

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Локтионова Оксана Геннадьевна
Должность: проректор по учебной работе
Дата подписания: 12.02.2021 17:07:50
Уникальный программный ключ:
0b817ca911e6668abb13a5d426d39e5f1c11eabbf73e945df4a4851fda56d089

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 (ЮЗГУ) 2017 г.



ПРОГРАММИРОВАНИЕ ЦИКЛИЧЕСКИХ 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 циклических алгоритмов на языке С++: 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. 18 с.

Содержат основные теоретические положения и приемы разработки циклических программ на языке С++, 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. 4,0 . Уч.-изд. л. 0,9 . Тираж 100 экз. Заказ 3481 . Бесплатно.

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

Программирование циклических алгоритмов на языке C++

Цель работы — изучение и приобретение навыков программирования циклических алгоритмов, освоение операторов цикла на языке C++.

Основные понятия

На языке C++ существуют три оператора цикла:

1. Цикл с предусловием (*while*).
2. Цикл с постусловием (*do*).
3. Цикл со счетчиком (*for*).

Оператор цикла с предусловием *while*

Рассмотрим схему алгоритма оператора *while* (рисунок 1).

while (условие) {тело цикла}

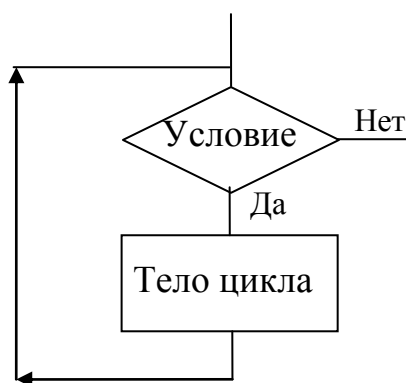


Рисунок 1 — Схема алгоритма оператора цикла *while*

Условие представляет собой выражение логического типа, тело цикла – один простой или составной оператор.

Если результат вычисления условия равен *true* (правда), то тело цикла выполняется и снова осуществляется переход к проверке условия. Если результат вычисления условия равен *false* (ложь), то происходит выход из цикла и переход к оператору, следующему за циклическим оператором *while*.

Если перед первым выполнением цикла значение выражения было *false*, то тело цикла вообще не выполняется и происходит переход на следующий оператор.

Пример:

Вычислите сумму нечетных чисел в интервале от 1 до 10.

```
{ int k=1, s=0;
  while (k<11)
  { s+=k;
    k+=2;
  }
}
```

Оператор цикла с постусловием **do**

Рассмотрим схему алгоритма оператора *do...while* (рисунок 2).

do { тело цикла } *while* (условие)

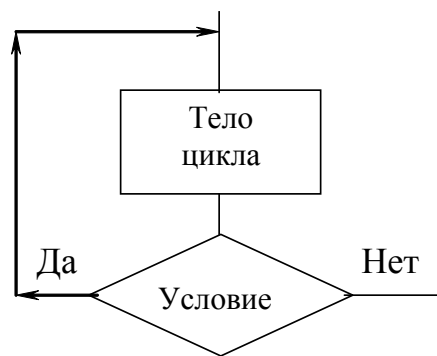


Рисунок 2 — Схема алгоритма оператора цикла *do...while*

Условие представляет собой выражение логического типа, тело цикла – один простой или составной оператор.

Тело цикла оператора *do...while* выполняется, по крайней мере, один раз, так как проверка условия выполняется после тела цикла. Поэтому данный оператор цикла используется реже по сравнению с другими операторами цикла. Тело цикла оператора *do...while* выполняется, пока условие равно *true* (не нулевое значение).

Пример:

Вычислите суммы чисел от 1 до 100.

```
int d=1, s=0;
```

```
do { s+=d;
    d+=1;
  }
while (d<100);
```

Рекомендуется использовать оператор *do...while* для контроля диапазона входных данных.

Оператор цикла со счетчиком *for*

Рассмотрим схему алгоритма оператора *for* (рисунок 3).
for (выражение 1; выражение 2; выражение 3) {тело цикла}

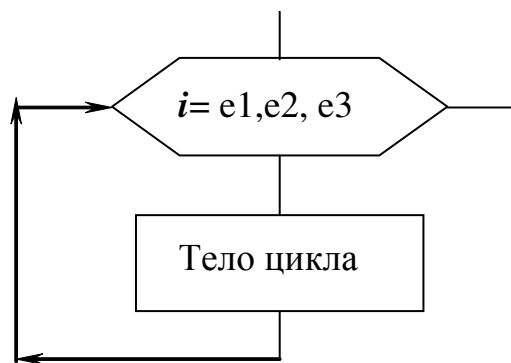


Рисунок 3 — Схема алгоритма оператора цикла *for*

Структура *for*: *e1* – выражение 1 задает начальное значение переменной, управляющей циклом; *e2* – выражение 2 является условием продолжения цикла; *e3* – выражение 3 изменяет управляющую переменную. Тело цикла – один простой или составной оператор.

Сначала выполняется выражение 1 (оно может состоять из нескольких выражений, разделенных запятой, т. е. может использоваться операция последования). Это выражение задает начальные значения переменной (или переменных) цикла. Затем проверяется выражение 2 – условие продолжения цикла. Если условие истинно (возвращает *true* – ненулевое значение), то выполняется тело цикла. После завершения тела цикла выполняется выражение 3, определяющее обычно изменение переменной цикла. Затем опять проверяется выражение 2, и при

его истинности выполнение цикла продолжается. Как только в каком-нибудь цикле выражение 2 вернет *false* (нулевое значение), цикл прерывается и управление передается оператору, расположенному следом за структурой *for*.

Приведённые ниже примеры операторов циклов демонстрируют возможности использования цикла *for*.

Операторы:	Результат:
1) <code>for (i=10; i<14; i++)</code> <code>Label1->Caption= Label1->Caption + IntToStr (i);</code>	10 11 12 13
2) можно применять операцию уменьшения для счета в порядке убывания: <code>for (n=10; n>5; n--)</code> <code>Label1->Caption= Label1->Caption+ IntToStr (n);</code>	10 9 8 7 6
3) можно вести счет двойками, десятками и т. д. <code>for (n=2; n<50; n+=13)</code> <code>Label1->Caption= Label1->Caption+IntToStr (n)</code>	2 15 28 41

В качестве третьего выражения можно использовать любое правильно составленное выражение. Обратите внимание, что в спецификации цикла проверяется значение *y*, а не *x*, т. е. в каждом из трех выражений цикла *for* могут использоваться любые переменные:

Операторы:	Результат:
<code>float x, y;</code> <code>for (x=1; y<=30; y=5*x++ +10)</code> { <code>Label2->Caption= Label2->Caption+FloatToStr (y);</code> }	1 2 3 4 5 15 20 25 30

Можно сделать так, чтобы значение некоторой величины возрастало в геометрической прогрессии:

Операторы:	Результат:
<code>float d;</code>	100 120 144

```

for (d=100.0; d<=150.0; d*=1.2)           a b c d e f
Label1->Caption= Label1->Caption+
FloatToStr (d);

```

Можно вести подсчет с помощью символов:

```

for (ch='a'; ch<='f'; ch++)
Label1->Caption= Label1->Caption+ch;

```

Можно опустить одно и более выражений, при этом предусмотрев завершение работы цикла:

```

ans=2;
for(n=3; ans<=25; )
ans*=n;

```

Пример использования в цикле *for* операции запятой (операции последования).

После цикла *for* стоит точка с запятой, что означает пустое тело цикла:

```

int sum, i;                               Результат:
for (sum=0, i=1; i<10; sum+=i++);         sum=45
Label1->Caption= "sum=" +IntToStr (sum);

```

Если переменная счетчика цикла объявлена в заголовке цикла *for*, то областью ее действия будет только этот цикл, а после завершения цикла эта переменная удаляется из памяти.

Выражения в структуре *for* являются необязательными. Иногда может отсутствовать выражение 1, если начальное значение управляющей переменной задано где-то в другом месте программы. Если отсутствует выражение 2, предполагается, что условие продолжения цикла всегда истинно и таким образом создается бесконечно повторяющийся цикл. Может отсутствовать в цикле *for* и выражение 3, если приращение переменной осуществляется операторами в теле цикла или если приращение не требуется. При пропуске какого-то из выражений, точка с запятой после пропущенного выражения (кроме третьего) должна писаться. Например:

```

int i=1; for (; i<10; ) i++;

```

Выйти из тела цикла можно, применив такие способы прерывания, как *break*, *continue*, *return*. Оператор *break* прерывает тело цикла и передает управление следующему за циклом оператору. Оператор *continue* прерывает только текущую итерацию цикла и передает управление на следующую итерацию. В отличие от оператора *break*, оператор *return* прерывает не только выполнение цикла, но и выполнение функции, в которой расположен цикл.

Пример программирования циклического алгоритма

Задание. Вычислите сумму целых чисел, значения которых лежат в диапазоне от x_1 до x_2 .

1. На рисунке 4 – разработка алгоритма:

- входные данные: x_1 , x_2 – целочисленные переменные;
- выходные данные: s – целочисленная переменная.

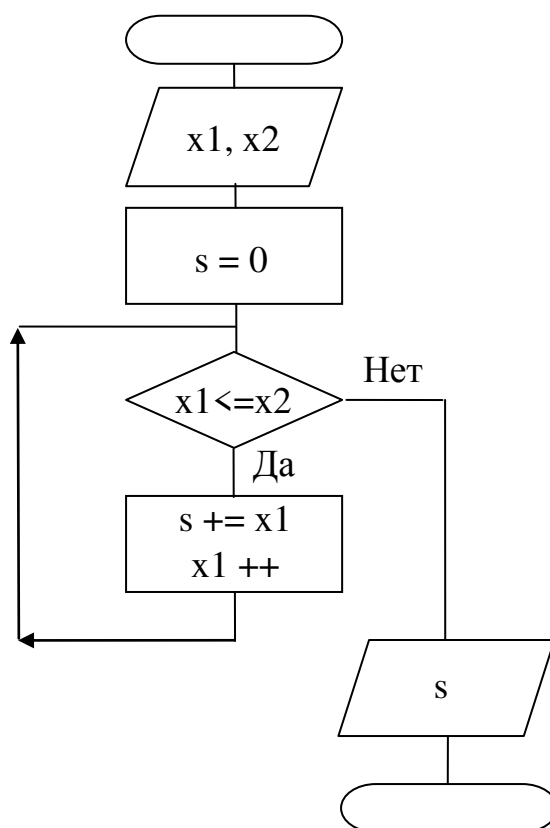


Рисунок 4 — Схема алгоритма вычисления суммы чисел

2. Разработка формы – рисунок 5, таблица 1.

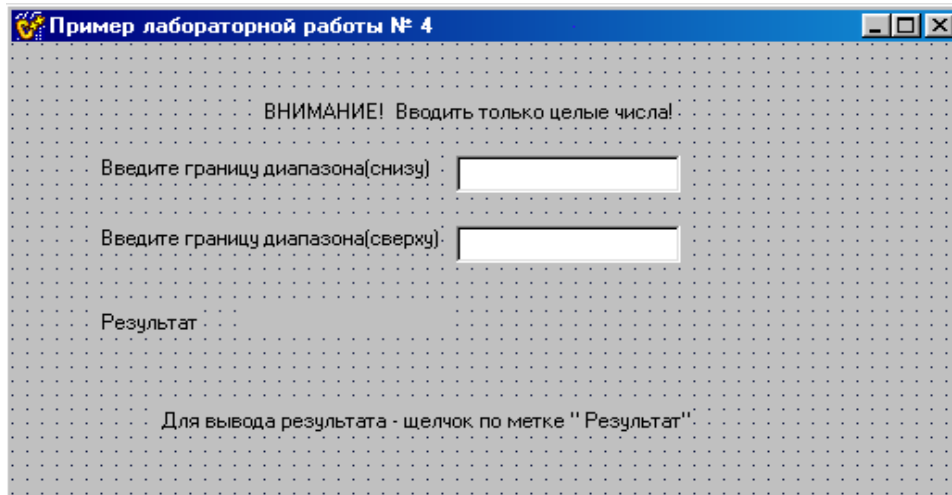


Рисунок 5 — Внешний вид формы

Таблица 1

Используемые компоненты

Имя компонента	Страница палитры компонента	Настраиваемые свойства	Значение
1. Form1	–	Caption	Пример лабораторной работы №4
2. Edit1	Standard	Text	
3. Edit2	Standard	Text	
4. Label1	Standard	Caption	ВНИМАНИЕ! Вводить только целые числа!
5. Label2	Standard	Caption	Введите границу диапазона (снизу)
6. Label3	Standard	Caption	Введите границу диапазона (сверху)
7. Label4	Standard	Caption	Результат
8. Label5	Standard	Caption	
9. Label6	Standard	Caption	Для вывода результата – щелчок по метке «Результат»

3. Текст программы:

```
//-----
#include <vcl.h>
#pragma hdrstop
#include "Unit1.h"
```

```
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TForm1::Label4Click(TObject *Sender)
{int x1, x2, s;
/*преобразуем текст из Edit1 в целое число */
x1=StrToInt (Edit1->Text);
/*преобразуем текст из Edit2 в целое число */
x2=StrToInt (Edit2->Text);
/*обнуление суммы */
s=0;
while (x1<=x2) // начало тела цикла
{ s+=x1; //прибавление к сумме текущего значения x1
  x1++; //увеличение значения x1 на 1
} // конец тела цикла
/*преобразуем число s в строку и покажем результат в метке
*/
Label5->Caption=IntToStr(s);
}
//-----
```

Индивидуальные задания

Решите две задачи из первого и второго уровней сложности.

Задачи первого уровня сложности

1. Дано натуральное число n . Вычислить $f=n!$
2. Подсчитайте k – количество цифр в десятичной записи целого неотрицательного числа n .
3. Логической переменной p присвоить значение *true*, если целое n ($n>1$) – простое число, а иначе – значение *false*.

4. Определите k – количество трехзначных натуральных чисел, сумма цифр которых равна n (значение n может быть в диапазоне $1 \leq n \leq 27$).

5. Напечатайте в возрастающем порядке все трехзначные числа, в десятичной записи которых нет одинаковых цифр.

6. Определите количество натуральных чисел, не превышающих n , которые не делятся нацело на 7.

7. Дано натуральное число n :

а) выясните, входит ли цифра 3 в запись числа n ;

б) поменяйте порядок цифр числа n на обратный.

8. Дано натуральное число n :

а) переставите первую и последнюю цифры числа n ;

б) припишите по единице в начало и в конец записи числа n .

9. Даны действительные числа x , dx . Вычислите

$$s = \sum_{i=1}^{10} (x_i^3 + 8.5), \text{ где } x_i = x_{i-1} + dx.$$

10. Даны действительные числа x , dx . Вычислите

$$s = \sum_{i=1}^{15} (x_i^2 + 1.5) / (|x_i| + 5), \text{ где } x_i = x_{i-1} + dx.$$

11. Даны действительные числа x , dx . Вычислите

$$y = \sum_{i=1}^{15} x_i^3, \text{ где } x_i = x_{i-1} + dx.$$

12. Дано натуральное число n . Подсчитайте количество единиц в записи данного числа в двоичной системе счисления.

13. Дано натуральное число n . Подсчитайте количество нулей в записи данного числа в двоичной системе счисления.

14. Дано натуральное число n . Подсчитайте количество нулей в записи данного числа в семеричной системе счисления.

15. Подсчитайте k – сумму четных цифр в десятичной записи целого неотрицательного числа n .

16. Дано натуральное число n . Подсчитайте количество цифры 3 в записи данного числа в пятеричной системе счисления.

17. Дано натуральное число n . Найдите самую меньшую по значению цифру числа n .

18. Дано натуральное число n . Найдите самую большую по значению цифру числа n .

19. Вычислите среднее арифметическое значение всех целых четных чисел, значения которых лежат в диапазоне от a до b .

20. Вычислите среднее геометрическое значение всех нечетных целых чисел, значения которых лежат в диапазоне от a до b .

21. Даны действительные числа x , dx . Вычислите

$$y = \sum_{i=1}^{12} (x_i^3 + 3.5), \text{ где } x_i = x_{i-1} + dx.$$

22. Даны действительные числа x , dx . Вычислите

$$s = \prod_{i=1}^{15} (5 + \sin x_i^2), \text{ где } x_i = x_{i-1} + dx.$$

23. Даны действительные числа x , dx . Вычислите

$$s = \sum_{i=1}^{20} (x_i^2 + x_i + 7), \text{ где } x_i = x_{i-1} + dx.$$

24. Дано натуральное число n . Подсчитайте количество единиц в записи данного числа в троичной системе счисления.

25. Дано натуральное число n . Подсчитайте количество нулей в записи данного числа в троичной системе счисления.

26. Вычислите среднее арифметическое значение всех целых чисел, значения которых лежат в диапазоне от a до b и кратны 3.

27. Вычислите среднее геометрическое значение всех целых чисел, значения которых лежат в диапазоне от a до b и не кратны 5.

28. Даны натуральные числа n и b . Число b задает основание системы счисления. Подсчитайте количество цифр в записи этого числа n в системе счисления с основанием b .

29. Подсчитать k – сумму нечетных цифр в десятичной записи целого неотрицательного числа n .

30. Дано натуральное число n . Выясните, есть ли среди цифр числа цифра b .

Задачи второго уровня сложности

1. Даны целые положительные числа x, n . Вычислите

$$\sum_{k=1}^n (k^3 \cdot \sum_{n=1}^{15} (k-1)^2 + x).$$

2. Даны целые положительные числа x, n . Вычислите

$$\sum_{i=1}^{10} \left(\sum_{j=1}^n \frac{x^3}{i+j^2} + 2 \right).$$

3. Даны целые положительные числа x, n . Вычислите

$$\sum_{i=1}^{10} \left(\sum_{j=1}^n \sin(x^3 + j^4) + 5 \right).$$

4. Даны целые положительные числа x, n . Вычислите

$$\sum_{i=1}^n \left(i + \sum_{j=n+1}^{n+10} \frac{x-j+1}{x+j} + n \right).$$

5. Даны целые положительные числа x, n . Вычислите

$$\sum_{i=1}^{100} \sum_{j=1}^i \left(\frac{i}{2 \cdot n + 1} + x^n \right).$$

6. Даны целые положительные числа x, n . Вычислите

$$\sum_{k=1}^n \sum_{m=k}^n \left(\frac{x+k}{m} + 2 \right).$$

7. Даны целые положительные числа x, n . Вычислите

$$\prod_{k=1}^{10} \left(k^3 + \prod_{i=1}^n (x^k - n)^2 + 1 \right).$$

8. Даны целые положительные числа x, n . Вычислите

$$\prod_{i=1}^n \left(j \cdot \prod_{j=1}^{n-1} \frac{x^2}{i+j^2} + n \right).$$

9. Даны целые положительные числа x, n . Вычислите

$$\prod_{i=1}^{10} \prod_{j=1}^n (\sin(x^n + j^4) + 1).$$

10. Даны целые положительные числа x, n . Вычислите

$$\prod_{i=1}^n \left(x^2 + \prod_{j=1}^{10} \frac{x-j+1}{i+j} \right).$$

11. Даны целые положительные числа x, n . Вычислите

$$\prod_{i=1}^n \prod_{j=1}^i \left(\frac{i^2}{2 \cdot j + 1} + x \right).$$

12. Даны целые положительные числа x, n . Вычислите

$$\prod_{k=1}^n \prod_{m=k}^{n+1} \frac{x^2 + k^5}{m^3}.$$

13. Даны целые положительные числа x, n . Вычислите

$$\sum_{k=1}^{n-1} \left(k^5 \cdot \prod_{i=1}^{n-2} (x^n - k)^2 + 2 \right).$$

14. Даны целые положительные числа x, n, k . Вычислите

$$\sum_{i=1}^{10} \left(k^3 \cdot \prod_{j=1}^n \frac{10x^k}{i+j^2} + n \right).$$

15. Даны целые положительные числа x, n . Вычислите

$$\sum_{i=1}^n \prod_{j=1}^{n+6} (\sin(x^7 + j^2) + n^x).$$

16. Дано целое положительное число y . Вычислите

$$\sum_{i=1}^y \prod_{j=y}^{y+10} \left(\frac{i^5 - j^2 + 1}{i+j} + 1 \right).$$

17. Даны целые положительные числа x, n . Вычислите

$$\sum_{i=1}^n \prod_{j=1}^i \left(\frac{2 + x^3}{2 \cdot j + i} + \sqrt[n]{x} \right).$$

18. Даны целые положительные числа x, n . Вычислите

$$\sum_{k=1}^n \prod_{m=k}^n \frac{x + k^2}{m}.$$

19. Даны целые положительные числа x, n . Вычислите

$$\prod_{i=1}^n \left(x^2 + \prod_{j=1}^{n+5} (\lg j^4) \right).$$

20. Даны целые положительные числа y, n . Вычислите

$$\prod_{k=1}^{10} \left(k^3 + \sum_{i=k+1}^n (i^2 - y) \right).$$

21. Даны целые положительные числа y, n . Вычислите

$$\prod_{k=1}^n (y + \sum_{i=k+1}^7 (i^2 + k^5)).$$

22. Даны целые числа y, x . Вычислите

$$\prod_{k=1}^{10} (\sqrt[3]{|y|} + \sum_{i=k+1}^9 (i^2 + x)).$$

23. Даны целые числа y, x . Вычислите

$$\prod_{k=1}^{10} (y^3 + \sum_{i=k+1}^7 (x^k)).$$

24. Даны целые положительные числа y, x . Вычислите

$$\prod_{k=1}^{15} (\sqrt{x} \cdot \sum_{i=k+1}^7 (4y^k)).$$

25. Даны целые положительные числа y и x . Вычислите

$$\prod_{k=1}^x (\sin y^3 + \sum_{i=2}^8 (\ln x^i + 2)).$$

26. Даны целые положительные числа x и y . Вычислите

$$\prod_{k=1}^{17} (\log_2 x + \sum_{i=1}^9 (\sqrt{x} + y)).$$

27. Даны целые положительные числа y, x . Вычислите

$$\prod_{k=1}^x (\lg y \cdot \sum_{i=1}^9 (k^i + x)).$$

28. Даны целые положительные числа y и x . Вычислите

$$\sum_{i=1}^x (y^3 + \prod_{j=1}^i (\frac{x^2 + 1}{x + j} + y^x)).$$

29. Задача Л. Эйлера: некий чиновник купил лошадей и быков на 1770 талеров. За каждую лошадь он уплатил по 31 талеру, а за каждого быка – по 21 талеру. Сколько лошадей и быков купил чиновник? Выясните, если решения в целых числах имеются, то сколько их – одно или несколько?

30. Математик Христиан Гольдбах (1690–1764) выдвинул гипотезу о том, что любое четное число, большее 2, представимо в

виде суммы двух простых чисел. Проверьте эту гипотезу для всех четных чисел, не превышающих число 30.

Контрольные вопросы к защите лабораторной работы

1. Как функционирует оператор цикла *for* и какая типовая схема алгоритма ему соответствует?

2. Как функционирует оператор цикла *while* и какая типовая схема алгоритма ему соответствует?

3. Как функционирует оператор цикла *do* и какая типовая схема алгоритма ему соответствует?

4. Сколько раз выполнится цикл *while* в данном примере?

`while (1)`

`Label5->Caption += "Привет!";`

5. Сколько раз выполнится цикл *while* в данном примере?

`while (0)`

`Label5->Caption += "Привет!";`

6. Запишите заголовок цикла *for* для счёта счетчика в порядке убывания с шагом 7.

7. Запишите заголовок цикла *for* для счёта счетчика в порядке возрастания с шагом 3.

8. Определите значение переменной *s* после выполнения следующих операторов:

```
int s=0, i=1;
while ( i!=6 )
{ i++;
  s+=2*i-1;
}
```

9. Определите значение переменной *s* после выполнения следующих операторов:

```
int s=0, i=1;
for (x=10; x!=0; x-=2)
s +=x;
```


10. Можно ли в заголовке цикла *for* для значения счетчика использовать символы?

11. Приведите пример заголовка цикла *for*, где упускалось хотя бы одно из выражений.

12. Определите значение переменной *s* после выполнения следующих операторов:

```
int s=1;  
do s+=2;  
while(s<3)
```

13. Запишите бесконечный цикл с помощью оператора *for*.

14. С помощью каких операторов можно прервать цикл?

15. Каким образом выполняются операторы *break* и *continue*?

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

5. Культин, Н. С++ 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.