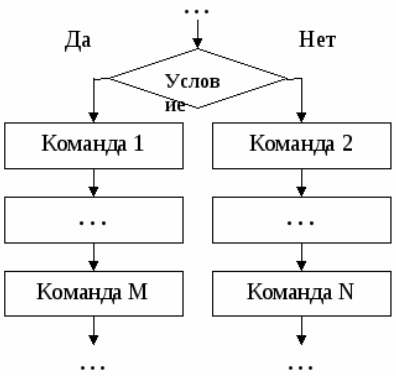
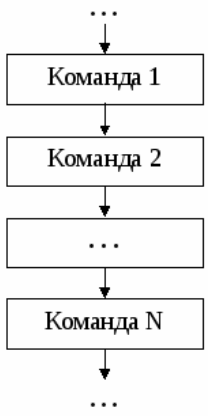


Структуры алгоритмов и программ



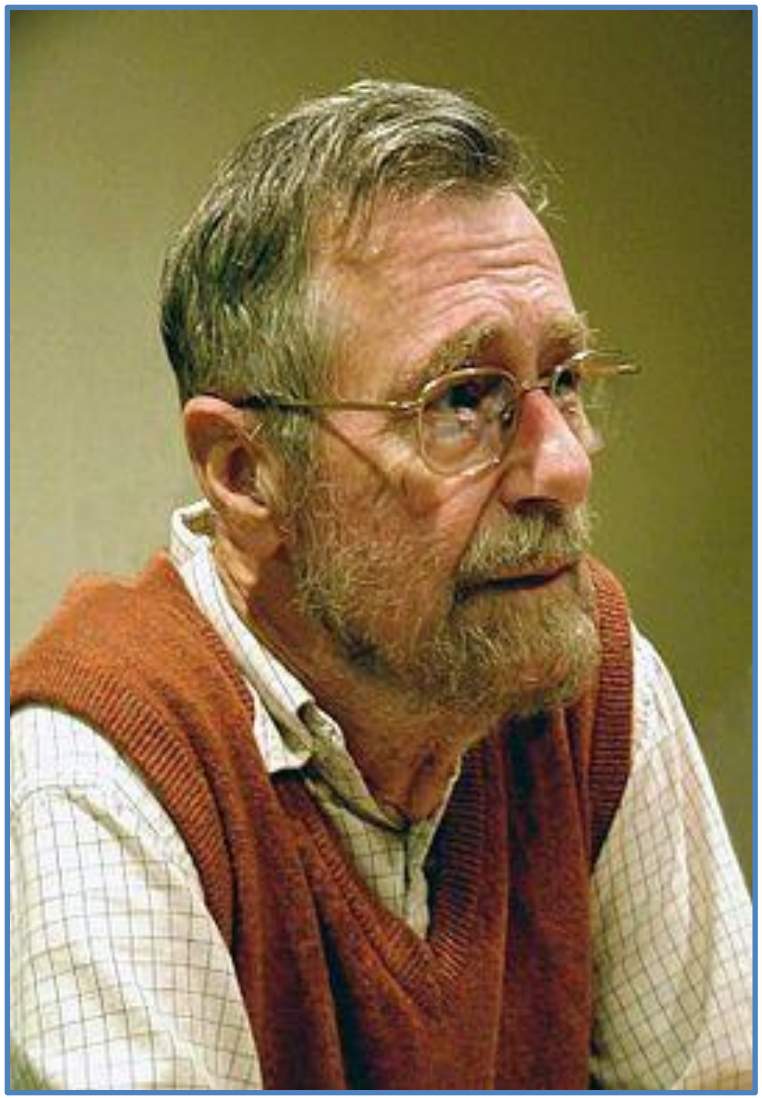
Немного истории

1. Программа решения системы двух линейных алгебраических уравнений с двумя неизвестными.
2. Программа вычисления значений тригонометрической функции с многократным повторением заданной последовательности вычислительных операций.
3. Нахождение чисел Бернулли.

Гипотеза

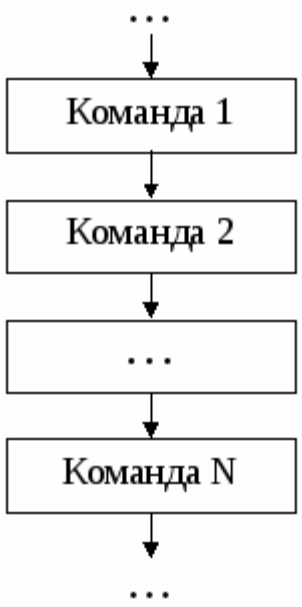
Сложность решения математических задач не зависит от сложности алгоритма для их решения

Теорема Дейстры



В 1969 году Эдсгер Дейкстра доказал теорему, суть которой заключается в следующем: «Алгоритм для решения любой логической задачи можно составить только из трех структур – СЛЕДОВАНИЕ, ВЕТВЛЕНИЕ и ЦИКЛ». Эти структуры называются **базовыми**.

Следование

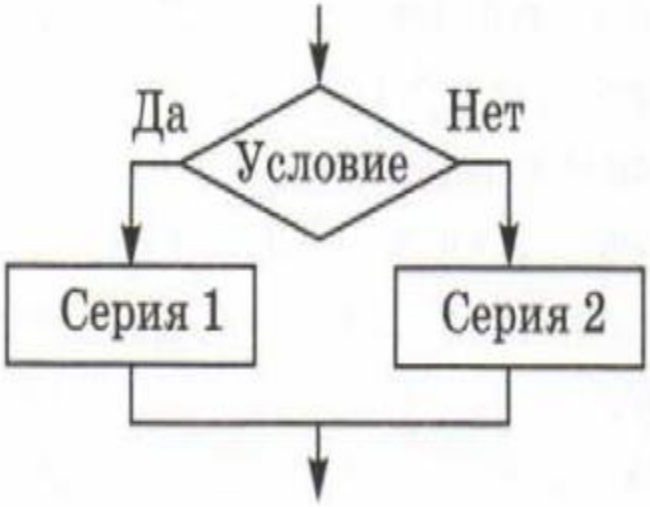



```
Begin
  a:=5.6;
  b:=a*7;
End.
```

Следование – это линейная последовательность действий

Ветвление

Полная форма

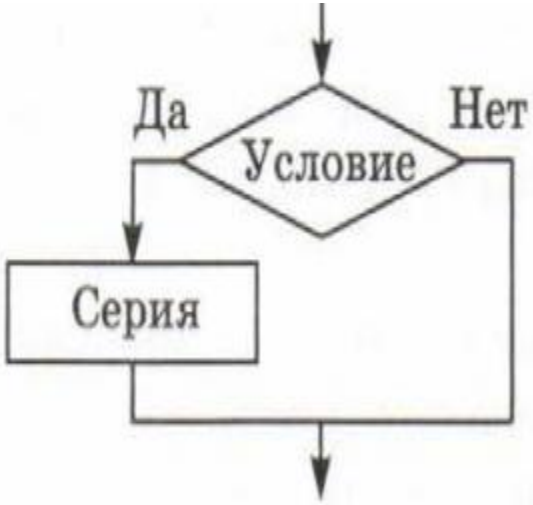



```
if <логическое выражение>
then <серия 1>
else <серия 2>;
```

Ветвление – алгоритмическая альтернатива. Управление передается одному из двух блоков в зависимости от истинности или ложности условия.

Ветвление

Неполная форма




```
if <логическое выражение>
then  <серия>;
```

Цикл

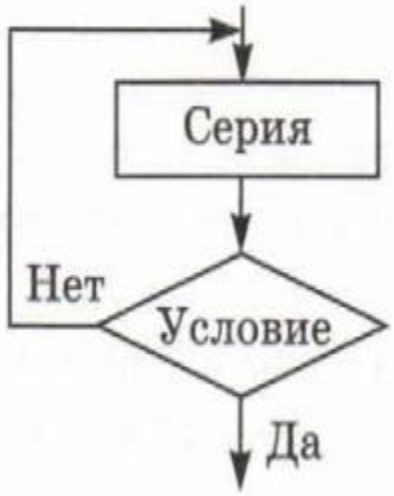
Цикл с предусловием (цикл-пока)

```
while <логическое выражение> do
  <серия>;
```

Пока условие истинно, выполняется серия, образующая тело цикла.

Цикл

Цикл с постусловием (цикл-до)




```
repeat
  <серия>
until <логическое выражение>
```

Тело цикла повторяется, если условие ложно.

Особенности структурного программирования

Это программирование без goto – оператора безусловного перехода.

Что приносит применение безусловного перехода:

- Программа лишается структурности;
- Перестает быть «прозрачной»
- Алгоритм перестает быть надежным.

Какого типа алгоритм надо использовать, чтобы решить следующие математическое задачи:

1. Нахождение среднего арифметического.
2. Подсчет суммы ста чисел.
3. Решение неравенства.
4. Решение квадратного уравнения

Комбинации базовых структур

Сложный алгоритм состоит из соединенных между собой базовых структур.
Соединяться эти структуры могут двумя способами:

1) Последовательно,
2) Вложено один в другой.

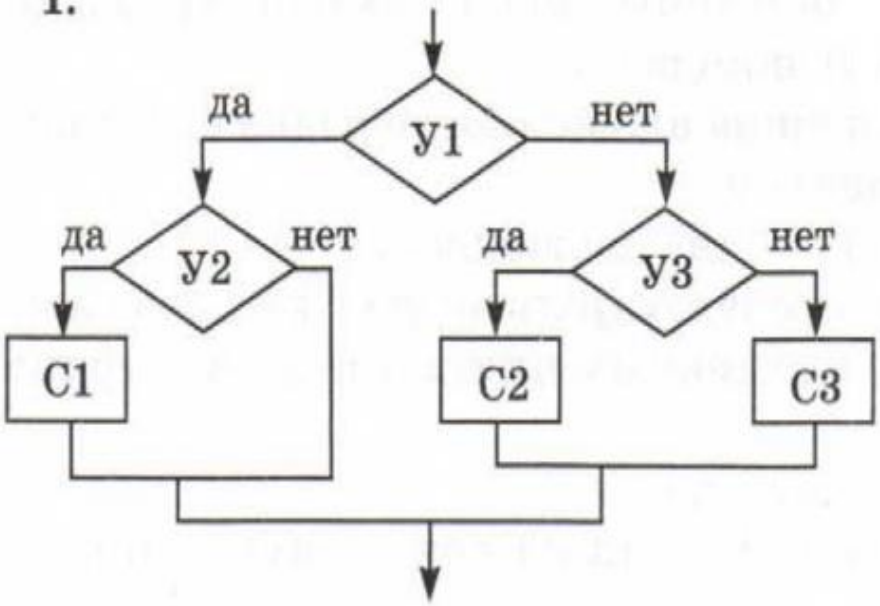
Комбинации базовых структур

Если блок, составляющий тело цикла, сам является циклом, то можно говорить о вложенных циклах. В свою очередь внутренний цикл может иметь внутри себя еще один цикл и т.д. В этом случае говорят о ***глубине вложенности цикла***.

А возможна ли подобная ситуация с уловным оператором?

Примеры комбинаций структур алгоритмов

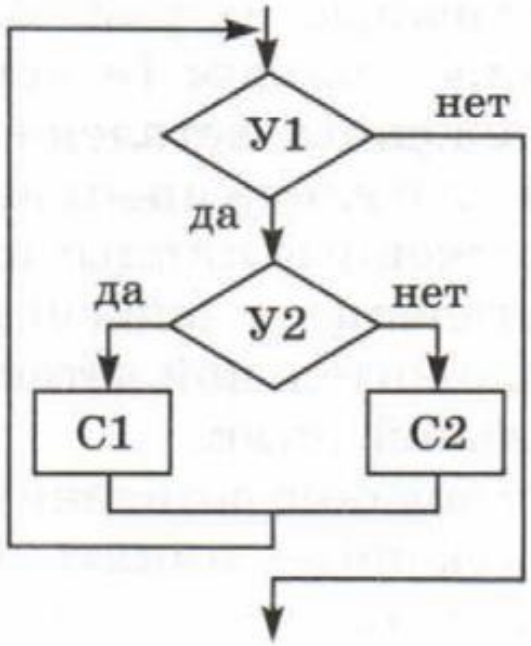
1.



1.

```
if <У1>
then
begin
   if <У2>
   then <С1>
end
else
   if <У3>
   then <С2>
   else <С3>;
```

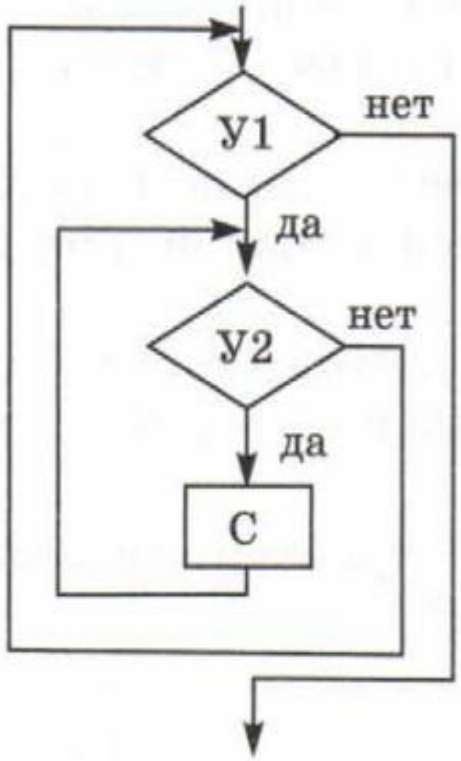
Примеры комбинаций структур алгоритмов



2.

```
while <У1> do
  if <У2>
  then <С1>
  else <С2>;
```

Примеры комбинаций структур алгоритмов

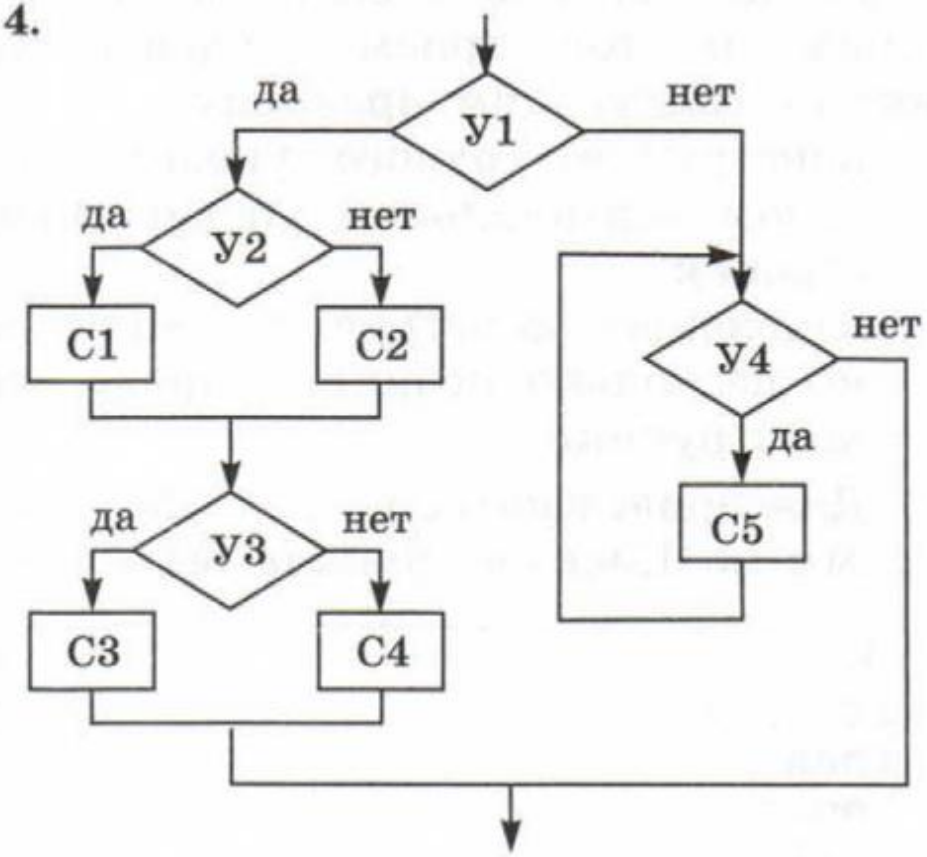


3.

```
while <У1> do
  while <У2> do
    <С>;
```

Примеры комбинаций структур алгоритмов

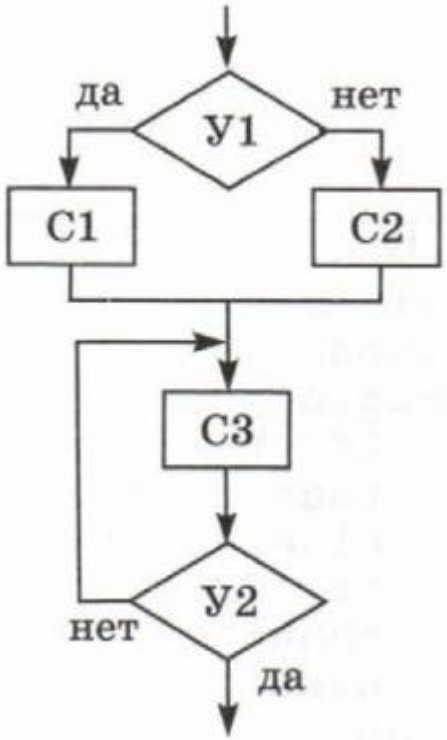
4.



4.

```
if <У1>
then
begin
   if <У2>
   then <С1>
   else <С2>;
   if <У3>
   then <С3>
   else <С4>
end
else while <У4> do <С5>;
```

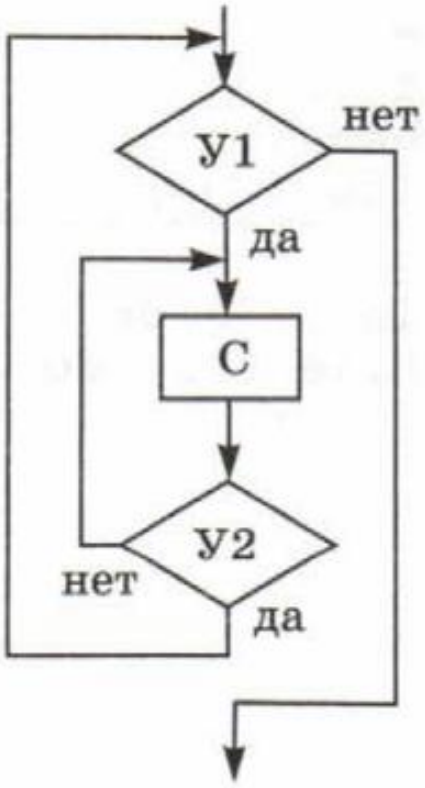
Примеры комбинаций структур алгоритмов



5.

```
if <У1>
then <С1>
else <С2>;
repeat
   <С3>
until <У2>;
```

Примеры комбинаций структур алгоритмов



6.

```
while <У1> do
  repeat
    <С1>
  until <У2>;
```

Правила написания программы

1. Конструкции одного уровня вложенности записываются на одном вертикальном уровне (начинаются с одной позиции в строке)
2. Вложенная конструкция записывается смещенной по строке на несколько позиций вправо относительно внешней для нее конструкции.

```
Procedure PROCESS(NODE)
begin
 If NODE intersects any non-FULL window then
  If NODE is TERMINAL then
   begin
    For WINDOW ← non-FULL intersecting window
     begin
       TERM(NODE,WINDOW)
     end
   end
  else
   begin
    PROCESS(CHILD(NODE))
   end
 else
  begin
   NODE ← NEXT(NODE)
  end
end

procedure TERM(NODE,WINDOW)
begin
 if WINDOW is EMPTY and enclosed by NODE then
  begin
   PAINT WINDOW
   mark WINDOW FULL
  end
 else
  begin
   for I:=1 to 4 do
    begin
     TERM (NODE,CHILD2(WINDOW(I)))
    end
  end
end

procedure CHILD(NODE)
 (* subdivides NODE into its children and returns
 first non-empty child *)

procedure CHILD2(WINDOW,I)
 (* returns child number I of WINDOW *)

procedure NEXT(NODE)
 (* returns next node in front-to-back traversal
sequence,
 i.e. the first non-empty node in sequence not a
 descendent of NODE *)
```

О структурном программировании

Структурное программирование – это не только форма описания алгоритма и программы, но еще и *способ мышления программиста*. Размышляя над алгоритмом, нужно стремиться составлять его из стандартных структур.

4. Нарисуйте блок-схемы и напишите на Паскале два варианта программы решения задачи: выбрать из трех числовых величин наименьшее значение. Первый вариант — с вложенными ветвлениями, второй вариант — с последовательными ветвлениями.
5. Для данного натурального числа N требуется вычислить сумму: $1 + 1/2 + + 1/3 + ... + 1/N$. Постройте алгоритм и напишите два варианта программы на Паскале: с циклом-до и с циклом-пока.
6. Какую структуру будет иметь алгоритм решения следующей задачи? Дано целое положительное число N. Если N — четное, то вычислить $N! = 1 \cdot 2 \cdot ... \cdot N$. Если N — нечетное, то вычислить сумму: $1 + 2 + ... + + N$. Составьте программу на Паскале.