

Занятие 8. Цикл с постусловием

Начнем с задачи. Требуется ввести целое положительное число и распечатать цифры его десятичного представления, начиная с младшей. Опорный пример может выглядеть так.

Введите натуральное число
125
Ответ:
5
2
1

Видим, что задача носит циклический характер. Ответ будет содержать хотя бы одну строчку с цифрой – если введенное число окажется однозначным.

Разработку алгоритма начнем с поиска и формулирования основного действия, которое должно находиться внутри цикла.

По смыслу задачи необходимой должна быть печать текущей цифры числа. Но чтобы цифру напечатать, ее необходимо из исходного числа извлечь. Зафиксируем первый результат в виде рис. 1.

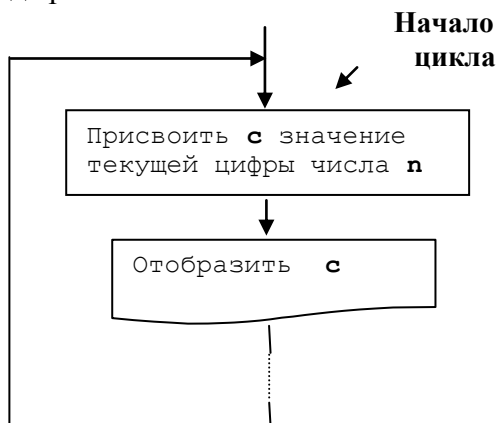


Рис. 1. Повторение основных действий для задачи печати цифр числа

Что же дальше? Во-первых, нужно придумать условие завершения цикла. Это вроде просто: будем продолжать повторения, пока остаются еще ненапечатанные цифры исходного числа.

Есть проблема посложнее. Дело в том, что после завершения действий на очередной итерации компьютер снова вернется в начало цикла, и будет выполнять *те же команды*, что и на предыдущей итерации, только результат этих команд должен быть другой. К примеру, если на первой итерации текущей цифрой является 5, то на следующей итерации – 2. Как же компьютеру объяснить необходимость перехода к следующей цифре? Трудновато ...

Попробуем по-другому. Видно, что после печати текущей цифры мы к ней больше не возвращаемся, то есть она нам больше не нужна. Тогда можно в качестве основного действия взять следующее: «Присвоить переменной *c* значение младшей цифры числа *n*». Что же сделать, чтобы на следующей итерации эта команда работала правильно? Надо в числе *n* стереть уже не нужную нам младшую цифру. Ура! Снова рисуем алгоритм – см. рис. 2.

Команда «Присвоить *n* значение *n* без младшей цифры» является вспомогательной, она обеспечивает правильную работу основной команды «Присвоить переменной *c* значение младшей цифры числа *n*» при переходе к следующей итерации цикла. Это один из самых важных моментов данного параграфа; он является наглядным примером общего правила разработки алгоритмов. В теле цикла должны быть:

1. Основные команды, создающие результаты в соответствии с постановкой задачи.
2. Вспомогательные команды, обеспечивающие правильность работы основных команд при переходе к следующей итерации цикла.
3. Команды, изменяющие переменные, входящие в условие завершения цикла. Эти переменные должны изменяться также в соответствии с постановкой задачи, но таким образом, чтобы цикл мог завершиться, а не остался бесконечным.

Для завершения задачи нам нужно разобраться с условием выхода из цикла и командами третьей группы в теле цикла. Здесь часто не обойтись без формализации, т.е. без замены формулировок, записанных на естественном языке, на строгие математические обозначения. Так, условие «Есть в *n* ненапечатанные цифры?» на языке математики можно выразить в виде отношения $n > 0$. Действительно, пока в положительном числе есть значащие цифры, его значение больше нуля.

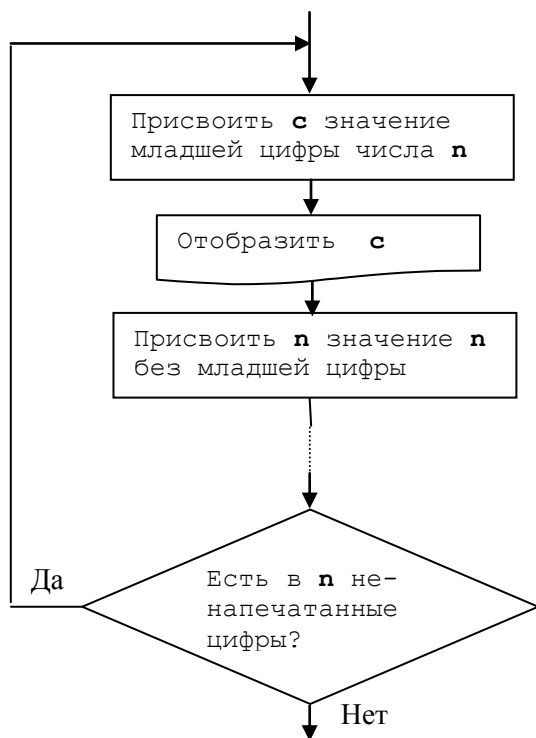


Рис. 2. Алгоритм для задачи печати цифр числа – уточнение 1

А как обеспечить выход из цикла? В данном примере дополнительной команды не требуется, потому что стирание цифр переменной n в конце концов приведет к тому, что ее значение станет равно 0. То есть произошло совпадение команд типа 2 и 3 внутри цикла.

Формализованная версия алгоритма для печати цифр числа приведена на рис. 3. Команды присваивания реализованы на основе команд работы с целыми числами: деления и взятия остатка.

Поскольку переменная c вычисляется внутри цикла, ее инициализация не требуется. Инициализация переменной n достигается за счет того, что ее значение вводится с клавиатуры.

Полученная циклическая структура называется циклом с постусловием, потому что оператор проверки завершения цикла находится после операторов тела цикла. Этот цикл обладает следующими особенностями.

1. Операторы тела цикла будут выполнены хотя бы один раз.
2. Данная конструкция является конструкцией структурного программирования: имеет один вход

и один выход.

3. Условие выхода из цикла нужно проектировать так, чтобы повторения происходили при истинном значении условия, а выход из цикла происходил при ложном значении условия.

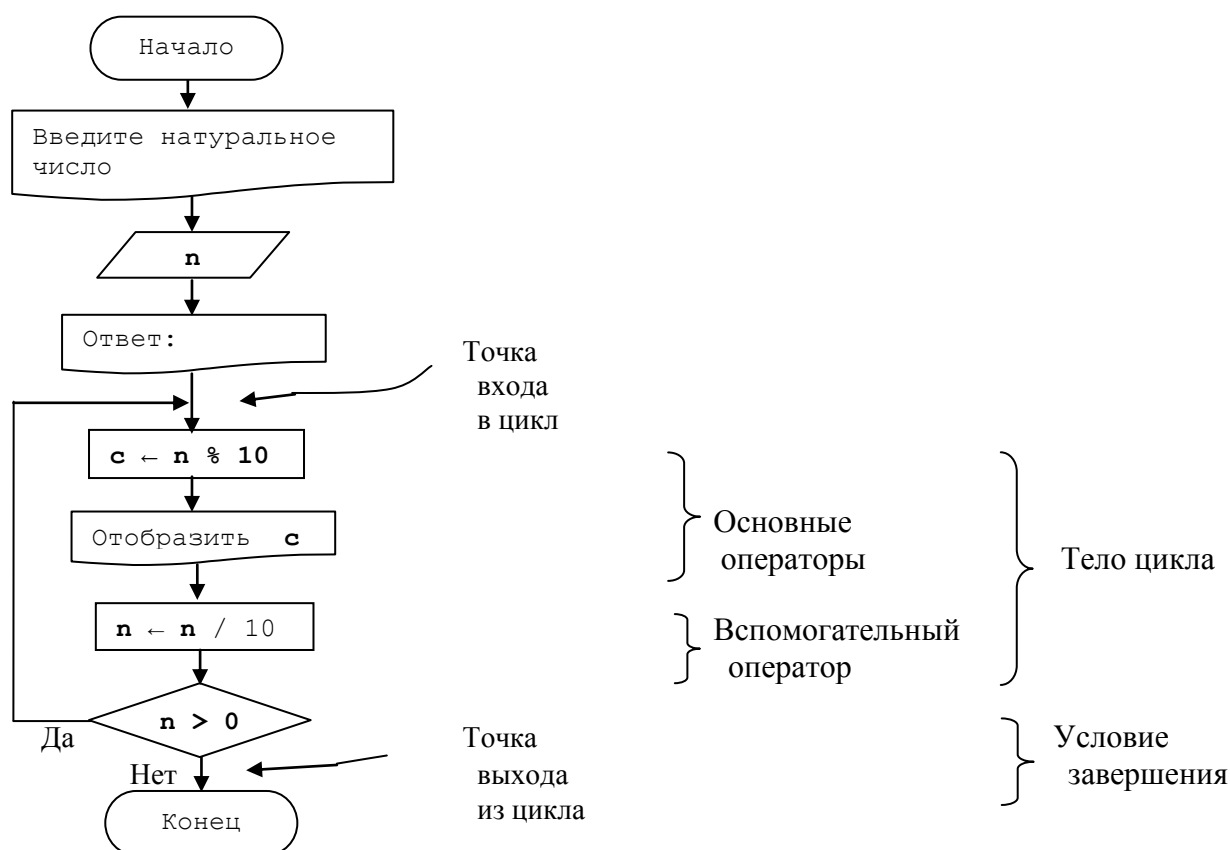


Рис. 3. Алгоритм для задачи печати цифр – формализованная версия

Результатом кодирования алгоритма является программа, показанная на рис. 4.

```
#include <iostream>
using namespace std;
int main()
{
    int n, c;
    cout << "Введите натуральное число\n";
    cin >> n;
    cout << "Ответ:\n";
    do
    {
        c = n % 10;
        cout << c << endl;
        n = n / 10;
    } while (n > 0);

    return 0;
}
```

Рис. 4. Программа печати цифр числа

Для реализации цикла с постусловием в языке C++ существует оператор do-while, имеющий следующий синтаксис

```
do { <операторы> } while (<условие>)
```

Операторы, включаемые в тело, - это все уже известные нам типы операторов. Также придем к вложенному циклу, если включим в тело цикла еще один оператор цикла. <Условие> записывается либо в виде отношения, либо в виде логического выражения. <Операторы> выполняются хотя бы один раз и затем повторяются до тех пор, пока значение <условия> остается истинным. Фигурные и круглые скобки в написании оператора являются обязательными.

Упражнения

1. Дано натуральное число. Напечатать его делители сначала в порядке возрастания, а затем в порядке убывания.
2. Даны два натуральных числа. Найти их наибольший делитель.