

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Программное обеспечение информационных технологий»

ПРОГРАММИРОВАНИЕ И ОСНОВЫ АЛГОРИТМИЗАЦИИ

*Методические рекомендации к лабораторным работам
для студентов направления подготовки
15.03.06 «Мехатроника и робототехника»
дневной формы обучения*



Могилев 2023

УДК 004.42
ББК 32.973-018
П78

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Программное обеспечение информационных технологий» «24» марта 2023 г., протокол № 8

Составитель канд. техн. наук, доц. Н. Н. Горбатенко

Рецензент канд. техн. наук, доц. Е. В. Ильюшина

Методические рекомендации разработаны на основе рабочей программы по дисциплине «Программирование и основы алгоритмизации» для студентов направления подготовки 15.03.06 «Мехатроника и робототехника» дневной формы обучения и предназначены для использования при выполнении лабораторных работ.

Учебное издание

ПРОГРАММИРОВАНИЕ И ОСНОВЫ АЛГОРИТМИЗАЦИИ

Ответственный за выпуск	В. В. Кутузов
Корректор	И. В. Голубцова
Компьютерная верстка	Н. П. Полевничая

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 21 экз. Заказ №

Издатель и полиграфическое исполнение:
Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет».

Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 07.03.2019.

Пр-т Мира, 43, 212022, г. Могилев.

© Белорусско-Российский
университет, 2023

Содержание

1 Лабораторная работа № 1. Разработка блок-схем алгоритмов.....	4
2 Лабораторная работа № 2. Среда программирования Visual Studio. Программирование линейных алгоритмов.....	14
3 Лабораторная работа № 3. Программирование ветвящихся алгоритмов.....	22
4 Лабораторная работа № 4. Разработка программ с использованием циклов.....	22
5 Лабораторная работа № 5. Использование циклов для решения задач численными методами.....	24
6 Лабораторная работа № 6. Разработка программ с использованием одномерных массивов.....	24
7 Лабораторная работа № 7. Разработка программ с использованием двумерных массивов.....	25
8 Лабораторная работа № 8. Программирование задач с использованием строк.....	25
9 Лабораторная работа № 9. Структуры.....	26
10 Лабораторная работа № 10. Файловый ввод/вывод. Текстовые файлы.....	26
11 Лабораторная работа № 11. Файловый ввод/вывод. Двоичные файлы.....	27
12 Лабораторная работа № 12. Программирование с использованием функций.....	27
Список литературы.....	28

1 Лабораторная работа № 1. Разработка блок-схем алгоритмов

Цель работы: овладение навыками составления блок-схем алгоритмов решения задач в соответствии с требованиями ГОСТ 19.701–90.

1.1 Краткие теоретические сведения

Алгоритм – это точная инструкция, а инструкции встречаются во всех областях человеческой деятельности. Однако не всякую инструкцию можно назвать алгоритмом. Алгоритм применяется к искомому набору исходных величин, называемых аргументами.

Цель исполнения алгоритма – получение определенного результата, если в результате исполнения алгоритма не достигнута определенная цель, значит, алгоритм либо неверен, либо не завершен.

Алгоритм – это метод (способ) решения задачи, записанный по определенным правилам, обеспечивающим однозначность его понимания и механического исполнения при всех значениях исходных данных за конечное число шагов.

Или более коротко: алгоритм – это строго определенная последовательность действий, необходимых для решения данной задачи.

Примером алгоритма может служить кулинарный рецепт приготовления блюда. Рассмотрим простейший алгоритм – алгоритм заварки чая.

- 1 Подготовить исходные величины – чай, воду, чайник, стакан, ложку.
- 2 Налить в чайник воду.
- 3 Довести воду до кипения и снять с огня.
- 4 Всыпать в чайник чай.
- 5 Довести воду до кипения (но не кипятить), снять с огня.
- 6 Чай готов. Процесс прекратить.

1.2 Основные свойства алгоритмов

Основными свойствами алгоритмов являются:

– универсальность (массовость) – применимость алгоритма к различным наборам исходных данных;

– дискретность – процесс решения задачи по алгоритму разбит на отдельные действия;

– однозначность – правила и порядок выполнения действий алгоритма имеют единственное толкование;

– конечность – каждое из действий и весь алгоритм в целом обязательно завершаются;

– результативность – по завершении выполнения алгоритма обязательно получается конечный результат;

– выполнимость – результата алгоритма достигается за конечное число шагов.

Алгоритм считается правильным, если его выполнение дает правильный результат.

Выделяют три крупных класса алгоритмов:

1) вычислительные алгоритмы – работающие с числами и матрицами, хотя сам процесс вычисления может быть долгим и сложным;

2) информационные алгоритмы – работающие с большими объемами информации (алгоритмы баз данных);

3) управляющие алгоритмы, генерирующие различные управляющие воздействия на основе данных, полученных от внешних процессов, которыми алгоритмы управляют.

1.3 Способы записи алгоритмов


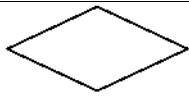
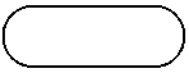
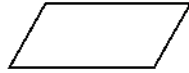
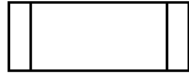
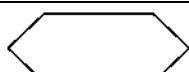
Выделяют следующие основные способы записи алгоритмов:

- вербальный, когда алгоритм описывается на человеческом языке;
- символьный, когда алгоритм описывается с помощью набора символов;
- графический, когда алгоритм описывается с помощью набора графических изображений.

Общепринятыми способами записи являются графическая запись с помощью блок-схем и символьная запись с помощью какого-либо алгоритмического языка.

Описание алгоритма с помощью блок схем осуществляется рисованием последовательности геометрических фигур, каждая из которых подразумевает выполнение определенного действия алгоритма. Далее приводятся изображения символов схемы программы по ГОСТ 19.701–90. Внешний вид основных блоков, применяемых при написании блок схем, приведен в таблице 1.1.

Таблица 1.1 – Графические изображения элементов схем алгоритмов

Номер	Изображение	Наименование этапа
1		Символ «Процесс» Он предназначен для изображения вычислительного действия или последовательности вычислительных действий, которые указываются внутри блока
2		Символ «Решение». Предназначен для проверки условия и передачи управления в соответствии с ним. Внутри блока должны быть указаны вопрос, решение, условие или сравнение
3		Символ «Терминатор»
4		Символ «Данные (ввод/вывод)»
5		Символ предопределенного процесса (вызова подпрограммы)
6		Символ «Подготовка (модификация)»

Окончание таблицы 1.1

Номер	Изображение	Наименование этапа
7		Символы «Границы цикла»
8		Символ «Соединитель»
9		Символ «Комментарий»

1.4 Основные алгоритмические конструкции

Линейный алгоритм – алгоритм или фрагмент алгоритма, в котором порядок исполнения инструкций соответствует порядку их записи.

Инструкции линейного алгоритма выполняются последовательно одна за другой в порядке их записи. Блок-схему линейного алгоритма обычно представляют в виде блок-символов, соединенных последовательно. В каждый блок-символ входит не более одной линии потока информации. Из каждого блок-символа выходит не более одной линии потока информации. Обычно блок-схему размещают таким образом, что блок-символы размещаются один под другим и нет необходимости обозначать линии потока информации стрелками.

Допускается изображать линейный алгоритм горизонтально или в виде изгибающейся под прямыми углами цепочки блок-символов. В этом случае линии потока информации, идущие снизу вверх и справа налево, следует изображать в виде стрелок.

Линейный фрагмент имеет любой алгоритм. В алгоритме «Кипячение воды» (рисунок 1.1) можно выделить два линейных фрагмента.

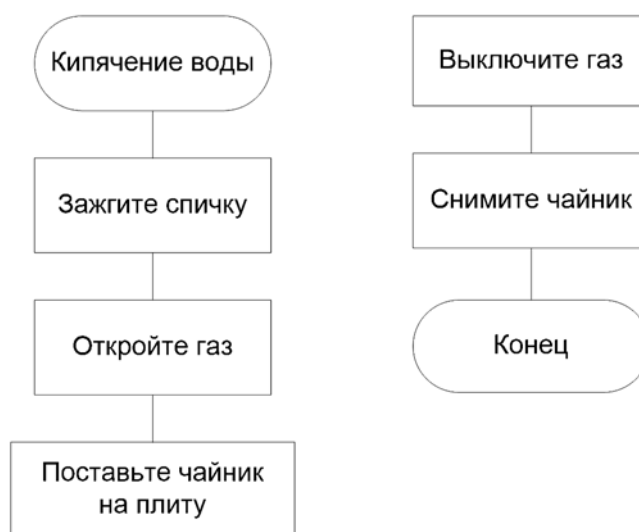


Рисунок 1.1 – Линейные фрагменты алгоритма «Кипячение воды»

Разветвляющийся алгоритм – алгоритм или фрагмент алгоритма, в котором порядок исполнения инструкций не определен изначально, имеет не менее двух вероятных направлений, каждое из которых соответствует одному из возможных исходов проверки логического условия.

Своим ветвлением разветвляющийся алгоритм обязан логическому условию, проверка которого на истинность обязательно приводит к возможности реализации одного из двух взаимоисключающих исходов (рисунок 1.2, а). Комбинируя проверку нескольких логических условий, можно получить многоуровневое ветвление с тремя и более вероятными направлениями (рисунок 1.2, б). Линии потока информации, выходящие из блока «Условие», могут быть отмечены парами противоположных по смыслу слов («Да», «Нет») или знаками («+», «-»):



а – стандартное ветвление; б – двухуровневое ветвление

Рисунок 1.2 – Разветвляющийся алгоритм

Циклический алгоритм – фрагмент алгоритма с ветвлением, в котором инструкция или группа инструкций исполняются более одного раза, т. е. повторяются.

Тело цикла – группа повторяющихся инструкций.

Основой циклического алгоритма является повторение инструкции или группы инструкций – тела цикла. Начало и завершение повторения тела цикла определяется итогом проверки логического условия.

Как правило, циклический алгоритм является фрагментом более сложного алгоритма. На рисунке 1.3 представлен фрагмент алгоритма «Кипячение воды». Инструкция «Ждите 1 мин» является телом цикла и будет повторяться до тех пор, пока вода не закипит.

Циклический алгоритм имеет две базовые структуры – *цикл с постусловием* и *цикл с предусловием*.

Циклический алгоритм с постусловием рисунок 1.4 характеризуется тем, что проверка условия расположена после тела цикла.

Циклический алгоритм с предусловием характеризуется тем, что проверка условия расположена перед телом цикла (рисунок 1.5).

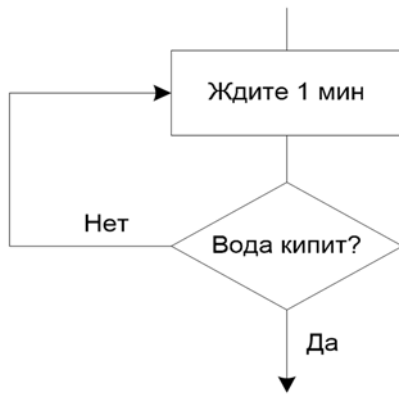


Рисунок 1.3 – Циклический фрагмент алгоритма «Кипячение воды»



Рисунок 1.4 – Циклический алгоритм с постусловием

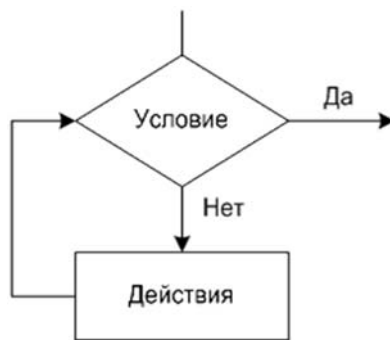


Рисунок 1.5 – Циклический алгоритм с предусловием

1.5 Примеры алгоритмов

Составить блок-схему алгоритма программы, которая запрашивает с клавиатуры целое число N . Если это число больше 10, то вычисляет и выводит на экран произведение всех целых чисел от 1 до N , иначе выводит на экран значение N .

Блок-схема алгоритма решения задачи показана на рисунке 1.6. В блоке 1 осуществляется ввод исходных данных (N). В блоке 2 производится проверка условия $N > 10$. В случае выполнения условия в блоке 3 переменной prN присваивается начальное значение, равное 1, далее в цикле (блоки 5, 6) вычисляется произведение всех целых чисел от 1 до N . В блоке 7 выводится на экран значение полученного произведения и завершается работа программы. Если условие $N > 10$ не выполняется, на экран выводится значение N (блок 4) и завершается работа программы.

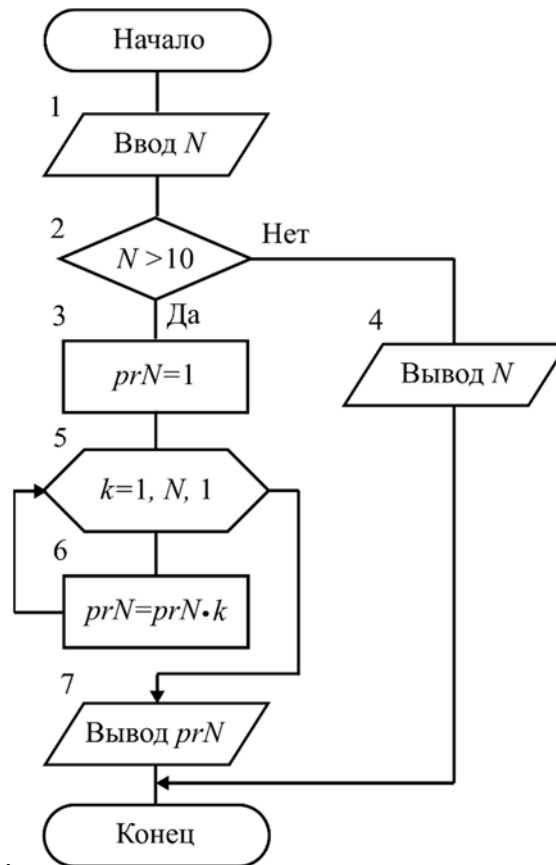


Рисунок 1.6 – Схема алгоритма

1.6 Индивидуальные задания

Задание 1

Составить блок-схему алгоритма решения задачи с условным переходом. Необходимо рассчитать значение искомой переменной по одному из двух альтернативных выражений в зависимости от переменной условия, значение которого необходимо предварительно вычислить согласно заданию. Значения переменной условия и переменной результата должны выводиться на экран. Исходные данные к заданию находятся в таблице 1.2.

Таблица 1.2 – Исходные данные к заданию 1

Но-мер	Данные 1	Данные 2	Переменная условия	Условие выполняется	Условие не выполняется
1	$A_1 = 2,35$	$A_2 = 5,89$	$B = (A_1^2 - A_2) > 0$	$C = 3,1B - A_1^2 A_2$	$C = 2,17B + A_1 A_2$
2	$X_1 = -8,4$	$X_2 = 1,84$	$Y = (4X_1 - X_2) < 0$	$Z = 5Y^3 - X_2 + 2X_1$	$Z = X_1 + X_2 - 5Y^2$
3	$E_1 = 7,54$	$E_2 = 3,62$	$F = (4X_2 - X_1) > 0$	$Y = 4F / (E_1 - E_2)$	$Y = 4F(E_2 - E_1)$
4	$B_1 = -6,7$	$B_2 = 4,57$	$E = (B_1 - B_2^2) < 0$	$X = 2E - B_1 + B_2^2$	$X = 2E - B_2 + B_1^2$
5	$D_1 = -9,08$	$D_2 = 6,35$	$C = (D_1^2 - D_2^2) > 0$	$B = 2,1D_1 - 3,4C$	$B = 1,2D_2 + 3,4C$
6	$C_1 = 3,26$	$D_2 = 5,41$	$X = (C_1 - C_2) < 0$	$E = 7,5X - C_1 / C_2$	$E = C_1 / C_2 - 7,5X$

Окончание таблицы 1.2

Но- мер	Данные 1	Данные 2	Переменная условия	Условие выполняется	Условие не выполняется
7	$F_1 = 5,66$	$F_2 = 3,48$	$A = (7F_1^2 - F_2) > 0$	$D = 3,1A - 7F_1F_2$	$D = 7F_1F_2 - 3,1A$
8	$H_1 = -0,9$	$H_2 = 7,94$	$D = (H_1 / H_2^2) < 0$	$F = 4D - 4,8H_1H_2$	$F = 8H_1H_2 - 6,2D$
9	$T_1 = 1,63$	$T_2 = -9,18$	$H = (3T_1^2 - T_2^2) > 0$	$A = H / (3,8T_1 - T_2)$	$A = H / (T_2 - 3,8T_1)$
10	$J_1 = -4,32$	$J_2 = 0,79$	$C = (J_1J_2 - 8) < 0$	$I = 3C - 4,1(J_1J_2)^2$	$I = 4(J_1J_2)^2 - 3C$
11	$K_1 = 7,45$	$K_2 = -3,8$	$J = (K_1 - 5K_2^2) > 0$	$H = 2,5J - K_1K_2$	$H = K_1K_2 - 2,5J$
12	$L_1 = -2,71$	$L_2 = 5,63$	$K = (L_1^2 / L_2) < 0$	$J = 9K - 3L_1 + L_2$	$J = 3L_1 - L_2 - 9K$
13	$M_1 = 5,23$	$M_2 = -1,1$	$L = (M_1 - M_2) > 0$	$K = L - 5,4M_1M_2$	$K = 5,4M_1M_2 - L$
14	$N_1 = -3,6$	$N_2 = 4,47$	$M = (N_1^2 - N_2) < 0$	$L = M / (N_1 - N_2)$	$L = M / (N_1 + N_2)$
15	$P_1 = 8,19$	$P_2 = 2,34$	$N = (2P_1 / P_2^2) > 0$	$M = (3P_1 - P_2) / N$	$M = (3P_2 - P_1) / N$

Задание 2

Составить блок-схему алгоритма решения нижеприведенной задачи.

1 Даны три целых числа. Найти количество положительных чисел в исходном наборе.

2 Даны три целых числа. Найти количество положительных и количество отрицательных чисел в исходном наборе.

3 Даны две переменные вещественного типа: А, В. Перераспределить значения данных переменных так, чтобы в А оказалось меньшее из значений, а в В – большее. Вывести новые значения переменных А и В.

4 Даны две переменные целого типа: А и В. Если их значения не равны, то присвоить каждой переменной сумму этих значений, а если равны, то присвоить переменным нулевые значения. Вывести новые значения переменных А и В.

5 Даны две переменные целого типа: А и В. Если их значения не равны, то присвоить каждой переменной большее из этих значений, а если равны, то присвоить переменным нулевые значения. Вывести новые значения переменных А и В.

6 Даны три числа. Найти наименьшее из них.

7 Даны три числа. Найти среднее из них (т. е. число, расположенное между наименьшим и наибольшим).

8 Даны три числа. Вывести вначале наименьшее, а затем наибольшее из данных чисел.

9 Даны три числа. Найти сумму двух наибольших из них.

10 Даны три переменные вещественного типа: А, В, С. Если их значения упорядочены по возрастанию, то удвоить их; в противном случае заменить значение каждой переменной на противоположное. Вывести новые значения переменных А, В, С.

11 Даны три переменные вещественного типа: А, В, С. Если их значения упорядочены по возрастанию или убыванию, то удвоить их; в противном случае

заменить значение каждой переменной на противоположное. Вывести новые значения переменных А, В, С.

12 Даны три целых числа, одно из которых отлично от двух других, равных между собой. Определить порядковый номер числа, отличного от остальных.

13 Даны четыре целых числа, одно из которых отлично от трех других, равных между собой. Определить порядковый номер числа, отличного от остальных.

14 На числовой оси расположены три точки: А, В, С. Определить, какая из двух последних точек (В или С) расположена ближе к А, и вывести эту точку и ее расстояние от точки А.

15 Даны координаты точки, не лежащей на координатных осях ОХ и ОУ. Определить номер координатной четверти, в которой находится данная точка.

Задание 3

Составить блок-схему алгоритма решения циклической задачи вычисления значения функции $y = f(x)$ в зависимости от значения аргумента x , изменяющегося от начального значения x_0 до конечного значения x_k с заданным шагом Δx . Исходные данные к заданию находятся в таблице 1.3.

Таблица 1.3 – Исходные данные к заданию 3

Но-мер	Начальное значение x_0	Конечное значение x_k	Значение шага Δx	Функция $y = f(x)$
1	-5	5	0,5	$y = \begin{cases} \ln(x^2), & x_0 \leq x < 0,5 \\ \operatorname{tg}(x^3) + x, & 0,5 \leq x \leq x_k \end{cases}$
2	1	4	0,01	$y = \begin{cases} 2,31x^2 + 12,47x + 17, & x_0 \leq x < 2 \\ 34x^3 + 15x^2 + 15x, & 2 \leq x \leq x_k \end{cases}$
3	-0,5	0,5	0,01	$y = \begin{cases} \cos(2\pi + x), & x_0 \leq x < 0 \\ \sin(2\pi + x), & 0 \leq x \leq x_k \end{cases}$
4	-2,5	5	0,05	$y = \begin{cases} -0,67x^4 - 2x - 123, & x_0 \leq x \leq 4 \\ 4,4x + 5,9x^2 + 6,7x^3, & 4 < x \leq x_k \end{cases}$
5	-0,4	0,4	0,01	$y = \begin{cases} \cos(\pi - x), & x_0 \leq x < 0 \\ \sin(\pi - x), & 0 \leq x \leq x_k \end{cases}$
6	1,5	4,5	0,1	$y = \begin{cases} (x + 1,25)^2 - \sqrt[3]{ x }, & x_0 \leq x < 2,5 \\ x^3 + \sin(x), & 2,5 \leq x \leq x_k \end{cases}$
7	0,1	0,9	0,05	$y = \begin{cases} \operatorname{tg}(\pi x), & x_0 \leq x < 0,2 \\ \operatorname{ctg}(\pi x), & 0,2 \leq x \leq x_k \end{cases}$
8	-0,3	0,3	0,01	$y = \begin{cases} 3e^{-0,05} \cos(\pi + x), & x_0 \leq x < 0 \\ 2e^{-0,1} \sin(\pi + x), & 0 \leq x \leq x_k \end{cases}$

Окончание таблицы 1.3

Но- мер	Начальное значение x_0	Конечное значение x_k	Значение шага Δx	Функция $y = f(x)$
9	1	8	1	$y = \begin{cases} x^3 + 5, & x_0 \leq x < 2 \\ 10,2 + \sqrt[3]{x}, & 2 \leq x \leq x_k \end{cases}$
10	-1	3	0,01	$y = \begin{cases} (\sqrt{ \sin(x) } + 2,5x)^2, & x_0 \leq x < 1,75 \\ 1,23x^2 + 0,15x, & 1,75 \leq x \leq x_k \end{cases}$
11	1	8	0,2	$y = \begin{cases} 10 \cos(\pi x / 2), & x_0 < x < 0 \\ 20 \sin(\pi x / 2), & 0 \leq x < x_k \end{cases}$
12	1	7	0,2	$y = \begin{cases} e^{-1,34x} / \sqrt{x^2 + 1,2}, & x_0 \leq x < 1,25 \\ \sqrt{2,2x} + \cos(2x), & 1,25 \leq x \leq x_k \end{cases}$
13	0,51	2	0,01	$y = \begin{cases} 1,67x^4 - 12,51, & x_0 \leq x < 1,34 \\ 10,2 + e^x, & 1,34 \leq x \leq x_k \end{cases}$
14	1	10	1	$y = \begin{cases} e^{-x} - \sqrt{100 - x}, & x_0 < x < 5,5 \\ \sqrt{x^2 + 10,25} + \operatorname{tg}(x), & 5,5 \leq x \leq x_k \end{cases}$
15	0,15	0,9	0,05	$y = \begin{cases} \ln(x^2), & x_0 \leq x < 0,5 \\ \operatorname{tg}(x^3) + x, & 0,5 \leq x \leq x_k \end{cases}$

Задание 4

Составить блок-схему алгоритма решения нижеприведенной задачи.

1 Дано целое число $N > 0$. Найти сумму $1 + 1/2 + 1/3 + \dots + 1/N$.

2 Дано целое число $N > 0$. Найти сумму $N^2 + (N + 1)^2 + (N + 2)^2 + \dots + (N + N)^2$.

3 Дано целое число $N > 0$. Найти произведение $1 \cdot 2 \cdot 3 \dots (N - 1) \cdot N$.

4 Дано целое число $N > 0$. Найти значение выражения $1,1 - 1,2 + 1,3 - \dots$ (N слагаемых, знаки чередуются).

5 Дано вещественное число A и целое число $N > 0$. Найти A в степени N (числа A перемножаются N раз).

6 Дано вещественное число A и целое число $N (> 0)$. Найти сумму $1 + A + A^2 + A^3 + \dots + A^N$.

7 Дано вещественное число A и целое число $N > 0$. Найти значение выражения $1 - A + A^2 - A^3 + \dots + (-1)^N A^N$.

8 Дано целое число $N > 0$. Найти произведение $N! = 1 \cdot 2 \cdot \dots \cdot N$ (N – факториал). Чтобы избежать целочисленного переполнения, вычислять это произведение с помощью вещественной переменной и вывести его как вещественное число.

9 Дано целое число $N > 0$. Найти сумму $1! + 2! + 3! + \dots + N!$ (выражение $N!$ – факториал числа N – обозначает произведение всех целых чисел от 1 до N). Чтобы избежать целочисленного переполнения, проводить вычисления с помощью вещественных переменных и вывести результат как вещественное число.

10 Дано целое число $N > 0$. Найти сумму $1 + 1/(1!) + 1/(2!) + 1/(3!) + \dots + 1/(N!)$ (выражение $N!$ – факториал числа N – обозначает произведение всех целых чисел от 1 до N). Полученное число является приближенным значением константы e .

11 Дано вещественное число X и целое число $N > 0$. Найти значение выражения $1 + X + X^2/(2!) + \dots + X^N/(N!)$ ($N! = 1 \cdot 2 \cdot \dots \cdot N$). Полученное число является приближенным значением функции e^x .

12 Дано вещественное число X ($|X| < 1$) и целое число $N > 0$. Найти значение выражения $X - X^2/2 + X^3/3 - \dots + (-1)^N \cdot X^{(N+1)}/(N+1)$. Полученное число является приближенным значением функции $\ln(1+x)$.

13 Дано вещественное число X ($|X| < 1$) и целое число $N > 0$. Найти значение выражения $X - X^3/3 + X^5/5 - \dots + (-1)^{N-1} \cdot X^{2N-1}/(2 \cdot N - 1)$. Полученное число является приближенным значением функции $\arctg(x)$.

14 Дано целое число $N > 1$ и две вещественные точки на числовой оси: A, B ($A < B$). Отрезок $[A, B]$ разбит на N равных отрезков. Вывести N – длину каждого отрезка, а также набор точек $A, A + H, A + 2H, A + 3H \dots B$, образующий разбиение отрезка $[A, B]$.

15 Дано целое число $N > 1$ и две вещественные точки на числовой оси: A, B ($A < B$). Отрезок $[A, B]$ разбит на N равных отрезков. Вывести N – длину отрезка, а также значения функции $F(X) = 1 - \sin(X)$ в точках, разбивающих отрезок $[A, B]$: $F(A), F(A + H), F(A + 2H) \dots F(B)$.

Контрольные вопросы

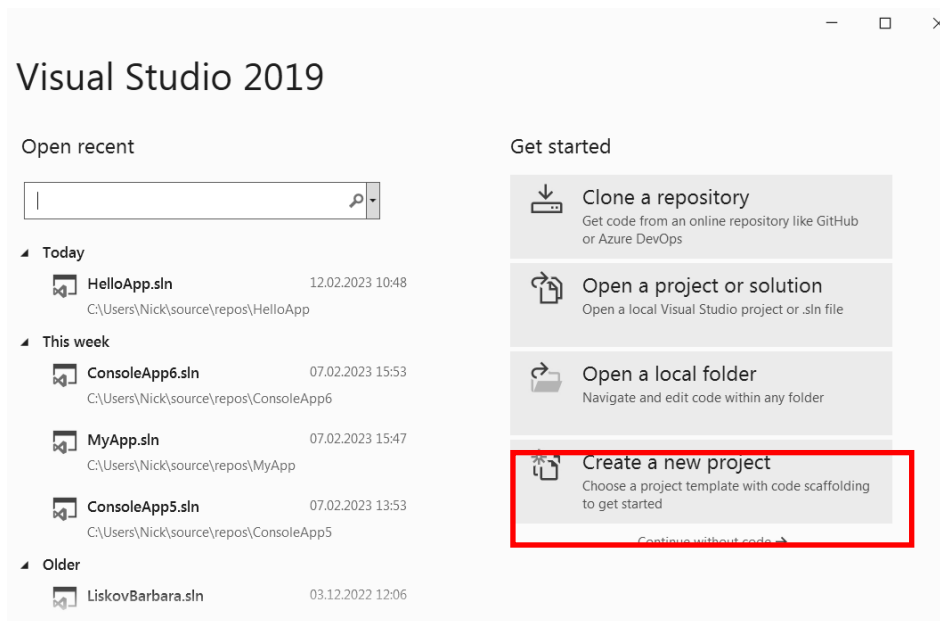
- 1 Дайте определение алгоритма.
- 2 Какими свойствами обладает алгоритм?
- 3 Каковы способы записи алгоритмов?
- 4 Перечислите основные свойства алгоритма.
- 5 Перечислите основные алгоритмические структуры и опишите их.
- 6 Что такое блок-схема?
- 7 Какие виды циклического алгоритма бывают?
- 8 Что такое тело цикла?
- 9 Для чего необходимо ветвление в алгоритмах?
- 10 Какие формы ветвления различают?

2 Лабораторная работа № 2. Среда программирования Visual Studio. Программирование линейных алгоритмов

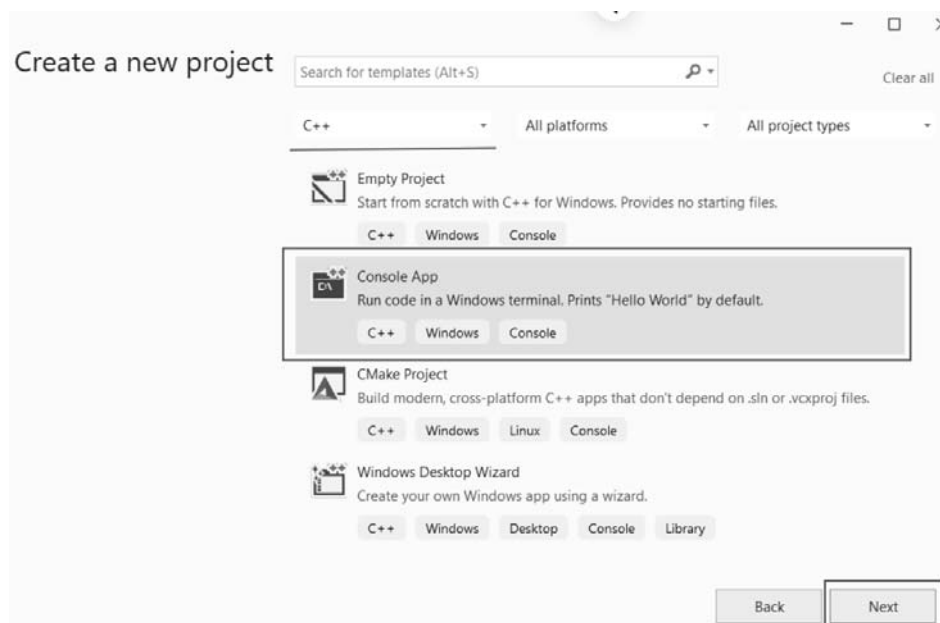
Цель работы: изучение среды программирования Microsoft Visual Studio 2019; получение навыков создания консольных приложений и программирования линейных алгоритмов.

2.1 Создание проекта консольного приложения

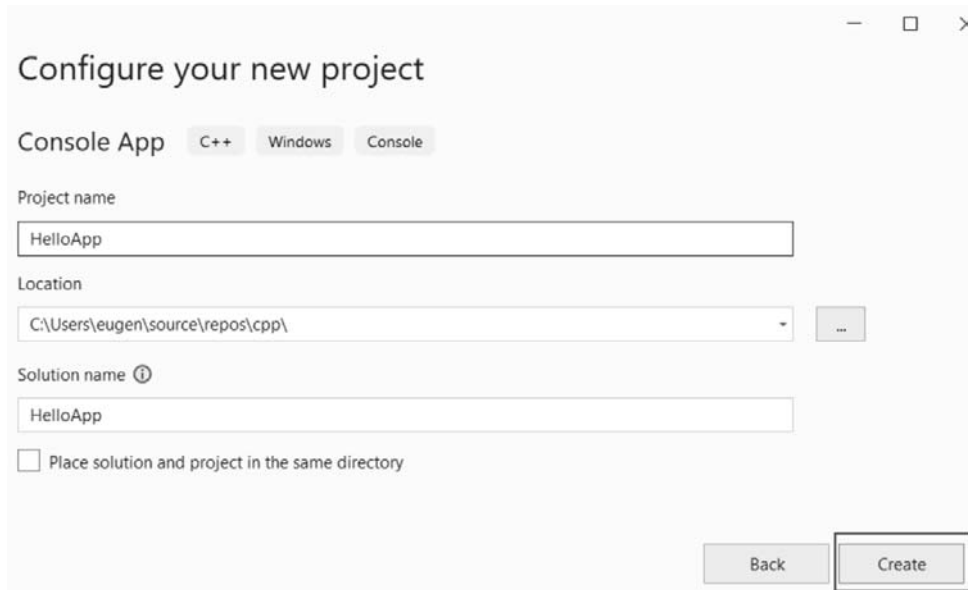
Запускаем среду Visual Studio 2019 и в стартовом окне выбираем Create a new project.



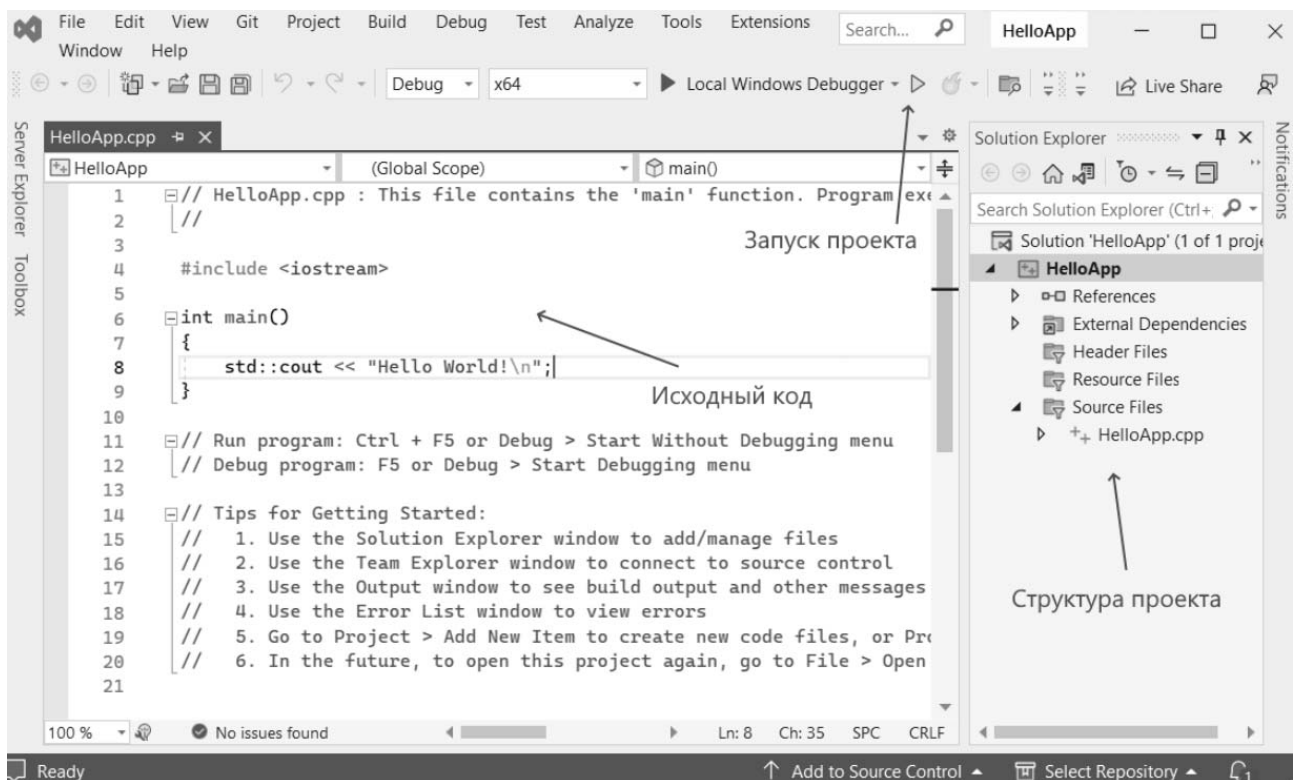
В появившемся окне выбираем язык программирования C++ и среди шаблонов проектов выбираем тип Console App.



В следующем окне в поле Project name вводим имя HelloApp, в поле Location указываем расположение проекта и нажимаем на Create для создания проекта.



После этого Visual Studio создаст типовой проект консольного приложения на C++.



Справа в окне Solution Explorer отображается структура проекта. В реальности окно Solution Explorer содержит решение. В данном случае оно называется HelloApp. Решение может содержать несколько проектов. По умолчанию имеется один проект с таким же именем – HelloApp. В проекте есть ряд каталогов:

- External Dependencies – содержит файлы, которые используются в файлах исходного кода, но не являются частью проекта;
- Header Files – предназначен для хранения заголовочных файлов с расширением .h;
- Resource Files – предназначен для хранения файлов ресурсов, например, изображений;
- Source Files – хранит файлы с исходным кодом проекта.

Каталог Source Files содержит файл HelloApp.cpp (название проекта + расширение файла .cpp). Этот файл виден слева в текстовом редакторе Visual Studio. По умолчанию он содержит следующий код:

```

1 | #include <iostream>
2 |
3 | int main()
4 | {
5 |     std::cout << "Hello World!\n";
6 | }
```

Здесь `iostream` – стандартный заголовочный файл библиотеки ввода-вывода. Эта библиотека содержит информацию о потоке `cout`, используемом в данной программе. `#include` является директивой препроцессора, заставляющей включить в программу текст из заголовочного файла `iostream`.

Функция `main()` – эта главная функция программы. Она вызывается операционной системой при запуске программы. Фигурные скобки `{ }` отмечают начало и конец тела функции `main()`.

В теле функции `main()` определена инструкция `std::cout << "Hello, world!\n"`, которая выводит в окно консоли сообщение `Hello, world!`.

`cout` – это выходной поток, направленный на терминал, `<<` – оператор вставки данных (строки `"Hello, world!"`) в выходной поток.

Префикс `std::` указывает, что объект `cout` определен в стандартном пространстве имен `std`.

Чтобы каждый раз не указывать название пространства, можно импортировать все идентификаторы в глобальное пространство имен с помощью инструкции `using namespace std`. В этом случае название пространства имен при обращении к объекту `cout` указывать не нужно:

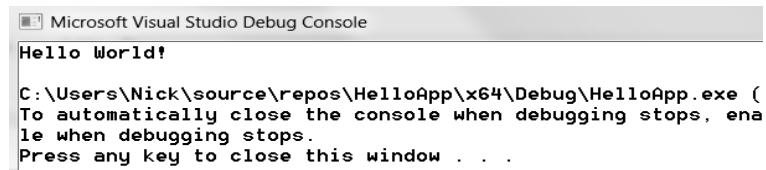
```

1 | #include <iostream>
2 | using namespace std;
3 |
4 | int main()
5 | {
6 |     cout << "Hello World!\n";
7 | }
```

Последовательность `\n` представляет собой символ перехода на новую строку.

2.2 Компиляция и запуск программы

Для компиляции и запуска программы нужно нажать сочетание клавиш Ctrl + F5 или выбрать пункт меню Debug → Start Without Debugging. В итоге Visual Studio передаст исходный код программы компилятору, который создаст из кода исполняемый файл .exe, который будет запущен на выполнение. В окне консоли появится сообщение Hello Word!.



```
Microsoft Visual Studio Debug Console
Hello World!
C:\Users\Nick\source\repos\HelloApp\x64\Debug\HelloApp.exe (
To automatically close the console when debugging stops, enable
when debugging stops.
Press any key to close this window . . .
```

Если в процессе компиляции были обнаружены синтаксические ошибки, то выводится соответствующее сообщение в окно Output. В этом случае необходимо последовательно исправлять ошибки и компилировать проект снова. Если синтаксических ошибок нет, но результат выполнения программы неверный, необходимо искать логические ошибки. Для этого следует использовать встроенный в Visual Studio отладчик (см. лабораторную работу № 3).

Задание

Удалить точку с запятой в 6-й строке файла HelloApp.cpp, затем запустить программу, найти в окне Output описание ошибки, исправить ее и снова запустить программу.

2.3 Переменные и типы данных

Типы данных определяют способ хранения чисел или символов в памяти компьютера. Они задают размер ячейки, в которую будет записано то или иное значение, определяя тем самым его максимальную величину. К основным типам данных языка C++ относят:

- int – целочисленный тип данных;
- float – тип данных с плавающей запятой;
- double – тип данных с плавающей запятой двойной точности;
- char – символьный тип данных;
- bool – логический тип данных.

Переменная – это именованный участок памяти, в котором хранится значение определенного типа. У переменной есть имя (идентификатор) и значение. Имя служит для обращения к области памяти, в которой хранится значение. Перед использованием любая переменная должна быть объявлена. Объявление переменной происходит таким образом: сначала указывается тип данных для этой переменной, а затем имя этой переменной.

Примеры объявления переменных.

int t; // объявление переменной t целого типа.

float b; // объявление переменной b типа данных с плавающей запятой.

double w = 14.2; // объявление и инициализация переменной w типа double.

`char d = 's';` // объявление и инициализация переменной `d` типа `char`.

`bool k = true;` // объявление и инициализация логической переменной `k`.

В C++ оператор присваивания (`=`) – не является знаком равенства и не может использоваться для сравнения значений. Оператор равенства записывается как «двойное равно» (`==`).

Присваивание используется для сохранения определенного значения в переменной. Например, запись вида `t = 10` задает переменной `t` значение числа 10.

2.4 Объектно-ориентированные средства ввода-вывода

Описание объектов для управления вводом выводом содержится в заголовочном файле `iostream`. При подключении этого файла с помощью директивы препроцессора `#include <iostream>` в программе автоматически создаются объекты-потоки – `cin` для ввода с клавиатуры и `cout` для вывода на экран, а также операции помещения в поток `<<` и чтения из потока `>>`.

С помощью объекта `cin>>` можно присвоить значение любой переменной. Например, если переменная `i` объявлена как целочисленная, то инструкция `cin>> i;` означает, что в переменную `i` будет записано некое целое число, введенное с клавиатуры. Если нужно ввести несколько переменных, следует написать `cin>>x>>y>>z;`.

Объект `cout` и операция `<<` позволяют вывести на экран значение любой переменной или текст. Текст необходимо заключать в двойные кавычки. Кроме того, допустимо применение символов управляющих последовательностей `\t` и `\n`. Инструкция `cout<<i;` означает вывод на экран значения переменной `i`, а инструкция `cout<<x<<"\t"<<y;` выведет на экран значения переменных `x` и `y`, разделенных символом табуляции.

Рассмотрим следующую задачу. Известны плотность ρ , высота h и радиус основания R цилиндрического слитка, полученного в металлургической лаборатории. Найти объем V , массу m и площадь S основания слитка.

Входные данные программы: ρ , h , R . Выходные данные: S , V , m . Учтывая, что $S = 2\pi R$, $V = \pi R^2 h$ и $m = \rho V$, набираем следующий текст программы:

```

1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      setlocale(0, ""); // Функция обеспечивает вывод русских символов в консоль
7
8      double R, h, rho, S, V, m; // Объявления переменных
9      const double pi = 3.14159; // Определение числа pi
10
11     cout << "R = ";           // Вывод на экран символов R =
12     cin >> R;                 // Ввод значения переменной R
13
14     cout << "h = ";           // Вывод на экран символов h =
15     cin >> h;                 // Ввод значения переменной h
16
17     cout << "rho = ";         // Вывод на экран символов rho =
18     cin >> rho;               // Ввод значения переменной rho
19
20     S = 2 * pi * R;           // Вычисление площади
21     V = pi * R * R * h;       // Вычисление объема
22     m = rho * V;              // Вычисление массы
23
24     cout << "\nПлощадь S = " << S; // Вывод на экран символов S= и значения переменной S
25     cout << "\nОбъем V = " << V; // Вывод на экран с новой строки V= и значения V
26     cout << "\nМасса m = " << m << "\n"; // Вывод на экран с новой строки m= и значения m
27 }
--

```

Исполнение программы начинается с выполнения первой инструкции функции `main()`, в данном случае – вызовом функции `setlocale()`. Затем одна за другой исполняются все дальнейшие инструкции, и, выполнив последнюю инструкцию функции `main()`, программа заканчивает работу.

2.5 Функции библиотеки *math*

Функции для расчета математических выражений находятся в библиотеке `math.lib` (подключение библиотеки: `#include <math.h>`). Все аргументы в тригонометрических функциях задаются в радианах. Параметры и аргументы всех остальных функций имеют тип `double` (кроме `abs(x)`) (таблица 2.1).

Таблица 2.1 – Основные функции библиотеки `math.lib`

Математическая функция	Функция библиотеки <code>math.lib</code>	Описание
$ x $	<code>abs(x)</code>	Вычисление абсолютного значения (только для целых чисел!)
$\arccos(x)$	<code>acos(x)</code>	Вычисление значения арккосинуса x
$\arctg(x)$	<code>atan(x)</code>	Вычисление значения арктангенса x
$\arctg(x/y)$	<code>atan2(x,y)</code>	Вычисление значения арктангенса двух аргументов x и y
Округление к большему	<code>ceil(x)</code>	Функция возвращает действительное значение, соответствующее наименьшему целому числу, которое больше или равно x
$\cos(x)$	<code>cos(x)</code>	Вычисление косинуса x
$\operatorname{ch}(x) = (e^x + e^{-x})/2$	<code>cosh(x)</code>	Вычисление косинуса гиперболического x
e^x	<code>exp(x)</code>	Вычисление экспоненты числа x
$ x $	<code>fabs(x)</code>	Вычисление абсолютного значения x
Округление к меньшему	<code>floor(x)</code>	Функция возвращает действительное значение, соответствующее наибольшему целому числу, которое меньше или равно x
Остаток от деления x на y	<code>fmod(x,y)</code>	Функция возвращает действительное значение, соответствующее остатку от целочисленного деления x на y
$\ln(x)$	<code>log(x)</code>	Вычисление натурального логарифма x
$\lg_{10}(x)$	<code>log10(x)</code>	Вычисление десятичного логарифма x
x^y	<code>pow(x, y)</code>	Возведение x в степень y
$\sin(x)$	<code>sin(x)</code>	Вычисление синуса x
$\operatorname{sh}(x) = (e^x - e^{-x})/2$	<code>sinh(x)</code>	Вычисление синуса гиперболического x
\sqrt{x}	<code>sqrt(x)</code>	Вычисление квадратного корня x
$\operatorname{tg}(x)$	<code>tan(x)</code>	Вычисление тангенса x
$\operatorname{tgh}(x)$	<code>tanh(x)</code>	Вычисление тангенса гиперболического x

2.6 Индивидуальные задания

Вычислить значение выражения при заданных исходных данных. Сравнить полученное значение с указанным правильным результатом.

$$1 \quad s = \frac{2 \cos\left(x - \frac{2}{3}\right)}{\frac{1}{2} + \sin^2 y} \left(1 + \frac{z^2}{3 - z^2/5}\right). \text{ При } x = 14,26; y = -1,22; z = 3,5 \cdot 10^{-2} \text{ ответ}$$

$$s = 0,749155.$$

$$2 \quad s = \frac{\sqrt[3]{9 + (x - y)^2}}{x^2 + y^2 + 2} - e^{|x-y|} \operatorname{tg}^3 z. \text{ При } x = -4,5; y = 0,75 \cdot 10^{-4}; z = -0,845 \cdot 10^2$$

$$\text{ответ } s = -3,23765.$$

$$3 \quad s = \frac{1 + \sin^2(x + y)}{\left|x - \frac{2y}{1 + x^2 y^2}\right|} x^{|y|} + \cos^2\left(\operatorname{arctg} \frac{1}{z}\right). \text{ При } x = 3,74 \cdot 10^{-2}; y = -0,825;$$

$$z = 0,16 \cdot 10^2 \text{ ответ } s = 1,05534.$$

$$4 \quad s = |\cos x - \cos y|^{(1+2\sin^2 y)} \left(1 + z + \frac{z^2}{2} + \frac{z^3}{3} + \frac{z^4}{4}\right). \text{ При } x = 0,4 \cdot 10^4; y = -0,875;$$

$$z = -0,475 \cdot 10^{-3} \text{ ответ } s = 1,98727.$$

$$5 \quad s = \ln\left(y^{-\sqrt{|x|}}\right) \left(x - \frac{y}{2}\right) + \sin^2(\operatorname{arctg}(z)). \text{ При } x = -15,246; y = 4,642 \cdot 10^{-2};$$

$$z = 21 \text{ ответ } s = -182,038.$$

$$6 \quad s = \sqrt{10(\sqrt[3]{x} + x^{y+2})} (\arcsin^2 z - |x - y|). \text{ При } x = 16,55 \cdot 10^{-3}; y = -2,75;$$

$$z = 0,15 \text{ ответ } s = -40,6307.$$

$$7 \quad s = 5 \operatorname{arctg}(x) - \frac{1}{4} \arccos(x) \frac{x + 3|x - y| + x^2}{|x - y|z + x^2}. \text{ При } x = 0,1722; y = 6,33;$$

$$z = 3,25 \cdot 10^{-4} \text{ ответ } s = -205,306.$$

$$8 \quad s = \frac{e^{|x-y|} |x-y|^{x+y}}{\operatorname{arctg}(x) + \operatorname{arctg}(z)} + \sqrt[3]{x^6 + \ln^2 y}. \text{ При } x = -2,235 \cdot 10^{-2}; y = 2,23;$$

$$z = 15,221 \text{ ответ } s = 39,3741.$$

$$9 \quad s = \left|x^{\frac{y}{x}} - \sqrt[3]{\frac{y}{x}}\right| + (y - x) \frac{\cos y - \frac{z}{(y-x)}}{1 + (y-x)^2}. \text{ При } x = 1,825 \cdot 10^2; y = 18,225;$$

$$z = -3,298 \cdot 10^{-2} \text{ ответ } s = 1,21308.$$

$$10 \quad s = 2^{-x} \sqrt{x + 4\sqrt{|y|}} \sqrt[3]{e^{x-1/\sin z}}. \text{ При } x = 3,981 \cdot 10^{-2}; y = -1,625 \cdot 10^3; z = 0,512$$

$$\text{ответ } s = 1,26185.$$

$$11 \quad s = y^{\sqrt[3]{|x|}} + \cos^3(y) \frac{|x-y| \left(1 + \frac{\sin^2 z}{\sqrt{x+y}} \right)}{e^{|x-y|} + \frac{x}{2}}. \text{ При } x = 6,251; y = 0,827; z = 25,001$$

ответ $s = 0,712122$.

$$12 \quad s = 2^{(y^x)} + (3^x)^y - \frac{y \left(\arctg z - \frac{1}{3} \right)}{|x| + \frac{1}{y^2 + 1}}. \text{ При } x = 3,251; y = 0,325; z = 0,466 \cdot 10^{-4}$$

ответ $s = 4,23655$.

$$13 \quad s = \frac{\sqrt[4]{y + \sqrt[3]{x-1}}}{|x-y|(\sin^2 z + \tg z)}. \text{ При } x = 17,421; y = 10,365 \cdot 10^{-3}; z = 0,828 \cdot 10^5$$

ответ $s = 0,330564$.

$$14 \quad s = \frac{y^{x+1}}{\sqrt[3]{|y-2|} + 3} + \frac{x + \frac{y}{2}}{2|x+y|} (x+1)^{-1/\sin z}. \text{ При } x = 12,3 \cdot 10^{-1}; y = 15,4;$$

$z = 0,252 \cdot 10^3$ ответ $s = 82,8256$.

$$15 \quad s = \frac{x^{y+1} + e^{y-1}}{1 + x|y - \tg z|} \left(1 + |y-x| \right) + \frac{|y-x|^2}{2} - \frac{|y-x|^3}{3}. \text{ При } x = 2,444;$$

$y = 0,869 \cdot 10^{-2}; z = -0,13 \cdot 10^3$ ответ $s = -0,498707$.

Контрольные вопросы

1 Назовите основные этапы создания консольного приложения в среде программирования Visual Studio.

2 Перечислите основные пункты меню среды программирования Visual Studio и их назначение.

3 Какова структура программы на языке C++?

4 Для чего необходима директива препроцессора #include?

5 Каким образом можно осуществить вывод информации на экран?

6 Как можно организовать ввод данных с клавиатуры?

7 Для чего надо подключать заголовочный файл iostream?

8 С какой целью в начале кода пишется строка using namespace std; и что надо делать, если эта строка отсутствует?

9 Что надо сделать, чтобы на консольном экране отображались русские буквы?

10 С чего начинается выполнение программ C++?

11 Что является признаком завершения инструкции в C++?

12 Дайте определение переменной и константы.

13 Объясните синтаксис объявления переменных и констант.

14 Что такое тип данных? Перечислите основные типы данных C++.

15 Дайте определение линейного алгоритма.

3 Лабораторная работа № 3. Программирование ветвящихся алгоритмов

Цель работы: получение навыков программирования ветвящихся алгоритмов.

Теоретические сведения и содержание работы приведены в [1, с. 17–23].

Индивидуальные задания [1, таблица 2.2, с. 20].

Контрольные вопросы

- 1 Объясните назначение оператора ветвления if.
- 2 Опишите синтаксис и работу полной, сокращённой и вложенной форм оператора if.
- 3 Нарисуйте структурные схемы различных форм оператора if.
- 4 Объясните назначение, синтаксис, работу оператора switch.
- 5 Нарисуйте структурную схему оператора switch.
- 6 Какую базовую структуру программирования реализует оператор if?
- 7 С помощью каких операций формируется условие-выражение в операторе if?
- 8 Для чего предназначены логические выражения? В чём заключается различие между логическими и арифметическими выражениями?
- 9 Как в логическом выражении распределяются приоритеты между арифметическими, логическими операциями и операциями отношения?
- 10 К какому типу данных принадлежит результат работы операций сравнения и логических операций?
- 11 Какие значения может принимать переменная логического типа данных?
- 12 Назовите основные логические операции, приведите таблицу истинности логических операций и примеры их использования.
- 13 Какие операции отношения определены в C++? Приведите примеры их использования.

4 Лабораторная работа № 4. Разработка программ с использованием циклов

Цель работы: получение навыков написания программ с применением операторов циклов for, while, do while; приобретение практических навыков отладки программ.

Теоретические сведения и содержание работы приведены в [1, с. 24–31]. Информация о приемах отладки программ на C++ приведена на сайте <https://radioprogram.ru/post/1085>.

Индивидуальные задания: задание 1 [1, таблица 3.1, с. 27]; задание 2 [1, таблица 3.2, с. 29].

Контрольные вопросы

- 1 Назовите операторы, предназначенные для организации циклических вычислений.
- 2 Что такое тело цикла?
- 3 Объясните назначение, синтаксис, работу оператора for.
- 4 Нарисуйте структурную схему оператора for.
- 5 Какую базовую структуру программирования реализует оператор for?
- 6 Какой тип имеет выражение-условие в операторе цикла for?
- 7 Когда вычисляется завершающее выражение цикла for?
- 8 Как выполняется вложение циклов?
- 9 Какие операторы могут прервать выполнение цикла до его завершения, запланированного выражением-условием?
- 10 Назовите назначение оператора break. Где его можно применять?
- 11 Где и когда используется оператор continue?
- 12 Объясните назначение, синтаксис, работу операторов цикла while и do ... while.
- 13 Когда рекомендуется использовать операторы цикла while, do ... while, for?
- 14 Нарисуйте структурные схемы операторов цикла while и do ... while.
- 15 Какую базовую структуру программирования реализуют операторы while и do ... while?
- 16 Какое имеется различие между выполнением операторов while и do ... while?
- 17 К какому типу принадлежит результат вычисления выражения-условия в операторах цикла while и do...while? С помощью каких операций формируются условия-выражения?
- 18 В каком случае тело цикла операторов while и do...while будет выполняться бесконечно долго?
- 19 Что необходимо предпринять, чтобы избежать бесконечного цикла при использовании операторов while и do...while?
- 20 Каково назначение встроенного отладчика среды Visual Studio? На примере разработанной программы продемонстрируйте основные приёмы работы с отладчиком.

5 Лабораторная работа № 5. Использование циклов для решения задач численными методами

Цель работы: получение навыков решения прикладных задач численными методами с использованием циклических алгоритмов.

Теоретические сведения и содержание работы приведены в [1, с. 32–41].

Индивидуальные задания: задание 1 [1, таблица 4.1, с. 38]; задание 2 [1, таблица 4.2, с. 40].

Контрольные вопросы

- 1 Каким образом цикл while может имитировать цикл for?
- 2 Каким образом цикл while может имитировать цикл do-while?
- 3 В каких случаях используются операторы break, continue, exit?
- 4 Почему в языке C++ нет необходимости использовать оператор goto?

6 Лабораторная работа № 6. Разработка программ с использованием одномерных массивов

Цель работы: изучение способов объявления, инициализации, организации ввода-вывода и обработки одномерных массивов; получение навыков алгоритмизации и программирования прикладных задач с применением одномерных массивов.

Теоретические сведения и содержание работы приведены в [1, с. 42–46].

Индивидуальные задания: задание 1 [1, таблица 5.1, с. 44]; задание 2 [1, таблица 5.2, с. 45].

Контрольные вопросы

- 1 Объясните, что такое массив.
- 2 Назовите способы объявления одномерного массива.
- 3 Назовите способы инициализации одномерного массива.
- 4 Какие значения принимают элементы массива при отсутствии в его объявлении инициализатора?
- 5 Можно ли изменить размер массива после его создания?
- 6 Что такое размерность массива и спецификатор размерности массива?
- 7 Допустимо ли динамическое определение размера массива?
- 8 Как осуществляется обращение к элементам одномерного массива?
- 9 Что произойдет с программой, если индекс массива выйдет за объявленные границы?
- 10 Какое значение имеет индекс при обращении к первому элементу массива?
- 11 Какой тип может иметь индекс массива?
- 12 Объясните процесс организации прохода по элементам массива.

7 Лабораторная работа № 7. Разработка программ с использованием двумерных массивов

Цель работы: изучение способов объявления, инициализации, организации ввода-вывода и обработки двумерных массивов; получение навыков алгоритмизации и программирования прикладных задач с применением двумерных массивов.

Теоретические сведения и содержание работы приведены в [1, с. 47–54].

Индивидуальные задания: задание 1 [1, таблица 6.1, с. 51]; задание 2 [1, таблица 6.2, с. 52].

Контрольные вопросы

- 1 Дайте определение двумерного массива.
- 2 Что является аналогом двумерного массива в математике?
- 3 Назовите способы объявления двумерного массива.
- 4 Как осуществляется обращение к элементам двумерного массива?
- 5 Назовите способы инициализации двумерного массива.
- 6 Какие значения принимают элементы двумерного массива при отсутствии в его объявлении инициализатора?
- 7 Можно ли изменить размер двумерного массива после его создания?
- 8 Объясните основные правила организации вложенных циклов, перечислите способы выхода из внутреннего цикла.
- 9 Что произойдет с программой, если индексы двумерного массива выйдут за объявленные границы?
- 10 Какое значение имеют индексы при обращении к первому элементу двумерного массива?

8 Лабораторная работа № 8. Программирование задач с использованием строк

Цель работы: получение навыков написания программ для обработки строк и символьной информации.

Теоретические сведения и содержание работы приведены в [1, с. 55–69].

Индивидуальные задания: задание 1 [1, таблица 7.1, с. 65]; задание 2 [1, таблица 7.2, с. 66].

Контрольные вопросы

- 1 Что представляют собой строки?
- 2 Каким образом строки описываются и определяются?
- 3 Какие функции используются для ввода строки и чем они отличаются друг от друга?

4 Какие функции используются для вывода строки?

5 Каким образом можно выделить слово из строки?

6 Где находится описание прототипов функции обработки строк?

7 Чем строки отличаются от обычного массива символов?

8 В чем заключается недостаток использования оператора ввода >> для строк?

9 Почему не рекомендуется смешивать для ввода оператор >> и getline? Как избежать ошибок при таком смешивании?

10 Какую библиотеку надо подключать, чтобы использовать функции для объединения строк?

9 Лабораторная работа № 9. Структуры

Цель работы: получение навыков разработки программ с использованием структурного типа данных.

Теоретические сведения и содержание работы приведены в [1, с. 70–80].

Индивидуальные задания: задание 1 [1, таблица 8.1, с. 75]; задание 2 [1, таблица 8.3, с. 77].

Контрольные вопросы

1 Какова область применения структур?

2 Каким образом определяется структура?

3 Как определяются переменные типа структура?

4 Как осуществляется доступ к структурным членам?

10 Лабораторная работа № 10ю Файловый ввод/вывод. Текстовые файлы

Цель работы: получение навыков разработки программ с использованием текстовых файлов.

Теоретические сведения и содержание работы приведены в [1, с. 81–91].

Индивидуальные задания: задание 1 [1, таблица 9.4, с. 85]; задание 2 [1, таблица 9.5, с. 87].

Контрольные вопросы

1 Что включает в себя понятие файла? Как оно связано со стандартными потоками ввода-вывода?

2 Как связаны между собой понятия логического и физического файлов?

3 Что такое режим доступа? Перечислите возможные режимы доступа при работе с текстовыми файлами.

11 Лабораторная работа № 11. Файловый ввод/вывод. Двоичные файлы

Цель работы: получение навыков разработки программ с использованием двоичных файлов.

Теоретические сведения и содержание работы приведены в [1, с. 92–102].

Индивидуальные задания: задание 1 [1, таблица 10.2, с. 98]; задание 2 [1, таблица 10.3, с. 99].

Контрольные вопросы

1 Какие функции осуществляют чтение и запись при работе с двоичными файлами?

2 В чем состоят особенности чтения массивов из двоичных файлов и записи массивов в двоичные файлы?

3 Что такое последовательный и произвольный доступ к компонентам файла?

4 Как «подойти» к предпоследнему элементу файла? Приведите все возможные варианты.

12 Лабораторная работа № 12. Программирование с использованием функций

Цель работы: получение навыков разработки программ с использованием функций.

Теоретические сведения и содержание работы приведены в [1, с. 103–121].

Индивидуальные задания: задание 1 [1, таблица 11.1, с. 115]; задание 2 [1, таблица 11.2, с. 116].

Контрольные вопросы

1 Каков допустимый уровень вложенности функции в языке С?

2 Как определить функцию? Что такое прототип функции? Всегда ли обязательно объявление прототипов?

3 Как передать информацию функции?

4 В чем разница между формальными и фактическими аргументами? Где описываются аргументы?

5 Где описываются локальные переменные функции?

6 Для чего служит оператор return? Обязательно ли его использование?

7 Проверьте, все ли правильно в следующем определении функции:

```
hallo (num)
{ int num, count;
for (count =1; count <= num; num++)
cout<<"Hello, my friend!"<<endl;
}
```

Список литературы

1 **Ашарина, И. В.** Язык C++ и объектно-ориентированное программирование в C++. Лабораторный практикум: учебное пособие / И. В. Ашарина, Ж. Ф. Крупская. – Москва: Горячая линия – Телеком, 2020. – 232 с.

2 **Павловская, Т. А.** C/C++. Программирование на языке высокого уровня: учебник / Т. А. Павловская. – Санкт-Петербург: Питер, 2020. – 432 с.

3 **Навроцкий, А. А.** Основы алгоритмизации и программирования в среде Visual C++: учебно-методическое пособие / А. А. Навроцкий. – Минск: БГУИР, 2021. – 160 с.

4 **Пацей, Н. В.** Основы алгоритмизации и программирования: учебно-методическое пособие для студентов специальности «Информационные системы и технологии (издательско-полиграфический комплекс)» / Н. В. Пацей. – Минск: БГТУ, 2022. – 289 с.