

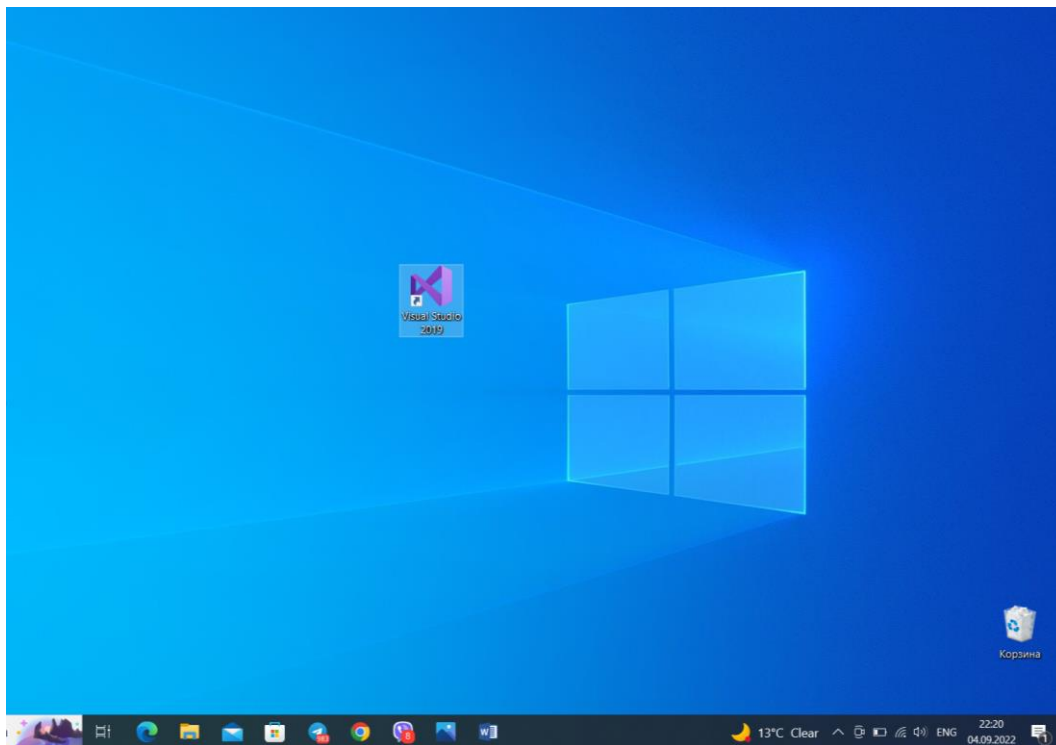
Тема 1. ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ ЛИНЕЙНОЙ СТРУКТУРЫ

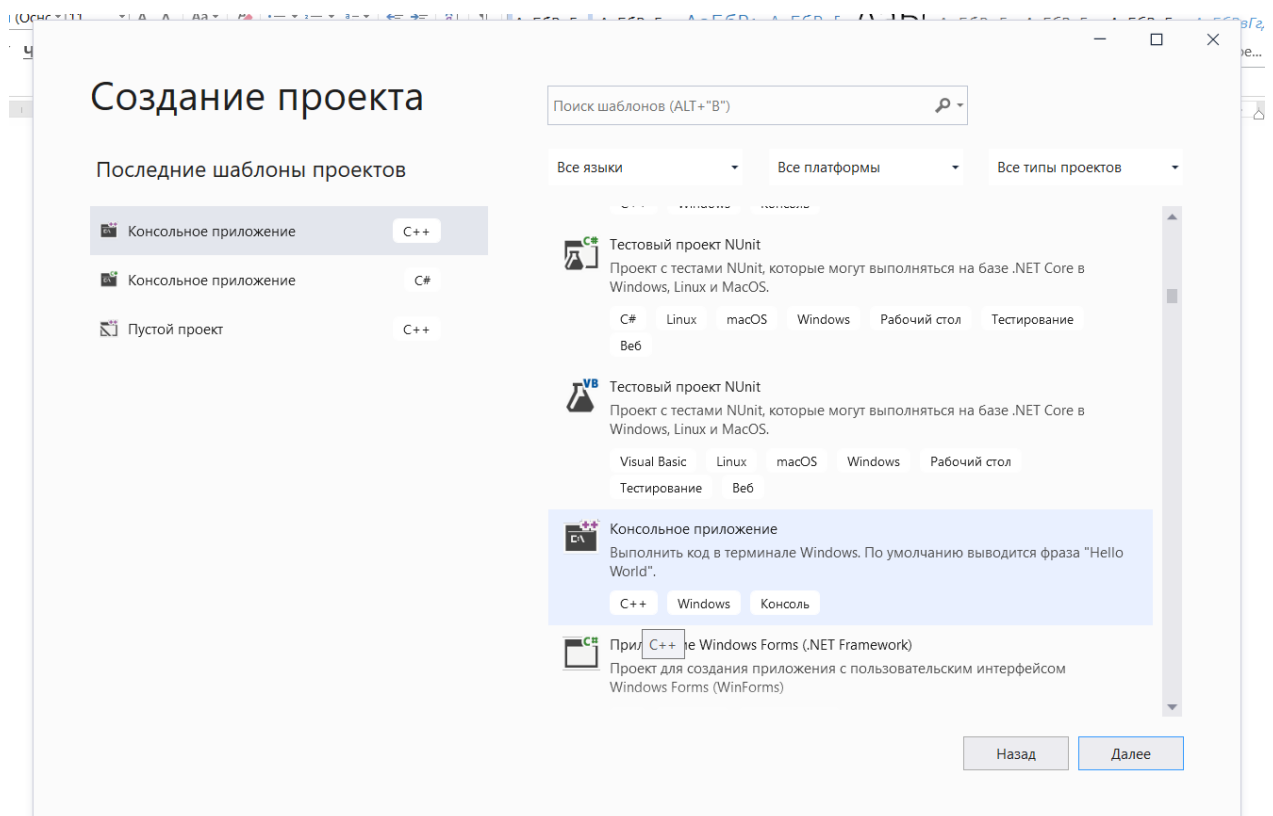
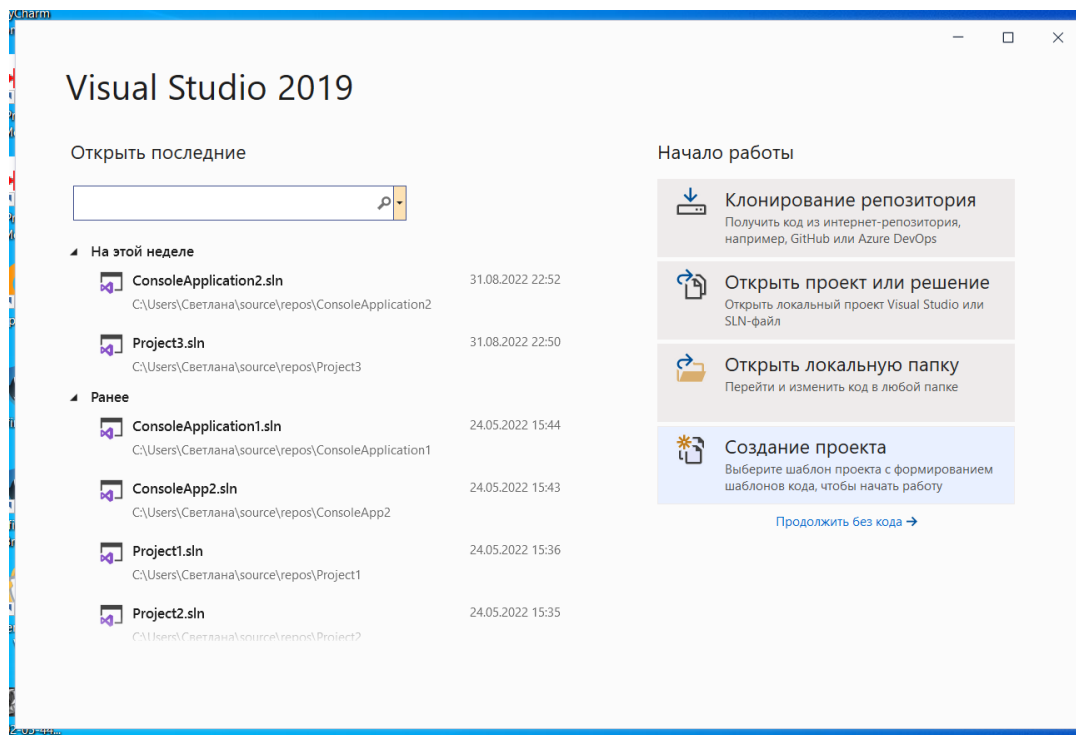
Цель занятия: изучить структуру программы на языке C++; приобрести навыки составления программ линейной структуры. Получить навыки работы со средой разработки Microsoft Visual Studio 2019.

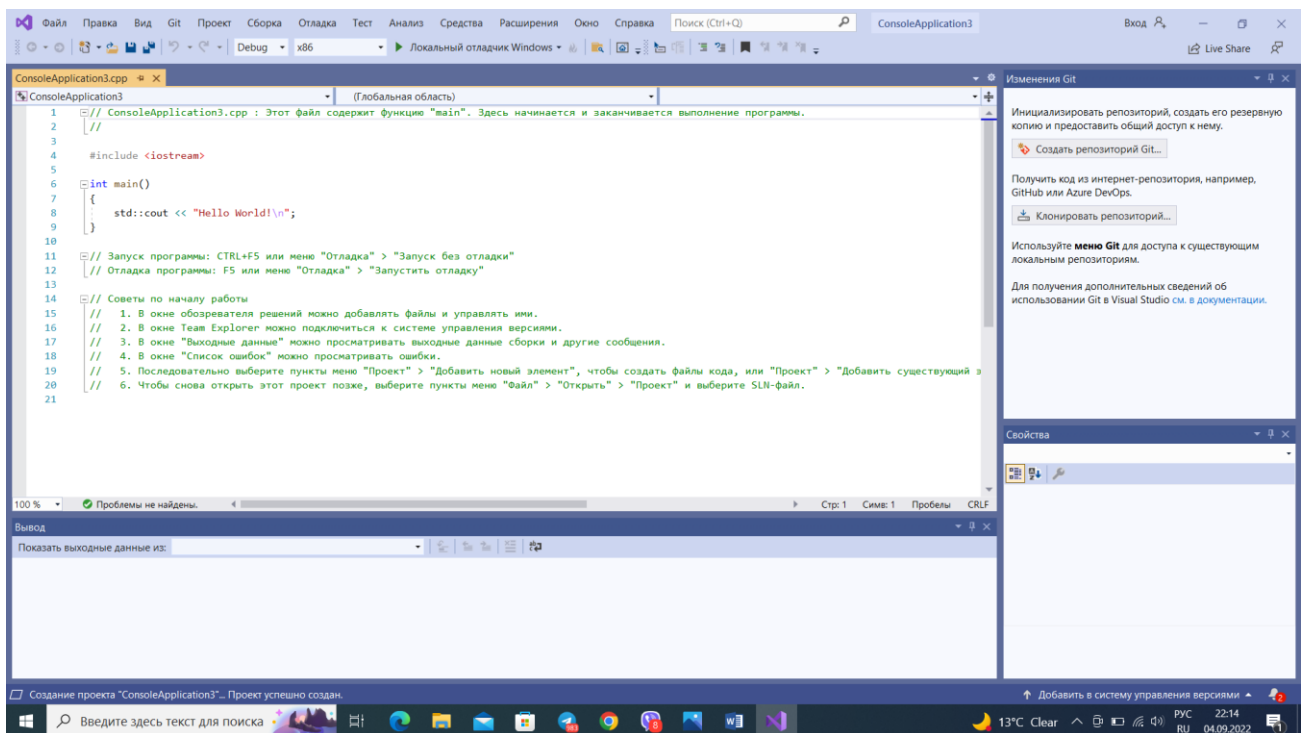
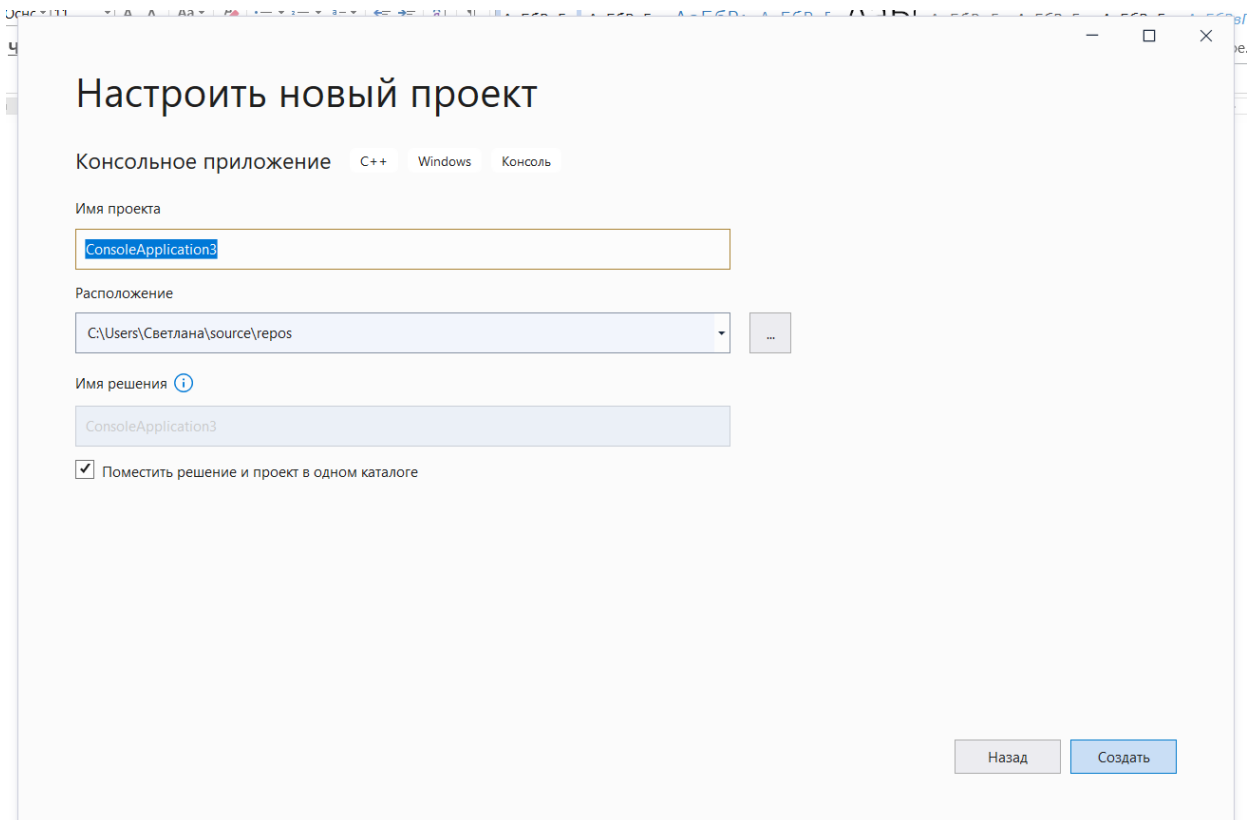
Указания по выполнению работы.

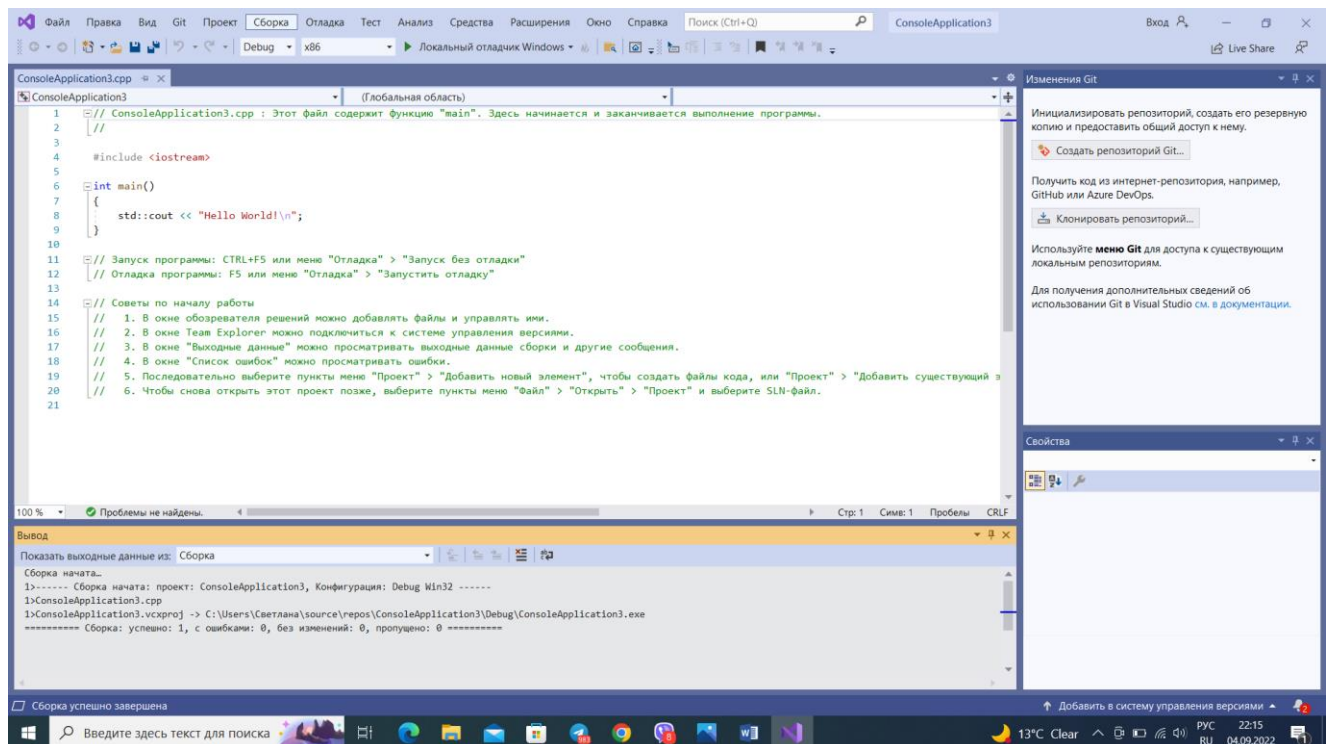
Для создания проекта нужно запустить MS Visual Studio 2019.

Запуск Visual Studio 2019









Теоретические сведения

Структура программы. Программа на языке С имеет следующую структуру [11]:

#директивы препроцессора

.....

#директивы препроцессора

функция a()

{ тело функции a }

.

.

функция b()

{тело функции b }

void main () //функция, с которой начинается выполнение программы
{последовательность определений, описаний и исполняемых операторов}

Правила препроцессорной обработки определяет программист с помощью директив препроцессора. *Директива* начинается с #.

Программа представляет собой набор описаний и определений, и состоит из набора функций.

Среди этих функций всегда должна быть функция с именем **main**. Без нее программа не может быть выполнена. Это точка входа в программу.

Перед именем функции помещаются сведения о типе возвращаемого функцией значения (тип результата): **double sqrt (double arg)**

Если функция ничего не возвращает, то указывается тип **void**: **void main ();**

Каждая функция, в том числе и **main** должна иметь набор параметров, он может быть пустым.

Если функция не объявлена как **void**, она должна возвращать значение. Возвращаемое значение задается инструкцией **return e**; где **e** – выражение, значение которого возвращается в качестве результата работы функции.

В программах на C++ используется два способа **комментариев**:

- первый способ для многострочных комментариев (начинается с комбинации символов косой черты и звёздочки (/*), и заканчивается обратной комбинацией этих же символов (*/*). Не может быть вложенным.),
- второй – для коротких замечаний (начинается с двух символов косой черты (//) и заканчивается концом строки).

Правила задания имен объектов (идентификаторов) С-программ. В идентификаторе могут быть использованы латинские буквы, цифры и знак подчеркивания. Первым символом не может быть цифра. Прописные и строчные буквы различаются. (PROG1, prog1 и Prog1 – три различных идентификатора). Пробелы не допускаются.

_PROG1, prog_1, P1ro2g1 - правильные идентификаторы.
3PROG1, prog_*, №5_Prog1 – ошибка!

Стандартные типы данных. Одним из важнейших понятий в программировании является переменная. Переменная – это поименованная область оперативной памяти компьютера, где хранится значение некоторой величины. Переменная обладает такими свойствами: название (имя), значение, тип. Количество переменных и их свойства указывает пользователь. Тип переменной определяет её допустимые значения, а также операции, которые можно над ней выполнять.

Рассмотрим стандартные типа данных

В C++ определены следующие простые **типы данных**:

int (целый)

char (символьный)

bool (логический)

float (вещественный)

double (вещественный с двойной точностью)

Логические, символьные и целые типы все вместе обобщённо называются **целыми** (строго говоря, **интегральными**) **типами**. Целые типы совместно с типами с плавающей точкой называются **арифметическими типами**.

Существует 4 спецификатора типа, уточняющих внутреннее представление и диапазон стандартных типов:

1) **short** (короткий)

2) **long** (длинный)

3) **signed** (знаковый)

4) **unsigned** (беззнаковый)

Операции присваивания имеют следующие виды:

=, +=, -=, *= и т.д.

Сокращённые формы арифметических действий, иногда нужно сделать какое либо одно арифметическое действие над одной переменной, прибавить, умножить.

Например:

$S = S + 32$; в сокращённой форме это будет $S += 32$;

$F = F - k$; в сокращённой форме это будет $F -= k$;

$N = N / 2$; в сокращённой форме это будет $N /= 2$;

Формат операции простого присваивания:

операнд1=операнд2

Действие оператора. Вычисляется <выражение>, и его значение присваивается

<переменной>. Выражение служит для описания формул, по которым выполняются вычисления, и может состоять из чисел, имён переменных, констант, функций, соединённых символами операций.

Если выражение формирует целое или вещественное число, то оно *называется арифметическим*. Пара арифметических выражений, объединённая операцией сравнения, называется *отношением*. Если отношение имеет ненулевое значение, то оно – истинно, иначе – ложно. Приоритеты операций в выражениях представлены в табл.1.1

Таблица 1.1

Приоритеты операций в выражениях	
Ранг	Операции
1	() [] -> .
2	! ~ - ++ -- & * (тип) sizeof тип()
3	* / % (мультипликативные бинарные)
4	+ - (аддитивные бинарные)
5	<< >> (поразрядного сдвига)
6	< > <= >= (отношения)
7	== != (отношения)
8	& (поразрядная конъюнкция «И»)
9	^ (поразрядное исключающее «ИЛИ»)
10	(поразрядная дизъюнкция «ИЛИ»)
11	&& (конъюнкция «И»)
12	(дизъюнкция «ИЛИ»)
13	?: (условная операция)
14	= *= /= %= -= &= ^= = <<= >>= (операция присваивания)
15	, (операция запятая)

Операции выполняются с учетом их приоритета (1 – самый высокий). Для изменения естественного порядка выполнения операций используют круглые скобки, например;

$2 * -3 - 2 = -8$; $2 * (-3 - 2) = -10$; $5 * (2 + 13) = 75$;
 $20 + 100 / 20 * 5 = 45$; $20 + 100 / (20 * 5) = 21$.

Операторы инкремента (++) и декремента (--) увеличивают или уменьшают значение операнда на единицу. Операнд может быть целого типа или типа с плавающей точкой, или типа указатель и должен быть модифицируемым. В языке имеется **префиксная** (++t, увеличивает операнд до его использования) и **постфиксная формы** (t++, увеличивает операнд после его использования) операторов инкремента и декремента.

Оператор умножения (*) выполняет умножение операндов:

```
int i = 5;
float f = 0.2;
double g;
g = f * i;
```

Тип результата умножения f на i преобразуется к типу double, затем результат присваивается переменной g.

Оператор деления (/) выполняет деление первого операнда на второй. Если две целые величины не делятся нацело, то результатом будет целая часть от деления (дробная часть отбрасывается):

```
int i = 49, j = 10, n, m;
```

```

n = i / j;           // результат  4
m = i / (-j);        // результат -4

```

Оператор остаток от деления (%) дает остаток от деления первого операнда на второй (только для целых операндов). **Знак результата совпадает со знаком делимого:**

```

int n = 49, m = 10, i, j, k, l;
i = n % m;           // результат  9
j = n % (-m);         // результат  9
k = (-n) % m;         // результат -9
l = (-n) % (-m);      // результат -9

```

Логическое выражение – это способ записи на языке программирования условий для поиска необходимых данных. Логическое выражение может принимать значения **true** (истина) или **false** (ложь). Логические выражения бывают простые и сложные. Простое выражение – это два арифметических выражения, соединённых символом отношения, а сложное – это простые логические выражения, соединённые логическими операциями **!** (логическое отрицание НЕ), **&&** (логическое И) или **||** (логическое ИЛИ). Приоритет выполнения логических операций такой:

1) **!**, 2) **&&**, 3) **||**.

В табл. 1.2 приведены определения логических операций.

Таблица 1.2

Логические операции

Выражение	Значение	Выражение	Значение
! true	false	! false	true
true && true	true	true true	true
true && false	false	false true	true
false && true	false	true false	true
false && false	false	false false	false

Пример. Пусть $x=3$, $y=-9$. рассмотрим некоторые логические выражения и их значения.

Простые выражения	Значения	Сложные выражения	Значения
$x = 3$	true	! ($y \leq -50$)	true
$x > y$	true	$(1 < x) \&\& (x < 5)$	true
$7 \% 3 = 1$	true	$(x > 4) \parallel (y < -15)$	false
$y - 2 = 4$	false	$(x > 4) \parallel (y > -15)$	true

Двойное неравенство $1 < x < 5$ как сложное логическое выражение записывают так: $(1 < x) \&\& (x < 5)$. Совокупность неравенств вида $x < 1$; $x > 5$ так: $(x < 1) \parallel (x > 5)$. Простые логические выражения, из которых состоят сложные в круглые скобки можно не брать.

Математические библиотечные функции, основные из которых перечислены в табл.3, дают возможность выполнять определенные типовые математические вычисления. Как правило, функции из математической библиотеки возвращают как результат числа с плавающей точкой типа *double*.

При использовании функций математической библиотеки в программу нужно включить соответствующий заголовочный файл с помощью директивы препроцессора:

`#include <math.h>`

Таблица 3

Часто используемые математические функции

Прототип функции	Вычисление	Примеры вычислений
<code>double sqrt(double);</code>	Корень квадратный	<code>sqrt(900.0) = 30.0</code> <code>sqrt(9.0) = 3.0</code>
<code>double exp(double);</code>	Экспоненциальная функция e^x	<code>exp(1.0) = 2.718282</code> <code>exp(2.0) = 7.389056</code>
<code>double log(double);</code>	Логарифм натуральный (по основанию e)	<code>log(2.718282) = 1.0</code> <code>log(7.389056) = 2.0</code>
<code>double log10(double);</code>	Логарифм десятичный (по основанию 10)	<code>log10(1.0) = 0.0</code> <code>log10(10.0) = 1.0</code> <code>log10(100.0) = 2.0</code>
<code>double fabs(double);</code>	Абсолютное значение	если $x > 0$, то <code>fabs(x) = x</code> если $x = 0$, то <code>fabs(x) = 0.0</code> если $x < 0$, то <code>fabs(x) = -x</code>
<code>double ceil(double);</code>	Округление аргумента до наименьшего целого, не меньшего чем аргумент	<code>ceil(9.2) = 10.0</code> <code>ceil(-9.8) = -9.0</code>
<code>double floor(double);</code>	Округление аргумента до наибольшего целого, не большего чем аргумент	<code>floor(9.2) = 9.0</code> <code>floor(-9.8) = -10.0</code>
<code>double pow(double x, double y);</code>	x в степени y	<code>pow(2, 7) = 128.0</code> <code>pow(9, 0.5) = 3.0</code>
<code>fmod(double x, double y);</code>	Остаток от x/y , как число с плавающей точкой	<code>fmod(13.657, 2.333) = 1.992</code>
<code>double sin(double x);</code>	Синус (x в радианах)	<code>sin(0.0) = 0.0</code>
<code>double cos(double x);</code>	Косинус (x в радианах)	<code>cos(0.0) = 1.0</code>
<code>double tan(double x);</code>	Тангенс (x в радианах)	<code>tan(0.0) = 0.0</code>

Вызов функций:

```
cout << "Модуль -10: " << abs(-10) << " Модуль -10.0: " << fabs(-10.0) << "\n";  
cout << "sin(90): " << sin(3.14/2) << "\n";
```

Остальные математические функции можно выразить через основные, например,

$$\log_b a = \ln(a) / \ln(b).$$

Операторы ввода (>>) и вывода (<<) не являются встроенными для языка C++, а обеспечиваются стандартной библиотекой с помощью потоков ввода – вывода. При запуске программы на выполнение автоматически открываются три стандартных потока языка C++:

cin – стандартный поток ввода (с клавиатуры);

cout – стандартный поток вывода (на дисплей);

cerr – стандартный поток для выдачи сообщений (на дисплей). Вместо потока cout для вывода сообщений об ошибках может использоваться поток cerr.

При выводе отдельных символов каждый символ должен заключаться в **одиночные кавычки** в отличие от вывода строк символов, когда **выводимая строка** заключается в **двойные кавычки**.

Для выполнения операторов потокового ввода – вывода в программу должна быть включена инструкция препроцессора:

#include <iostream>

Задача 1.1. Вычислить высоты треугольника со сторонами a , b , c , используя формулы 1.1, 1.2, 1.3:

$$h_a = \frac{2}{a} \sqrt{p(p-a)(p-b)(p-c)}; \quad 1.1$$

$$h_b = \frac{2}{b} \sqrt{p(p-a)(p-b)(p-c)}; \quad 1.2$$

$$h_c = \frac{2}{c} \sqrt{p(p-a)(p-b)(p-c)}, \quad 1.3$$

где $p = (a + b + c)/2$.

Схема алгоритма решения задачи приведена на рис. 1.1.

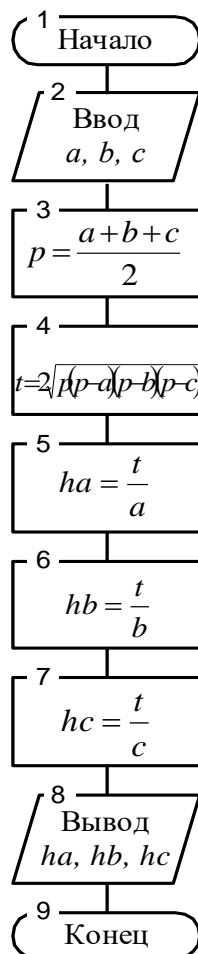


Рис. 1.1. Схема алгоритма решения задачи 1.1

Исходными данными для решения являются значения длин сторон треугольника – a , b , c . В программе используются переменная p для вычисления полупериметра и вспомогательная переменная t для исключения повторений. Вычисленные значения высот h_a , h_b , h_c необходимо вывести со своими именами, каждую на одной строке.

Программа имеет вид:

```
#include <iostream>
#include <conio.h> // файл, где определена функция getch()
using namespace std;
void main()
{
    double a, b, c, p, t, ha, hb, hc; //Объявление переменных
    cout<<"\nEnter a, b, c";
    cin>>a>>b>>c; //ввод с клавиатуры значения сторон
    p = (a + b + c)/2; //вычисление полупериметра
    t= 2*sqrt(p * (p - a) * (p - b) * (p - c));
    ha=t/a; // вычисление высоты ha
    hb=t/b; // вычисление высоты hb
    hc=t/c; // вычисление высоты hc
    cout<<"\n" <<" ha ="<< ha; //Вывод высоты ha
    cout<<"\n" <<" hb ="<< hb; //Вывод высоты hb
    cout<<"\n" <<" hc ="<< hc; //Вывод высоты hc
    getch(); // ждать нажатия любой клавиши
}
```

Задание 1.1. Известна длина окружности (l). Найти площадь круга (S), ограниченного этой окружностью ($S = \pi R^2$, $l = 2\pi R$).

Задача 1.2. Написать программу преобразования дюймов в сантиметры (1 дюйм равен 2.54 сантиметра). Вывести на экран сообщение вида: “12 in = 30.48 sm”

Программа имеет вид:

```
#include <iostream>
#include <conio.h> // файл, где определена функция getch()
#include <windows.h> // файл, где определена функция SetConsoleOutputCP(1251) для
//вывода текста кириллицей;
using namespace std; // использовать стандартное пространство имен
int main()
{
    SetConsoleOutputCP(1251);
    float inches = 0.0, sentimeters = 0.0;
    cout<<"Введите длину в дюймах:\n";
    cin>>inches;
    sentimeters = inches*2.54;
    cout<<inches<<" in = "<<sentimeters<<" sm ";
    getch(); // ждать нажатия любой клавиши
    return 0;
}
```

Задача 1.3. Пользователь вводит с клавиатуры денежную сумму в рублях и копейках (рубли и копейки вводятся в разные переменные). Сумма может быть введена как правильно (например, 19 руб. 90 коп), так и неправильно (например 22 руб. 978 коп). Написать программу, которая, используя только линейный алгоритм, осуществит

корректировку введенной денежной суммы в правильную форму. Например, если пользователь ввел 11 руб. 150 коп., программа должна вывести на экран сумму 12 руб. 50 коп.

Программа имеет вид:

```
#include <iostream>
#include <locale>           // файл, где определена функция setlocale (LC_CTYPE, "rus")
                           // для вывода текста кириллицей;
#include <conio.h>          // файл, где определена функция getch()
using namespace std;
void main()
{
    setlocale (LC_CTYPE, "rus");
    int rub, cent, rcent, rrub;
    cout<<"Введите рубли = ";
    cin>>rub;
    cout<<"Введите копейки = ";
    cin>>cent;
    rcent=cent%100;
    rrub=(cent-rcent)/100+rub;
    cout<<"Сумма = "<<rrub<<" руб. "<<rcent<<" коп.\n";
    getch();               // ждать нажатия любой клавиши
}
```

Задание 1.2. Написать программу, вычисляющую, с какой скоростью бегун пробежал дистанцию. Рекомендуемый вид экрана во время выполнения программы приведен ниже:

Вычисление скорости бега

Введите длину дистанции (метров) = 1000

Введите время (мин.сек) = 3.25

Дистанция: 1000 м

Время: 205 сек

Вы бежали со скоростью 17.56 км/ч

Задача 1.4. Ввести целое четырехзначное число. Найти сумму первого и последнего, второго и предпоследнего разрядов этого числа.

```
#include <iostream>
#include <locale>
#include <conio.h>
using namespace std;
void main()
{
    setlocale (LC_CTYPE, "rus");
    int num, d1,d2,d3,d4,s1,s2;
    cout<<"Введите четырехзначное число = ";
    cin>>num;
    // поиск разрядов четырехзначного числа
    d1=num/1000;
    d2=num/100%10;
    d3=num/10%10;
    d4=num%10;
    // вывод разрядов четырехзначного числа
```

```

cout<<d1<<"\n";
cout<<d2<<"\n";
cout<<d3<<"\n";
cout<<d4<<"\n";
// вычисление сумм
s1=d1+d4;
s2=d2+d3;
cout<<"сумма первого и последнего разрядов =" <<s1;
cout<<"сумма второго и предпоследнего разрядов =" <<s2;
getch();      // ждать нажатия любой клавиши
}

```

Задание 1.3. Дано натуральное трехзначное число n . Чему равно произведение его цифр?

Примечание: С целью сокращения текстов программ в языке C++ допускается размещать на одной строке несколько операторов. Однако этого следует избегать, так как такое размещение приводит к плохому восприятию программы.

Методические указания

- При подготовке к занятию необходимо изучить:
 - разделы описаний данных, стандартные функции, правила записи арифметических выражений;
 - операторы: присваивания, составной;
 - организацию ввода-вывода данных.
- Функции, отсутствующие в списке стандартных функций языка C++, следует выразить через имеющиеся.

Аудиторные и домашние задания

- Для следующих формул записать соответствующие арифметические выражения на языке C++:

а) $\frac{c}{ab} + \frac{ab}{c}$; б) $\frac{x+y}{\sqrt{a_1}} \cdot \frac{a_2}{x-y}$; в) $10\alpha - 3\frac{1}{5}\beta$; г) $\frac{(1 + \frac{x}{2!} + \frac{y}{3!})}{1 + \frac{2}{3+xy}}$

- Даны x, y, z . Вычислить a, b , если:

а) $a = \frac{\sqrt{|x-1|} - \sqrt[3]{|y|}}{1 + \frac{x^2}{2} + \frac{y^2}{4}}, \quad b = x(\arctg z + e^{-(x+3)})$

б) $a = \frac{2 \cos\left(x - \frac{\pi}{6}\right)}{\frac{1}{2} + \sin y^2}, \quad b = 1 + \lg^2 \frac{z}{2}.$

- Пользователь указывает цену одной минуты исходящего звонка с одного мобильного оператора другому, а также продолжительность разговора в минутах и секундах. Необходимо вычислить денежную сумму, на которую был произведен звонок.

4. Пусть заданы координаты трёх вершин треугольника $A(1;1)$, $B(2;2)$, $C(-1;2)$. Вычислить биссектрису W_a ($W_a = \frac{2}{b+c} \sqrt{bcp(p-a)}$) и радиус описанной окружности R ($R = abc / (4s)$), где: s – площадь треугольника; p – полупериметр треугольника.
5. Ввести размер ребра куба. Определить объем и площадь боковой поверхности куба.
6. Сколько секунд в сутках, неделе, году?
7. Цены на два вида товаров возросли на p процентов. Вывести старые и новые цены.
8. Ввести целое трехзначное число. Определить первую цифру числа.
9. Ввести целое пятизначное число. Найти сумму цифр этого числа.
10. Ввести 2 целых числа X, Y . Поменять местами значения этих переменных.
11. Ввести 2 целых числа X, Y . Вычислить остаток от деления одного числа на другое, разделить числа двумя различными операторами деления. Вывести все результаты.
12. По заданному времени начала решения задачи (часы, минуты, секунды) и времени выполнения задачи (секунды) определить время окончания решения задачи.
13. Составить линейную программу, печатающую значение `true` или `false` в зависимости от того, истинно или ложно указанное высказывание:
 - а) целые n и k имеют одинаковую четность;
 - б) целое n кратно четырем и не оканчивается нулем;
 - в) число c является средним арифметическим чисел a и b ;
 - г) каждое из чисел x, y, z отрицательное;
 - д) только одно из чисел x, y, z четное;
 - е) хотя бы одно из чисел x, y, z больше ста;
 - ж) число x принадлежит отрезку $[2,5]$ или $[-1,1]$;
 - з) число x лежит вне отрезка $[2,5]$ и $[-1,1]$.

Контрольные вопросы

1. Поясните структуру программы на языке C++.
2. Назовите основные типы данных языка C++.
3. Назовите основные операторы языка C++.
4. Что такое идентификатор, как он формируется?
5. Поясните форму объявления констант и переменных.
6. Можно ли менять значения констант?
7. Поясните основные характеристики вещественных, целых, логических, символьных, строковых, перечисляемых констант.
8. Что такое оператор присваивания?
9. Как подразделяются операции C++ по приоритету?
10. С помощью каких процедур выполняется стандартный ввод, вывод?
11. Какая инструкция препроцессора должна быть включена в программу для выполнения операторов потокового ввода – вывода?
12. Какие стандартные потоки языка C++ автоматически открываются при запуске программы на выполнение?