени вылета, полное расписание самолетов, отсортированное в алфавитном порядке по пункту назначения.

6. Создать два массива, содержащие сведения о пяти нападающих каждой из хоккейных команд «СПАРТАК» и «ДИНАМО». Структурный шаблон должен содержать: фамилию; число заброшенных шайб; число, сделанных голевых передач; штрафное время. Программа по данным из этих массивов должна создавать

новый массив, содержащий фамилию; команду; сумму очков (голы + передачи) для шести лучших игроков обеих команд, и выводить его содержимое на экран. Предусмотреть просмотр состава команд.

7. Создать структурный шаблон, содержащий сведения о погоде: дата (день, месяц), дневная и ночная температуры, осадки, атмосферное давление, влажность воздуха. По требованию пользователя определить самый холодный зимний и теплый день определенного времени года, среднюю температуру указанного месяца, количество дней месяца с осадками.

8. Создать структурный шаблон, содержащий: фамилию, имя, пол, рост, дату рождения. По запросу пользователя предусмотреть вывод всех данных, отсортированных в алфавитном порядке: фамилии; средний рост мужчин, фамилии и имени самой молодой женщины, из тех, чей рост превосходит средний рост мужчин.

9. Структурный шаблон должен содержать сведения о работниках предприятия: фамилия, имя, дата рождения, должность, название отдела, дата устройства на работу, оклад. Создать программу, которая позволяет получить в алфавитном порядке список сотрудников пенсионного возраста с указанием стажа работы (высчитывать программно) и должности, а также средний заработок работающих в указанном отделе.

10. Структурный шаблон должен содержать сведения об автомобилях: марка, страна-производитель, год выпуска, стоимость. По запросу пользователя организовать вывод на экран полного ассортимента автомобилей российского производства, марок, отсортированных в алфавитном порядке, вывод по неполным данным (по дате выпуска, марке, марке и дате, стоимости и др.).

11. Структурный шаблон должен содержать сведения о коллекции книголюбца: шифр книги, автор, название, год издания, жанр.

По требованию пользователя организовать вывод на экран списка всех книг в алфавитном порядке с указанием автора; число книг издания указанного пользователем года; список книг указанного автора в алфавитном порядке, выбор книг по жанру.

12. Структурный шаблон должен содержать данные о пациентах скорой помощи: фамилия, пол, год рождения, домашний адрес, диагноз, информацию о госпитализации (да, нет). По требованию пользователя организовать вывод на экран списка госпитализированных больных в алфавитном порядке с указанием диагноза; а также список детей с указанным диагнозом, список не госпитализированных больных с указанием домашнего адреса.

13. Структурный шаблон должен содержать сведения об игрушках магазина: название, стоимость, возрастные границы, для которых предназначена игрушка. По требованию пользователя организовать вывод на экран: названий игрушек, цена которых не превосходит указанной пользователем суммы и которые подходят детям определенного возраста (список игрушек отсортировать в порядке возрастания их цены); вывод перечня игрушек по названию; названий игрушек, которые подходят как детям 4 лет, так и детям 10 лет.

14. Структурный шаблон должен содержать сведения багажной квитанции: фамилия, количество вещей, указание веса каждой вещи отдельно и всего багажа в целом (общий вес высчитывать программно). По запросу пользователя организовать вывод на экран общего веса багажа каждого пассажира; фамилии владельцев багажа (в алфавитном порядке), средний вес одной вещи в котором отличается не более чем на 0,3 кг от общего среднего веса вещей. Определите количество пассажиров, имеющих более трех вещей в багаже.

15. Создать структурный шаблон, хранящий сведения о квартире: площадь, количество комнат, этаж, стоимость за квадратный метр, адрес. По требованию пользователя рассчитать среднюю стоимость всех квартир, организовать поиск данных о квартирах с требуемым количеством комнат и указанием этажа (данные отсортировать по стоимости квартиры), данные о квартирах, стоимость которых не превышает указанной.

16. Создать программу справочной службы железнодорожного вокзала. Структурный шаблон должен содержать: номер поезда, станцию назначения, название остановок (ограничить шестью), время отправления, наличие билетов.

По запросу пользователя предусмотреть вывод информации об отправлении поездов в указанный пункт назначения (пунктом назначения может служить как конечная станция, так и промежуточная), в указанный временной интервал, а также информации о наличии билетов на поезд с указанным номером.

17. Создать структурный шаблон, содержащий сведения о комплектации набора подарочных конфет: название набора, массивы наименований конфет, входящих в его состав, соответствующего веса определенного наименования (в граммах), стоимости за один кг, а также общая стоимость набора конфет и его масса. Поля стоимости и массы набора должны рассчитываться вручную. Количество наименований конфет ограничить десятью. По запросу пользователя предусмотреть вывод следующих данных: перечень всех наборов, отсортированных по возрастанию стоимости; наименование набора, наибольшего по весу; перечень наборов из заданного интервала стоимости.

18. Структурный шаблон должен содержать сведения о продаваемых магазином дисках: название, категория (DVD, CD), жанр, стоимость, количество. По требованию пользователя организовать вывод на экран: отдельных перечней DVD и CD, отсортированных в алфавитном порядке; результатов поиска по названию и жанру.

19. Структурный шаблон должен содержать сведения о туристических путевках: страна, город, дата, длительность пребывания, стоимость, наличие путевок (количество). По запросу пользователя предусмотреть вывод информации о путевках до заданного пункта назначения с указанием даты, времени прибытия, номере рейса, наличие путевок и стоимости. Список отсортировать по стоимости, указать полный перечень путевок, имеющихся в наличии, результаты поиска путевки в конкретный город и в указанную дату (ввод с клавиатуры).

20. Структурный шаблон должен содержать сведения о комплексном обеде, состоящем из 4 блюд (салат, первое, второе, третье), о наличии хлеба (да, нет). Каждое блюдо должно содержать название, массу в гр., стоимость, программно должна быть рассчитана стоимость всего обеда. По запросу пользователя предусмотреть вывод следующей информации: списка всего перечня, отсортированного

по стоимости, полный перечень самого дорогого обеда, результаты поиска обеда с указанным блюдом.

Тема 6.ФАЙЛОВЫЙ ВВОД-ВЫВОД

Задание 6.1. С помощью текстового редактора Блокнот создать файл, содержащий текст, длина которого не превышает 1000 символов (длина строки не превышает 70 символов). Имя файла должно иметь расширение txt. Написать программу, которая выводит содержимое файла на экран и выполняет действия в соответствии с вариантом.

Варианты

1. Определить количество слов в тексте; по нажатию произвольной клавиши выделить цветом десятое слово.
2. Определить количество слов в тексте, начинающихся с гласной буквы; выделить шестое слово в тексте по нажатию произвольной клавиши.
3. Определить количество слов в тексте, у которых первый и последний символы совпадают; выделить по нажатию произвольной клавиши первое из найденных слов.
4. Определить количество предложений, начинающихся с гласной буквы; по нажатию произвольной клавиши выделить первое из найденных предложений.
5. Отредактировать текст, удаляя лишние символы пробелов между словами; по нажатию произвольной клавиши выделить цветом первое предложение текста.
6. Отредактировать текст, заменяя буквы «о» на «а»; по нажатию произвольной клавиши выделить цветом первое исправленное слово.
7. Отредактировать текст, заменяя двойную букву «н» на одинарную. По нажатию произвольной клавиши выделить цветом первое исправленное слово.
8. Определить количество восклицательных предложений; выделить цветом первое найденное предложение по нажатию произвольной клавиши.

9. Подсчитать количество слов в самом длинном предложении; выделить цветом по нажатию произвольной клавиши найденное предложение.

10. Определить количество слов в тексте, совпадающих с введенным с клавиатуры словом. По нажатию произвольной клавиши выделить цветом последнее из найденных слов.

11. Отредактировать текст, заменяя в нем повествовательные предложения восклицательными, а восклицательные – вопросительными. По нажатию произвольной клавиши выделить цветом последнее отредактированное предложение.

12. Определить количество слов в тексте, у которых первый и последний символы совпадают; по нажатию произвольной клавиши выделить цветом первое из найденных слов.

13. Определить количество слов, оканчивающихся гласной буквой; по нажатию произвольной клавиши выделить первое из найденных слов.

14. Определить количество символов в самом большом предложении; по нажатию любой клавиши выделить цветом найденное слово.

15. Определить количество повествовательных предложений в тексте; по нажатию произвольной клавиши выделить цветом третье повествовательное предложение.

16. Определить символ, с которого начинается самое длинное слово текста. По нажатию произвольной клавиши выделить все вхождения данного символа в тексте.

17. Определить в тексте количество симметричных слов, выделить их цветом (например, шалаш).

18. Определить в тексте количество слов, начинающихся с гласной буквы, выделить цветом последнее из них.

19. Определить в тексте количество слов, начинающихся и оканчивающихся гласными. Выделить цветом последнее из найденных слов

20. Определить в тексте симметричные предложения, например, кабан упал и лапу на бак. Если таковые имеются – выделить их цветом на экране.

Задание 6.2. В каждом из заданий исходный файл формировать программно. Количество элементов исходного файла неизвестно. Целесообразно для создания бинарного файла создать отдельную программу. Числа могут вводиться либо пользователем, либо счетчиком случайных чисел. Количество чисел в файле задавать случайным образом, и оно должно быть неизвестно в основной программе, т.е. цикл выполняется при условии «пока не конец файла». Для проверки правильности работы программы на экран выводить первоначальное содержимое файла, а также результат.

Варианты

1. В файле содержится некоторое количество чисел. Сформировать из них матрицу, содержащую 4 столбца. Недостающие элементы последней строки задать равными нулю. Вывести на экран матрицу в общепринятом виде, ее размерность и суммы элементов главной диагонали и побочной диагонали.

2. В файле содержатся числа. Сформировать из них матрицу, содержащую пять элементов в строке. Числа неполной строки отбросить. Вывести на печать матрицу в общепринятом виде и посчитать сумму элементов, содержащихся в предпоследней строке.

3. В файле содержатся числа. Сформировать квадратную матрицу из 25 последних чисел файла. Посчитать сумму элементов над главной диагональю полученной матрицы.

4. Из цифр, содержащихся в файле сформировать новый файл следующим образом, сначала расположить трехзначные цифры, затем двузначные. Найти максимальное и минимальное число в полученном наборе.

5. В файле содержатся числа, сформировать из них матрицу 4X4, взяв первые 8 чисел начала файла, а остальные – с конца файла. Вычислить произведение элементов под главной диагональю матрицы.

6. В файле содержатся числа. Сформировать из них матрицу, содержащую пять элементов в строке. Первые десять элементов взять из начала файла, последние пять с его конца. Вывести на печать матрицу в общепринятом виде и вычислить сумму элементов, содержащихся в последней строке.

7. Описать структурный шаблон для представления комплексного числа, содержащий поля: вещественная и мнимая части. Написать программу, записывающую данные о комплексных числах в бинарный файл и отображающую на экран информацию о числах, модуль которых меньше десяти.

8. Описать структурный шаблон для представления комплексного числа, содержащий поля: вещественная и мнимая части. Написать программу, записывающую данные о комплексных числах в бинарный файл и отображающую на экран информацию о числах, аргумент которых меньше $\pi/2$.

9. Описать структурный шаблон для представления комплексного числа, содержащий поля: вещественная и мнимая часть. Написать программу, записывающую данные о комплексных числах в бинарный файл и отображающую на экран информацию о числах, мнимая и вещественная части которых имеют одинаковый знак.

10. В файле заданы множество точек A и точка d вне его. Найти все пары точек, лежащих с точкой d на одной прямой.

11. В файле задано множество точек на плоскости. Найти все точки, лежащие внутри некоторого треугольника. Треугольник задавать с клавиатуры координатами вершин.

12. В файле задано множество точек на плоскости. Подсчитать количество точек, лежащих внутри указанной окружности. Окружность задавать с клавиатуры координатами центра и радиусом. Вывести на экран координаты найденных точек.

13. В файле задано множество точек на плоскости. Найти из данного множества равноудаленные точки от заданной с клавиатуры точки.

14. В файле задано множество точек в пространстве. Определить количество точек, лежащих в одной плоскости.

15. В файле задано множество точек в пространстве. Подсчитать количество точек, лежащих внутри указанной сферы. Сферу задавать с клавиатуры координатами центра и радиусом. Вывести на экран координаты найденных точек.

16. Описать структуру с именем ZNAK, содержащую следующие поля: фамилия, имя; знак зодиака; дата рождения (массив из трех чисел).

Написать программу, записывающую данные в бинарный файл и отображающую на экран информацию о людях, родившихся под знаком, название которого введено с клавиатуры.

17. Описать структурный шаблон, содержащий поля: название начального пункта маршрута; название конечного пункта маршрута; номер маршрута. Написать программу, записывающую данные в бинарный файл и отображающую на экран информацию о маршрутах, которые начинаются в пункте, название которого введено с клавиатуры.

18. В файле содержится некоторое количество чисел. Сформировать из них матрицу, содержащую 6 столбцов. Недостающие элементы последней строки задать равными нулю. Вывести на экран матрицу в общепринятом виде, ее размерность и суммы положительных элементов каждой второй строки.

19. В файле содержатся числа. Сформировать из них матрицу, содержащую пять элементов в строке. Лишние числа отбросить. Вывести на печать матрицу в общепринятом виде и вычислить сумму элементов, содержащихся в предпоследней строке.

20. В файле содержатся числа. Сформировать квадратную матрицу из 25 элементов, начиная элементы с элемента файла, позиция которого указывается пользователем. Вычислить сумму отрицательных элементов под главной диагональю полученной матрицы.

ТЕМА 7. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

Задание 7.1. В соответствии с вариантом составить описание класса. Класс должен содержать не менее двух конструкторов – по умолчанию и с параметрами, деструктор. Объявить объекты класса и продемонстрировать работу методов.

Варианты

1. Организовать класс *матрица*, содержащий методы вывода матрицы в общепринятом виде, нахождения транспонированной матрицы и определителя матрицы.

2. Организовать класс *треугольник*, определенный по длинам трех сторонам, содержащий методы нахождения периметра и площади (по формуле Герона).
3. Организовать класс *треугольник*, определенный по координатам вершин и содержащий конструктор, деструктор, функции нахождения длин сторон, периметра и высоты на большую сторону.
4. Организовать класс *параллелограмм*, определяемый длиной сторон и меньшим углом и содержащий методы нахождения периметра и площади параллелограмма, длин его диагоналей.
5. Организовать класс *окружность*, определяемый координатой центра и длиной радиуса. Класс должен содержать методы вычисления площади круга и длины окружности.
6. Организовать класс *дробь*, содержащий методы вывода дроби в общепринятом виде и функцию выделения целой части.
7. Организовать класс *дробь*, содержащий методы вывода дроби в общепринятом виде и функцию приведения дроби к несократимому виду.
8. Организовать класс *дата*, содержащий данные – число, месяц, год. Создать метод, проверяющий правильность введенной даты, метод вывода даты на экран в формате 12.03.2005.
9. Описать класс *треугольник*, содержащий длины сторон, функцию, определяющую правильность введения данных, т.е. возможность построения треугольника по заданным вершинам, и функцию, рисующую треугольник на экране.
10. Описать класс *вектор* на плоскости, содержащий координаты его начала и конца и метод нахождения длины.
11. Описать класс *многочлен* с полями степень, аргумент и коэффициенты. Создать метод вычисления значения многочлена от аргумента и вывода многочлена в общем виде на экран.
12. Организовать класс *прямоугольник*, содержащий методы нахождения его площади, периметра, длин диагоналей.
13. Описать класс *множество* с полями элементы множества и количество элементов, позволяющий добавлять и удалять элементы из множества, вывод всей элементов множества на экран.

14. Описать класс *карта* (масть и достоинство), содержащий функцию вывода значения карты на экран.
15. Описать класс *время* с полем количество минут. Определить конструктор, задающий время в минутах, прошедших после полуночи. Определить функцию определения текущего времени в часах и минутах (для двадцати четырех часового представления времени в сутках).
16. Создать класс *комплексное число* с методами нахождения модуля и аргумента комплексного числа и вывода числа на экран в общепринятом виде.
17. Описать класс *адрес*, включающий полный почтовый адрес: индекс, область, город, улица дом, квартира. Определить разные виды конструктора.
18. Организовать класс *ромб*, содержащий методы нахождения его площади, периметра, длин диагоналей.
19. Организовать класс *эллипс*, определяемый координатами фокусов и эксцентриситетом. Класс должен содержать методы нахождения канонического уравнения эллипса и длин его полуосей.
20. Организовать класс *вектор*, содержащий его координаты в трехмерном пространстве и методы нахождения длины вектора, вывода на экран координат.

Задание 7.2. Изменить программу предыдущего задания, организовав на основе существующего класса производный класс. Производный класс также должен содержать не менее двух конструкторов.

Исходная программа должна содержаться в двух файлах. В первом описание классов, во втором – реализация, т.е. работа с объектами. Продемонстрировать работу всех методов базового и производного классов.

Варианты

1. Организовать производный класс, дополнительно содержащий функции нахождения произведения и сложения матриц.
2. Организовать производный класс, содержащий дополнительно функцию нахождения углов треугольника, высоты и новой функции нахождения площади по основанию и высоте.

3. Организовать производный класс, содержащий дополнительно функцию нахождения площади по высоте и основанию.

4. Описать производный класс *параллелепипед*, наследующий свойства базового класса и определяющий объем параллелепипеда, длины диагоналей основания и самого параллелепипеда.

5. Организовать производный класс *конус*, определенный по радиусу основания и координатой вершины и содержащий функции нахождения площади поверхности и объема конуса.

6. Организовать производный класс, содержащий функции вычисления сложения и вычитания дробей.

7. Организовать производный класс, содержащий функции вычисления деления и умножения дробей, функцию приведения дроби к несократимому виду.

8. Опишите производный класс, включив в дату день недели. Опишите функцию, для определения дня недели, на который приходится введенная дата, если считать, что 1-е января 1-го года нашей эры – понедельник, функцию вывода даты на экран.

9. Описать производный класс *треугольник + цвет*, дополнительно содержащую цвет и функцию, закрашивающую треугольник в заданный цвет.

10. Описать производный класс, дополнительно содержащий функцию умножения вектора на число и сложения двух векторов.

11. Описать производный класс, позволяющий умножать многочлен на число и складывать многочлены.

12. Описать производный класс *параллелепипед*, наследующий свойства базового класса и определяющий объем параллелепипеда, длины диагоналей основания и самого параллелепипеда.

13. Описать производный класс, дополнительно содержащий функции объединения и пересечения множеств.

14. Описать производный класс, содержащий поле козырь и функцию, проверяющую кроет ли карта, указанную в качестве параметра другую карту с учетом козыря.

15. Организовать производный класс с функцией отображения времени в двенадцати часовом представлении с комментарием до или после полудня.

16. Организовать производный класс, содержащий функции нахождения умножения, деления и сложения комплексных чисел.

17. Объявить производный класс *международный адрес*, игнорирующий индекс, содержащий поле название страны и метод определения кода страны по названию.

18. Объявить производный класс, который в дополнении к базовому классу позволяет определять величину углов ромба.

19. Объявить производный класс, который в дополнении к базовому классу позволяет изображать эллипс на экране.

20. Организовать производный класс, наследующий все свойства базового, и дополнительный метод нахождения скалярного произведения двух векторов, в котором второй вектор является параметром.

Задание 7.3. В зависимости от варианта использовать третье свойство объектно-ориентированного программирования – полиморфизм, создавая виртуальную функции или осуществляя перегрузку операций.

Варианты

1. Изменить программу предыдущей лабораторной работы, используя для сложения и умножения матриц перегрузку соответствующих операций.

2. Изменить программу предыдущей лабораторной работы, определив виртуальную функцию для нахождения площади таким образом, чтобы в базовом и производном классе тела функций определялись по-разному.

3. Описать абстрактный класс *животное*. Класс должен содержать характеристики животных: название, вид, местообитание, функцию вывода всех данных на экран. На его основе реализовать классы *млекопитающее*, *рыба*, *птица*. Отдельными характеристиками классов являются: для млекопитающих – травоядное, хищник или всеядное; для рыб – морская или пресноводная; для птиц – дикая, домашняя, если дикая перелетная, или нет.

4. Создать абстрактный класс *средство передвижения*. На его основе реализовать классы *самолет*, *машина*, *корабль*. Все классы должны хранить параметры средств передвижения: скорость, расход топлива, наименование производителя, год выпуска, метод вывода на экран всех данных, определения срока службы. Индивидуально для самолета указать высоту и максимальную дальность полета, для машины – объем двигателя, для самолета и корабля – количество посадочных мест, для корабля – водоизмещение.

5. Создать абстрактный класс *правильный многоугольник*. На его основе создать классы *треугольник*, *квадрат*, *пятиугольник*. Предусмотреть методы создания объектов, вычисления их периметра, площади, величины угла.

6. Изменить программу предыдущей лабораторной работы, используя для вычитания и сложения дробей соответствующие перегруженные операции.

7. Изменить программу предыдущей лабораторной работы, используя для умножения и деления дробей соответствующие перегруженные операции.

8. Описать абстрактный класс *фигура на плоскости*. На его базе создать классы *круг*, *треугольник*, *прямоугольник*. Предусмотреть методы создания объектов, вычисление площади фигур, периметра для треугольника и прямоугольника, длины окружности – для круга.

9. Изменить программу предыдущей лабораторной работы. Функцию перемещения окна на экране, определить в базовом классе как виртуальную и переопределить ее в производном классе с возможностью перемещения вместе с заголовком.

10. Изменить программу предыдущей лабораторной работы. Функцию сложения векторов оформить через перегрузку операций.

11. Изменить программу предыдущей лабораторной работы. Функцию сложения и умножения многочленов оформить через перегрузку операций.

12. Создать абстрактный класс *правильный многоугольник*. На его основе создать классы *треугольник*, *квадрат*, *осьмиугольник*. Предусмотреть методы создания объектов, вычисления их периметра, площади, величины угла.

13. Изменить программу предыдущей лабораторной работы. Функции добавления и удаления элементов оформить через перегруженные операции сложения и вычитания.

14. Изменить программу предыдущей лабораторной работы. Функцию, определяющую бьет ли первая карта вторую, определить в базовом классе как виртуальную, переопределив в производном классе с учетом козыря.

15. Изменить программу предыдущей лабораторной работы. Функцию определения текущего времени определить в базовом классе как виртуальную. Переопределить ее в производном классе.

16. Изменить программу предыдущей лабораторной работы, используя перегрузку операций для умножения, деления и сложения комплексных чисел.

17. Создать абстрактный класс *правильный многогранник* с полями *длина ребра* и *число ребер*. На его основе создать классы *тетраэдр*, *куб*, *октаэдр* (*осьмигранник*). Предусмотреть методы создания объектов, вычисления их площади поверхности и объема.

18. Изменить программу предыдущей лабораторной работы. Функцию нахождения площади треугольника определить как виртуальную. В базовом классе определять площадь через длины диагоналей, а в производном – через длину ребра и угол.

19. Изменить программу предыдущей лабораторной работы. Функцию, рисующую эллипс на экране, определить в базовом классе как виртуальную и переопределить ее в производном классе с закрашиванием эллипса.

20. Изменить программу предыдущей лабораторной работы, используя перегрузку операции умножения, для метода нахождения скалярного произведения.

ТЕСТОВЫЕ ЗАДАНИЯ

Модуль 1. ОСНОВЫ ПРОГРАММИРОВАНИЯ В C++

1. При объявлении переменной какого типа используется ключевое слово double?

- a) длинное целое b) короткое целое
- c) беззнаковое вещественное d) вещественное с двойной точностью.

2. Как называется законченное описание некоторого действия на языке программирования?

- a) выражением b) оператором
- c) лексемой d) переменной

3. Как называется программа, в которую компилятор транслирует программу на исходном языке?

- a) исходной программой b) объектной программой
- c) выходной программой d) листингом

4. Выберите верный результат выполнения фрагмента программы:

```
#define sqr(y) y*y
```

```
int main( ){
```

```
    int x = 1;    printf("sqr x = %d", sqr(x+3));
```

```
}
```

- a) sqr x = 16 b) sqr x = 7 c) sqr x = 9 d) sqr x = 8

5. Какие файлы заголовков нужно включить с помощью директивы #include, чтобы использовать в программе оператор: cout<< cos(y)<<endl;

- a) conio.h iostream.h b) stdio.h math.h
- c) iostream.h stdio.h d) iostream.h math.h

6. Выберите результат выполнения программы:

```
#include <iostream.h>
```

```
int main( ) {
```

```
int x=8, y=2;
```

```
cout<< x/(y+1); }
```

- a) 2 b) 5 c) 2.666666 d) 3

7. Определяя переменную, можно задать ее значение. Как называется это действие?

- a) объявлением b) инициализацией c) вводом d) выводом

8. Как называется множество допустимых данных и набор операций, применимых к этому значению?

- a) описанием данных b) типом данных
- c) идентификатором d) переменной

9. Какие файлы заголовков нужно включить с помощью директивы #include, чтобы использовать в программе оператор: cout<< setw(4)<<y<<endl;

- a) iomanip.h iostream.h b) iomanip.h math.h
- c) iostream.h stdio.h d) iostream.h math.h

10. Напишите результат выполнения программы

```
#include <iostream.h>
```

```
int main( ) {
```

```
int x=8, y=2;
```

```
y++;
```

```
cout<< "x="<<x<<"y="<<y<<endl<<<"x/y="<<x/y; }
```

11. Выберите недопустимую конструкцию языка C++

- a) a=c=b=5 b) x++
c) m=++k*1 d) p*k=h

12. Выберите недопустимую конструкцию языка C++

- a) a*=b b) --u
c) m+=k*1 d) p=*k

13. Поставьте соответствие между терминами и их определениями:

- a) символьное наименование, состоящее из букв, цифр и символа подчеркивания, не начинающее с цифры;
b) последовательность операторов, заключенная в блок;
c) последовательность команд, выполняемая компьютером для решения некоторой задачи;
d) место в памяти для хранения изменяемого значения.

переменная _____
блок _____
идентификатор _____
программа _____

Модуль 2. РАЗВЕТВЛЯЮЩИЕСЯ ПРОГРАММЫ

1. Вычислите значение логического выражения Не (А или не В и С) или С при значениях A=B=C=истина

- a) истина b) ложь

2. Найдите логическое выражение, имеющее значение истина только при выполнении условия $10 < x \leq 25$

- a) $x > 10 \ \&\& \ x \leq 25$ b) $x > 10 \ || \ x \geq 25$
c) $x < 10 \ \&\& \ x \geq 25$ d) $x > 10 \ \&\& \ x > 25 \ \&\& \ x = 25$

3. Определите результат выполнения фрагмента программы

```
float fvalue= -15.7;
if ( fvalue>=0.0 ) fvalue = fvalue; else fvalue = -fvalue;
cout<< fvalue;
```

4. Определите результат выполнения фрагмента программы

```
int x= 1, y=2, s;
if ( x*x + y*y < 1 ) s = 10;
else if ( x*x + y*y < 9 && x*y>0 ) s = 5;
else s = 0;
cout<<x+y*s;
```

5. Выберите из вариантов верный результат выполнения фрагмента программы:

```
int i = 1;
if ((i <=1) && (i ==2)) i = 3;
printf ("i = %d", i);
```

- a) фрагмент скомпилирован не будет b) i = 1
c) i = 2 d) i = 3

6. Выберите верный результат выполнения фрагмента программы:

```
int i = 1;
if ((i <=1) && (i ==2)) i = 3;
printf ("i = %d", i);
```

- a) фрагмент скомпилирован не будет b) i = 1
c) i = 2 d) i = 3

7. Определите результат выполнения фрагмента программы


```
int x= 1, y=2, s
```

```
if ( x*x + y*y < 1 ) {s = 10; x++;} else {s = 5; x+=2;}
```

```
s = 0
```

```
cout<<x+y*s
```

8. Выберите выражение, соответствующее истине, если переменным x, y, z присвоены следующие значения: x=true, y=true, z=false

- a) z&& x&& y b) !(!x || y)
c) !z || (x && (y || z)) d) x || (!z || y)

9. Выберите выражение, соответствующее значению ложь, если переменным x, y, z присвоены следующие значения: x=true, y=true, z=false

- a) z && y | x b) !(x || z)
c) z || (x && (y || z)) d) x || (z || y)

10. Поставьте в соответствие выражениям эквивалентные им выражения:

- a) !(y==x) 1) x!=y || z<=x
b) !(x ==y||x== z) 2) x!= y && a!=z
c) !(x==y && (z>x) 3) x!= y
d) z>x && z==y 4) z>=y

Модуль 3. ЦИКЛЫ

1. Сколько раз распечатается слово Hello в процессе работы ниже приведенного фрагмента?

```
int i = 0;  
while (i < 10)  
    printf ("Hello");  
i++;
```

- a) ни одного раза b) 9 раз c) 10 раз d) бесконечное число раз

2. В каких случаях используется оператор break?

- a) для обеспечения перехода в точку программы, находящуюся непосредственно за оператором, внутри которого он находится;
b) для пропуска всех операторов, оставшихся до конца цикла, и передачи управления на начало следующей итерации;
c) для завершения выполнения функции и передачи управления в точку ее вызова;
d) для передачи управление на помеченный оператор.

3. В каких случаях используется оператор continue?

- a) для обеспечения перехода в точку программы, находящуюся непосредственно за оператором, внутри которого он находится;
b) для пропуска всех операторов, оставшихся до конца цикла, и передачи управления на начало следующей итерации;
c) для завершения выполнения функции и передачи управления в точку ее вызова;
d) для передачи управление на помеченный оператор.

4. Сколько раз распечатается слово Hello в процессе работы ниже приведенного фрагмента

```
int i = 10;  
do    cout<<"Hello";  
i++;  
while (i < 10);
```

- a) ни одного раза b) 9 раз c) 1 раз d) бесконечное число раз

5. Сколько раз распечатается слово Hello в процессе работы ниже приведенного фрагмента?

```
for (int i = 10; i > 1; i /= 2)    cout <<"Hello";
```

Модуль 4. МАССИВЫ

- a) 4 раза b) 5 раз c) 3 раза d) бесконечное число раз

6. Сколько раз распечатается слово Hello в процессе работы ниже приведенного фрагмента?

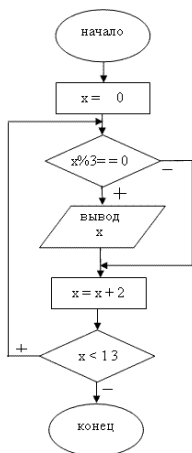
```
for ( int i = 5; i < 100; i*=5);  
cout << "Hello";
```

- a) ни одного раза b) 3 раза c) 1 раз d) бесконечное число раз

7. Выберите верное утверждение из перечисленных:

- a) цикл с постусловием выполняется хотя бы один раз
b) тело цикла не может содержать более одного оператора
c) цикл for реализован как цикл с постусловием
d) вложение цикла в цикл недопустимо

8. Каков результат выполнения алгоритма, представленного на блок-схеме, если переменная x является типа int и может принимать только целое значение?



1. Объявлен массив `int plan [3][3]={ { 1, -2, 8}, {0, 7}, {-1, 9, -3} }`;

Какое значение имеет элемент массива `plan[2][1]`?

- a) 0 b) 9 c) -1 d) -2

2. Объявлен массив `float X []={ 1.3, -2. 8, 0. 7, -1, 9, -0.3};`

Какой индекс имеет элемент массива, 4 равный -0.3?

- a) 6 b) 5 c) 4 d) в объявлении ошибка

3. Выберите верное объявление массива элементов типа float, содержащего 30 строк и 10 столбцов

- a) `float plan[10][30];` b) `float plan [29][9];`
c) `float plan [31][11];` d) `float plan [30][10];`

4. Выберите верное объявление строки символов, предназначенной для хранения слова “студент”

- a) `char string[];` b) `char string[7];` c) `char string[8];` d) `char string[6];`

5. Выберите верный фрагмент программы, предназначенный для подсчета суммы и количества положительных элементов массива A.

- a) `int i, s=0, k=0, A[]={1, -3, 0, 8};` b) `int i, s=0, k=0, A[]={1, 0, 6, 8};`
for (i=1; i<=4; i++) for (i=0; i<4; i++)
if (A[i]>0) {s+=A[i]; k++;} if (A[i]>0) s+=A[i]; k++;
- c) `int i, s=0, k=0, A[]={ -3, 0, 6, 8};` d) `int i, s=0, k=0, A[]={1, -3, 6, 8};`
for (i=0; i<4; i++) for (i=0; i<=4; i++)
if (A[i]>0) {s+=A[i]; k++;} if (A[i]>0) {s+=A[i]; k++;}

6. Выберите верный результат работы фрагмента программы:

```
char s[ ]= "session";
cout<< strcmp(s, "session");
```

- a) 0 b) -10 c) 5 d) session e) sessionsession

7. Выберите верный результат работы фрагмента программы:

```
char s[ ]= "сессия";
strcat(s, "зачтено");
cout<< s;
```

- a) 0 b) сессия c) зачтеносессия d) сессиязачтено

8. Выберите верный результат работы фрагмента программы:

```
char s[20]= "сессия";
strcat (s, "зачтено");
cout<< s;
```

- a) 0 b) сессия c) зачтеносессия d) сессиязачтено

9. Выберите результат выполнения фрагмента программы

```
char r[5]= "test", t[]="yes!";
if (strlen(r)>4) strcpy(r, t);
printf( "%s %s", r, t )
```

- a) "test" "yes!" b) test yes! c) yes! yes! d) test test

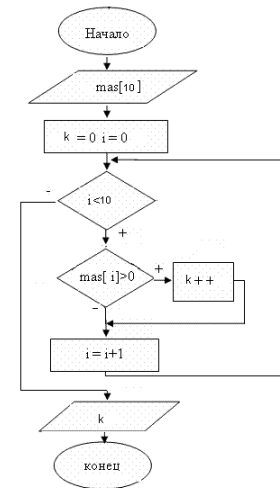
10. Даны объявления: int sample[8], k

Выбрать содержимое массива sample после выполнения программного следующего кода:

```
for (k=0; k<8; k++)
if ( k%2==0 )    sample[k]=k;
else            sample[k]= 10 - k;
```

- a) 0 9 2 7 4 5 6 3 b) 0 9 8 7 6 5 4 3 c) 0 9 1 8 3 7 4 6

11. Изучите блок-схему. Какой алгоритм она описывает?



- a) нахождение суммы положительных элементов массива
 b) нахождение суммы неотрицательных элементов массива
 c) определение количества положительных элементов массива
 d) определение количества положительных элементов массива

12. Массив инициализирован при объявлении следующим образом:

```
int array[ ]={ 1, 6, 8, 2, 9, 4, 0};
```

Чему равен индекс последнего элемента?

13. Массив вещественных чисел инициализирован следующим образом:

```
float matrix [ ] [2] ={{ 1.2, 6.5}, {-4.0}, {9.6, 2.3}, {0.0, -2.7}};
```

Значение 0.0 имеет элемент:

- a) matrix[3][1] b) matrix[2][1] c) matrix[3][0] d) matrix[0][3]

14. Каков результат выполнения программы?

```
#include <iostream.h>
```

```
int main {
```

```
int i, j, s=0, M[ ] [3]={ 1, 2, 3, 4, 5, 6, 7, 8, 9};
```

```
for (i=0; i<2; i++)
```

```
for (j= 1; j<3; j++) s+=M[i][j];
```

```
cout<<s;
```

```
}
```

15. Каков результат выполнения программы?

```
#include <iostream.h>
```

```
int main {
```

```
int i, j, s=0, M[ ][3]={1, 2, 3, 4, 5, 6, 7, 8, 9};
```

```
for (j=0; j<2; j++)
```

```
    for (i= 2; i<3; i++) s+=M[i][j];
```

```
cout<<s;
```

```
}
```

Модуль 5. УКАЗАТЕЛИ И ССЫЛКИ

1. Для приведенного фрагмента программы выберите среди приведенных вариантов верный результат:

```
int t[ ] = {0, 1, 2, 3, 4 };
```

```
for (int i = 0; i < sizeof(str)/sizeof(int); i+=2 )
```

```
    cout<<t[i]<< *(t+i)<<endl;
```

a) 024024

b) 002244

c) 00

d) 00

11

22

22

44

33

44

2. В приведенном фрагменте программы подсчета суммы элементов массива, возможно, содержится ошибка. Выберите среди приведенных ответов верный:

```
int arr[6] = {0, 1, 2, 3, 4, 5 };
```

```
int i, s = 0;
```

```
for (i = 0; i < 6; i++)
```

```
    s += *arr++;
```

a) всё правильно

b) допущена ошибка, следует писать `s += *(arr+i);`

c) допущена ошибка, следует писать `s += arr++;`

d) допущена ошибка, следует писать `s += *arr+i`

3. Объявлены переменные `int x, y, z[5]`

и прототип функции `int mean (int* a, int &b)`

Выберите правильный вариант вызова данной функции:

a) `y=mean(z, x);`

b) `y=mean(z, &x);`

c) `x=mean(z[], y);`

d) `x=mean(*z, y);`

4. Дано объявление функции

```
void function ( int &x, float &y)
```

```
{ x--; y=x/4+y*y -1 ;
```

```
if (!x) return x-3;
```

```
}
```

5. Выберите фрагмент программы с верным вариантом вызова функции

a) `int a=1; float b=0.5; function (a, b);`

b) `int a=1,c; float b=0.5; c= function (&a, &b);`

c) `int a=1,c; float b=0.5; c=function (a, b);`

6. Напишите результат работы фрагмента кода:

```
int function ( int &x, float &y)
```

```
{ x++;
```

```
    y=x/3+sqr(y) -1 ;
```

```
return 1;
```

```
}
```

```
int main{
```

```
int a=1,c; float b=0.25;
```

```
c=function (a, b);
cout<< "a="<<a<< " b="<<b;
}
```

7. Объявлен массив `int A[3][2]={1, 2, 3, 4, 5, 6};`

Чему равно значение выражения `*A[1]+1`?

8. Объявлен массив `int A[3][2]={{1, 2}, {-3}, {5, 6}};`

Поставьте в соответствие выражения их значениям

- | | |
|---------------------------|-------|
| a) <code>*A+3</code> | 1) 5 |
| b) <code>*(A[2]+1)</code> | 2) -3 |
| c) <code>*A[1]</code> | 3) 3 |
| d) <code>*(A+5)</code> | 4) 6 |

9. Выберите недопустимое выражение с использованием переменной типа

указатель `int *p, *t;`

- | | |
|------------------------|------------------------|
| a) <code>p++</code> | b) <code>p+t</code> |
| c) <code>&p</code> | d) <code>*(t+5)</code> |

10. Выберите неверное утверждение:

- a) в C++ разрешается объявление указателей любой степени вложенности;
- b) указатель может быть объявлен типа `void`;
- c) указатель не является самостоятельным типом, он всегда связан с каким-либо другим;
- d) указатель хранит в себе адрес другой переменной.

11. Выберите верное утверждение:

- a) при передаче массива в функцию требуется передать значение его нулевого элемента;
- b) массивы нельзя передавать в функцию;

c) при передаче массива функцию следует передавать адрес его нулевого элемента;

d) при передачи массива в функцию достаточно указать количество его элементов.

12. Объявлен массив `int A[3][2]={{1, 2}, {-3}, {5, 6}};`

Поставьте в соответствие выражения-синонимы

- | | |
|---------------------------|----------------------------------|
| a) <code>*A+2</code> | 1) <code>A[2][0]</code> |
| b) <code>A[2]+1</code> | 2) <code>2+*(&A[0])</code> |
| c) <code>*(A[2]+1)</code> | 3) <code>(&A[2][0])++</code> |
| d) <code>*(A+2)</code> | 4) <code>A[2][1]</code> |

Модуль 6. ПОЛЬЗОВАТЕЛЬСКИЕ ФУНКЦИИ

1. Функция с именем `prim` является не возвращающей значение и имеет два аргумента целого типа, какой из вариантов прототипа функции является правильным

- | | |
|---|--|
| a) <code>prim (int x, int y);</code> | b) <code>void prim (int x, y);</code> |
| c) <code>void prim (int , int);</code> | d) <code>void prim(int x; int y);</code> |

2. Дано описание функции:

```
void sred (int k)
{
    int s, sum=0;
    for int i=1 to k
    {
        cout<<"Введите целое число"<<endl;
        cin>>s;
        sum+=s;
    }
```

Выберите верный вариант вызова данной функции:

3. Выберите верное утверждение:

4. Выберите верное утверждение;

- а) количество и тип аргументов в прототипе функции и в ее описании могут

б) функции усложняют алгоритм программы;

- c) количество вызовов функции ограничено;
- d) функция с помощью оператора return может возвращать только одно зна-

5. Дано описание функции и вызов функции:

```
float a (int b) {
```

```
float c, d;
```

```
cin>>d;
```

```
c=d+b;
```

```
return c;
```

}

Поставьте соответствие между идентификаторами и их семантикой.

- 4) d d) возвращаемое значение

6. Дано описание функции и вызов функции:

```
int change (int b, int a) {
```

```
if (a>b&& a>0) return a;
```

```
return b;
```

$$\}$$

Выберите верный прототип функции.

7. Дано описание функции и вызов функции:

```
void change (float b, int a) {
```

```
if (a>b && a!=0) cout<< b/a;
```

}

Выберите верный прототип функции.

8. Дана рекурсивная функция:

```
int R(int a)
```

```

{ if (a==0) return 1; else return R(a-2);

```

}

Каков результат функции при значении фактического параметра равного 5?

9. Дано описание функции:

```
int S( int *p, int n)
{ for (int i=0; i<n; i++)
if (p[i]>0) s+=p[i];
return s;}
```

Выберите верный вариант вызова функции для int A[8];

- a) cout<< “сумма положительных элементов равна”<<S(A[], 8);
- b) cout<< “сумма положительных элементов равна”<<S(A[0], 8);
- c) cout<< “сумма положительных элементов равна”; S(A, 8);
- d) cout<< “сумма положительных элементов равна”<<S(A, 8);

10. Выберите правильное использование функции, описанной предыдущем задании для строк массива int B[4][7];

- a) for (i=0; i<4; i++)
cout<< “сумма положительных элементов”<<i<< “строки”<<S(B, 7)<<endl;
- b) for (i=0; i<4; i++)
cout<< “сумма положительных элементов”<<i<< “строки”<<S(B[i], 7)<<endl;
- c) for (i=0; i<7; i++)
cout<< “сумма положительных элементов”<<i<< “строки”<<S(B[i], 4)<<endl;
- d) for (i=0; i<7; i++)
cout<< “сумма положительных элементов”<<i<< “строки”<<S(B, 7)<<endl;

Модуль 7. ТИПЫ ДАННЫХ, ОПРЕДЕЛЯЕМЫЕ ПОЛЬЗОВАТЕЛЕМ

1. Изучите код программы и напишите результат ее выполнения.

```
#include <iostream.h>
struct name {
char first[20];
char second[20];
};
struct rezume{
```

```
struct name fi;
int age;
char proff[15];
};
int main( )
{
struct rezume n1=( “Ivan”, “Ivanov”, 27, “doctor”);
struct rezume *pt=&n1;
cout<< n1.fi.second<< “-”<<pt->proff;
}
```

2. Дано объявление структурных шаблонов:

```
struct name {
char first[10], second[15];
};
struct student{
struct name f;
int age;
char special[15];
};
```

Какое из объявлений переменных типа заданных структурных шаблона является неправильным?

- a) struct student st[];
- b) struct student *st;
- c) student st1, st2;
- d) struct student st = { “Петров”};

3. После объявления переменной st типа student (шаблон предыдущего задания) требуется задать значение полю first. Выберите верный вариант:

- a) st.name.fist= “John”;
- b) st.f.fist= “John”;

- c) strcpy(st.name.fist, "John");
- d) strcpy (student.name.fist, "John");

4. Дано объявление struct worker { char family[30];

int day, code;

float sale;};

Выберите правильный вариант объявления переменных данного типа:

- a) struct worker1, *work;
- b) worker worker1; *work;
- c) worker worker1, work[];
- d) struct worker worker1[5];

5. Дано описание: enum numbers {two = 2, tree, four, ten = 10};

Каково значение переменной four?

- a) не определено
- b) равно 4
- c) равно 1
- d) равно 9

6. Дано описание:

struct N {int x, y} array[5];

Для задания значений элементам массива следует писать:

- a) for (int k=0; k<5: k++) cin>>array[k].x>>array[k].y;
- b) for (int k=0; k<5: k++) cin>>x.array[k]>>y.array[k];
- c) for (int k=0; k<5: k++) cin>>array.x[k]>>array.y[k];
- d) for (int k=0; k<5: k++) cin>>array.x>>array.y;

7. Выберите неверное утверждение:

- a) в C++ допускается совмещение объявления структурного шаблона с объявлением структурной переменной
- b) ИМЯ структурного шаблона не является обязательным
- c) допускается объявление массива структурных переменных
- d) нельзя объявлять указатели на структуры

8. Выберите верное утверждение:

- a) в качестве полей структурного шаблона нельзя использовать структурные переменные других шаблонов
- б) в качестве полей структурного шаблона нельзя использовать указатели на собственный структурный шаблон
- с) в качестве полей структурного шаблона нельзя использовать сложные типы данных
- d) в качестве полей структурного шаблона нельзя использовать константы.

9. Определите результат работы фрагмента программы

struct A {int x, y;}B[4];

for (int k=0, int s=0; k<4; k++) { B[k].x=k; B[k].y=k-1;

s+= B[k].x*B[k].y;}

cout<<s;

10. Как называются элементы структуры?

- a) значениями
- б) записями
- c) полями
- d) указателями

Модуль 8. РАБОТА С ФАЙЛАМИ

1. Изучите код программы. Каково будет содержимое файла text2.txt, если первоначально в файле text1.txt содержалось слово timetable?

```
#include<stdio.h>
```

```
int main( )
```

```
{
    FILE *f1, *f2; char ch;
    f1=fopen("text1.txt", "r");
    f2=fopen("text2.txt", "w");
    while (ch=getch(f1)!=NULL)
        if (ch> 'p') putchar(ch, f2);
```



```
fclose(f1);
fclose(f2);
}
```

2. Изучите код программы. Каково будет содержимое файла text2.txt, если первоначально в файле text1.txt содержалось слово stipend.

```
#include<stdio.h>
int main( )
{
    FILE *f1, *f2; char ch;
    f1=fopen("text1.txt", "r");
    f2=fopen("text2.txt", "w");
    while (ch=getch(f1)!=EOF)
        if (ch<'p' && ch!= 'i') putch(ch, f2);
    fclose(f1);
    fclose(f2);
}
```

3. Функция FSEEK() реализует следующее действие:

- a) считывает заданное число элементов из файла;
- b) перемещает внутренний указатель файла на требуемую позицию;
- c) записывает заданное число элементов в файл;
- d) дополняет содержимое файла значением, заданным как ее параметр.

4. Приведенный фрагмент программы

```
struct point //шаблон для хранения координат точки
{
    int x, y;
    } z;
FILE *f;
f = fopen( "point .bbb", "rb");
fread (&z, sizeof(point), 1, f);
```

- a) организует запись координат одной точки в файл;
- b) организует чтение координат одной точки из файла;
- c) организует запись координат точек в файл;
- d) открывает файл для дополнения координатами точки.

Модуль 9. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

1. Выберите верное определение понятия «абстрактный класс»

- a) класс, не содержащий встроенные методы;
- b) класс, содержащий встроенные методы;
- c) класс, имеющий чистые виртуальные функции;
- d) класс, не имеющий виртуальные функции.

2. Что такое инкапсуляция?

- a) создание новых классов на базе уже имеющихся (базовых) классов, причем новые классы обладают всеми свойствами базовых классов и имеют новые, свойственные только им;
- b) создание сложных типов данных, включающих данные и методы для их обработки;
- c) обозначение одним именем общего действия во всей иерархии типов;
- d) определение функций, вызывающих саму себя.

3. Изучите программный код и сделайте выбор среди приведенных результатов

```
class A {
int x;
public:
    A ( int y=5) {x=y;}
    void PrintA (int _x) { cout<< "x = "<<x<<endl; }
```

```
};
int main
{ A a;
a.PrintA( );
}
```

- a) компилятор выдаст ошибку b) x = 0
c) x = 5 d) 5

4. Для фрагмента программы из предыдущего пункта какой спецификатор доступа поле x имеет?

- a) public b) private c) protected d) void

5. Что называю методом класса?

- a) элемент класса, содержащий данные;
b) элемент класса, содержащий функцию;
c) функцию, отвечающую за инициализацию данных в классе;
d) функцию, отвечающую за уничтожение данных в классе.

6. Выберите верное утверждение о множественном наследовании.

- a) Создание новых классов на основе одного базового класса
b) Создание в одной программе нескольких новых классов на основе множества классов
c) Создание нового класса на основе нескольких базовых классов
d) Создание нескольких классов в одной программе

7. Какой принцип ООП означает обозначение одним именем общего действия во всей иерархии типов?

- a) инкапсуляция б) полиморфизм в) наследование г) рекурсия

8. Выберите правильный вариант результата работы программы

```
class A { int a;
public: A( ) {cout<< "construct A ";}
~A( ) {cout<< "destruct A ";}
};
class B : class A { int a;
public: B( ) {cout<< "construct B ";}
~B( ) {cout<< "destruct B ";}
};
int main ( )
{ B b;}
```

- a) ничего не выведется
b) construct B destruct B
c) construct A destruct A construct B destruct B
d) construct A construct B destruct B destruct A
e) construct B construct A destruct A destruct B

9. Выберите верное утверждение:

- a) закрытые элементы базового класса в производном классе недоступны
b) обращение к закрытым элементам класса осуществляется только через методы производных классов
c) открытые элементы класса доступны только производным классам
d) спецификатор доступа может обозначаться с помощью ключевых слов private, public, virtual, protected

10. Выберите верное утверждение:

- a) указание ключа доступа необязательно, по умолчанию для классов используется ключ доступа public
b) производный класс наследует из базового класса поля и методы, а также деструктор, конструктор и операцию присваивания

с) защищенные элементы при наследовании с ключом доступа `private` становятся закрытыми, а в остальных случаях доступ к ним не меняется

д) доступ к открытым элементам остается открытым вне зависимости от ключа доступа.

11. Выберите верное утверждение:

а) с помощью перегрузки можно создавать новые операции и операторы

б) Обращение к виртуальным функциям возможно только через указатели на экземпляр класса

в) в перегрузке функций участвуют параметры и тип перегруженной функции

д) при использовании перегрузки операции можно полностью изменить смысл операции

12. Изучите программный код и сделайте выбор среди приведенных вариантов оценки

```
class A {  
    int x;  
    public:  
        A ( ) {cout<<"x-?"; cin>>x;}  
        void SetX (int _x) { x = _x; }  
};  
int main () {  
    A *a;  
    SetX (10);  
}
```

а) для доступа следует писать `a -> SetX (10);`

б) для доступа следует писать `A.SetX (10);`

в) для доступа следует писать `a.SetX (10);`

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Дейл, Н. Программирование на C++ / Дейл Н., Уимз Ч., Хеддингтон М. – М.: ДМК, 2000. – 672 с.

2. Лаптев, В.В. C++ Объектно-ориентированное программирование: задачи и упражнения (Допущено Мин. обр. РФ) / В.В. Лаптев, А.В. Морозов, А.В. Бокова. – СПб.: Питер, 2007. – 281 с.

3. Лаптев, В.В. C++ Объектно-ориентированное программирование: учеб. пособие. – СПб.: Питер, 2008. – 458 с.

4. Павловская, Т.А. C/C++. Программирование на языке высокого уровня (Допущено МинОбр РФ) – СПб.: Питер, 2010. – 461 с.

5. Павловская, Т.А. C++. Объектно-ориентированное программирование. Практикум (Допущено МинОбр РФ) / Т.А. Павловская, Ю.А. Щупак. – СПб.: Питер, 2004. – 265 с.

6 Холзнер, С. Visual C++ 6. учебное пособие/ С. Холзнер. – СПб.: Питер, 2008. – 571 с.

7 Шамис, В.А. Borland C++ Builder 6 / В.А. Шамис – СПб.: Питер, 2005. – 798 с.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
ЗАДАЧИ ПО ПРОГРАММИРОВАНИЮ	4
Тема 1. РАЗВЕТВЛЯЮЩИЕСЯ АЛГОРИТМЫ	4
Задание 1.1.	4
Задание 1.2.	6
Задание 1.3.	7
Тема 2. ЦИКЛИЧЕСКИЕ ПРОГРАММЫ	10
Задание 2.1.	10
Задание 2.2.	12
Задание 2.3.	13
Тема 3. ОБРАБОТКА МАССИВОВ	13
Задание 3.1.	13
Задание 3.2.	15
Задание 3.3.	16
Задание 3.4.	19
Тема 4. ПОЛЬЗОВАТЕЛЬСКИЕ ФУНКЦИИ	22
Задание 4.1.	22
Задание 4.2.	24
Задание 4.3.	27
Тема 5. СТРУКТУРЫ	29
Задание 5.1.	29
Тема 6. ФАЙЛОВЫЙ ВВОД-ВЫВОД	34
Задание 6.1.	34
Задание 6.2.	36
Тема 7. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ	38
Задание 7.1.	38
Задание 7.2.	40
Задание 7.3.	42

ТЕСТОВЫЕ ЗАДАНИЯ	45
Модуль 1. Основы программирования в C++	45
Модуль 2. Разветвляющиеся программы	47
Модуль 3. Циклы	49
Модуль 4. Массивы	52
Модуль 5. Указатели и ссылки	55
Модуль 6. Пользовательские функции	58
Модуль 7. Типы данных, определяемые пользователем	61
Модуль 8. Работа с файлами	64
Модуль 9. Объектно-ориентированное программирование	66
Библиографический список	70

Галаган Татьяна Алексеевна,

доцент кафедры ИиУС АмГУ, канд. техн. наук

Язык C++. Сборник задач и тестовых заданий. Учебное пособие

Заказ 474.