МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

Юго-Западный государственный университет

(ЮЗГУ)

Кафедра вычислительной техники

ПРОГРАММИРОВАНИЕ ЦИКЛИЧЕСКИХ АЛГОРИТМОВ

Методические указания по выполнению лабораторной работы

для студентов направления подготовки 09.03.01

Курск 2016

УДК 621.3
Составитель: Э.И. Ватутин

Рецензент
Кандидат технических наук, доцент *В.С. Панищев*

Программирование циклических алгоритмов: методические указания по выполнению лабораторных работ по дисциплине «Программирование» / Юго-Зап. гос. ун-т; сост.: Э.И. Ватутин; Курск, 2016. 13 с.: ил. 2.

Методические рекомендации содержат сведения по разработке циклических программ на современных языках программирования высокого уровня.

Предназначены для студентов направления подготовки 09.03.01 «Информатика и вычислительная техника».

Текст печатается в авторской редакции

Подписано в печать _____. Формат 60x84 1/16.
Усл. печ. л. Уч. – изд.л. Тираж 30 экз. Заказ . Бесплатно.
Юго-Западный государственный университет
305040, Курск, ул. 50 лет Октября, 94.

Содержание

Введение 4
Индивидуальные задания 9
Содержание отчета 12
Контрольные вопросы 12
Библиографический список 12

Введение

Целью работы является получение практических навыков при программировании циклических алгоритмов с использованием операторов циклов с предусловием, с постусловием и со счетчиком.

Многие программы требуют циклического (итерационного) повторения выполнения группы операторов. Для их реализации в составе современных языков программирования высокого уровня предусмотрены специальные языковые конструкции.

Граф-схема *цикла с предусловием* приведена на рис. 1.

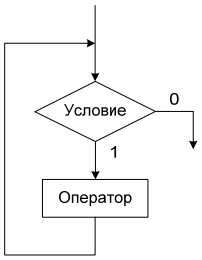

Рис. 1. Граф-схема цикла с предусловием

На языках программирования Delphi с C++ такой цикл записывается следующим образом:

Delphi

```
while Условие do
  Оператор;
```

C++

```
while (Условие)
  Оператор;
```

В качестве условия выступает выражение булевского типа. Тело цикла выполняется до тех пор, пока значение условия истинно. Как только условие становится ложным, управление передается следующему за циклом оператору. Значение условия вычисляется на каждой итерации цикла. Тело цикла может не выполниться ни разу, если при первом проходе условие ложно. При необходимости записи нескольких операторов в теле цикла необходимо использовать операторные скобки.

В качестве примера рассмотрим нахождение суммы квадратов целых чисел от 1 до *N*.

$$s = \sum_{i=1}^{N} i^2.$$

Для использования данного цикла в задаче нахождения суммы необходимо определиться с инициализирующими действиями, телом цикла и условием завершения. При вычислении необходимо использовать вспомогательную переменную `I`, в которой хранится текущее число. Изначально оно равно 1, на каждом шаге производится переход к следующему слагаемому путем увеличения его значения на единицу. Значение суммы изначально равно нулю, на каждом шаге оно увеличивается на выражение `I*I`. В теле цикла два оператора, поэтому необходимо использовать операторные скобки. Цикл продолжается до тех пор, пока значение переменной `I` не больше `N`. Соответствующие программы с использованием цикла `while` представлены ниже.

Delphi

```
var
  I, N, S: Integer;

begin
  N := 10;

  I := 1;
  S := 0;
  while I <= N do begin
    S := S + I*I;
    Inc(I);
  end;

  Writeln('S = ', S);
  Readln;
end.
```

C++

```
void main()
{
	int N = 10;
	int S = 0;
	int i = 1;

	while (i <= N)
	{
		S += i*i;
		i++;
	}

	cout << S << endl;
	getchar();
}
```

Еще одним видом цикла является *цикл с постусловием*. В нем условие располагается после тела цикла.

Рис. 2. Граф-схема цикла с постусловием

В языках программирования Delphi и C++ такой цикл записывается следующим образом:

Delphi

```
repeat
  Оператор1;
  Оператор2;
  ...
  ОператорN;
until Условие;
```

C++

```
do {
    Оператор1;
    Оператор2;
    ...
    ОператорN;
} while (Условие);
```

В качестве условия также выступает выражение булевского типа, ложное значение которого производит переход на новую итерацию цикла, а истинное является завершением цикла (наоборот по отношению к циклу `while`). Фактически цикл выполняется до тех пор, пока не станет истинным условие. В языке C++ цикл с постусловием имеет противоположное условие завершения.

Значение условия вычисляется на каждой итерации цикла. Тело цикла всегда выполняется хотя бы один раз, т.к. проверка условия завершения производится после тела цикла. В Delphi при необходимости записи нескольких операторов в теле цикла операторные скобки не применяются.

Для нахождения рассмотренной выше суммы квадратов чисел используются те же самые действия инициализации и операторы тела цикла. Программы с использованием цикла с постусловием выглядят следующим образом:

Delphi

C++

```
var
  I, A, B, S: Integer;

begin
  A := 3;
  B := 10;

  I := A;
  S := 0;
  repeat
    S := S + I*I;
    Inc(I);
  until I > B;

  Writeln('S = ', S);
  Readln;
end.
```

```
void main()
{
        int N = 10;
        int S = 0;
        int i = 1;

        do {
                S += i*i;
                i++;
        } while (i <= N);

        cout << S << endl;
        getchar();
}
```

Еще одним типом цикла является *цикл со счетчиком*. Данная форма цикла является достаточно удобной при записи цикла в случае, когда заранее известны пределы изменения переменной-счетчика (фактически количество итераций цикла) и шаг изменения счетчика равен +1 или –1. В программах на Delphi данный цикл записывается в следующем виде:

```
for Переменная := Начальное_значение to/downto Конечное_значение do
  Оператор;
```

В качестве переменной (называемой *счетчиком цикла*) можно использовать переменные порядкового типа. Если цикл используется в подпрограмме, то переменная должна быть локальной, иначе возможно появление предупреждений или ошибок компилятора. Это сделано с одной стороны для повышения эффективности, т.к. переменные-счетчики обычно размещаются в регистрах процессора, с другой стороны это ограждает от ряда ошибок, запрещая использовать одну и ту же переменную-счетчик, например, в вызывающих друг друга подпрограммах. Значения выражений «Начальное значение» и «Конечное значение» вычисляются однократно перед выполнением цикла и определяют диапазон изменения переменной-счетчика. Если используется ключевое слово «to», то на каждой итерации переменная-счетчик увеличивается на единицу; при использовании слова «downto» – уменьшается на единицу. Выполнение цикла завершается и управление передается на следующий за циклом оператор при выходе значения переменной-счетчика за пределы диапазона (при использовании

ключевого слова «to» переменная-счетчик становиться больше конечного значения, «downto» – меньше).

Во время выполнения тела цикла переменную-счетчик запрещается модифицировать непосредственно в теле цикла. Если возникает подобная необходимость, то это свидетельствует об ошибке в логике: необходимо либо модифицировать алгоритм, либо использовать другой тип цикла.

После цикла значение переменной-счетчика в общем случае является неопределенным, его не рекомендуется использовать, о чем обычно в виде предупреждения сообщает компилятор.

В языке C++ цикл со счетчиком имеет больше возможностей и записывается в следующем виде:

```
for (Инициализация; Условие_завершения; Действие_между_итерациями)
    Оператор;
```

Действие «Инициализация» выполняется однократно перед выполнением тела цикла. Цикл выполняется до тех пор, пока выражение «Условие_завершения» истинно. При переходе от итерации к итерации выполняются операторы, записанные в секции «Действие_между_итерациями».

Для нахождения рассмотренный выше суммы квадратов с использованием цикла со счетчиком можно использовать следующую последовательность действий:

Delphi

```
var
  A, B, S, I: Integer;

begin
  A := 3;
  B := 10;

  S := 0;
  for I := A to B do
    S := S + I*I;

  Writeln('For S = ', S);
  Readln;
end.
```

C++

```
void main()
{
        int N = 10;
        int S = 0;
        int i = 1;

        for (int i = 1; i <= N; i++)
                S += i*i;

        cout << S << endl;
        getchar();
}
```

Иногда в процессе выполнения цикла возникает необходимость в его прерывании либо досрочном переходе к следующей итерации цикла. Для прерывания цикла используется процедура `break`, для досрочного перехода к следующей итерации – `continue`. Поддержка данных процедур интегрирована в компилятор, делая их похожими на операторы. Процедурами `break` и `continue` можно пользоваться только в пределах тела цикла, в противном случае компилятор выдаст сообщение об ошибке. Если имеют место вложенные циклы, то процедуры `break` и `continue` оказывают влияние на изменение выполнения наиболее вложенного цикла, в котором они расположены.

Достаточно часто в программах используются циклы, условие выхода из которых никогда не выполняется (умышленно). Подобные циклы называются *бесконечными* и записываются следующим образом:

Delphi

```
while True do
  Оператор;

repeat
  Оператор;
until False;
```

C++

```
while (1)
       Оператор;

do {
       Оператор;
} while (1);
```

Если программа попадает в такой цикл, то она перестает реагировать на действия пользователя – «повисает», что является крайне нежелательным. Поэтому обычно бесконечные циклы используются в совокупности с процедурой `break`, которая выполняется по какому-либо условию в теле цикла и тем самым прерывает его.

Индивидуальные задания

1. Вывести на экран все простые числа в диапазоне от 1 до *N*. Число называется простым, если оно делится без остатка только само на себя и на 1.

2. Найти натуральное число в диапазоне от 1 до N с максимальной суммой делителей.

3. Представить заданное число в виде произведения простых чисел (например, $5040 = 2^4 \cdot 3^2 \cdot 5 \cdot 7$).

4. Найти наименьшее натуральное число N из диапазона $[A, B]$, представимое в виде суммы кубов двух натуральных чисел $N = x^3 + y^3$.

5. Натуральное число из N цифр называется числом Армстронга, если оно может быть представлено в виде суммы его цифр, возведенных в N-ую степень (например, $153 = 1^3 + 5^3 + 3^3$). Найти числа Армстронга в заданном диапазоне.

6. Указать все способы представления заданного числа N в виде суммы квадратов 3 натуральных чисел $N = x^2 + y^2 + z^2$.

7. Согласно теореме Лагранжа о сумме четырех квадратов любое число можно представить в виде суммы не более чем 4 квадратов натуральных чисел. Проверить справедливость теоремы для чисел из заданного диапазона.

8. (Плоские числа) Найти и вывести на экран все числа, меньшие N, единственным образом представимые в виде произведения простых чисел A и B (например, $985 = 5 \cdot 197$).

9. (Телесные числа) Найти и вывести на экран все числа, меньшие N, единственным образом представимые в виде произведения простых чисел A, B и C (например, $385 = 5 \cdot 7 \cdot 11$).

10. Согласно теории чисел НОД (наибольший общий делитель) двух чисел представим в виде их линейной комбинации. Проверить справедливость теоремы для двух заданных чисел.

11. В приведенной последовательности сумм

$$S_1 = 1,$$
$$S_2 = 3 + 5 = 8,$$
$$S_3 = 7 + 9 + 11 = 27,$$
$$\ldots$$

вычислить значение *i*-ой суммы. Вывести на экран все промежуточные значения.

12. Найти наилучшее приближение к заданному иррациональному числу (π, e, $\sqrt{2}$ и т.д.) в виде рациональной дроби $\frac{A}{B}$ в заданном диапазоне ($1 \le A \le N$, $1 \le B \le N$).

13. Проверить справедливость соотношения $\sum_{i=0}^{n} C_n^i = 2^n$, где $C_n^m = \frac{n!}{m!(n-m)!}$ – число сочетаний из *n* по *m*.

14. Проверить справедливость соотношения $\sum_{i=0}^{n} (-1)^i C_n^i = 0$, где $C_n^m = \frac{n!}{m!(n-m)!}$ – число сочетаний из *n* по *m*.

15. (Тождество Абеля) Убедиться в справедливости соотношения $2(n-1)n^{n-2} = \sum_{\substack{k+m=n,\\ k>1,\, m>1}} C_n^k k^{k-1} m^{m-1}$, где $C_n^m = \frac{n!}{m!(n-m)!}$ – число сочетаний из *n* по *m*.

16. Проверить справедливость соотношения $n! = 1 + \sum_{i=1}^{n-1} i \cdot i!$.

17. Проверить справедливость соотношения $\sum_{i=1}^{n} i^3 = \left(\sum_{i=1}^{n} i \right)^2 = \frac{n^2 (n+1)^2}{4}$.

18. Проверить справедливость соотношения $\sum_{k=0}^{n-1} 2^k = 2^n - 1$.

19. Проверить справедливость соотношения $(x-a)^n \equiv x^n - a \pmod{n}$, где x, a, n – целые положительные числа, n – простое число, $a \le x$, с использованием датчика случайных чисел.

20. (малая теорема Ферма) Если a не делится на простое число p, то $a^{p-1}-1$ делится на p. Проверить справедливость теоремы с использованием датчика случайных чисел.

Содержание отчета

1. Титульный лист.
2. Индивидуальное задание.
3. Краткое описание стратегии решения.
4. Листинг программы.
5. Тестовые примеры, результаты тестирования.
6. Выводы.

Контрольные вопросы

1. Для чего применяются цикла в программах?
2. Какие типы циклов существуют? Как они записываются в конструкциях языков высокого уровня?
3. Что является условие завершения цикла каждого конкретного типа?
4. Какое минимальное и максимальное количество раз может быть выполнен цикл каждого из типов?
5. Что такое бесконечные циклы?
6. Какие средства языка применяются для досрочного прерывания тела цикла?
7. Какие средства языка применяются для досрочного перехода к следующей итерации цикла?

Библиографический список

1. Емельянов С.Г., Ватутин Э.И., Панищев В.С., Титов В.С. Процедурно-модульное программирование на Delphi: учебное пособие. М.: Аргамак-Медиа, 2014. 352 с.

2. Зотов И.В., Ватутин Э.И., Борзов Д.Б. Процедурно-ориентированное программирование на C++: учебное пособие. Курск: КурскГТУ, 2008. 211 с.