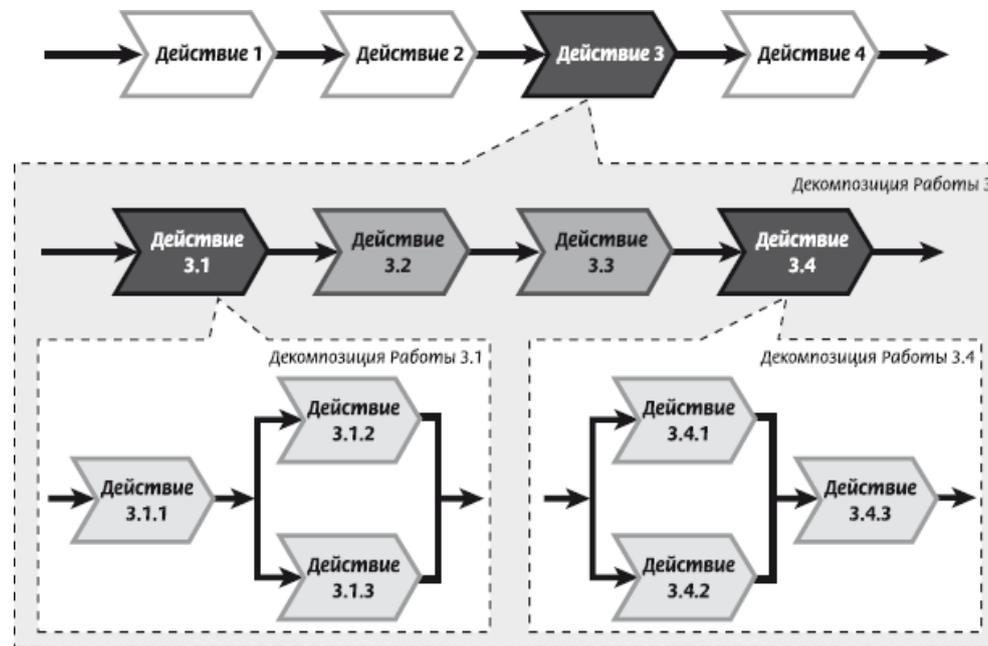


Основы алгоритмизации

Часть 2



Содержание лекции

1. Основные принципы алгоритмизации.
2. Линейные алгоритмы.
3. Разветвляющиеся алгоритмы.
4. Циклические алгоритмы.
5. Виды циклов.
6. Задачи «Использование циклов».

Основные принципы алгоритмизации

1. **Выявить:** исходные данные
результаты
назначить имена переменных.
2. **Выбрать:** метод (порядок) решения задачи.
3. **Разбить:** метод решения задачи на этапы.

Основные принципы алгоритмизации

4. **Изобразить**: каждый этап в виде блока алгоритма
указать стрелками порядок их
выполнения.

5. В полученной схеме при любом варианте
вычислений:

предусмотреть: выдачу результатов или
сообщений об их отсутствии.

Линейный алгоритм

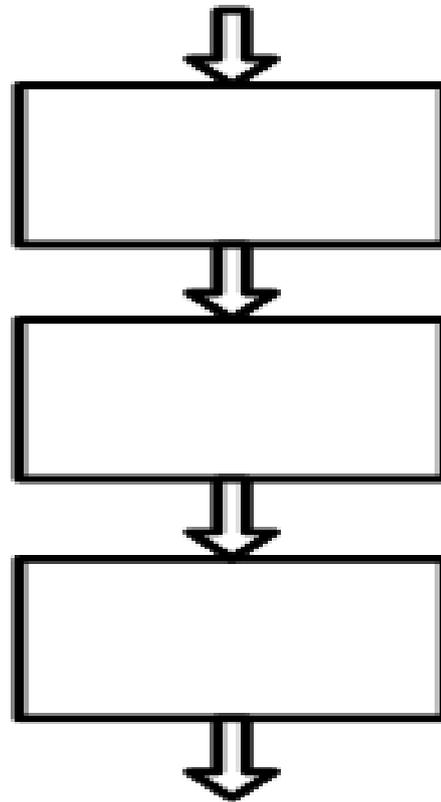
алгоритм

строгая последовательность
выполнения

Линейные алгоритмы

Линейный алгоритм – это алгоритм, в котором все этапы решения задачи выполняются **строго последовательно**.

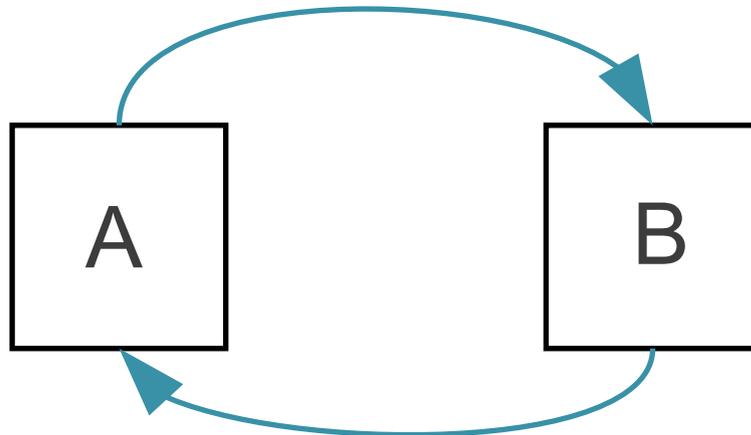
Структура алгоритма:



Линейные алгоритмы

Задача:

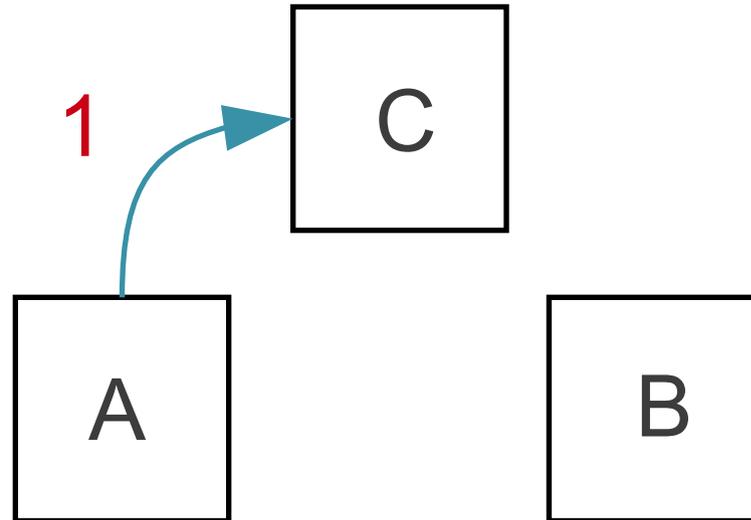
Имеются две клетки – в одной находится волк, в другой – заяц. Требуется поменять их местами, т. е. пересадить их из одной клетки в другую.



Линейные алгоритмы

Решение:

шаг 1

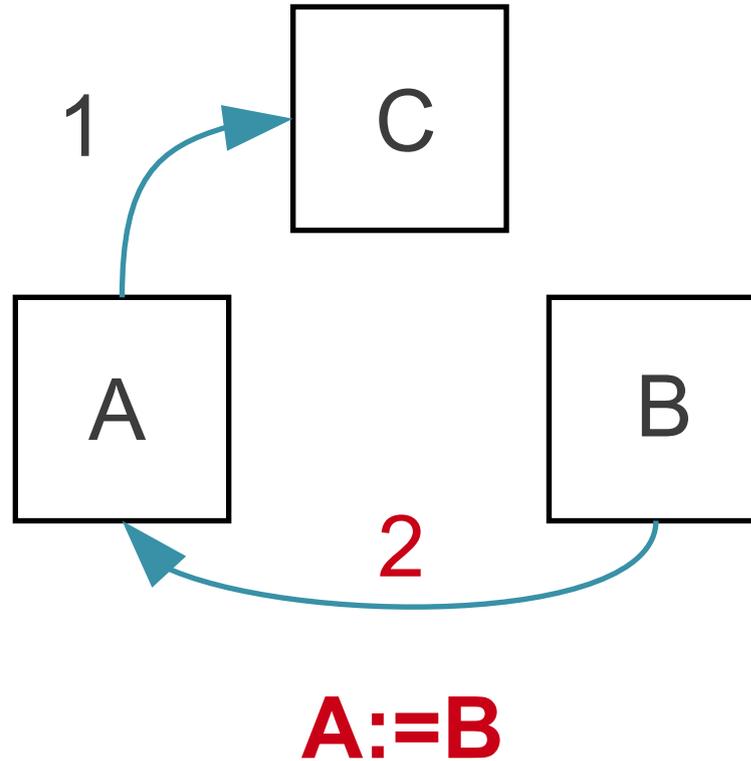


C:=A

Линейные алгоритмы

Решение:

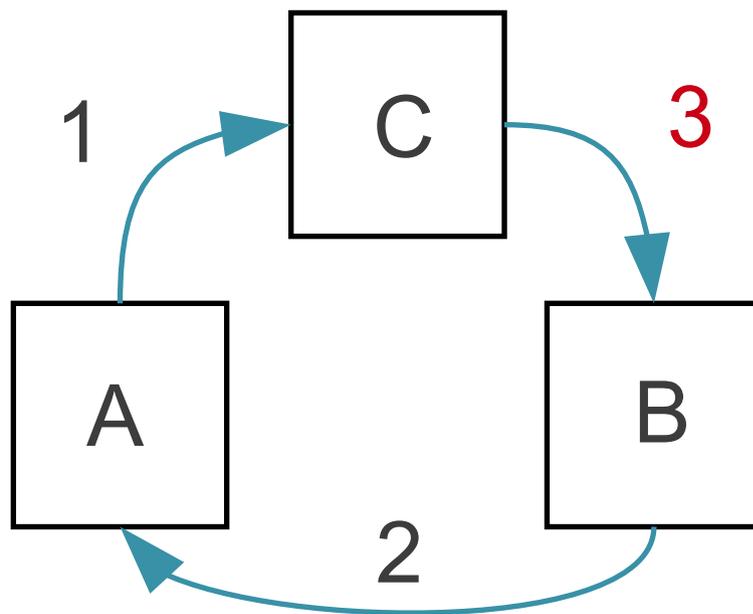
шаг 2



Линейные алгоритмы

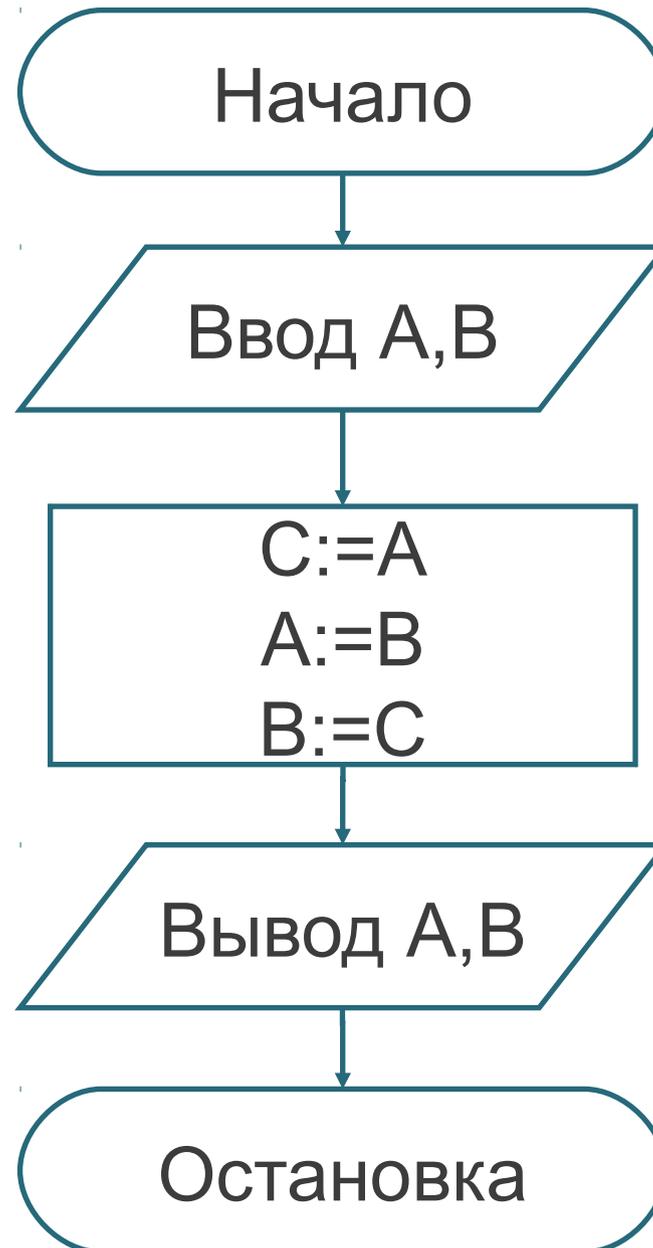
Решение:

шаг 3



B:=C

Линейные алгоритмы



Разветвляющиеся алгоритмы

Разветвляющийся алгоритм

алгоритм

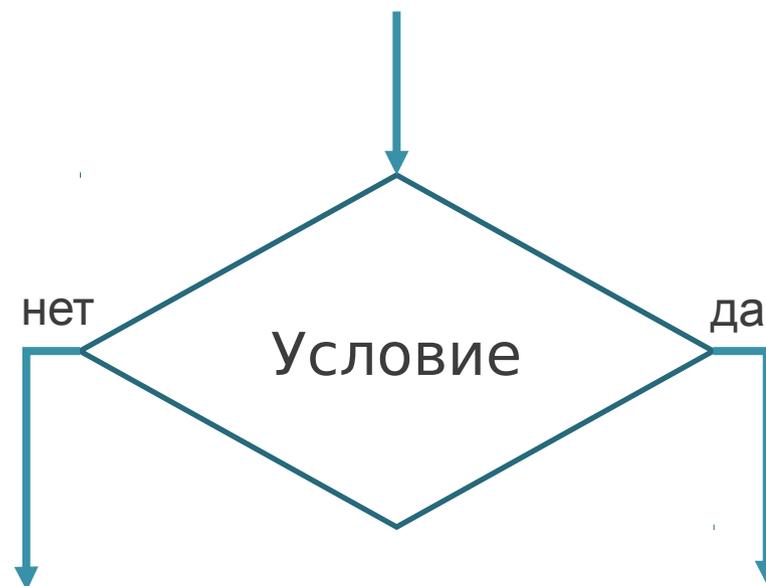
выбор одного из вариантов

Разветвляющиеся алгоритмы

Разветвляющийся алгоритм – это алгоритм, в котором **выбирается один из** возможных путей (**вариантов**) вычислительного процесса.

Каждый путь – ветвь алгоритма.

Блок проверки условия:



Разветвляющиеся алгоритмы

Условие – это логическое выражение, имеющее значение типа **Boolean**: **True** (Истина) или **False** (Ложь).

Логическое выражение (условие) состоит из:

- логических отношений;
- логических операций.

Разветвляющиеся алгоритмы

Логические отношения – это сравнение двух выражений.

Результат сравнения имеет тип **Boolean**:
True (Истина) или **False** (Ложь).

Для сравнения двух выражений используются **операторы сравнения**:

< (Меньше)

<= (Меньше или равно)

> (Больше)

>= (Больше или равно)

= (Равно)

< > (Не равно)

Разветвляющиеся алгоритмы

Логические операции – это операции над логическими отношениями.

Логические операция осуществляются с помощью **логических операторов**:

Не - логическое отрицание

И - логическое умножение

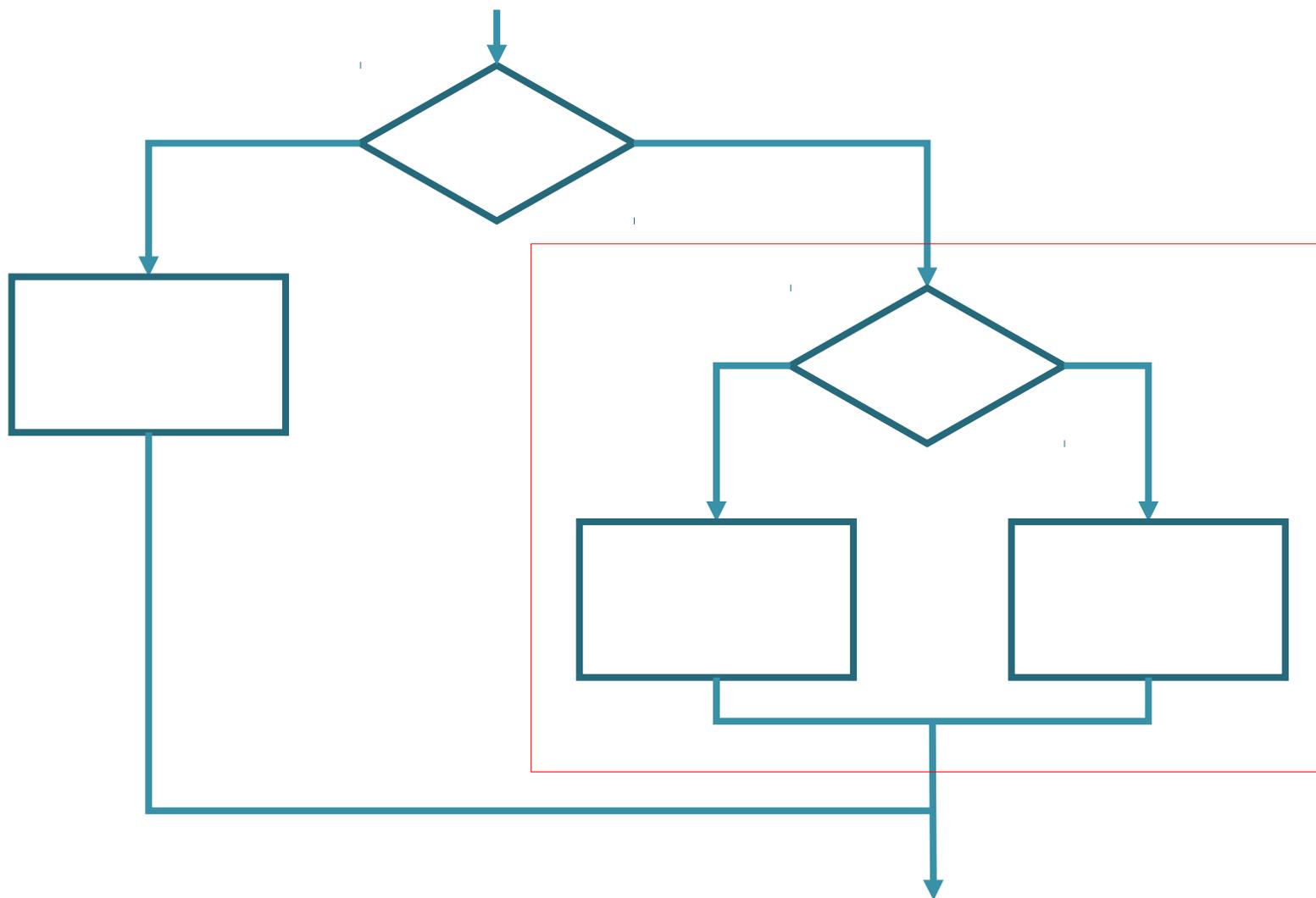
Экв. - эквивалентность

Или - логическое сложение

Искл. Или – исключающее логическое сложение

Разветвляющиеся алгоритмы

Структура разветвляющегося алгоритма



Циклические алгоритмы

Циклический алгоритм

алгоритм

многократная повторяемость

Циклические алгоритмы

Цикл – это многократно исполняемая последовательность инструкций, организованная любым способом (например, с помощью условного перехода).

Тело цикла – последовательность инструкций, предназначенная для многократного исполнения.

Циклические алгоритмы

Итерация – единичное выполнение тела цикла.

Условие выхода (условие окончания цикла) – выражение определяющее, будет в очередной раз выполняться итерация, или цикл завершится.

Счётчик – переменная, хранящая текущий номер итерации.

Виды циклов

1. Безусловные циклы – циклы, выход из которых не предусмотрен логикой программы.

Проверка условия либо отсутствует, либо заменяется константным значением.

Виды циклов

2. Цикл с предусловием – цикл, который выполняется пока истинно некоторое условие, указанное перед его началом.

Это условие проверяется **до** выполнения тела цикла, поэтому тело может быть не выполнено ни разу.



Виды циклов

3. Цикл с постусловием – цикл, в котором условие проверяется после выполнения тела цикла.

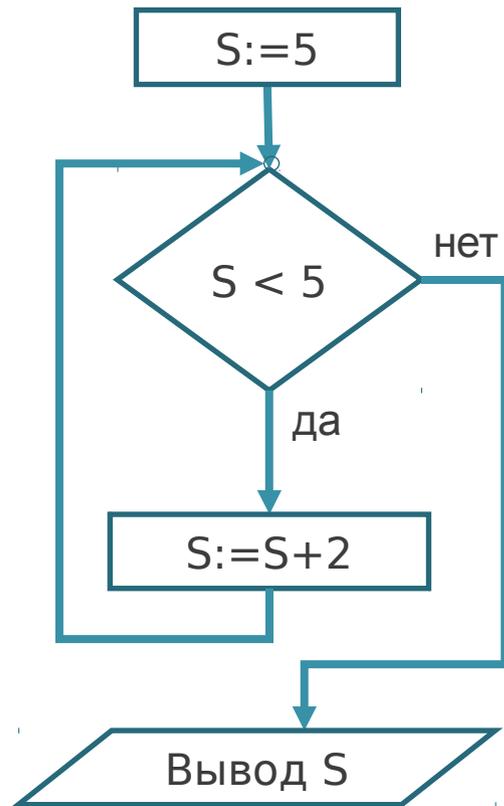
Тело цикла всегда выполняется **хотя бы один раз**.



Виды циклов

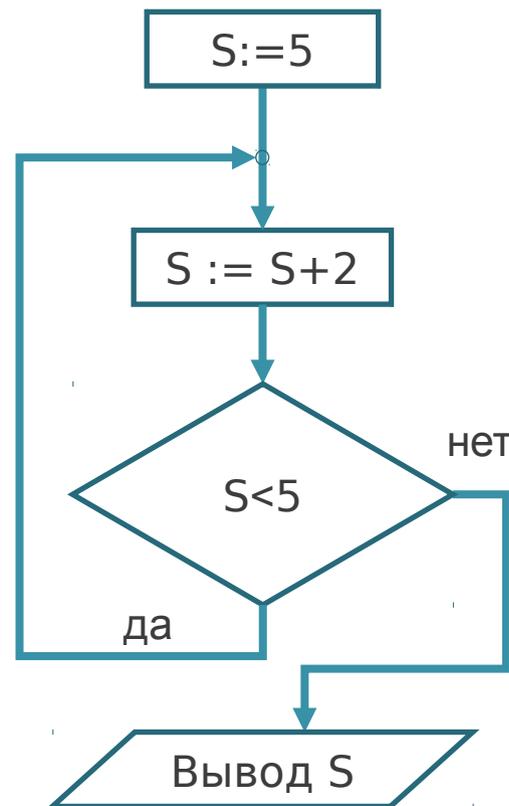
Различия между циклами с предусловием и постусловием

Цикл с предусловием:



Результат: $S = 5$

Цикл с постусловием:



Результат: $S = 7$

Виды циклов

4. Цикл со счетчиком – цикл, в котором некоторая переменная изменяет своё значение

от заданного начального значения

до конечного значения с некоторым шагом,

и для каждого значения этой переменной тело цикла выполняется один раз.

Виды циклов

Вложенный цикл – цикл внутри тела другого цикла.

Количество уровней вложенности не ограничивается.

Задачи «Использование циклов»

Задача:

Подсчитать сумму всех нечетных чисел от 101 до 301.

Исходные данные:

X – начальное значение

Результат:

S – сумма

Задачи «Использование циклов»

Решение.

1. Начальное значение $X = 101$,
начальное значение суммы $S = 0$.

2. Организуем цикл с предусловием

Если X меньше или равно 301 то

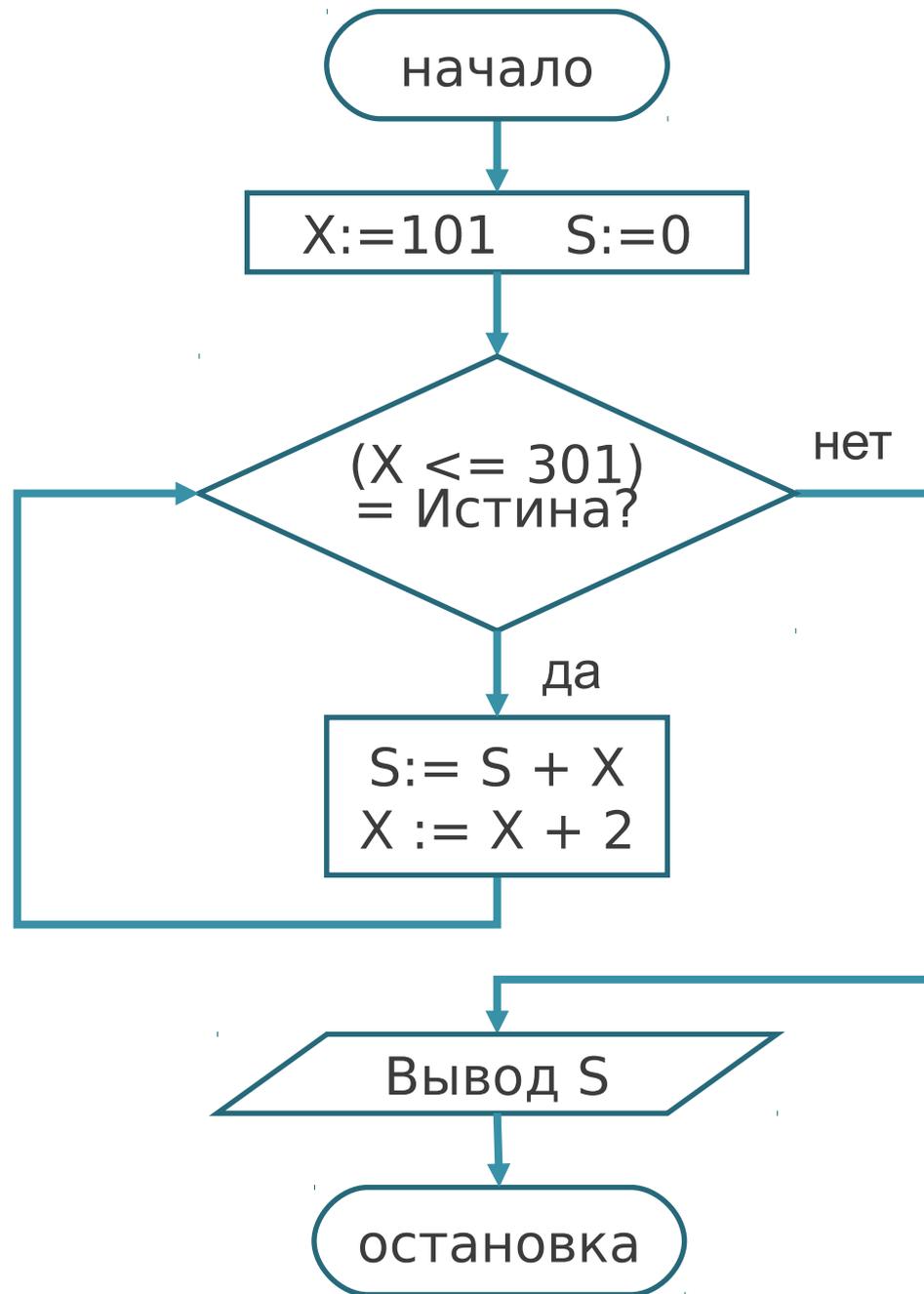
- считаем сумму: $S = S + X$

- увеличиваем значение X на 2: $X = X + 2$

Если X станет больше чем 301, то заканчиваем цикл.

3. Выводим значение суммы.

Задачи «Использование циклов»



Задачи «Использование циклов»

Задача:

Вычислить значение переменной S после выполнения программы при $n=7$.

