

Министерство науки и высшего образования Российской Федерации
Казанский национальный исследовательский технический университет –
КАИ им. А.Н. Туполева

Институт компьютерных технологий и защиты информации
Отделение СПО ИКТЗИ «Колледж информационных технологий»

ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЕ

Методические указания к лабораторным работам

Казань 2020

Составитель преподаватель СПО ИКТЗИ Мингалиев Заид Зульфатович

Методические указания к лабораторным работам по дисциплине «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЕ» предназначены для студентов направления подготовки 09.02.07 «Информационные системы и программирование»

КНИТУ-КАИ, 2020

ОГЛАВЛЕНИЕ

ПРОЦЕСС СДАЧИ ВЫПОЛНЕННОЙ РАБОТЫ	4
ЛАБОРАТОРНАЯ РАБОТА №3.	5
Программирование алгоритмов циклической структуры.	

ПРОЦЕСС СДАЧИ ВЫПОЛНЕННОЙ РАБОТЫ

По итогам выполнения работы студент:

1. демонстрирует преподавателю правильно работающие программы;
2. демонстрирует приобретённые знания и навыки отвечает на пару небольших вопросов преподавателя по составленной программе, возможностям её доработки;
3. демонстрирует отчет по выполненной лабораторной работе.

Итоговая оценка складывается из оценок по трем указанным составляющим.

Шаблон оформления отчета представлен в приложении 1. Требования к формированию отчета представлены в приложении 2.

ЛАБОРАТОРНАЯ РАБОТА №3.

Программирование алгоритмов циклической структуры.

ЦЕЛЬ РАБОТЫ

Приобрести умения и практические навыки для программирования алгоритмов циклической структуры.

ХОД РАБОТЫ

1) Цикл while

Циклический алгоритм – это описание действий, которые в зависимости от исходных данных могут повторяться многократно. Последовательность действий, предназначенная для многократного исполнения, называется **телом цикла**.

Итерационные циклы – это алгоритмы, в которых тело цикла выполняется в зависимости от какого-либо условия. Различают циклы с **предусловием** и **постусловием**.

Цикл с предусловием (цикл while) – цикл, который выполняется, пока истинно некоторое условие, указанное перед его началом. Это условие проверяется до выполнения тела цикла, поэтому последнее может быть ни разу не выполнено (если условие с самого начала ложно).

Цикл `while` является самым простым циклов, которые есть в языке C++. Он очень похож на ветвление `if/else`:

1	<code>while (<условие>)</code>
2	<code>{</code>
3	<code><тело цикла>;</code>
4	<code>}</code>

Цикл `while` объявляется с использованием ключевого слова `while`. В начале цикла обрабатывается условие. Если его значением является `true` (любое ненулевое значение), то тогда выполняется тело цикла.

Однако, в отличие от оператора `if`, после завершения выполнения тела цикла, управление возвращается обратно к `while` и процесс проверки условия

повторяется. Если условие опять является `true`, то тогда тело цикла выполняется еще раз.

Например, следующая программа выводит все числа от 0 до 9:

1	<code>#include <iostream></code>
2	<code>using namespace std;</code>
3	<code>int main()</code>
4	<code>{</code>
5	<code>int count = 0;</code>
6	<code>while (count < 10)</code>
7	<code>{</code>
8	<code>cout << count << " ";</code>
9	<code>++count;</code>
10	<code>}</code>
11	<code>cout << "done!";</code>
12	<code>return 0;</code>
13	<code>}</code>

Результат выполнения программы:

0 1 2 3 4 5 6 7 8 9 done!

Рассмотрим детально эту программу. Во-первых, инициализируется переменная-счетчик: `int count = 0;`. Условие `0 < 10` имеет значение `true`, поэтому выполняется тело цикла. В первом выражении (строка 8) мы выводим 0, а во втором (строка 9) — выполняем инкремент переменной `count`. Затем управление возвращается к началу цикла `while` для повторной проверки условия. Условие `1 < 10` имеет значение `true`, поэтому тело цикла выполняется еще раз. Тело цикла будет повторно выполняться до тех пор, пока переменная `count` не будет равна 10, только в том случае, когда результат условия `10 < 10` будет `false`, цикл завершится.

2) Цикл do-while

Цикл с постусловием (цикл do-while) – цикл, в котором условие проверяется после выполнения тела цикла. Отсюда следует, что тело данного цикла всегда выполняется хотя бы один раз.

Синтаксис цикла имеет следующий вид:

1	do
2	{
3	<тело цикла>;
4	}
5	while (<условие>;

Тело цикла do while всегда выполняется хотя бы один раз. После выполнения тела цикла проверяется условие. Если оно истинно, то выполнение переходит к началу блока do и тело цикла выполняется снова. Если оно ложно, то происходит выход из цикла.

Ниже приведен пример использования цикла do while для отображения меню:

1	#include <iostream>
2	using namespace std;
3	int main()
4	{
5	// Переменная choice должна быть объявлена вне цикла do while
6	int choice;
7	do
8	{
9	cout << "Please make a selection: \n";
10	cout << "1) Addition\n";
11	cout << "2) Subtraction\n";
12	cout << "3) Multiplication\n";

13	<code>cout << "4) Division\n";</code>
14	<code>cin >> choice;</code>
15	<code>}</code>
16	<code>while (choice != 1 && choice != 2 &&</code>
17	<code>choice != 3 && choice != 4);</code>
18	<code>// Что-то делаем с переменной choice, например,</code> <code>используем оператор switch</code>
19	<code>cout << "You selected option #" << choice << "\n";</code>
20	<code>return 0;</code>
21	<code>}</code>

Переменная `choice` должна быть объявлена вне блоков `do while`. Если бы переменная `choice` была объявлена внутри блока `do`, то она была бы уничтожена при завершении этого блока еще до проверки условия `while`. Но нам нужна переменная, которая будет использоваться в условии `while`, следовательно, переменная `choice` должна быть объявлена вне блока `do`.

3) Цикл for

Цикл со счетчиком (цикл `for`) – это циклический алгоритм, в котором тело цикла выполняется заранее известное число раз.

Параметр цикла (счетчик) изменяет свое значение от заданного начального до заданного конечного, и для каждого значения этой переменной тело цикла выполняется один раз.

Цикл `for` в языке C++ идеален, когда известно необходимое количество итераций. Выглядит он следующим образом:

1	<code>for(<объявление переменных>; <условие>;</code> <code><инкремент/декремент счетчика>)</code>
2	<code>{</code>
3	<code><тело цикла>;</code>
4	<code>}</code>

Переменные, определенные внутри цикла `for`, имеют специальный тип области видимости: область видимости цикла. Такие переменные существуют только внутри цикла и недоступны за его пределами.

Цикл `for` в C++ выполняется в 3 шага:

Шаг №1: Объявление переменных. Как правило, здесь выполняется определение и инициализация счетчиков цикла, а точнее — одного счетчика цикла. Эта часть выполняется только один раз, когда цикл выполняется впервые.

Шаг №2: Условие. Если оно равно `false`, то цикл немедленно завершает свое выполнение. Если же условие равно `true`, то выполняется тело цикла.

Шаг №3: Инкремент/декремент счетчика цикла. Переменная увеличивается или уменьшается на единицу. После этого цикл возвращается к шагу №2.

Рассмотрим пример цикла `for` и разберемся детально, как он работает:

1	<code>#include <iostream></code>
2	<code>using namespace std;</code>
3	<code>int main()</code>
4	<code>{</code>
5	<code>for (int count = 0; count < 10; ++count)</code>
6	<code>{</code>
7	<code>cout << count << " ";</code>
8	<code>}</code>
9	<code>return 0;</code>
10	<code>}</code>

Сначала мы объявляем переменную `count` и присваиваем ей значение 0. Далее проверяется условие `count < 10`, а так как `count` равен 0, то условие `0 < 10` имеет значение `true`. Следовательно, выполняется тело цикла, в котором мы выводим в консоль переменную `count` (т.е. значение 0).

Затем выполняется выражение `++count`, т.е. инкремент переменной. Затем цикл снова возвращается к проверке условия. Условие `1 < 10` имеет значение `true`, поэтому тело цикла выполняется опять. Выводится 1, а переменная `count` увеличивается уже до значения 2. Условие `2 < 10` является истинным, поэтому выводится 2, а `count` увеличивается до 3 и так далее.

В конце концов, `count` увеличивается до 10, а условие `10 < 10` является ложным, и цикл завершается. Следовательно, результат выполнения программы:

0 1 2 3 4 5 6 7 8 9

4) Задание на лабораторную работу

Обе части лабораторной работы должны быть выполнены либо на языке программирования C++, либо на языке программирования C#.

Часть 1

В соответствии с вариантом необходимо написать консольную программу.

Варианты	Индивидуальное задание
1	Задана последовательность чисел от 1 до 200. Найти сумму квадратов этих чисел.
2	Необходимо протестировать группу из n человек. Каждый из них вводит: 1 – если он изучал английский язык, 2 – если немецкий, 3 – если французский, 0 – если не изучал никакой. Составить программу, определяющую, сколько человек в каждой языковой группе.
3	Имеется часть катушки с автобусными билетами. Номер билета шестизначный. Составить программу, определяющую

Варианты	Индивидуальное задание
	количество счастливых билетов на катушке, если меньший номер билета – n, больший – m.
4	<p>Вычислить</p> $1 + \frac{1}{3 + \frac{1}{5 + \frac{1}{7 + \frac{1}{\dots}}}}$ $101 + \frac{1}{103}$
5	Дано натуральное число n. Вычислить $1 \cdot 2 + 2 \cdot 3 \cdot 4 + \dots + n \cdot (n + 1) \cdot \dots \cdot 2n$.
6	<p>Вычислить</p> $\sum_{a=3}^{103} \frac{1}{(2a)^2}$
7	Дано натуральное число n. Проверить, является ли оно совершенным (число называется совершенным, если оно равно сумме всех своих делителей).
8	<p>Дано натуральное число i. Вычислить</p> $\sum_{i=1}^{40} (a_i - b_i)^2$ <p>где</p> $a_i = \begin{cases} i, & \text{при нечетном } i \\ i - 1, & \text{в противоположном случае} \end{cases}$ $b_i = \begin{cases} i^2, & \text{при нечетном } i \\ i^3, & \text{в противоположном случае} \end{cases}$
9	<p>Дано натуральное число x. Вычислить</p> $\frac{(x - 2)(x - 4)(x - 8) \dots (x - 128)}{(x - 1)(x - 3)(x - 7) \dots (x - 127)}$

Варианты	Индивидуальное задание
10	Вычислить k – количество точек с целочисленными координатами, попадающих в круг радиуса R ($R > 0$) с центром в начале координат.
11	Даны действительное число a , натуральное число n . Вычислить $\frac{1}{a} + \frac{1}{a^2} + \frac{1}{a^4} + \dots + \frac{1}{a^{2n}}$
12	Даны действительное число a , натуральное число n . Вычислить $\frac{1}{a} + \frac{1}{a(a+1)} + \frac{1}{a(a+1)(a+2)} + \dots + \frac{1}{a(a+1) \dots (a+n)}$
13	Вычислить $(1 + \sin 0.1)(1 + \sin 0.2) \dots (1 + \sin 10)$.
14	Дано натуральное число n . Получить все простые делители этого числа.
15	Составьте программу, подсчитывающую количество цифр вводимого вами целого неотрицательного числа. Используйте операцию целочисленного деления.
16	Вывести в порядке убывания все делители данного числа.
17	Составьте программу, позволяющую найти все числа ряда Фибоначчи, меньше заданного числа N . Ряд Фибоначчи – это бесконечная последовательность чисел, каждое из которых является суммой двух предыдущих элементов. 1,1,2,3,5,8,13,21,34,55,89,144, ...
18	Вводятся по очереди данные о росте N учащихся в классе. Определить средний, минимальный и максимальный рост учащихся в классе.
19	Дано натуральное число n ($n < 99999$). Найти предпоследнюю цифру введенного числа (в предположении, что $n > 10$).

Варианты	Индивидуальное задание
20	<p>Пусть $a_1 = u; b_1 = v; a_k = 2b_{k-1} + a_{k-1}; b_k = 2a_{k-1}^2 + b_{k-1}; k = 2, 3, \dots$</p> <p>Даны действительные числа u, v и натуральное число n. Найти</p> $\sum_{k=1}^n \frac{a_k b_k}{(k+1)!}$

Часть 2.

В соответствии с вариантом необходимо написать консольную программу.

Варианты	Индивидуальное задание
1	<p>Вычислить</p> $\frac{1}{n!} \sum_{k=1}^n (-1)^k \frac{x^k}{(k! + 1)!}$
2	<p>Вычислить</p> $\sum_{k=1}^{10} k^3 \sum_{l=1}^{15} (k-l)^2$
3	<p>Вычислить</p> $\sum_{i=1}^{50} \sum_{j=1}^{100} \frac{j-i+1}{i+j}$
4	<p>Дано натуральное число N. Найти все натуральные числа меньшие N, которые являются палиндромами.</p>
5	<p>Дано натуральное число N. Найти все натуральные числа меньшие N, которые при возведении в квадрат дают палиндром.</p>
6	<p>Вычислить y – первое из чисел $\sin(x), \sin(x) \sin(x), \sin(x) \sin(x) \sin(x), \dots$, меньшее по модулю 10^{-4}.</p>

Варианты	Индивидуальное задание
7	Остров Манхэттен был приобретен поселенцами \$24 в 1826 году. Каково было в настоящее время состояние их счета, если бы эти \$24 были помещены тогда в банк под 8% годового дохода?
8	Перевести целое число из десятичной системы счисления в двоичную, используя алгоритм деления на 2.
9	Перевести целое число из десятичной системы счисления в шестнадцатеричную, используя алгоритм деления на 2.
10	Дано натуральное число. Найти, сколько раз в нем встречается каждая цифра.
11	Спортсмен-лыжник начал тренировки, пробежав в первый день 10 км. Каждый следующий день он увеличивал длину пробега P процентов от пробега предыдущего дня (P — вещественное, $0 < P < 50$). По данному P определить, после какого дня суммарный пробег лыжника за все дни превысит 200 км, и вывести найденное количество дней K (целое) и суммарный пробег S (вещественное число).
12	Найти в промежутке от 1 до N все числа, у которых пять делителей.
13	Найти все четырехзначные числа, у которых все цифры различны.
14	<p>Для заданных значения переменной x и количества слагаемых ряда n вычислить по приведённой приближённой формуле значение указанной функции и путём сравнения с точным значением функции (левой части) найти погрешность вычислений.</p> $\cos x \approx 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + (-1)^n \frac{x^{2n}}{(2n)!} \pm \dots;$

Варианты	Индивидуальное задание
15	<p>Составить программу по проверке знаний таблицы умножения.</p> <p>Программа должна вывести по очереди 10 примеров и выставить оценку. Если 10 правильных ответов – оценка «отлично», если 8-9 правильных ответов – оценка «хорошо», если 6-7 правильных ответов – оценка «удовлетворительно», все остальное – оценка «неудовлетворительно».</p>
16	<p>Для заданных значения переменной x и количества слагаемых ряда n вычислить по приведённой приближённой формуле значение указанной функции и путём сравнения с точным значением функции (левой части) найти погрешность вычислений.</p> $\ln x \approx (x - 1) - \frac{(x - 1)^2}{2} + \frac{(x - 1)^3}{3} - \frac{(x - 1)^4}{4} + \dots + (-1)^{n+1} \frac{(x - 1)^n}{n} \pm \dots; 0 < x \leq 2;$
17	<p>Коля очень любит наблюдать за электронными часами. Он целыми днями смотрел на часы и считал, сколько раз встречается каждая цифра. Через несколько месяцев он научился по любому промежутку времени говорить, сколько раз на часах за это время встретится каждая цифра, и очень гордился этим.</p> <p>На входе вводится два значения: начало и конец промежутка времени. Время задается в формате чч:мм:сс. На выходе надо получить количество попаданий цифр в данный временной промежуток.</p>
18	<p>Требуется вывести в консоль от 1-го до 6-ти прямоугольных треугольников, в зависимости от того какое число введет пользователь. Например, если на вход в программу будет отправлено число 3, то необходимо напечатать 3 треугольника, если пользователь введет 6, то напечатать 6-ть треугольников.</p>

Варианты	Индивидуальное задание
	Треугольники нужно выводить не в столбик, а в строчку. При этом размер треугольников также формируется по введенному пользователем числу. Если пользователь введет число 3, то должно напечататься три прямоугольных треугольника, катеты которых равны — 3-м. Для печати треугольников можно выбрать любой символ.
19	<p>Для заданных значения переменной x и количества слагаемых ряда n вычислить по приведённой приближённой формуле значение указанной функции и путём сравнения с точным значением функции (левой части) найти погрешность вычислений.</p> $\sin x \approx 1 - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + (-1)^n \frac{x^{2n+1}}{(2n+1)!} \pm \dots;$
20	Поле шахматной доски определяется парой натуральных чисел, каждое из которых не превосходит 8: первое число — номер вертикали (при счете слева направо), второе — номер горизонтали (при счете снизу-вверх). Даны натуральные числа a, b, c, d, e, f , каждое из которых не превосходит 8. Определить, может ли белая ладья, расположенная на поле (a, b) , одним ходом пойти на поле (e, f) , не попав при этом под удар черной ладьи, находящейся на поле (c, d) .

5) Контрольные вопросы

1. Что такое цикл?
2. Чем отличается цикл while от цикла do-while?
3. Из каких блоков состоит цикл с параметром?
4. Что такое бесконечный цикл? Как избежать заикливания?
5. Что такое вложенный цикл? Для чего используются вложенные циклы?
6. Для чего используются операторы continue и break?