



РАНХиГС
РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА
И ГОСУДАРСТВЕННОЙ СЛУЖБЫ
ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ

**НИЖЕГОРОДСКИЙ
ИНСТИТУТ
УПРАВЛЕНИЯ**



Введение в специальность Алгоритмы и структуры данных

Цветкова Ирина Николаевна,
Зав. кафедрой информатики и ИТ
к. ф.-м.н., доцент
i.tsvetkova@niu.ranepa.ru

<http://niu.ranepa.ru/>



РАНХиГС
РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА
И ГОСУДАРСТВЕННОЙ СЛУЖБЫ
ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ

**НИЖЕГОРОДСКИЙ
ИНСТИТУТ
УПРАВЛЕНИЯ**

"Алгоритмы + структуры данных = программы".

Вирт, Н. Алгоритмы и структуры данных

<http://www.iprbookshop.ru/63821.html>

Никлаус Вирт - знаменитый швейцарский
специалист по программированию, автор языка
Паскаль.



РАНХиГС
РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА
И ГОСУДАРСТВЕННОЙ СЛУЖБЫ
ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ

**НИЖЕГОРОДСКИЙ
ИНСТИТУТ
УПРАВЛЕНИЯ**

Основы алгоритмики.

Понятие алгоритма - одно из основных понятий программирования и математики.

Мухаммед ибн Муса аль-Хорезми (Alhorithmi), 783-850 гг.,
«Книга о сложении и вычитании»

Алгоритм - это последовательность команд, предназначенная исполнителю, в результате выполнения которой он должен решить поставленную задачу.

Программа - конкретная формулировка абстрактных алгоритмов, основанная на конкретных представлениях и структурах данных.



РАНХиГС
РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА
И ГОСУДАРСТВЕННОЙ СЛУЖБЫ
ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ

**НИЖЕГОРОДСКИЙ
ИНСТИТУТ
УПРАВЛЕНИЯ**

«Алгоритм — это конечный набор правил, который определяет последовательность операций для решения конкретного множества задач и обладает пятью важными чертами: конечность, определённости, ввод, вывод, эффективность».

(Кнут, Д.Э. Искусство программирования. Том 1 : Основные алгоритмы / Д. Э. Кнут; под общей редакцией Ю.В. Козаченко; перевод с английского С.Г. Тригуб, Ю.Г. Гордиенко, И.В. Красикова. - Москва : Вильямс, 2004. - 720 с. - ISBN [5-8459-0080-8](#))



РАНХиГС
РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА
И ГОСУДАРСТВЕННОЙ СЛУЖБЫ
ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ

**НИЖЕГОРОДСКИЙ
ИНСТИТУТ
УПРАВЛЕНИЯ**

- **Конечность.** Алгоритм должен всегда заканчиваться после выполнения конечного числа шагов.
- **Определенность.** Действия, которые необходимо произвести на каждом шаге, должны быть определены строго и недвусмысленно в каждом возможном случае.
- **Вход (input).** Алгоритм всегда имеет некоторое (иногда равное нулю) количество входных данных, то есть величин, передаваемых ему до начала работы.
- **Выход (output).** Алгоритм всегда обязан иметь одну или несколько выходных величин. Алгоритмы, не имеющие выходных данных, бесполезны на практике.
- **Эффективность.** От алгоритма требуется, чтобы он был эффективным. Это означает, что все операции, которые необходимо произвести в алгоритме, должны быть достаточно простыми, чтобы их в принципе можно было выполнить точно и за конечное время.



РАНХиГС
РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА
И ГОСУДАРСТВЕННОЙ СЛУЖБЫ
ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ

**НИЖЕГОРОДСКИЙ
ИНСТИТУТ
УПРАВЛЕНИЯ**

Формы представления алгоритмов:

словесная - запись на естественном языке;

псевдокоды - полужформализованные описания алгоритмов на условном алгоритмическом языке, включающие в себя как элементы языка программирования, так и фразы естественного языка, общепринятые математические обозначения и др.;

графическая – блок-схемы (текст и графика);

программная - тексты на языках программирования.



РАНХиГС
РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА
И ГОСУДАРСТВЕННОЙ СЛУЖБЫ
ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ

**НИЖЕГОРОДСКИЙ
ИНСТИТУТ
УПРАВЛЕНИЯ**

Словесный способ. Алгоритм может быть следующим:

1. задать любое целое число;
2. задать счетчик, равный 1;
3. задать число для хранения произведения, равное 1;
4. заменить произведение, умножив произведение на счетчик;
5. если значения счетчика и целого числа равны, то взять произведение в качестве ответа и остановиться, в противном случае продолжить выполнение алгоритма;
6. увеличить счетчик на 1;
7. повторить алгоритм с шага 4.



РАНХиГС

РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА
И ГОСУДАРСТВЕННОЙ СЛУЖБЫ
ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ

**НИЖЕГОРОДСКИЙ
ИНСТИТУТ
УПРАВЛЕНИЯ**

Псевдокод. Общий вид алгоритма:

алг название алгоритма (аргументы и результаты)

дано условия применимости алгоритма

надо цель выполнения алгоритма

нач описание промежуточных величин

| последовательность команд (тело алгоритма)

кон

алг Факториал (арг цел N, рез цел F)

дано | N

надо | $F = 1 * 2 * 3 * \dots * N$

нач цел i

ввод N; F:=1

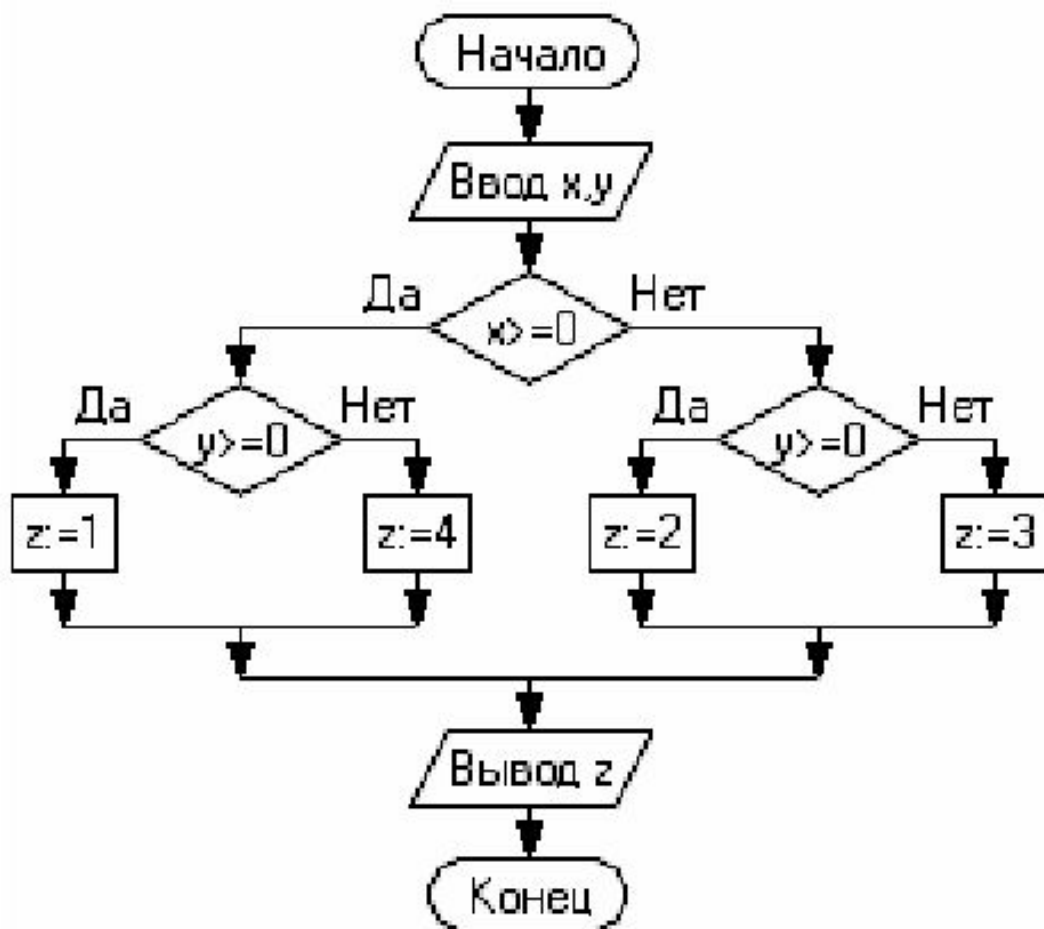
нц для i от 1 до N

F:=F*i

кц

вывод "F = ", F

кон



**Алгоритм
присвоения
переменной
демонстрирует
блок-схема
программы
(графическая форма)**



РАНХиГС

РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА
И ГОСУДАРСТВЕННОЙ СЛУЖБЫ
ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ

**НИЖЕГОРОДСКИЙ
ИНСТИТУТ
УПРАВЛЕНИЯ**

Пример программы вычисления факториала числа N на языке C#:

```
using System;
namespace Factorial
{
    class Program
    {
        static void Main(string[] args)
        {
            int N, F = 1, i;
            Console.WriteLine("Расчет факториала. Введите число N = ");
            N = Convert.ToInt32(Console.ReadLine());
            for (i = 2; i <= N; i++)
                F = F * i;
            Console.WriteLine("F = " F);
            Console.ReadKey();
        }
    }
}
```



РАНХиГС
РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА
И ГОСУДАРСТВЕННОЙ СЛУЖБЫ
ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ

**НИЖЕГОРОДСКИЙ
ИНСТИТУТ
УПРАВЛЕНИЯ**

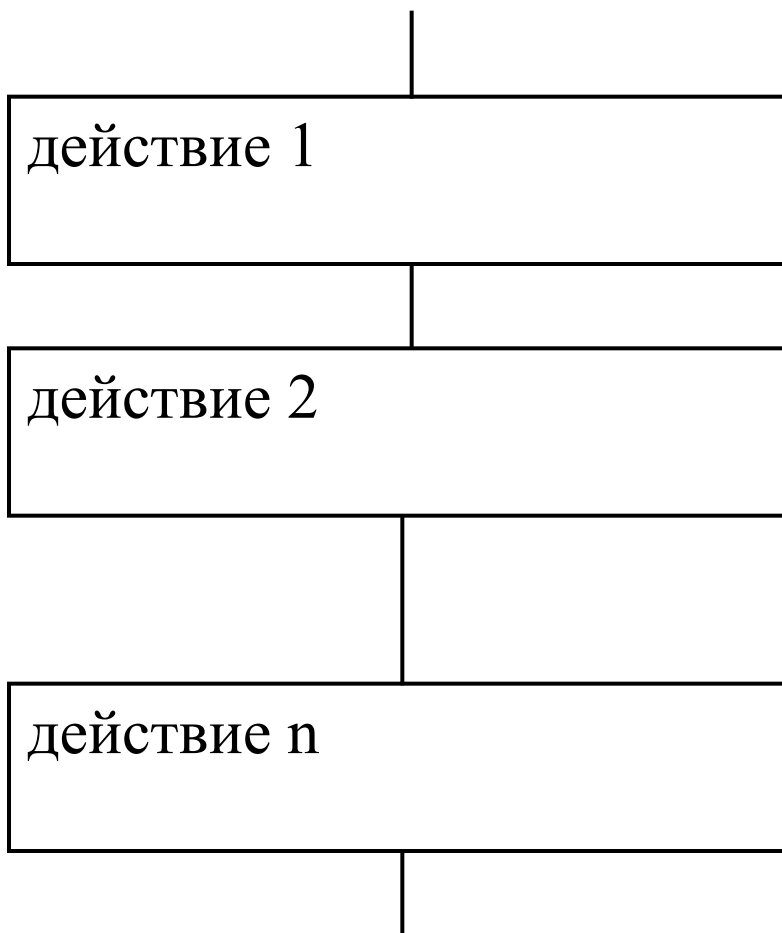
Блок-схема – совокупность символов, соответствующих этапам работы алгоритма и соединяющих их линий.

- **ГОСТ 19.701-90 (переиздан в 2010г).** Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения

Название символа	Обозначение и пример заполнения	Пояснение
Процесс		Вычислительное действие или последовательность действий
Решение		Проверка условий
Модификация		Начало цикла
Предопределенный процесс		Вычисления по подпрограмме (вызов внешней процедуры)
Ввод-вывод		Ввод-вывод в общем виде
Пуск-остановка		Начало, конец алгоритма, вход и выход в подпрограмму
Документ		Вывод результатов на печать



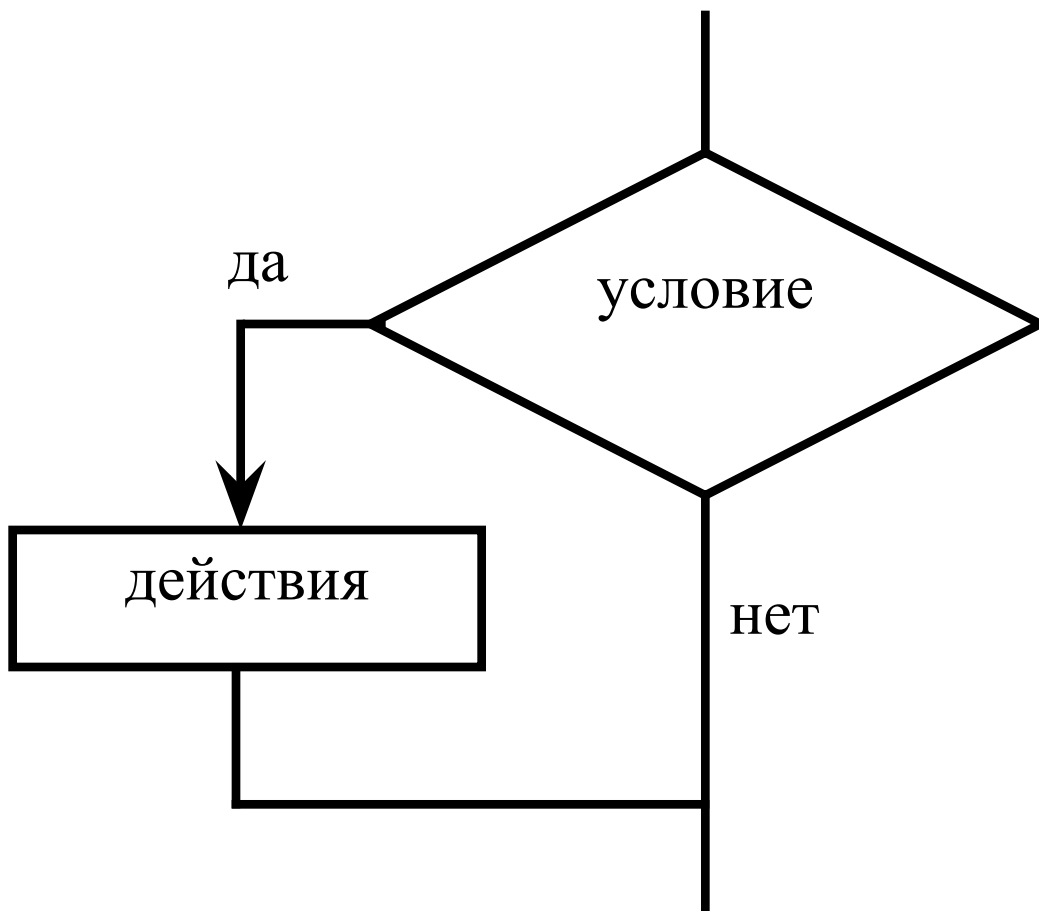
1. Базовая структура "следование".





2. Базовая структура "ветвление".

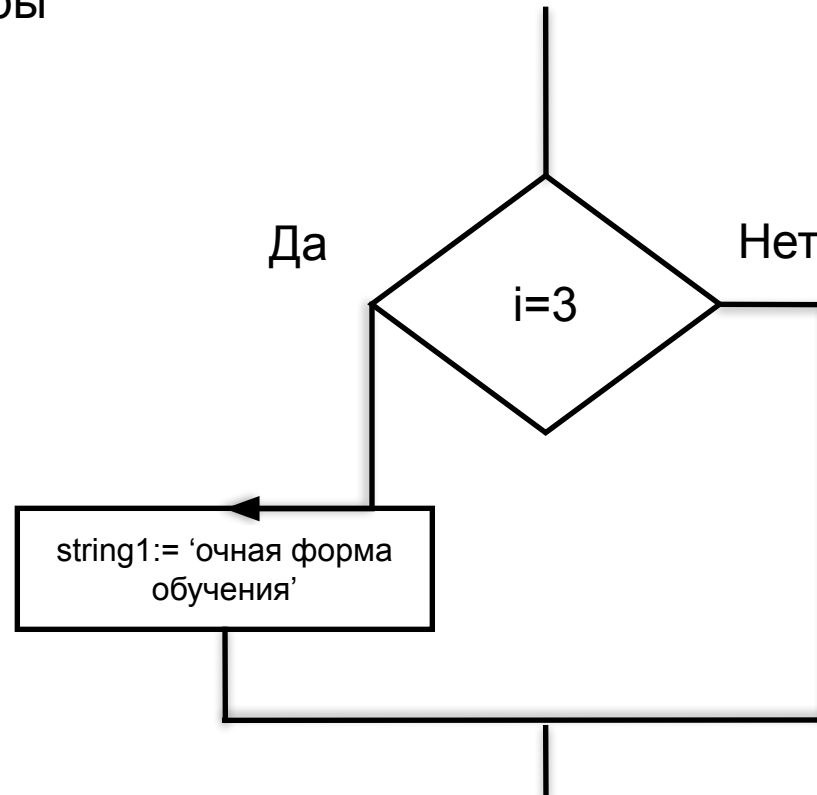
1) если-то





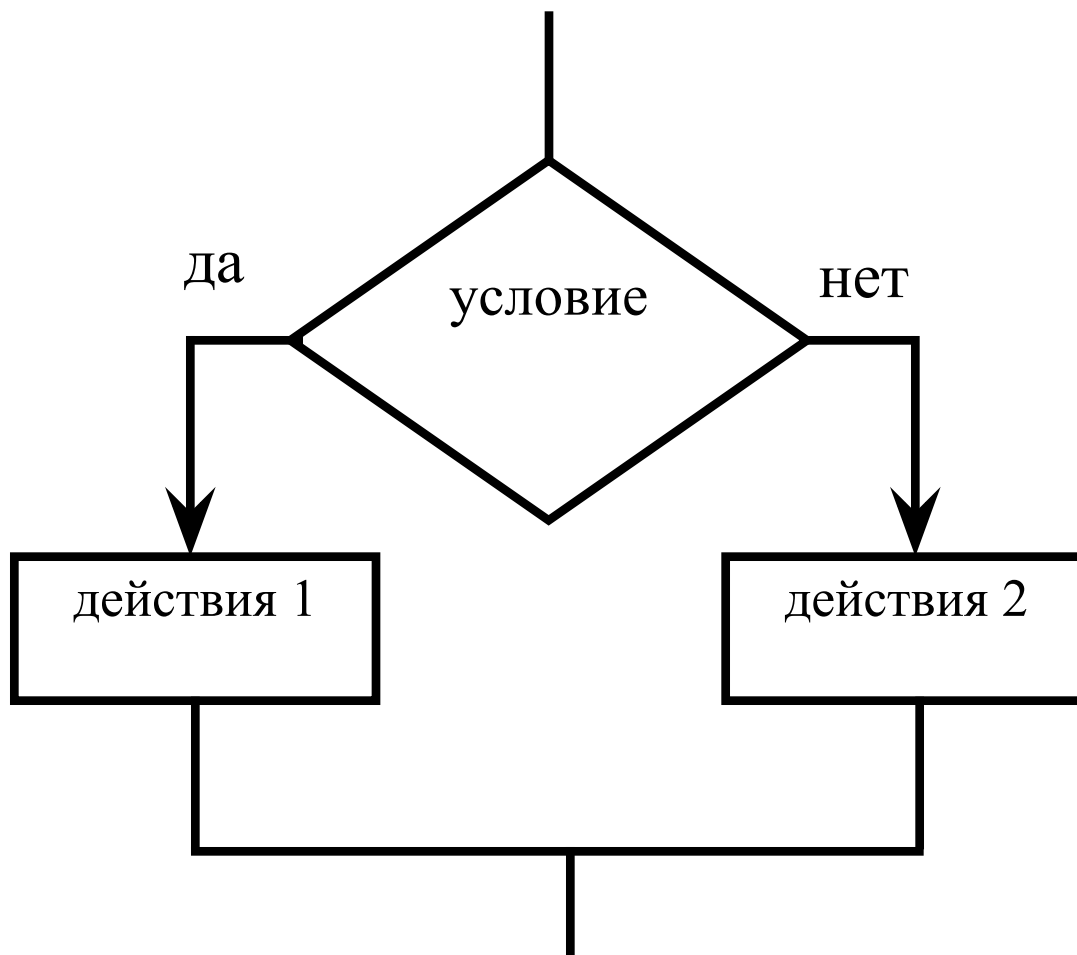
Пример. Формирование 1 цифры
в нумерации групп

если $i=3$
то $string1 :=$ 'очная форма
обучения'
все





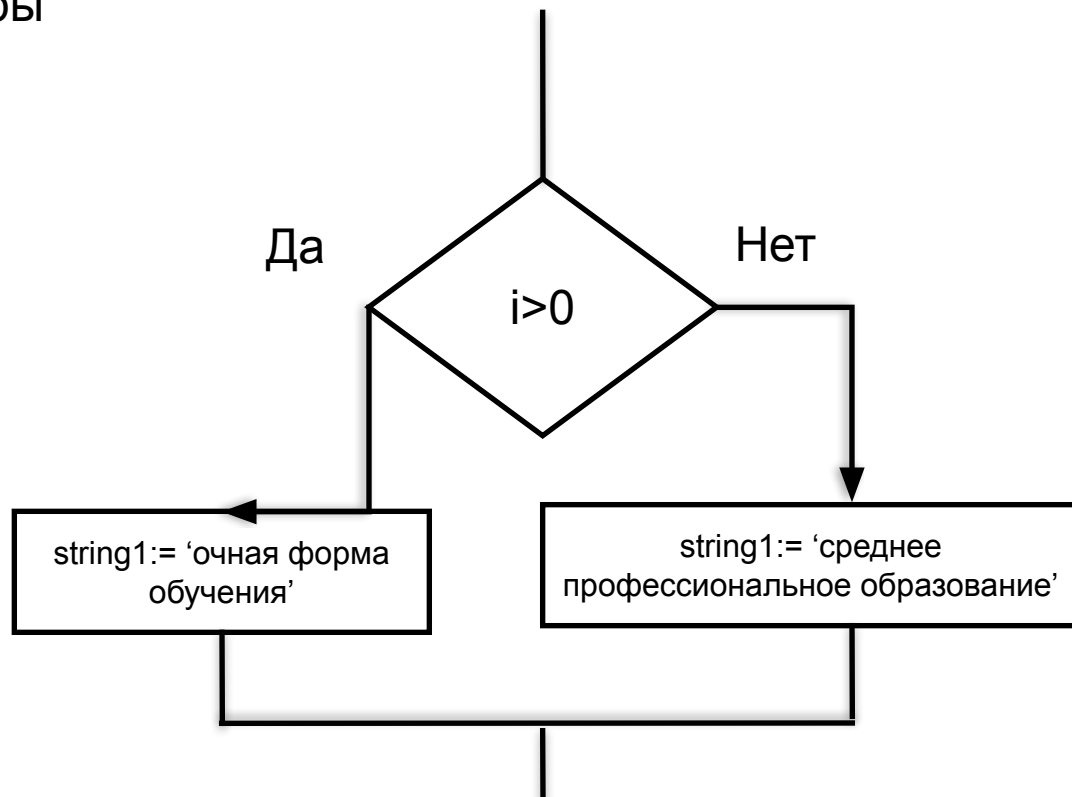
2. Базовая структура "ветвление". 2) если-то-иначе





Пример. Формирование 1 цифры
в нумерации групп

если $i > 0$
то $string2 :=$ 'высшее образование'
иначе $string2 :=$ 'среднее
профессиональное образование'
все



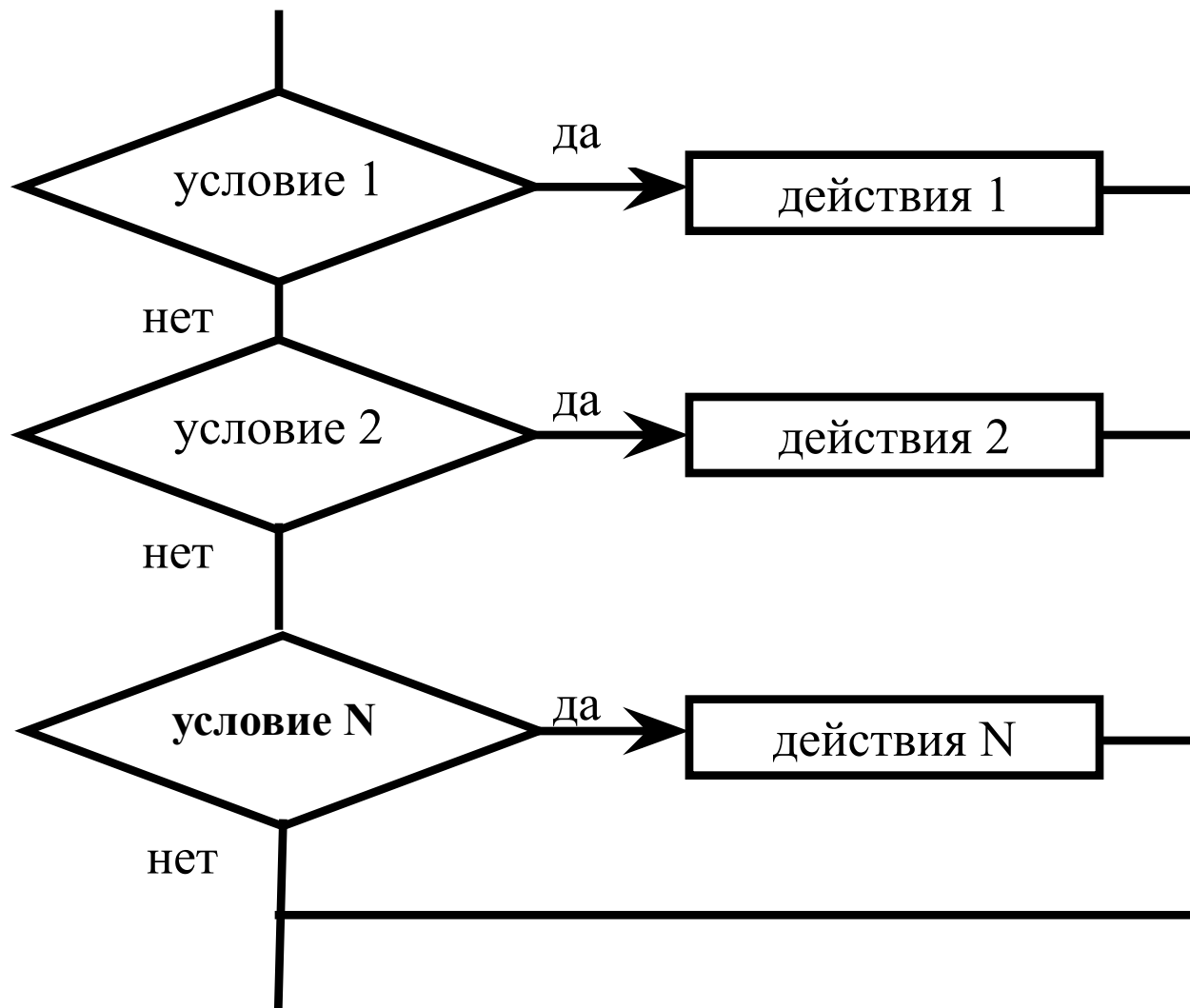


РАНХиГС

РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА
И ГОСУДАРСТВЕННОЙ СЛУЖБЫ
ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ

**НИЖЕГОРОДСКИЙ
ИНСТИТУТ
УПРАВЛЕНИЯ**

2. Базовая структура "ветвление". 3) выбор





РАНХиГС

РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА
И ГОСУДАРСТВЕННОЙ СЛУЖБЫ
ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ

НИЖЕГОРОДСКИЙ
ИНСТИТУТ
УПРАВЛЕНИЯ

Пример. Формирование 2 цифры
в нумерации групп

выбор

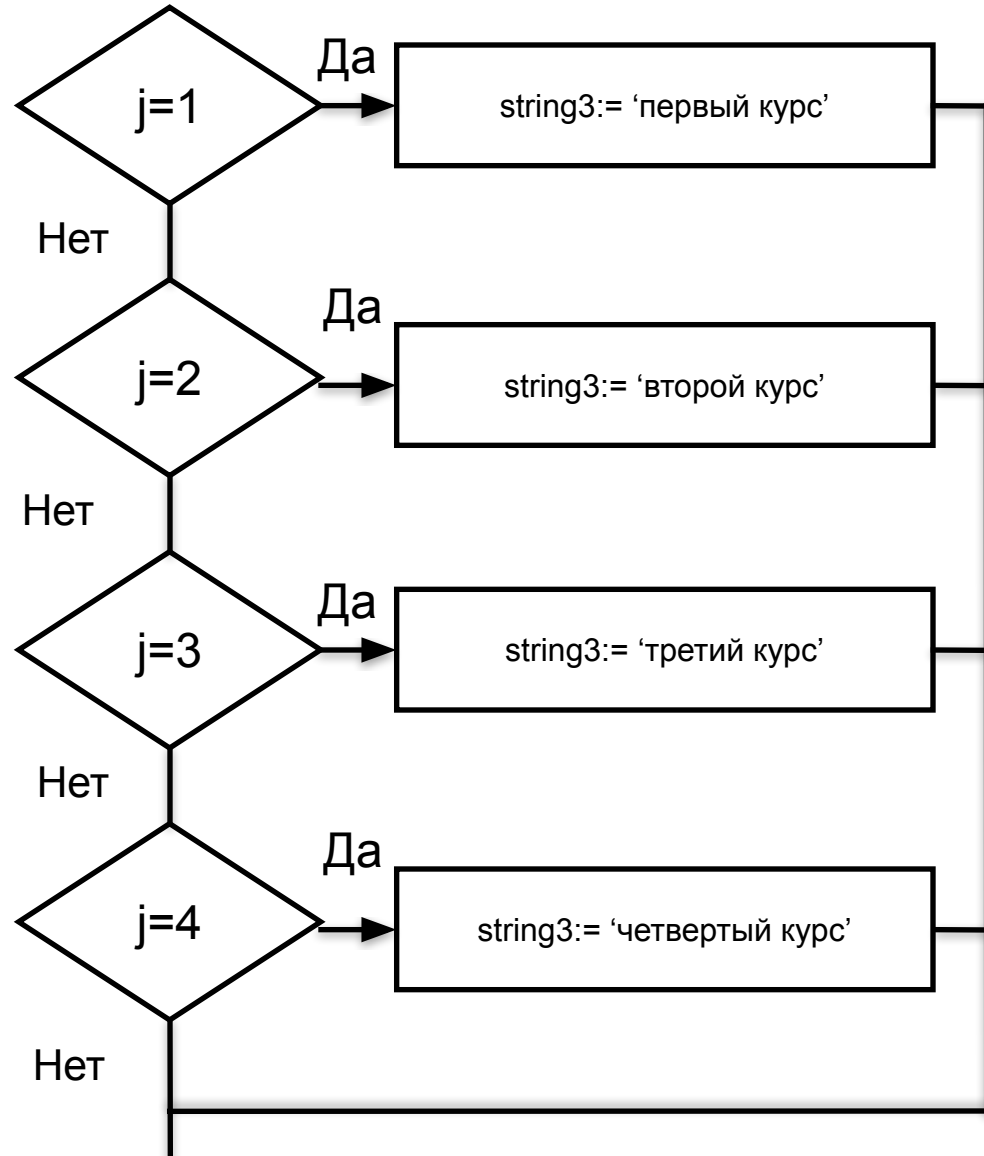
при $j=1$ $string3:=$ 'первый курс'

при $j=2$ $string3:=$ 'второй курс'

при $j=3$ $string3:=$ 'третий курс'

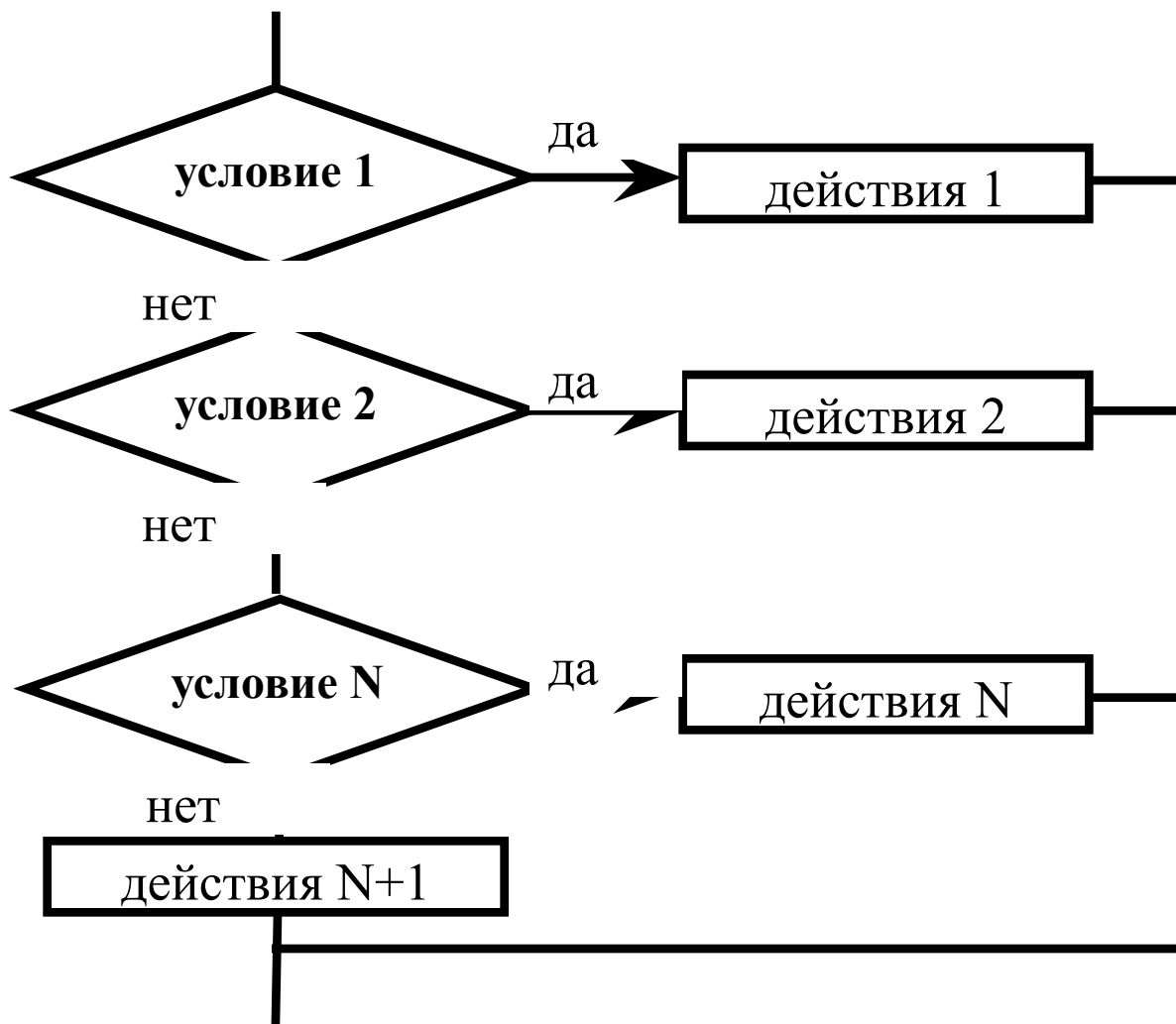
при $j=4$ $string3:=$ 'четвертый курс'

все





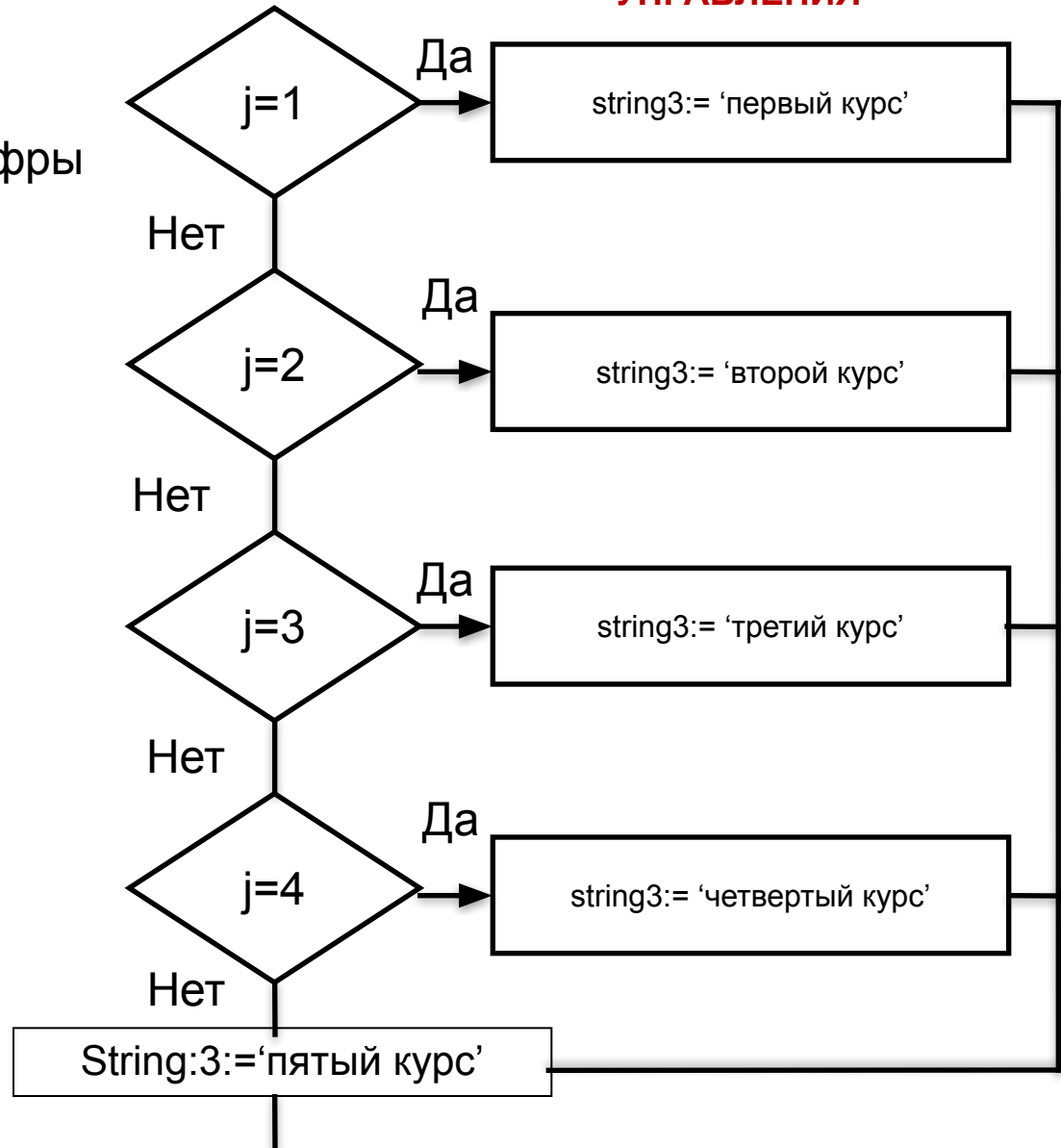
2. Базовая структура "ветвление". 3) выбор-иначе





Пример. Формирование 2 цифры
в нумерации групп

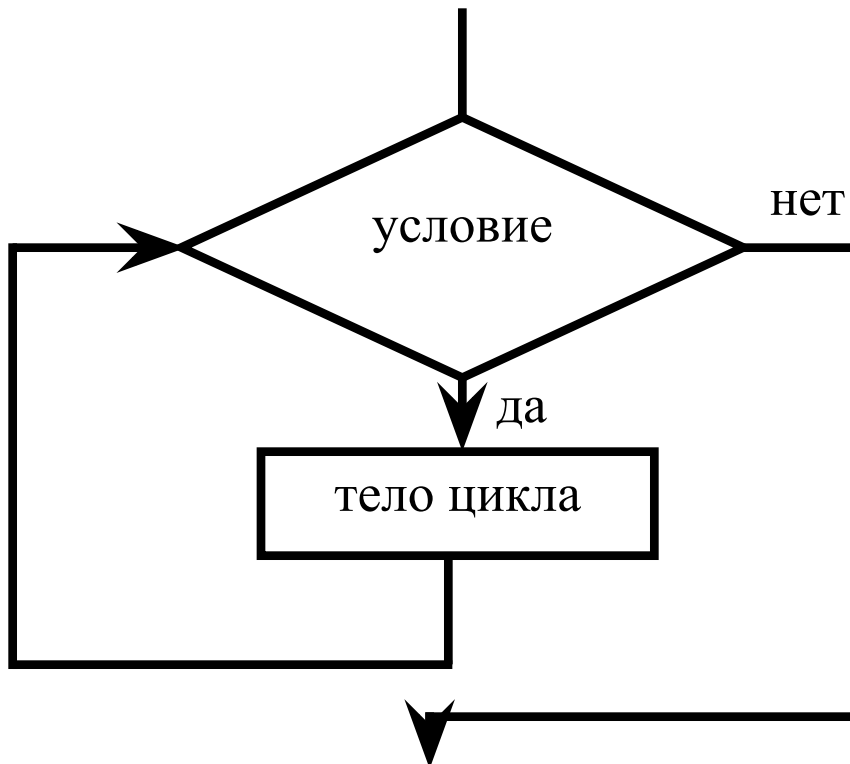
выбор
при $j=1$ $string3:=$ 'первый курс'
при $j=2$ $string3:=$ 'второй курс'
при $j=3$ $string3:=$ 'третий курс'
при $j=4$ $string3:=$ 'четвертый курс'
иначе $string3:=$ 'пятый курс'
все

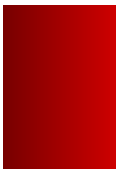




3. Базовая структура "цикл". Обеспечивает многократное выполнение некоторой совокупности действий, которая называется телом цикла

1) **Цикл типа пока.** Предписывает выполнять тело цикла до тех пор, пока выполняется условие, записанное после слова «пока».



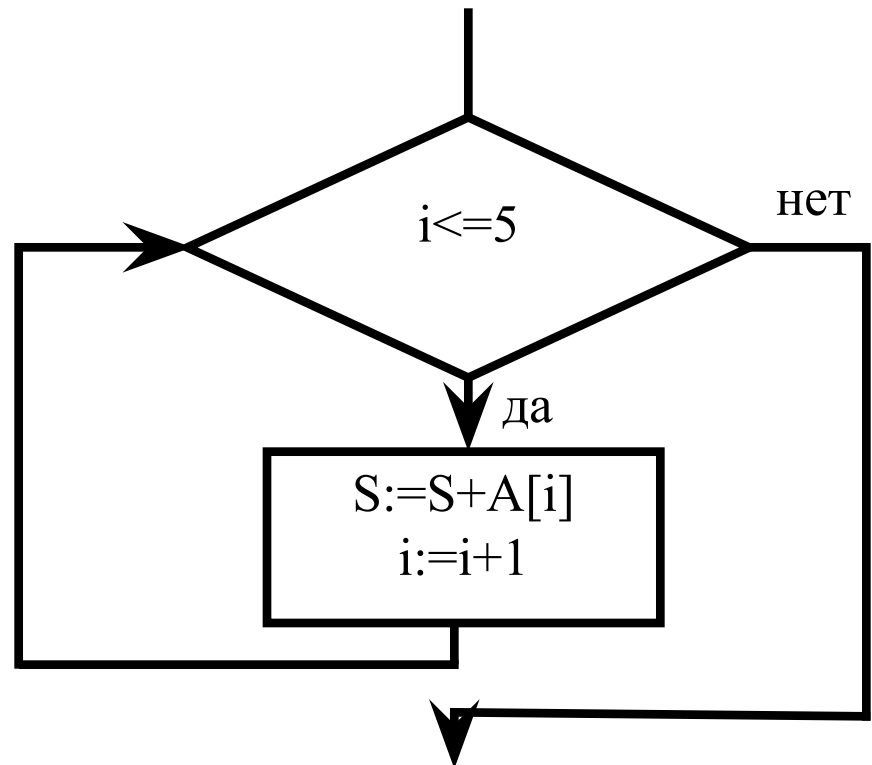


нц пока $i \leq 5$

$S := S + A[i]$

$i := i + 1$

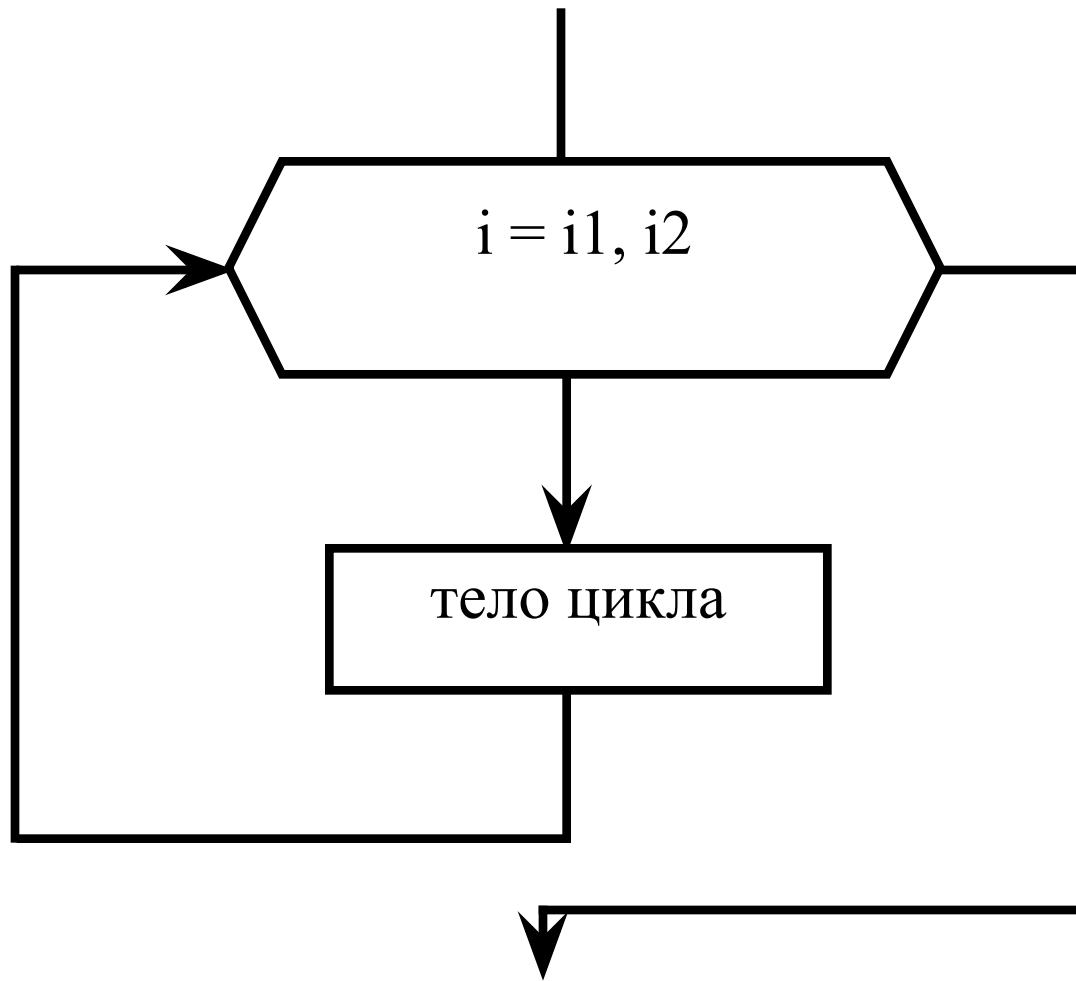
кц





3. Базовая структура "цикл".

2) **Цикл типа для.** Предписывает выполнять тело цикла для всех значений некоторой переменной (параметра цикла) в заданном диапазоне.



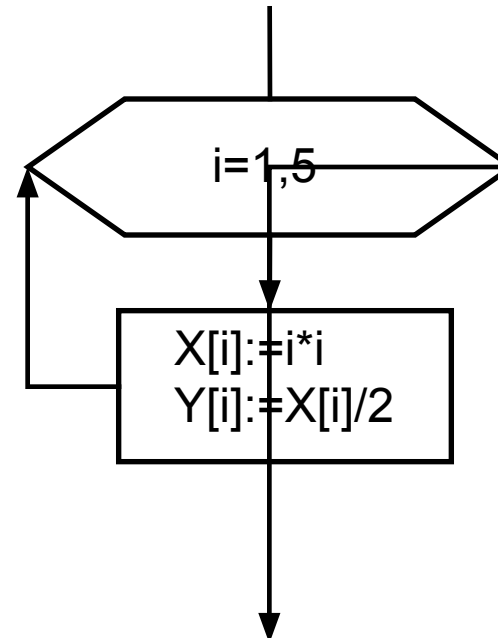


НЦ для i от 1 до 5

$X[i] := i * i$

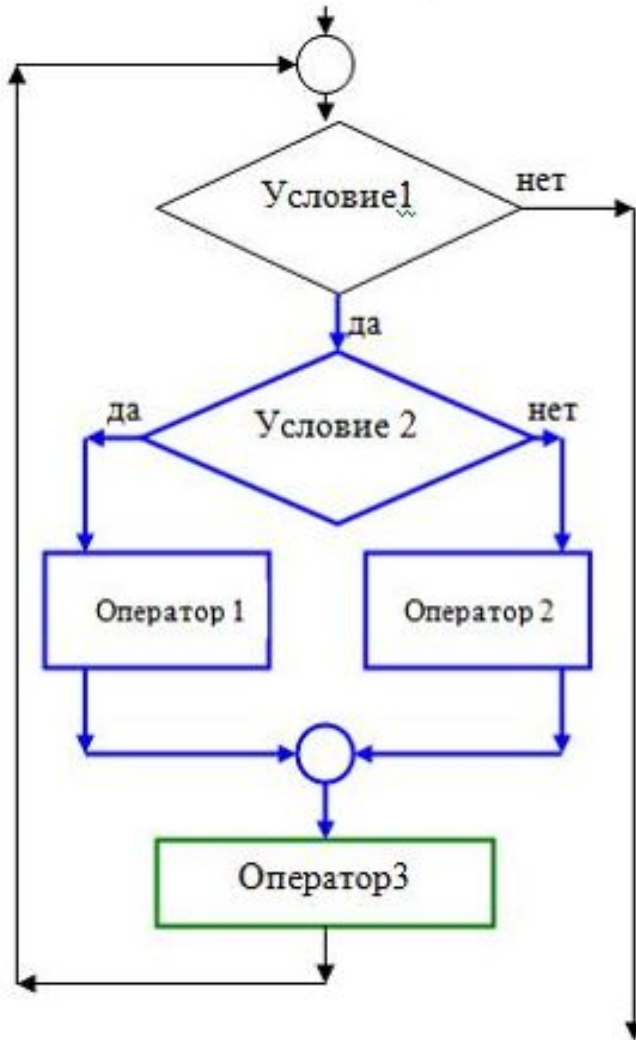
$Y[i] := X[i] / 2$

КЦ



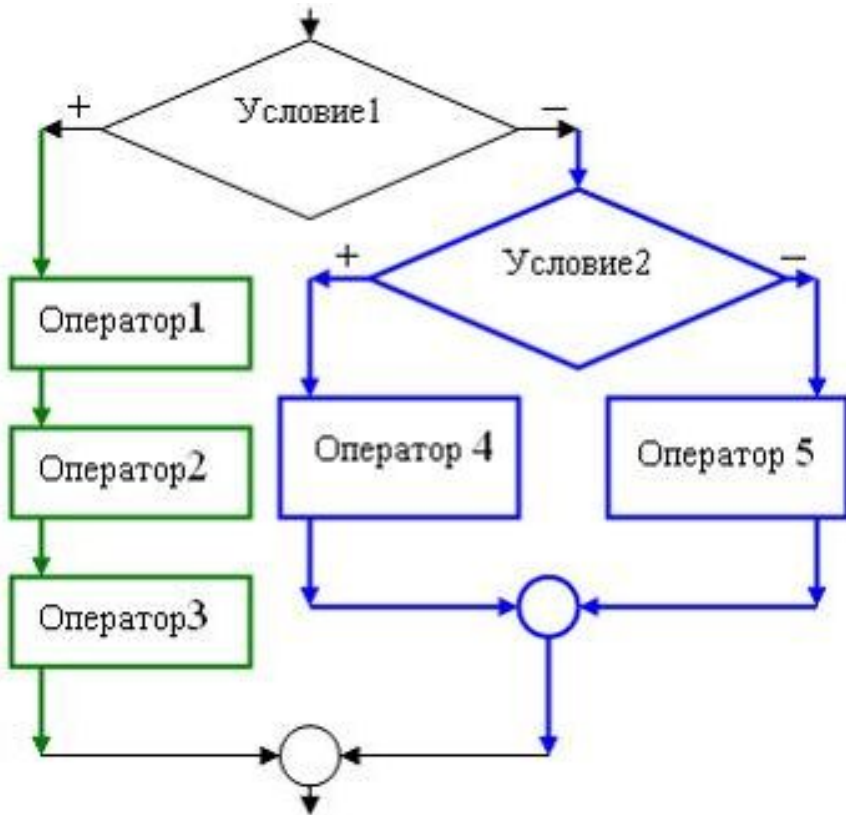


Условие1 в приведенном ниже алгоритме задает ...



1. полное ветвление;
2. цикл с предусловием;
3. цикл с постусловием;
4. цикл с заданным числом повторений.

Приведенной блок-схеме соответствует фрагмент программы ...



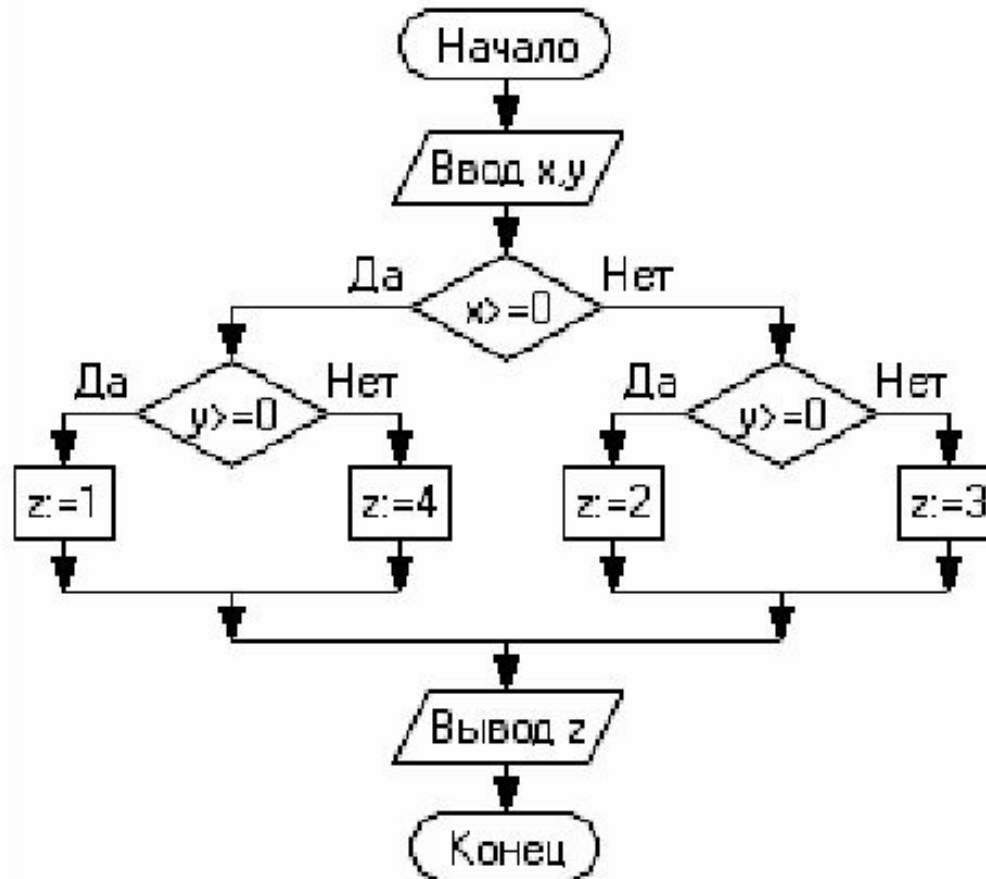
а. если условие 1 то
 если условие 2 то оператор 4
 иначе
 начало
 оператор 1
 оператор 2
 оператор 3
 конец
 иначе оператор 5

б. если условие 1 то
 оператор 1
 оператор 2
 оператор 3
 если условие 2 то оператор 4
 иначе оператор 5

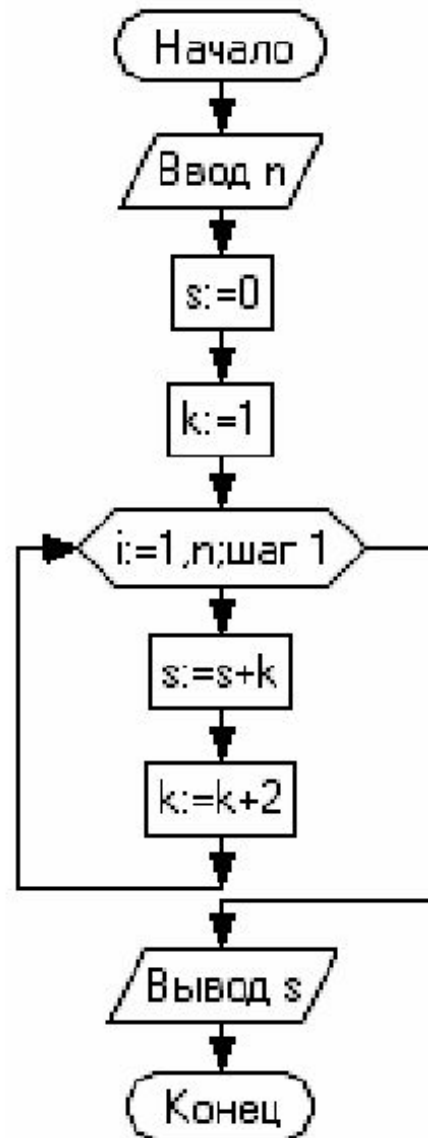
с. если условие 1 то
 начало
 оператор 1
 оператор 2
 оператор 3
 конец
 иначе
 если условие 2 то оператор 4
 иначе оператор 5



При выполнении приведенного ниже алгоритма с исходными данными $x = 14$, $y = -5$ значение переменной z будет равно ...



При выполнении приведенного ниже алгоритма с исходными данными $n = 6$ значение переменной s будет равно ...



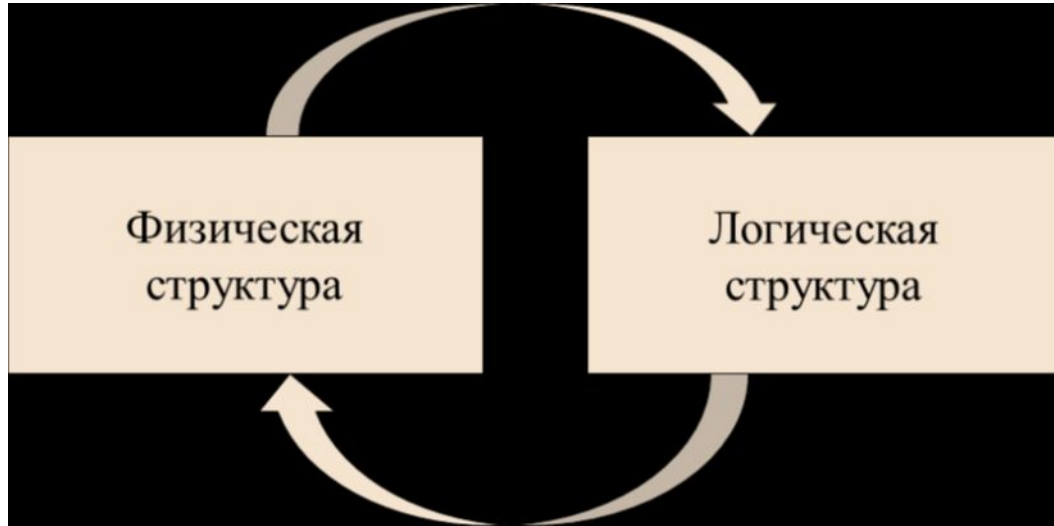


Понятие структуры данных

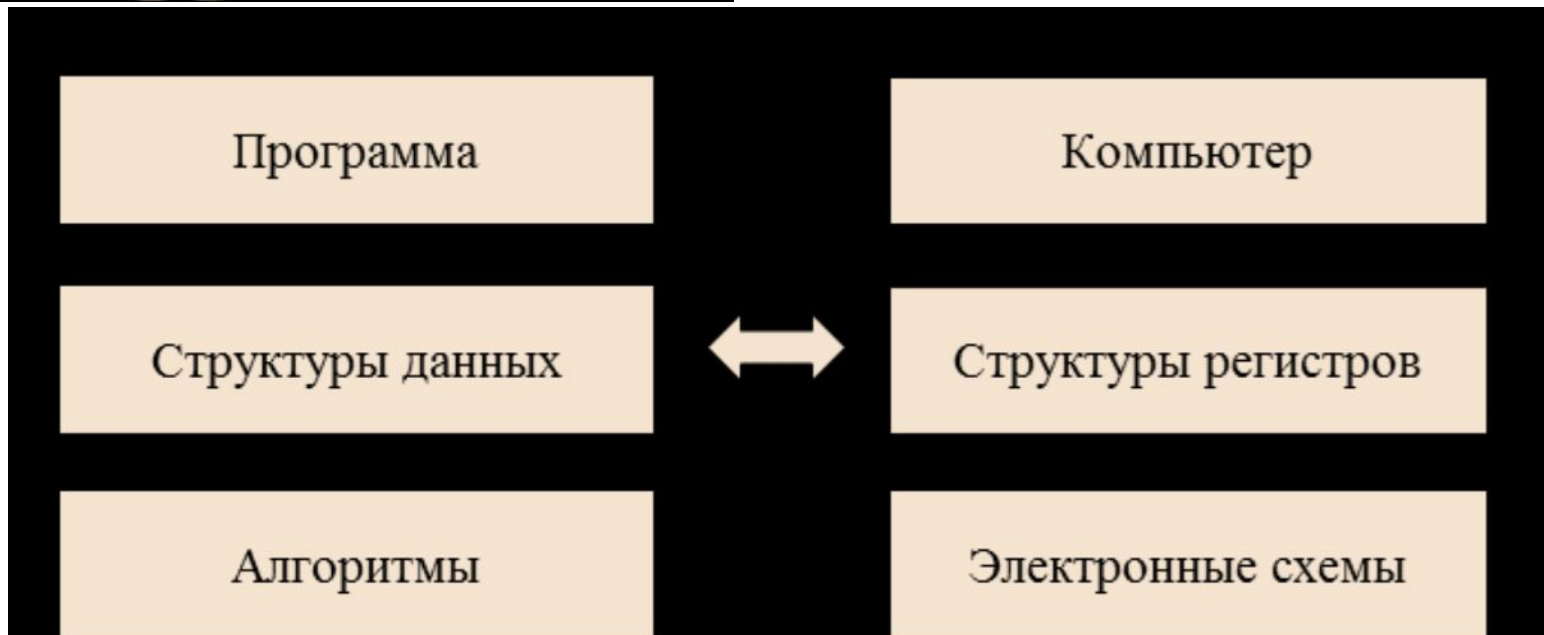
Структура данных — множество элементов данных и множество связей между ними.

Структура данных — программная единица, позволяющая хранить и обрабатывать множество однотипных и/или логически связанных данных.

Для добавления, поиска, изменения и удаления данных структура данных предоставляет некоторый набор функций, составляющих её интерфейс.



Способ представления структур данных



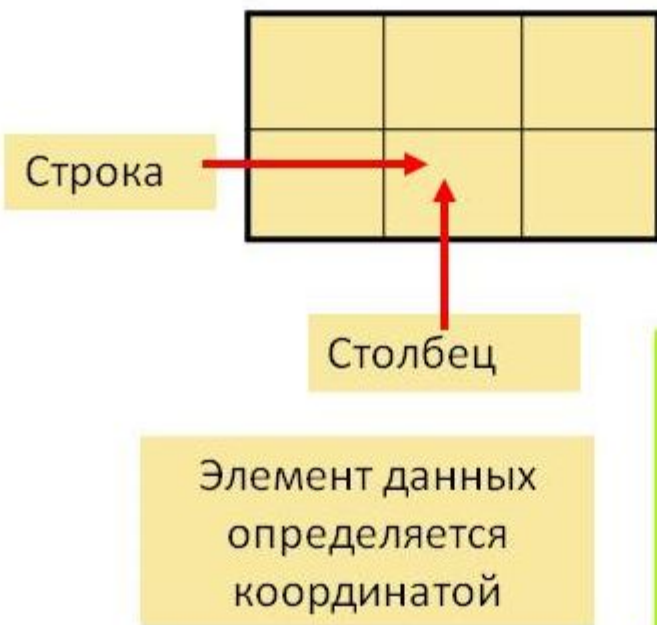


Классификация структур данных

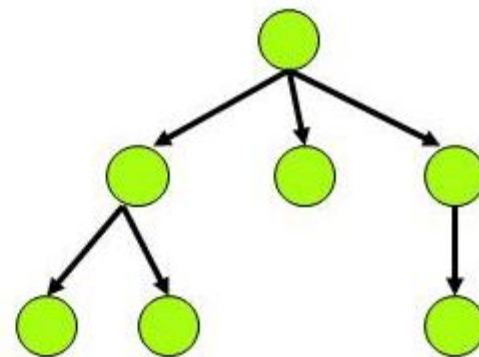
Линейная структура



Табличная структура



Иерархическая структура



Данные делят на уровни, каждый предшествующий элемент иерархии порождает не одну, а несколько линейных зависимостей.



ОПЕРАЦИИ НАД СТРУКТУРАМИ ДАННЫХ

Создание – выделение памяти для структуры данных.

Уничтожение – противоположна по своему действию операции создания.

Выбор – обеспечивает доступ к данным внутри самой структуры.

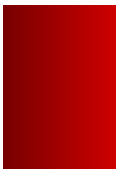
Обновление – позволяет изменять значения данных в структуре и добавлять (*удалять*) выбранные данные.



Как бы сложна ни была задача, блок-схема соответствующей программы (алгоритма) всегда может быть представлена с использованием ограниченного числа элементарных управляющих структур (последовательность, ветвление, цикл).

Идея доказательства этого утверждения:

- преобразование каждой части алгоритма в одну из трех основных структур или их комбинацию;
- после достаточного числа таких преобразований оставшаяся неструктурированная часть либо исчезнет, либо станет ненужной;
- в результате получится алгоритм, эквивалентный исходному и использующий лишь 3 управляющие структуры.



РАНХиГС
РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА
И ГОСУДАРСТВЕННОЙ СЛУЖБЫ
ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ

**НИЖЕГОРОДСКИЙ
ИНСТИТУТ
УПРАВЛЕНИЯ**

Спасибо за внимание!

**<http://niu.ranepa.ru>
i.tsvetkova@niu.ranepa.ru**