

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Таныгин Максим Олегович  
Должность: и.о. декана факультета фундаментальной и прикладной информатики  
Дата подписания: 21.09.2023 13:19:53  
Уникальный программный ключ:  
65ab2aa0d384efe8480e6a4c688eddbc475e411a

**МИНОБРНАУКИ РОССИИ**  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Юго-Западный государственный университет»  
(ЮЗГУ)

Кафедра программной инженерии

УТВЕРЖДАЮ  
Проректор по учебной работе

  
О.Г. Локтионова  
«15» 12 2017 г.



**ПРОГРАММИРОВАНИЕ ЛИНЕЙНЫХ АЛГОРИТМОВ НА  
ЯЗЫКЕ C++**

Методические указания по выполнению лабораторной работы по  
дисциплине "Программирование на языках высокого уровня" для  
студентов направления подготовки 09.03.04 "Программная  
инженерия"

Курск 2017

УДК 681.3.06(071.8)

Составители: Т.М. Белова, В.Г. Белов

Рецензент

Кандидат технических наук, доцент кафедры программной инженерии ЮЗГУ И.Н. Ефремова

**Программирование линейных алгоритмов на языке C++:** методические указания по выполнению лабораторной работы по дисциплине "Программирование на языках высокого уровня" для студентов направления подготовки 09.03.04 "Программная инженерия" / Юго-Зап. гос. ун-т; сост. Т.М. Белова, В.Г. Белов. Курск, 2017. 25 с.

Содержат основные теоретические положения и приемы разработки линейных программ на языке C++, пример решения типовой задачи, индивидуальные задания и контрольные вопросы к защите лабораторной работы.

Методические рекомендации соответствуют требованиям рабочей программы по дисциплине "Программирование на языках высокого уровня".

Предназначены для студентов направления подготовки 09.03.04 «Программная инженерия» дневной и заочной форм обучения.

Текст печатается в авторской редакции.

Подписано в печать 15.12.17 . Формат 60x84 1/16.

Усл. печ. л. 1,5. Уч.-изд. л. 1,3. Тираж 100 экз. Заказ 3483. Бесплатно.

Юго-Западный государственный университет  
305040, Курск, ул.50 лет Октября, 94.

## Программирование линейных алгоритмов на языке C++

**Цель работы** — изучение и приобретение навыков программирования линейных алгоритмов, освоение оператора присваивания, типов данных, операторов ввода и вывода, арифметических операций и элементарных математических функций языка C++.

### Основные понятия

Модуль – программная единица, предназначенная для размещения объектов программы. С помощью содержащегося в нем программного кода реализуется вся поведенческая структура программы.

Заголовочный файл:

```
//-----
#ifndef Unit1H // Директивы препроцессора
#define Unit1H
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
//-----
class TForm1 : public TForm
{
published: // IDE-managed Components
private: // User declarations
public: // User declarations
    __fastcall TForm1(TComponent* Owner);
};
//-----
extern PACKAGE TForm1 *Form1;
//-----
#endif
```

Файл реализации:

```
//-----
#include <vcl.h>
#pragma hdrstop

#include "Unit1.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
}
/* В этом месте размещается программный код, реализующий
алгоритм программы */
```

Любые данные (константы, переменные, свойства, значения функций или выражения) в C++ характеризуются своими типами. Тип определяет множество допустимых значений, а также множество допустимых операций, которые применимы к нему. Кроме того, тип определяет также и формат внутреннего представления данных в памяти ПК (рисунок 1).

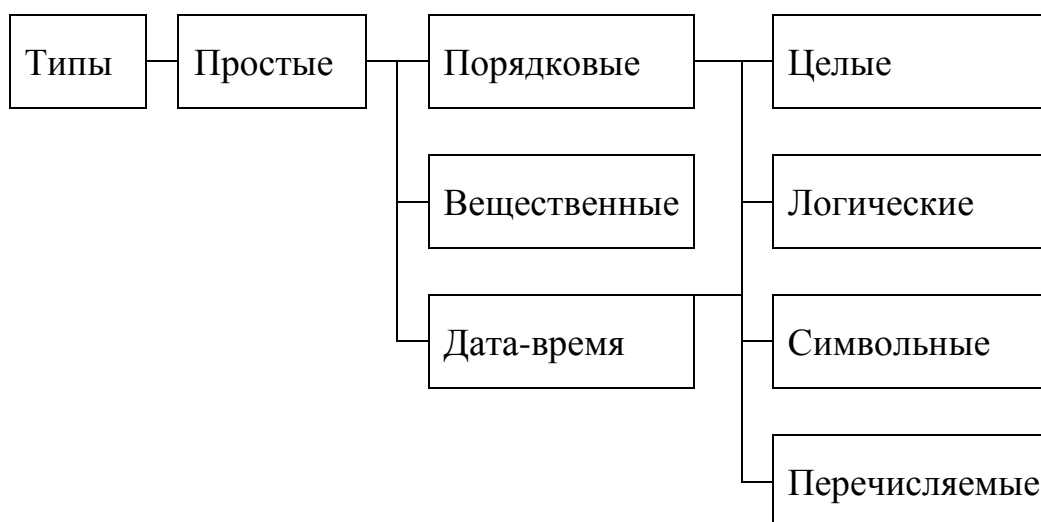


Рисунок 1 — Структура простых типов данных

## Порядковые типы

*Порядковые типы* отличаются тем, что каждый из них имеет конечное число возможных значений. С каждым значением можно сопоставить некоторое целое число – порядковый номер значения. К ним относят целые, символьные, перечисляемые типы.

## Арифметические типы данных

К ним относят целые и действительные типы.

## Целые типы

В таблице 1 приводятся названия *целых типов* данных, длина их внутреннего представления в байтах и диапазон возможных значений. Стандартные функции, применимые к целым типам, показаны в таблице 2. При работе с математическими функциями для подключения стандартных библиотек необходимо вручную вводить директивы препроцессора:

```
#include <math.h>
#include <Math.hpp>
#include <stdlib.h>
```

Таблица 1

Целые типы

Название	Длина, байт	Диапазон
char	1	-128...127
unsigned char	1	0...255
signed char	1	-128...+127
short	2	-32768...+32767
unsigned short	2	0...65535
int	4	-2147483648...+2147483647
unsigned int	4	0...4294967295
long	4	-2147483648...+2147483647
unsigned long	4	0...4294967295

## Стандартные функции, применимые к целым типам

Функция	Синтаксис	Описание	Файл
abs	int abs(int x)	Абсолютное значение	stdlib.h
_lrotl	unsigned long _lrotl(unsigned long val, int count)	Циклический сдвиг val влево на count битов	stdlib.h
_lrotr	unsigned long _lrotr(unsigned long val, int count)	Циклический сдвиг val вправо на count битов	stdlib.h
div	div_t div(int t, int d) typedef struct{ int quot; //частное int rem; //остаток } div_t	Целочисленное деление n/d	math.h
random	int random(int num)	Псевдослучайное число, равномерно распределенное в диапазоне 0 ... num-1	stdlib.h

**Логические типы**

К логическим типам данных относят тип *bool* (1 байт), *unsigned char* (1 байт), *unsigned short* (2 байта), *BOOL*(WinAPI) (4 байта). Значениями логического типа *bool* может быть одна из предварительно объявленных констант *false* (ложь) или *true* (истина). В качестве логического типа используется целый тип. Если значение целого числа равно 0, то это интерпретируется как ложь. Если любое ненулевое целое значение, то это интерпретируется как истина.

**Символьный тип**

Значением символьного типа является множество всех символов ПК. Каждому символу приписывается целое число в диапазоне 0...255. К *символьным типам* данных относятся типы *char*, *wchar\_t*. *Символьный тип* – *char*, предназначен для хранения одного символа. В памяти ПК символ занимает один байт. Один

байт – это 256 ( $2^8$ ) различных символов, которые можно хранить в переменной *char*. Эти символы составляют символы *ASCII* от 0 до 255. Символьный тип – *wchar\_t*, представляет собой 16-битный символьный тип стандарта *Unicode*.

### Перечислимый тип

*Перечислимый тип* задается перечислением тех значений, которые он может получать. Каждое значение именуется некоторым идентификатором и располагается в списке, заключенном в фигурные скобки:

`enum {<константа 1>, ..., <константа n>} <имена переменных>;`

Соответствие между значениями *перечислимого типа* и порядковыми номерами этих значений устанавливается порядком перечисления: первое значение в списке получает порядковый номер 0, второе – 1 и т.д. Максимальная мощность перечисляемого типа составляет 65536 значений.

### Вещественные типы

В отличие от порядковых типов, значения которых всегда сопоставляются с рядом целых чисел и представляются в ПК абсолютно точно, значения *вещественных типов* определяют произвольное число лишь с некоторой конечной точностью, зависящей от внутреннего формата вещественного числа.

В таблице 3 приводятся названия вещественных типов, длина их внутреннего представления в байтах и диапазон возможных значений.

Таблица 3

Вещественные типы

Название	Длина, байт	Количество знач. цифр	Диапазон
float	4	7...8	$3,4 \cdot 10^{-38} \dots 3,4 \cdot 10^{38}$
double	8	15...16	$1,7 \cdot 10^{-308} \dots 1,7 \cdot 10^{308}$
long double	10	19...20	$3,4 \cdot 10^{-4932} \dots 1,1 \cdot 10^{4932}$
comp	8	19...20	$-2^{63} \dots +2^{63}-1$

Для работы с вещественными данными могут использоваться математические функции, представленные ниже (табл. 4).

Таблица 4

## Стандартные математические функции

Функция	Синтаксис	Описание	Файл
ceil	double ceil(double x)	Округление вверх: наименьшее целое, не меньшее x	math.h
floor	double floor(double x)	Округление вниз: наибольшее целое, не большее x	math.h
exp	double exp(double x)	Экспонента	math.h
abs	int abs(int x)	Модуль аргумента	stdlib.h
fabs	double fabs(double x)	Модуль аргумента	math.h
fmod	double fmod(double x, double y)	Остаток от деления x/y	math.h
frexp	double frexp(double x, int *exponent)	Разделяет x на мантиссу (возвращает) и степень exponent	math.h
ldexp	double ldexp(double x, int exp)	$x \cdot 2^{\text{exp}}$	math.h
log	double log(double x)	Натуральный логарифм	math.h
log10	double log10(double x)	Десятичный логарифм	math.h
Log2	Extended Log2 (Extended X)	Логарифм по основанию 2	Math.hpp
LogN	Extended LogN(Extended Base, Extended X)	Логарифм X по основанию Base	Math.hpp
max	max(a,b)	Макрос возвращает максимальное значение из a и b любых типов	stdlib.h
min	min(a,b)	Макрос возвращает минимальное значение из a и b любых типов	stdlib.h
modf	double modf(double x, double *ipart)	Разделяет x на целую часть ipart и возвращаемую дробную часть	math.h



Функция	Синтаксис	Описание	Файл
pow	double pow(double x, double y)	$x^y$	math.h
sqrt	double sqrt(double x)	Корень квадратный	math.h
acos	double acos(double x)	Арккосинус в диапазоне $[0, \pi]$	math.h
asin	double asin(double x)	Арсинус в диапазоне $[-\pi/2, \pi/2]$	math.h
atan	double atan(double x)	Арктангенс в диапазоне $[-\pi/2, \pi/2]$	math.h
atan2	double atan2(double y, double x)	Арктангенс $y/x$ в диапазоне $[-\pi/2, \pi/2]$	math.h
DegToRad	Extended DegToRad (Extended Degrees)	Вычисляет угол в радианах по его значению в градусах Degrees	Math.hpp
RadToDeg	Extended RadToDeg (Extended Radians)	Вычисляет угол в градусах по его значению в радианах Radians	Math.hpp
hypot	double hypot(double x, double y)	Гипотенуза треугольника с катетами $x$ и $y$	math.h
cos	double cos(double x)	Косинус, угол в радианах	math.h
sin	double sin(double x)	Синус, угол $x$ в радианах	math.h
tan	double tan(double x)	Тангенс, угол в радианах	math.h

### Константы, используемые в математических выражениях

M\_PI      число  $\pi$

M\_E        число  $e$

### Тип дата-время

Тип *дата-время* определяется идентификатором *TDateTime* и предназначен для одновременного хранения и даты, и времени. Во внутреннем представлении он занимает 8 байт и представляет собой вещественное число с фиксированной дробной частью: в целой части числа хранится дата, в дробной – время.

Над данными типа *TDateTime* определены те же операции, что и над вещественными числами, а в выражениях этого типа

могут участвовать константы и переменные целого и вещественного типов.

### Тип `void`

Тип `void` называется *пустым* типом. Используется в случае, если параметры без типа или если функция не возвращает значения. Данный тип используется в некоторых функциях (табл. 5).

Таблица 5

Функции, использующие пустой тип данных

Функция	Синтаксис/Описание	Файл
<code>_lrand()</code>	<code>long _lrand(void)</code> Псевдослучайное целое, диапазон от 0 до $2^{31}-1$	<code>stdlib.h</code>
<code>rand()</code>	<code>int rand(void)</code> Псевдослучайное целое, диапазон от 0 до <code>RAND_MAX</code> (32767)	<code>stdlib.h</code>
<code>randomize()</code>	<code>void randomize (void)</code> Рандомизация генераторов случайной величиной	<code>stdlib.h</code>

Пользователь может вводить в программу свои собственные типы. Объявления типов могут делаться в различных местах кода.

Синтаксис объявления типа:

```
typedef <тип> <идентификатор>;
```

Объявление типа с помощью *typedef* можно использовать и для создания нового типа, имя которого будет являться псевдонимом стандартного типа C++, например:

```
typedef bool boolean;
typedef unsigned int uint;
typedef unsigned char byte;
typedef unsigned long long ul;
```

В `C++Builder` определены следующие операции над данными:

#### 1. Унарные:

! отрицание

- & адресация
- \* косвенная адресация
- ~ поразрядное отрицание
- > доступ к элементам структур и классов

## 2. Мультипликативные:

- \* умножение
- / деление
- % остаток от целочисленного деления
- << поразрядный сдвиг влево
- >> поразрядный сдвиг вправо
- & поразрядное И
- && логическое И

## 3. Аддитивные:

- ++ инкремент
- декремент
- + сложение
- вычитание
- ^ поразрядное исключающее ИЛИ
- | поразрядное ИЛИ
- || логическое ИЛИ

## 4. Отношения:

- == равно
- != не равно
- < меньше
- > больше
- <= меньше или равно
- >= больше или равно

*Логические операции* над логическими данными дают результат логического типа по правилам, указанным ниже (табл. 6).

Таблица 6

Логические операции над данными логического типа

Операнд 1	Операнд 2	!	&&	
true	-	false	-	-
false	-	true	-	-

Операнд 1	Операнд 2	!	&&	
false	false	-	false	false
false	true	-	false	true
true	false	-	false	true
true	true	-	true	true

### Оператор присваивания

С помощью этого оператора переменной или функции присваивается значение выражения. Для этого используется знак присваивания =, слева от которого записывается имя переменной или функции, которой присваивается значение, а справа – выражение, значение которого вычисляется перед присваиванием.

Допустимо присваивание значений переменным или функциям любого типа, за исключением типа файл. Тип выражения и тип переменной должны быть совместимы для присваивания.

В C++ определен ряд операций присваивания, представленных ниже (табл. 7).

Таблица 7

#### Операции присваивания

Обозначение	Операция	Типы операндов и результата	Пример
=	Присваивание	Любые	X=Y
+=	Присваивание со сложением	Арифметические, указатели, структуры, объединения	X+=Y
--	Присваивание с вычитанием	Арифметические, указатели, структуры, объединения	X-=Y
*=	Присваивание с умножением	Арифметические	X*=Y
/=	Присваивание с делением	Арифметические	X/=Y
%=	Присваивание остатка целочисленного деления	Целые	X%=Y
<=	Присваивание со сдвигом влево	Целые	X<=Y
>=	Присваивание со сдвигом вправо	Целые	X>=Y

Обозначение	Операция	Типы операндов и результата	Пример
&=	Присваивание с поразрядной операцией И	Целые	X&=Y
^=	Присваивание с поразрядной операцией исключающее ИЛИ	Целые	X^=Y
=	Присваивание с поразрядной операцией ИЛИ	Целые	X =Y

**Примечание.** Выражение  $x+=y$  эквивалентно выражению  $x=x+y$ , но записано компактнее. Аналогично определяются и другие операции присваивания.

Операции присваивания возвращают как результат присвоенное значение, поэтому допускается сцепление

$$a = b = c = 1;$$

### Операции увеличения и уменьшения: ++ и --

Операции инкремента (++) и декремента (--) сводятся к увеличению (++) или уменьшению (--) операнда на единицу. Эти операции выполняются быстрее, чем обычное сложение и вычитание.

Если операция инкремента или декремента помещена перед переменной, говорят о префиксной форме записи (++n или --n). Если операция инкремента или декремента записана после переменной, то говорят о постфиксной форме записи (n++ или n--). При префиксной форме переменная сначала увеличивается или уменьшается на единицу, а затем ее новое значение используется. При постфиксной форме в выражении используется текущее значение переменной, а после ее значение изменяется на единицу.

Пример:

1) int i=1, j;  
j = i++ \* 5;

Результат:

j = 1 \* 5  
i = 2, j = 5.

2) int i=1, j;  
j = ++i \* 5

Результат:

j = 2 \* 5  
i = 2, j = 10.

Рассмотрим приоритеты операций (табл. 8).

Таблица 8

### Приоритеты операций

Операция	Комментарии
() [] -> ::	:: операция разрешения области видимости
! ~ ++ -- & *	Унарные операции: & операция адресации * операция разыменования
* / %	Арифметические операции
+ -	Арифметические операции
>> <<	Поразрядные сдвиги
< <= > >= == !=	Операций отношений
&&	Логическое И
	Логическое ИЛИ
?:	Условная операция – тернарная (трехчленная) операция: Условие? Выражение 1: Выражение 2
= += -= *= /= %= <.= >.= &.= ^=  =	Операции присваивания
,	Операция «запятая» соединяет два произвольных выражения

### Приведение типов

В арифметических выражениях, содержащих элементы различных арифметических типов, **C++Builder** в процессе вычислений автоматически осуществляет преобразование типов по принципу: если операция имеет операнды разных типов, то менее точный тип операнда приводится к более точному. Таким образом, оба операнда и результат применения операции оказываются одного типа. Это правило действует для арифметических операций, но не относится к операции присваивания. Присваивание сводится к приведению типа результата выражения к типу левого операнда.

Рассмотрим примеры неявного преобразования типов по умолчанию:

1) double a=5.4, b=2;  
int c=a\*b;

2) int m=1, n=2;  
double a=m/n;

Результат:

$c = 10$

в результате умножения  $a*b = 10.8$ , но затем дробная часть будет отброшена, так как  $c$  – целая переменная.

```
3) int m=1;
   double n=2;
   double a=m/n;
```

Результат:

$a = 0.5$

Результат:

$a = 0$

$m$  и  $n$  – целые переменные, поэтому деление  $m/n$  сведется к целочисленному делению с отбрасыванием дробной части.

```
4) double a=300, b=200;
   short c=a*b;
```

Результат:

$c = -5536$ , вместо  $60\ 000$ , так как переменная типа *short* принимает значения не больше, чем  $32767$ .

Результат неявного автоматического приведения типов можно исправить, применив операцию явного приведения типов.

Пример:

```
int m=1, n=2;
double a=(double)m/n;
```

Результат:

$a=0.5$

### Ввод исходных данных

На странице *Standard* палитры компонентов размещена компонента *Edit* – текстовый редактор, предназначенный для ввода и/или отображения одной текстовой строки. Щелкнув мышью по соответствующей пиктограмме, перенесите компоненту *Edit* на форму. Изменить свойства компоненты (имя, размер и т.п.) можно, используя *Object Inspector/Properties*. Например, чтобы очистить строку ввода, в свойстве *Text* компоненты *Edit* удалите содержащуюся там запись. Так как компонента *Edit* работает со строками, то чтобы преобразовать текст из компоненты *Edit* (имя *Edit1*) в число, потребуются специальные функции.

Например, выражение

```
y = StrToInt(Trim(Edit1->Text));
z = StrToFloat(Edit2->Text);
```

преобразует строку Edit1->Text в переменную типа Int и строку Edit2->text в действительную переменную. Возможно и обратное преобразование:

$$y = \text{IntToStr}(s);$$

$$z = \text{FloatToStr}(s1);$$

где  $s$  – переменная типа *int*,

$s1$  – переменная типа *float*.

Функции преобразования чувствительны к возможным ошибкам символьного представления числа (ошибки в программе можно блокировать с помощью редактора *MaskEdit* и функции *Trim*).

### Вывод данных

На странице *Standard* палитры компонентов размещена компонента *Label* – метка. Эта компонента используется для размещения в окне не очень длинных однострочных надписей. Щелкнув мышью по соответствующей пиктограмме, перенесите компоненту *Label* на форму. Изменить свойства компоненты (имя, размер и т.п.) можно, используя *Object Inspector/ Properties*. Например, чтобы метка содержала текст, в свойстве *Caption* компоненты *Label* удалите содержащуюся там запись и наберите необходимый текст. Обработчик события *OnClick* поможет вывести на форму необходимую информацию. Для этого нужно щелкнуть дважды по метке, и **C++Builder** автоматически подготовит заготовку функции – заголовок функции и операторные скобки. После этого следует добавить необходимые операторы в тело функции.

### Пример программирования линейного алгоритма

**Задание.** Даны целые числа  $x$  и  $y$ .

Вычислите  $F = (x^2 + \sqrt{|y|}) / (1 + |x * y|)$ .

1. На рисунке 2 – разработка алгоритма:

– входные данные:  $x$ ,  $y$  – целые числа;

– выходные данные:  $f$  – действительное число.



17

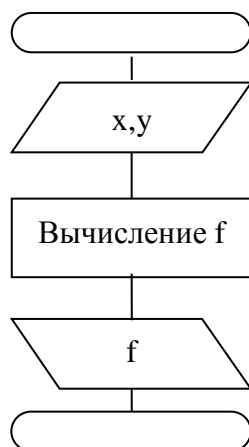


Рисунок 2 — Схема алгоритма решения задачи  
2. На рисунке 3 – разработка формы.

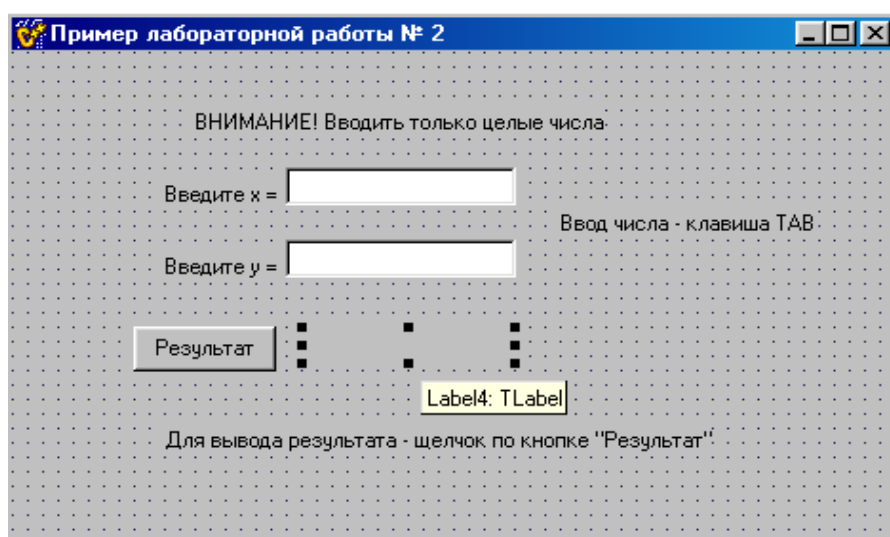


Рисунок 3 — Внешний вид формы

В таблице 9 представлены используемые компоненты, их свойства и значения.

Таблица 9

### Используемые компоненты

Имя компонента	Страница палитры <i>Компонент</i>	Настраиваемые свойства	Значения
1. Form1	-	Caption	Пример лабораторной работы № 2
2. Edit1	Standard	Text	
3. Edit2	Standard	Text	

Имя компонента	Страница палитры <i>Компонент</i>	Настраиваемые свойства	Значения
4. Label1	Standard	Caption	ВНИМАНИЕ! Вводить только целые числа!
5. Label2	Standard	Caption	Введите x
6. Label3	Standard	Caption	Введите y
7. Label4	Standard	Caption	
8. Label5	Standard	Caption	Ввод числа – клавиша TAB
9. Label6	Standard	Caption	Для вывода результата – щелчок по кнопке «Результат»
10. Button1	Standard	Caption	Результат

Текст программы:

```
//-----
#include <vcl.h>
#pragma hdrstop
#include <math.h>
#include "Unit1.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
/*обработчик события OnClick*/
void __fastcall TForm1::Button1Click(TObject *Sender)
```

```

{ int x,y; //описание переменных
float f;
//преобразование текста из Edit1 в целое число
x=StrToInt(Edit1->Text);
//преобразование текста из Edit2 в целое число
y=StrToInt(Edit2->Text);
f=(pow(x,2)+sqrt(abs(y)))/(1+abs(x*y));
/* преобразование числа f в строковый формат и вывод
результата в метку Label4*/
Label4->Caption=FloatToStr(f);
}

```

### Индивидуальные задания

Решите две задачи из первого и второго уровней сложности.

#### Задачи первого уровня сложности

Вычислите значения f и g, если:

Вариант	f	g
1	$3\sin(2^x)+0.35x^3-3.8$	$\ln(x)-x+1.8$
2	$0.25x^4+3x+2.5$	$x^8 * \operatorname{tg} x - 1/3$
3	$\sin(\ln x )-\cos(\ln x )$	$\operatorname{tg}(x/2)-\operatorname{ctg}(x/2)+x$
4	$x-2+\sin(1/x)$	$0.4x^{12}+\operatorname{arctg}(\sqrt[3]{x})-x$
5	$e^{2x+4}+\ln x -10x^5+3x$	$1-2x^7-\cos(3x)$
6	$\cos(2x)-5x^9+1$	$3.6^x-2.3x^5-3$
7	$1-x^7 + \sin x - \ln 1+x $	$3x - \sqrt{3.7x} + e^{x+5}$
8	$3 \ln(x^2) + 6 \ln x  - 6$	$x + 3.6x - \sin x^7$
9	$x - 2x * \sin x - \cos x^9$	$\cos(1.3x) + \sqrt[3]{x} * \ln x$

Вариант	f	g
10	$x^2 - \ln 1+x  - 3$	$\ln(x^2+5) - \cos^3 x$
11	$x - 1/(3 + \sin 3.6x)$	$1/(x^6 + 13x) - \operatorname{arctg} x$
12	$0.1x^4 - x \cdot \ln x $	$\cos(\operatorname{arctg}^5 x) - 3$
13	$\operatorname{tg} x - 3\operatorname{tg}^2 x + 0.2\operatorname{tg}^3 x$	$3 \cdot \ln(x^4 + 2.8) - x$
14	$\arccos x - \sqrt{ 1 - 0.3x }$	$1/\operatorname{arctg} x + \cos x - 4.2$
15	$3x^5 - 4\ln x-5 $	$(1-2x)/(x^3+5.4)$
16	$\cos(2/x) - 2\sin(1/x)$	$\operatorname{arctg} x - \sin^2 x + 5$
17	$e^x - e^{-x+2} + 5\ln x-3 $	$13x^6 + 13\sin x + 13\cos x$
18	$\sin(\ln x ) + \sqrt[3]{x}$	$\ln 0.7x - \cos^7 x$
19	$1 + \sin^2(x + y)$	$2 +  x - 2x/(1 + x^y) $
20	$\cos^2(\operatorname{arctg}(1/y))$	$2  \cos(x - \pi/6) $
21	$1/(2x) + \sin(y)$	$\sqrt[5]{x + \ln x}$
22	$y^2 +  x^2/y + x/4 $	$\sqrt[3]{5x + \lg x} + 8.2$
23	$5 \lg(x^3) + 3.2 \lg x  - 2$	$7.2^x - 5x - 9.8$
24	$\sqrt[3]{x + \lg x }$	$ y^5 - x + \sin x^2 $
25	$\sqrt[3]{2x + \sin x }$	$5 \cdot \lg(x^3 + 9.8) - x^7$
26	$\sqrt[7]{e^{3x+5} +  \operatorname{tg} x^3 }$	$2  \sin(x^3 - \pi/3) $
27	$2x^5 +  x + 7x/(\operatorname{tg}^3 x + x^y) $	$\sqrt[5]{4x^3 +  \cos^9 x^7 }$
28	$\sqrt[9]{3x^2 + \sqrt{ 3\operatorname{tg} x }}$	$\lg(3x^2 + 6x + 81) - y^8$
29	$\ln(xy^2 + 6^x + 31x) + 71.5^x$	$\sqrt[3]{4x^5 \cdot \lg \sqrt{ x }}$
30	$5 \ln(x^3) + 7.2 \log_2 x /3$	$\sqrt[3]{5e^{9x+5} +  \lg^7 x^3 }$

### Задачи второго уровня сложности

1. Вычислите дробную часть среднего геометрического трех заданных положительных целых чисел.
2. По заданным коэффициентам и правым частям системы уравнений 
$$\begin{cases} a_1x + b_1y = c_1 \\ a_2x + b_2y = c_2 \end{cases}$$
 найдите ее решение в предположении, что определитель системы не равен нулю.
3. По координатам вершин некоторого треугольника найдите его площадь и периметр.
4. По длинам двух сторон некоторого треугольника и углу (в градусах) между ними найдите длину третьей стороны и площадь этого треугольника.
5. Найдите произведение цифр заданного четырехзначного числа.
6. Определите число, полученное выписыванием в обратном порядке цифр заданного трехзначного числа.
7. Присвойте целой переменной  $d$  первую цифру из дробной части положительного вещественного числа  $x$  (так, если  $x=32.597$ , то  $d=5$ ).
8. Целой переменной присвойте сумму цифр трехзначного целого числа  $k$ .
9. Идет  $k$ -я секунда суток. Определите, сколько полных часов ( $h$ ) и полных минут ( $m$ ) прошло к этому моменту (например,  $h=3$  и  $m=40$ , если  $k=3*3600+40*60+57$ ).
10. Даны катеты прямоугольного треугольника. Найдите его периметр и площадь.
11. Даны два действительных числа. Найдите среднее арифметическое и среднее геометрическое этих чисел.
12. Даны гипотенуза и катет прямоугольного треугольника. Найдите второй катет и радиус вписанной окружности.
13. Известна длина окружности. Найдите площадь круга, ограниченного этой окружностью.
14. Найдите площадь кольца, внутренний радиус которого равен 20, а внешний заданному числу  $r$  ( $r>20$ ).
15. Даны основания и высота равнобедренной трапеции. Найдите ее периметр и площадь.

16. Вычислите дробную часть среднего арифметического трех заданных положительных чисел.

17. По координатам вершин некоторого прямоугольника найдите его площадь и периметр.

18. Дана сторона равностороннего треугольника. Вычислите площадь и периметр треугольника.

19. Дана сторона квадрата. В квадрат вписана окружность. Найдите сторону и площадь квадрата, вписанного в эту окружность.

20. Дан радиус окружности. В окружность вписан квадрат. Найдите площади окружности и квадрата.

21. Равносторонний треугольник задан координатами вершин. Найдите площадь и периметр треугольника.

22. Вычислите целую часть среднего геометрического трех заданных положительных действительных чисел.

23. Вычислите целую часть среднего арифметического четырех заданных действительных чисел.

24. Дано целое трехзначное число. Найдите вторую цифру дробной части среднего арифметического цифр этого числа.

25. Дано целое четырехзначное число. Вычислите дробную часть среднего арифметического цифр этого числа.

26. Дано целое четырехзначное число. Вычислите дробную часть среднего геометрического цифр этого числа.

27. Дано целое пятизначное число. Вычислите среднее геометрическое и среднее арифметическое 1, 3, 5 цифр этого числа.

28. Дано целое пятизначное число. Вычислите сумму и произведение 1, 3, 5 цифр этого числа.

29. Дано целое пятизначное число. Вычислите среднее арифметическое и среднее геометрическое 2, 4 цифр этого числа.

30. Даны  $x, y, z$ . Вычислите  $a, b$ , если:

$$a = \frac{3 + e^{y-1}}{1 + x |y - \operatorname{tg} z|} ; \quad b = 1 + |y-x| + \frac{(y-x)^2}{2} + \frac{|y-x|^3}{3} .$$

**Примечания:**

1. Сторона треугольника  $c = \sqrt{a^2 + b^2 - 2 \times a \times b \times \cos(g)}$ , где  $a, b$  – стороны треугольника;  $g$  – угол между сторонами  $a$  и  $b$ .

2. Площадь треугольника  $s = \sqrt{p \times (p-a) \times (p-b) \times (p-c)}$ , где  $a$ ,  $b$ ,  $c$  – стороны треугольника;  $p$  – полупериметр.

3. Радиус вписанной в прямоугольный треугольник окружности  $r = s/p$ , где  $s$  – площадь треугольника;  $p$  – полупериметр треугольника.

4. Площадь окружности  $s = \pi r^2$ , длина окружности  $l = 2\pi r$ , где  $r$  – радиус окружности.

### Контрольные вопросы к защите лабораторной работы

1. Какую структуру имеет программа?

2. Верно ли, что в программе, написанной на языке **C++Builder**, надо описывать все используемые в ней переменные?

3. Записать на **C++** следующие числа:

$-25,8 \cdot 10^{-7}$ ;  $10^6$ ;  $0,5 \cdot 10^6$ ; 7,48; 2; 4/1000.

4. Записать следующие числа без десятичного порядка:

$-0.00027E+4$ ;  $759E-3$ ;  $1E1$ .

5. Записать на **C++** следующие формулы:

$a+bx+cyz$ ;  $(1+x)^2$ ;  $(1+x)^{0.5}$ ;  
 $\cos 3x^2$ ;  $|a+bx|$ ;  $\sin 8$ ;  $\log_2 0.4x$ ;  
 $\text{arcctg } 10^3$ ;  $\text{tg } x$ ;  $\arcsin x$ ;  $x^5$ ;  $\ln(x+3)$ ;  
 $x\sqrt{2}$ ;  $\sqrt[3]{|x|}$ ;  $x^{-2}$ ;  $e^{|x-y|}$ ;  $\ln(1+3.3x)$ .

6. Вычислить значения выражения:

$\text{ceil}(6.3)$ ;  $\text{max}(2.3, 9.6)$ ;  $\text{pow}(3, 2)$ ;  
 $\text{floor}(2.7)$ ;  $\text{min}(0, 5)$ ;  $\text{sqrt}(81)$ .

7. Вычислить значение выражений:

$20 / 5$ ;  $20 \% 7$ ;  
 $2 / 5$ ;  $2 \% 7$ .

8. Указать порядок выполнения операций в выражении

$a \% b + a / b * c / a$ ;

9. Какое значение будет иметь переменная  $x$  после выполнения операторов?  $x=10$ ;  $x+=3$ ;

10. Какое значение будет иметь переменная  $x$  после выполнения операторов?  $y=3$ ;  $x=++y * 3$ ;  $x=y++ * 2$ ;

11. Какое значение будет иметь переменная  $y$  после выполнения операторов?  $x=5$ ;  $y = --x * 7$ ;  $y = x -- + 3$ ;

12. С помощью каких средств языка **C++ Builder** можно осуществить ввод данных?

13. С помощью каких средств языка **C++ Builder** можно осуществить вывод данных?

14. Каким образом можно отредактировать форму, предложенную **C++ Builder** с тем, чтобы создать свой проект?

### Содержание отчета

Отчет по лабораторной работе включает:

- титульный лист;
- условие задания;
- алгоритм решения задачи;
- текст программы;
- результаты тестирования программы.

### Список используемых источников

1. Белов В.Г. Основы программирования на языке **C++ Builder** [Текст]: учеб. пособие / В.Г. Белов, Т.М. Белова; Юго-Зап. гос. ун-т. – Курск, 2015. – 160 с.

2. Белов В.Г. Основы программирования на языке **C++ Builder** [Электронный ресурс]: учеб. пособие / В.Г. Белов, Т.М. Белова; Юго-Зап. гос. ун-т. – Курск, 2015. – 160 с.

3. Архангельский, А.Я. Программирование в **C++ Builder** [Текст] / А.Я. Архангельский. – М.: Изд-во БИНОМ, 2010. – 1304 с.

4. Дэвид Р. Мюссер. **C++** и **STL**. Справочное руководство [Текст] / Дэвид Р. Мюссер, Жилмер Дж. Дердж, Атул Сейни. – М.: Вильямс, 2010. – 432 с.

5. Культин, Н. **C++ Builder** [Текст] / Н. Культин. – СПб.: БХВ-Петербург, 2012. – 464 с.



6. Лафоре, Р. Объектно-ориентированное программирование в С++ [Текст] / Р. Лафоре. – СПб.: ПИТЕР, 2013. – 924 с.

7. Прата, С. Язык программирования С++. Лекции и упражнения [Текст] / С. Прата. – М.: Вильямс, 2012. – 1244 с.

8. Стенли Б. Липпман. Язык программирования С++. Базовый курс [Текст] / Стенли Б. Липпман, Жози Лажойе, Барбара Э. Му. – М.: Вильямс, 2014. – 1120 с.

9. Страуструп, Б. Программирование. Принципы и практика использования С++ [Текст] / Б. Страуструп. – М.: Вильямс, 2011. – 1206 с.