

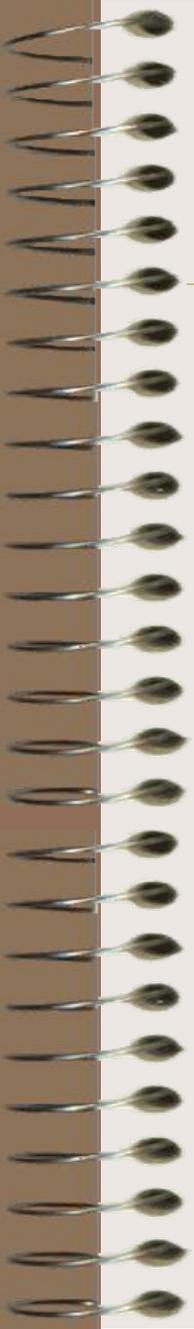
# Программирование

*Носова Людмила Сергеевна*

# Список литературы

---

- 1) Информатика: Учебное пособие для студентов пед.вузов. А.В. Могилев, Н.И.Пак, Е.К.Хеннер, 1999
- 2) Информатика: Учебник Под редакцией Н.В. Макаровой, 1998
- 3) Элементы информатики: Учебное пособие для пед.вузов. Фролов Г.Д., Кузнецов Э.Н., 1989
- 4) Семантический подход к языкам программирования.



---

# § 1 Алгоритмы и исполнители

Задача

---

Данные

Исполнитель

Алгоритм

**Исполнитель алгоритма - человек или автоматическое устройство, способное воспринять и выполнить предусмотренные в нем действия.**

**Конечное множество команд, которые воспринимает исполнитель – это система команд исполнителя (СКИ)**

# Алгоритм

Узбекский математик Аль-Хорезми IX в.

от лат. algorithmi

- **Алгоритм** – это упорядоченный набор из недвусмысленных и выполнимых этапов, определяющих некоторый конечный процесс. (Дж. Гленн Брукшир)
- **Алгоритм** – понятное и точное предписание исполнителю совершить последовательность действий, направленных на достижение поставленной цели (Могилев А.В. и др.)

Исполнитель действует

*формально*, т.е. он только строго выполняет команды алгоритма, не вникая в содержание поставленной задачи и только строго выполняет некоторые правила, инструкции.



# Свойства алгоритма

---

- 1) Дискретность
- 2) Понятность
- 3) Определенность
- 4) Конечность (результативность)
- 5) Массовость

## Способы описания алгоритмов

---

- словесно-формульный;
- структурный или блок-схемный;
- с помощью граф-схем;
- с помощью сетей Петри.
- языки программирования

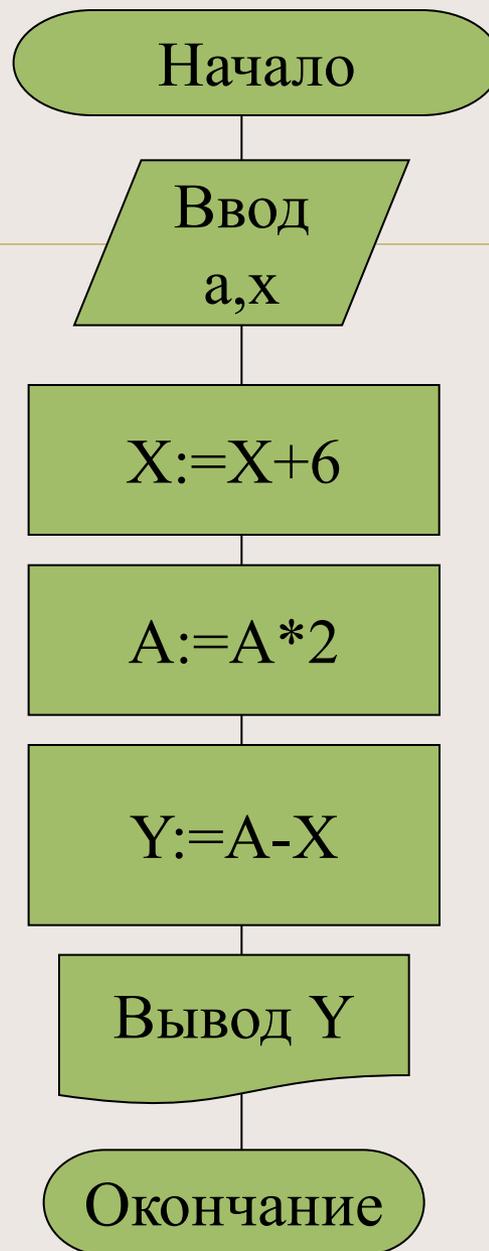
Запись алгоритма на языке  
программирования называется  
**программой**

$$y = 2a - (x+6)$$

Словесно-формульный способ:

1. Ввести значения  $a$  и  $x$ .
2. Сложить  $x$  и  $6$ .
3. Умножить  $a$  на  $2$ .
4. Вычесть из  $2a$  сумму  $(x+6)$ .
5. Вывести  $y$  как результат вычисления выражения.

Блок-  
схемный  
способ



# Язык программирования

---

**Programm** MyPr;

**Var** a,x,y:integer;

**Begin**

Writeln('input a,x');

Readln(a,x);

a:=a\*2; x:=x+6; y:=a+x;

Writeln('y=', y);

**End.**

# Этапы решения задачи

---

- Определение требования и целей
- Проектирование
- Кодирование
- Тестирование

# Примерная схема разработки алгоритма:

---

- 1) Постановка задачи
- 2) Спецификация. *Что дано, что найти, указание ограничений.*
- 3) Математическая модель. *Используемые математические формулы*
- 4) Общая запись алгоритма. *С помощью выбранного способа записи алгоритма.*
- 5) Детализированная запись алгоритма.

# Пример

1) Найти гипотенузу по двум катетам

2) Дано:  $a, b$  – вещественные числа  $>0$ , Найти:  $c$   
– вещественное число

3)  $c = \sqrt{a^2 + b^2}$

4) 1-й шаг

*Алгоритм* Гипотенуза

Объявить переменные

*Начало*

Подготовить исходные данные

Вычислить значение Гипотенузы

Вывести результат

*Конец*

## 2-й шаг

*Алгоритм* Гипотенуза

Вещ  $a, b, c$

*Начало*

Вывести заголовок задачи

Вывести запрос на значение  $a$

Ввести значение  $a$

Вывести запрос на значение  $b$

Ввести значение  $b$

$$c := c = \sqrt{a^2 + b^2}$$

Вывести заголовки выходных данных

Вывести значения  $c$

*Конец*

## §2 Семантический подход к языкам программирования

---

- 2.1. Классификация языков программирования высокого уровня
- 2.2. История языков программирования
- 2.3. Основные понятия языков программирования

**Язык** - знаковая система, служащая  
для хранения, переработки и  
передачи информации

Все языки делятся на

**естественные и искусственные**



---

**Языки программирования – знаковые системы, применяемые для описания процессов решения задачи на ЭВМ. Строго формализованы.**

# Классификация языков программирования,

основанная на степени приближенности  
языка к техническим ресурсам  
компьютера

1. Машинные или внутренние языки, связанные с системой команд компьютера.
2. Машинно–ориентированные языки.
3. Машинно–независимые языки.

# Запись алгоритмов на машинном языке

---

Программа сложения двух чисел,  
хранящихся в ячейках  
с адресами 6С и 6D:

156С 166D 5056      306E С000

Программа на языке с использованием  
мнемонической записи команд, регистров  
и описательных имен областей памяти

- LD R5, PRICE
- LD R6, TAX
- ADDI R0, R5, R6
- ST R0, TOTAL
- HLT

# Программа сложения на языке высокого уровня

---

TOTAL := PRICE + TAX

**Транслятор** - это специальная программа, переводящая текст программы на языке программирования в текст эквивалентной программы на языке машинных команд (ЯМК).

# Объектно-ориентированный подход

Под словом «объект» понимается структура, объединяющая в единое целое данные и программы их обработки.

Популярны: объектно-ориентированные операционные системы, прикладные программы, системы объектно-ориентированного программирования.

# Система программирования включает:

---

- 1) транслятор с языка программирования высокого уровня (обычно компилятор)
- 2) средства редактирования, компоновки и загрузки программы
- 3) макроассемблер (машинно-ориентированный язык)
- 4) отладчик программ

# Транслятор

---

Методы трансляции:

1. Компиляция

2. Интерпретация

Текст программы на ЯПВУ

Исходные данные

Компилятор

Компиляция

Программа на ЯМК

Результат

Текст  
программы  
на ЯПВУ

Исходные  
данные

Интерпретатор

Результат

Текст  
программы  
на ЯПВУ

Исходные  
данные

Интерпретатор

Результат

# Средства редактирования, компоновки и загрузки программ

---

- 1) Текстовый редактор (edit), осуществляющий функции записи и редактирования исходного текста программы.
- 2) Загрузчик программы (load), позволяющий открыть нужный файл программы
- 3) Запускатель программ (run), осуществляющий процесс выполнения программы и т.д.

# Отладчик (Debugger)

---

позволяет осуществить пошаговое выполнение программ (трассировку), идентификацию места и вида ошибок в программе, отслеживание значений переменных и т.д.

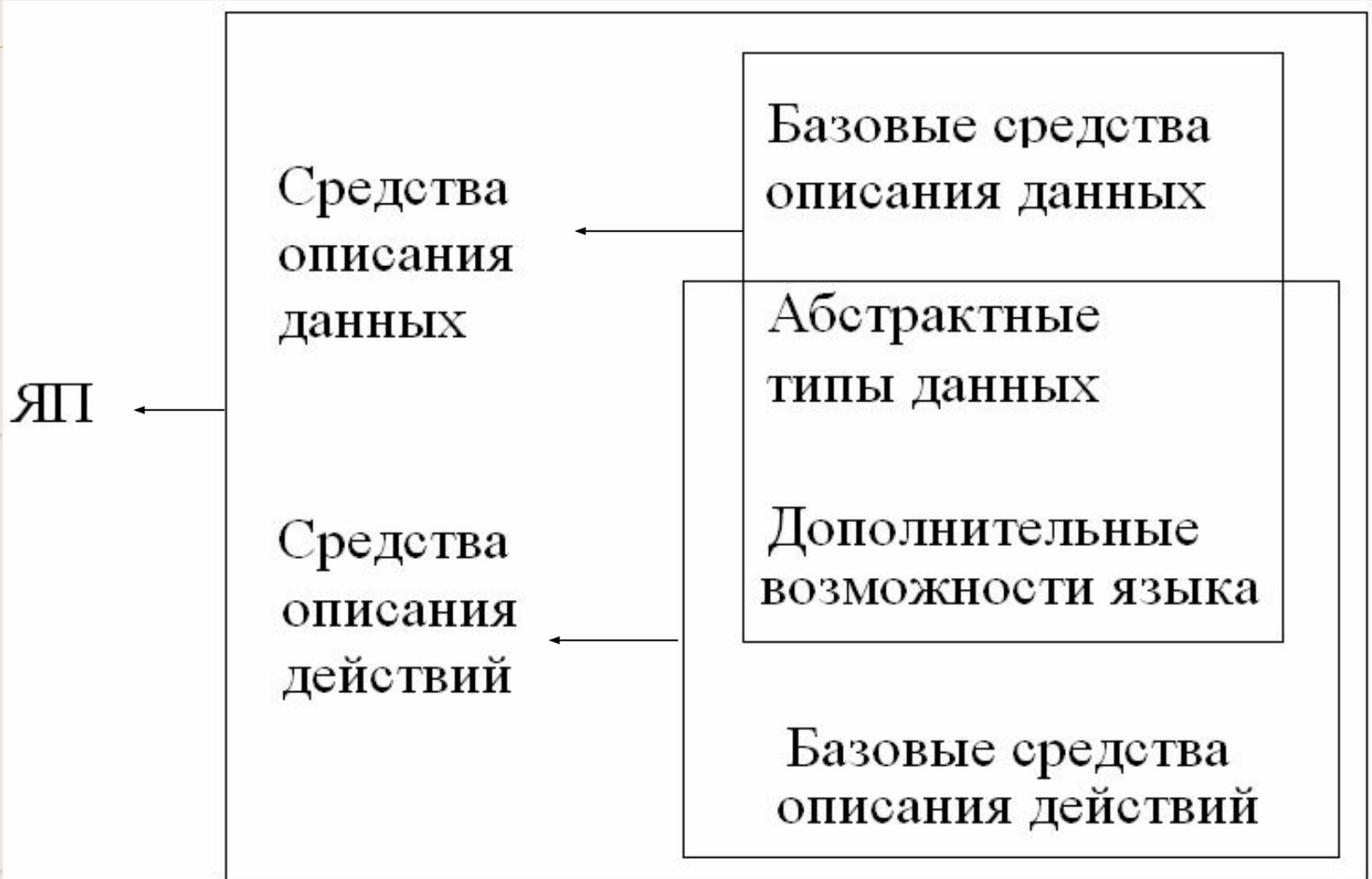
# Язык программирования как объект изучения:

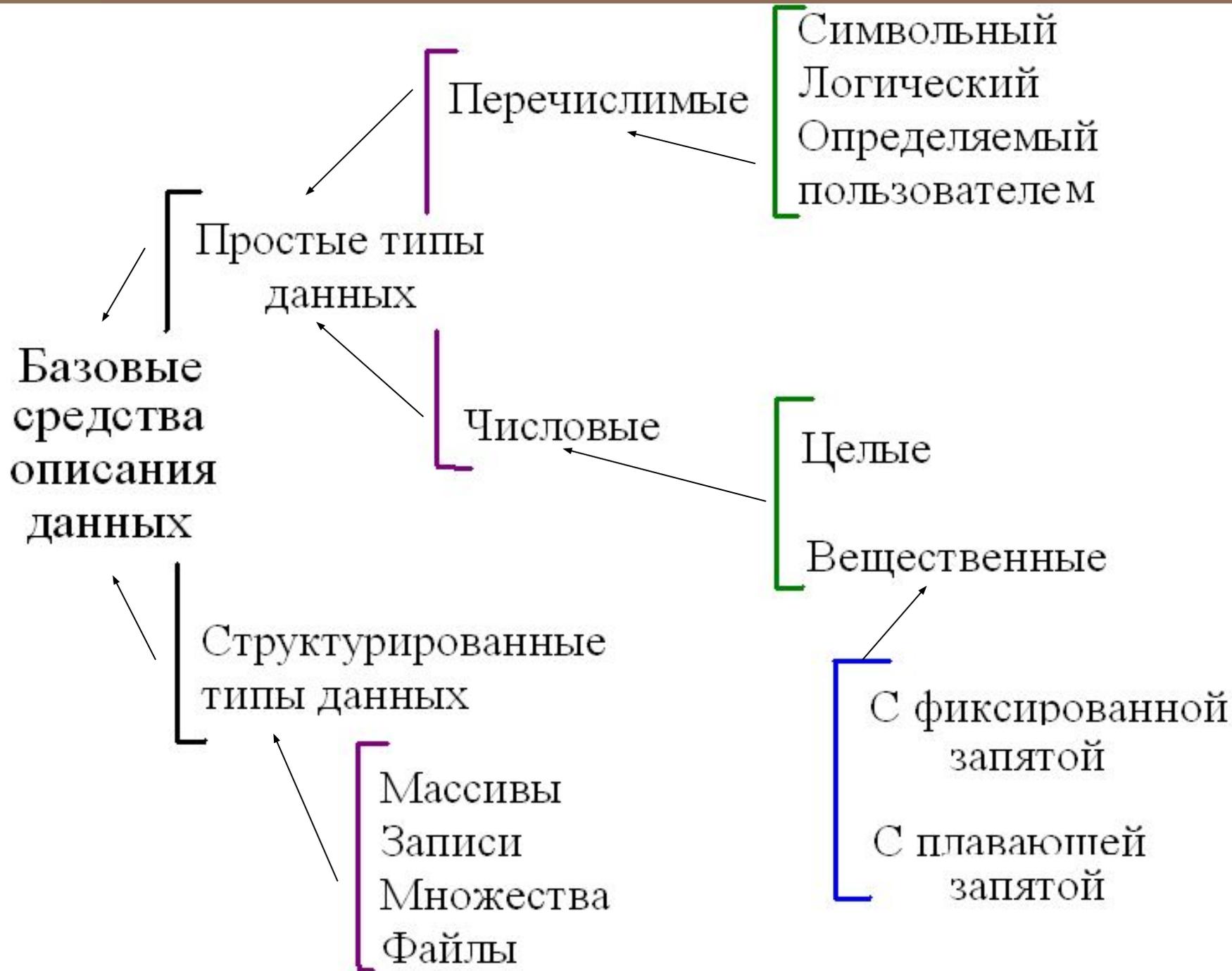
---

- элементы языка (алфавит, лексеммы, синтаксис)
- средства организации данных (типы и структуры данных)
- средства организации действий над данными

- **Алфавит** – фиксированный для данного языка набор основных символов, допускаемых для составления текста программы на этом языке.
- **Лексемы** – минимальные значимые единицы текста программы. Категории лексем: специальные символы, идентификаторы, зарезервированные слова, числа, метки, строки, комментарии.
- **Синтаксис** – система правил, определяющих допустимые конструкции ЯП из знаков алфавита.
- **Семантика** – система правил однозначного толкования отдельных языковых конструкций, позволяющих воспроизвести процесс обработки данных. Семантика языка программирования задается определением средств описания данных и действий (алгоритм).

# Схема семантики ЯП





Базовые  
средства  
описания  
действий

Выражения

Операторы действий

Операторы управления

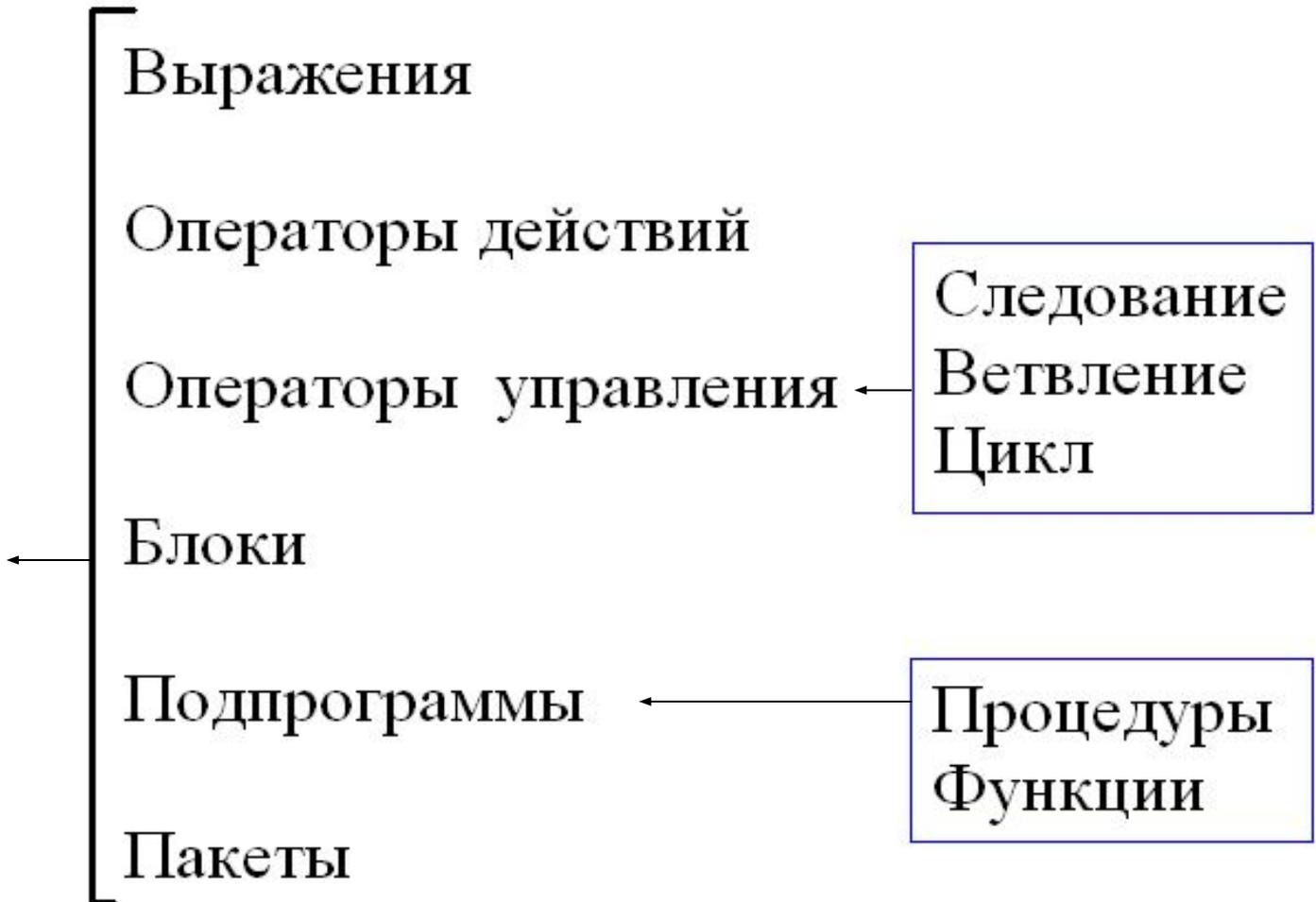
Блоки

Подпрограммы

Пакеты

Следование  
Ветвление  
Цикл

Процедуры  
Функции



# Объекты данных

---

Возможности доступа к объектам данных:

1. Ссылки на объекты данных
2. Указание типов данных

Переменная характеризуется:

1. Ссылкой
2. Значением
3. Именем

# Переменная



$A$  – целое,  $A = 7$



# Средства описания действий.

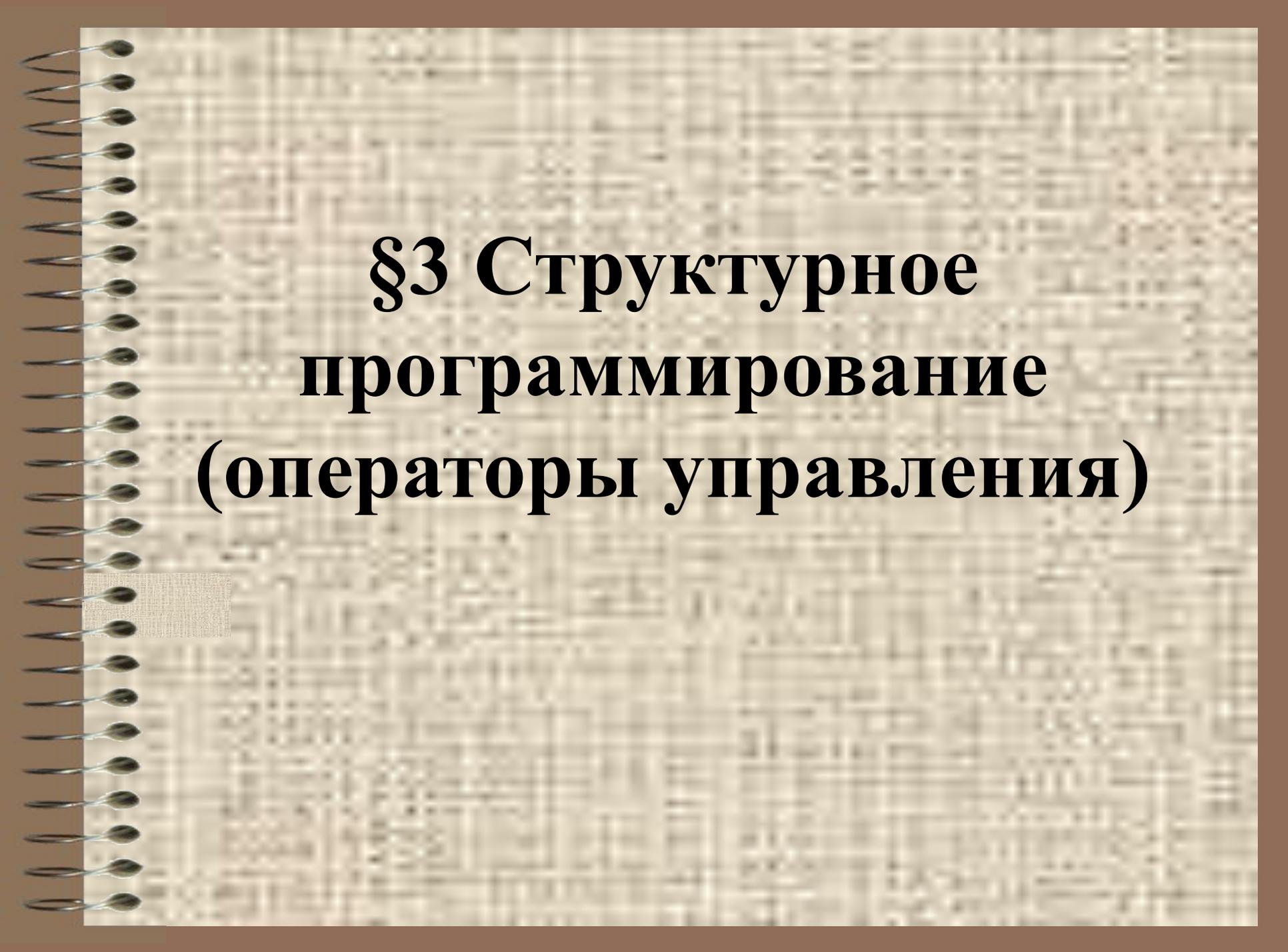
## Выражения и операторы

---

**Выражения** в программировании служат для определения действий, которые в математике обычно описываются формулами.

**Операторы действия** — элементы языка, предназначенные для изменения объектов данных, а также для выполнения некоторых вспомогательных действий

Средства, явно задающие последовательность выполнения операторов в ЯП — **операторы управления**

The background of the slide is a spiral-bound notebook with a light-colored, textured cover and a silver metal spiral binding on the left side. The text is centered on the page.

# **§3 Структурное программирование (операторы управления)**

# Подходы к созданию алгоритмов

---

- 1) **Операциональный** (ЭВМ 1, 2 поколения).  
Все шаги алгоритма имеют адрес (метку), естественный порядок нарушается переходами на команды с определёнными адресами. Переходы бывают условными и безусловными.
- 2) **Структурный**

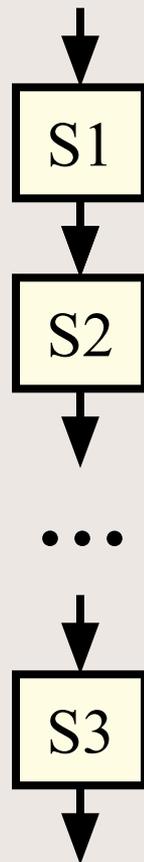
Пр: Дано 2. Найти разницу большего и меньшего

---

- 1) Ввести числа  $a$  и  $b$
- 2) Сравнить  $a$  и  $b$ , если  $a > b$  перейти к метки 3, иначе перейти к метке 5
- 3)  $d$  присвоить  $a-b$
- 4) Перейти к метке 7
- 5)  $d$  присвоить  $b-a$
- 6) Перейти к метке 7
- 7) Вывести  $d$
- 8) Стоп

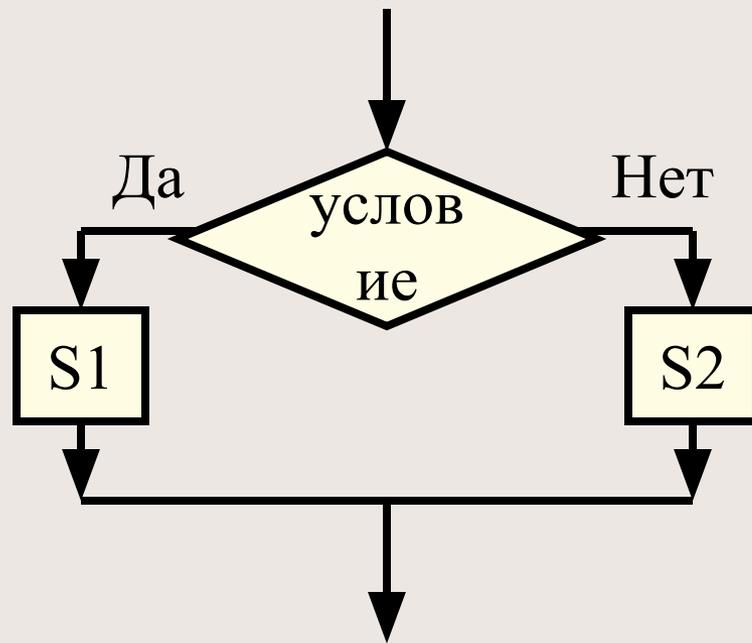
# Следование

Управление передается от одного блока к следующему – однозначная (линейная последовательность действий)

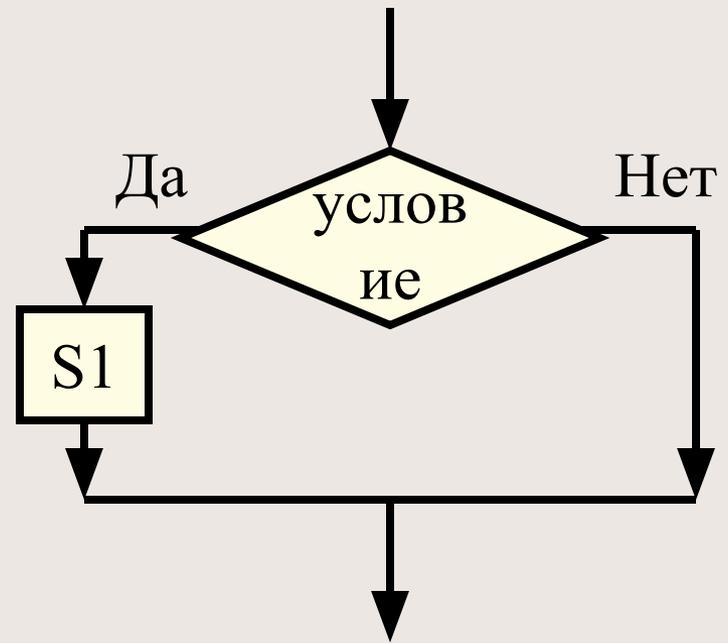


# Ветвление

Разделение алгоритма на два пути (две ветви) по некоторому условию с дальнейшим выходом на общее положение.



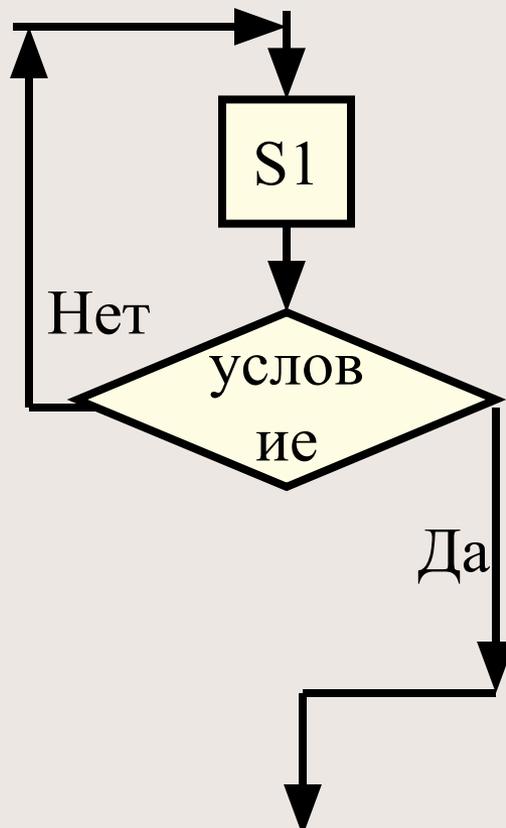
Полное ветвление



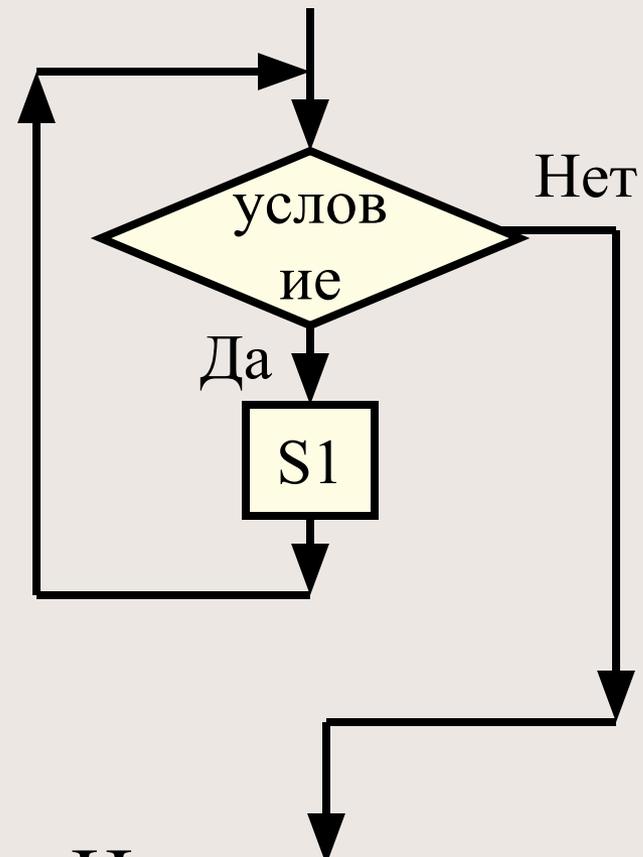
Неполное ветвление

# ЦИКЛ

Повторение некоторой группы действий по условию



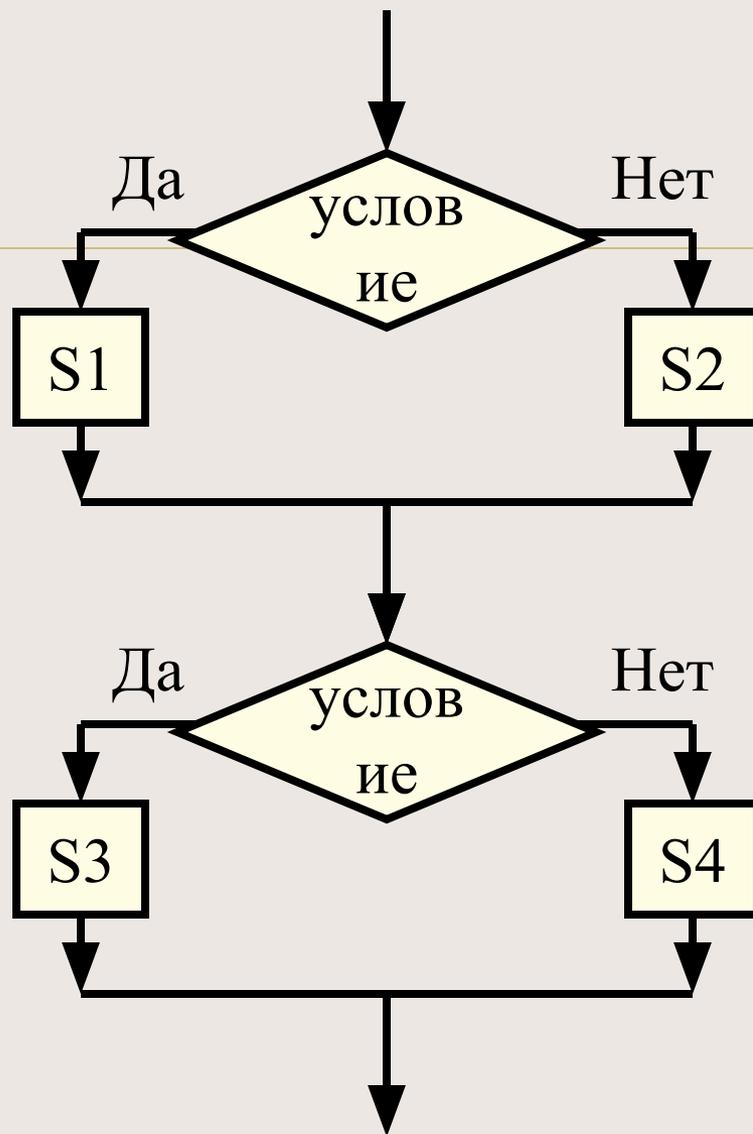
Цикл с постусловием



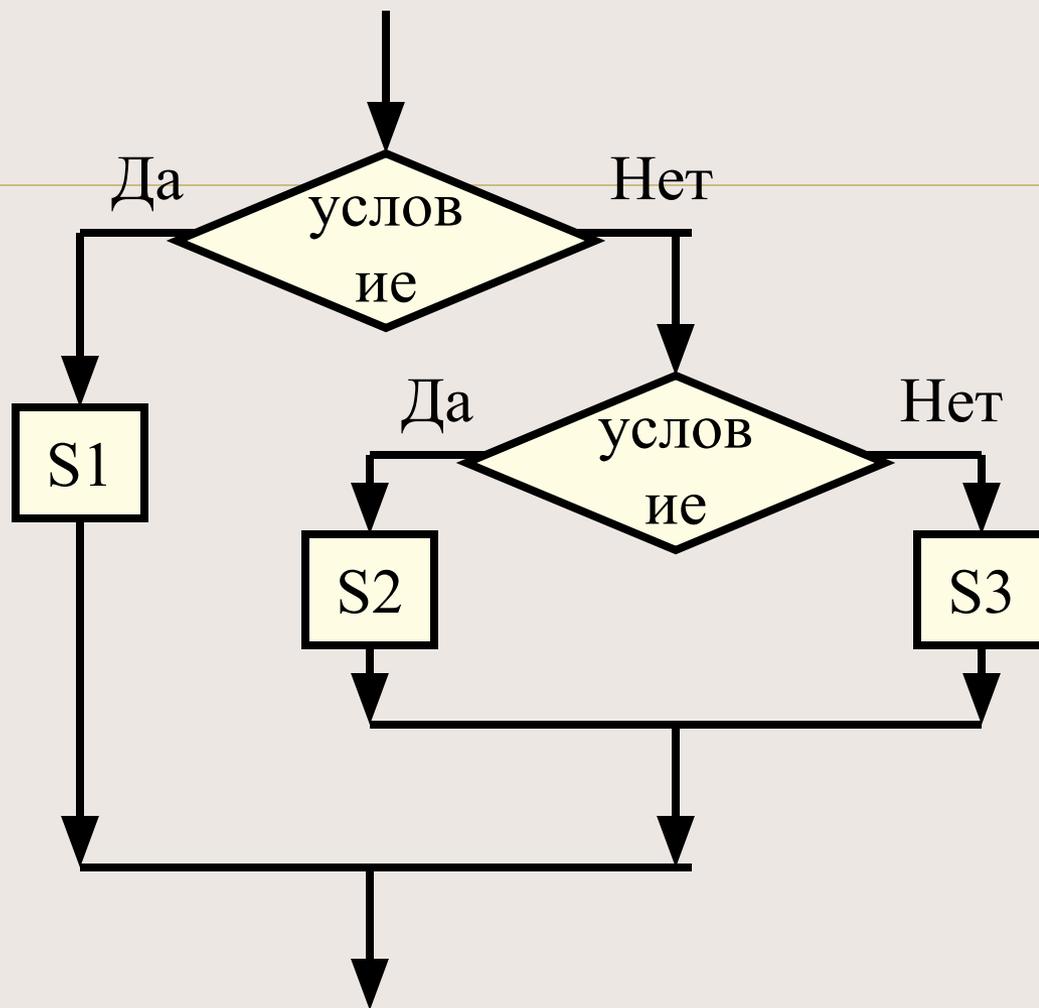
Цикл с предусловием

**Сложный алгоритм состоит из множества соединенных между собой базовых структур.**

**Два способа соединения структурных элементов алгоритма: последовательный и вложенный.**



последовательный



**ВЛОЖЕННЫЙ**

Второй фундаментальный  
принцип структурного  
программирования является  
**метод пошаговой детализации  
алгоритма**

# Метод пошаговой детализации

---

или построение алгоритма  
«сверху вниз»

Сначала строиться основной алгоритм, состоящий из больших частей, затем каждая из этих частей конкретизируется.