Алгоритмизация и программирование

Этапы решения задачи на ПК

Подготовка и решение инженерных задач на компьютере включает последовательность этапов, составляющих *жизненный цикл* программного продукта.

1. **Постановка задачи**. На этом этапе формулируется цель решения задачи, подробно описывается содержание задачи, анализируются условия и требования, при которых решается задача, а также выявляется область определения входных параметров, которые называются исходными

2. **Формальное описание задачи**. На этом этапе выявляются существенные элементы и взаимосвязи между ними, составляется формальная модель задачи в виде схемы, диаграммы или сжатого лаконичного описания.
3. **Математическое моделирование**. Выполняется *математическая формализация* задачи. Описательная модель записывается с использованием наиболее адекватного математического языка, осуществляется выбор метода решения задачи. Как правило, математическая модель строится приближенно, то есть с определенной

4. **Алгоритмизация**. Составляется алгоритм решения задачи в соответствии с выбранной математической моделью. При разработке алгоритма вычислительный процесс разбивается на отдельные самостоятельные блоки и определяется последовательность выполнения этих блоков.
5. **Программирование**. Выполняется запись разработанного алгоритма на конкретном языке программирования.
6. **Тестирование и отладка программы**. Проводится устранение синтаксических и логических ошибок, которые возникают в процессе трансляции и тестирования программы.

7. **Анализ и интерпретация результатов**. Первоначально выполняется многократное решение задачи для разных наборов данных. Из анализа полученных результатов проводится формулировка выводов и выработка мер по реализации выбранного решения.
8. **Внедрение и сопровождение.** Если заказчик удовлетворен качеством программного продукта, то наступает период его внедрения в эксплуатацию. Как правило, сотрудничество исполнителя с заказчиком продолжается. Такое взаимодействие исполнителя и заказчика называется *сопровождением программы*.

Алгоритм и его свойства

Алгоритмизация – это совокупность приемов и способов составления алгоритмов для решения алгоритмических задач.

Алгоритм – *точное предписание*, задающее процесс, обеспечивающий получение результатов, соответствующих определенным входным данным.

Основными свойствами алгоритмов являются:

- **Массовость (универсальность)** – применимость алгоритма для решения класса задач, различающихся исходными данными.
- **Дискретность** – процесс решения задачи разбит на отдельные действия (шаги).

- **Определенность (детерминированность)** – правила и порядок выполнения действий алгоритма имеют *единственное толкование*.
- **Результативность** – результат алгоритма достигается за конечное число шагов.

Способы записи алгоритма

- **Словесное описание** – запись алгоритма на естественном языке или в виде структурированной записи на ***псевдокоде*** (частично формализованном языке).
- **Графическая форма** – представление алгоритма в виде блок–схемы.

Блок-схема – описание структуры алгоритма с помощью геометрических фигур (блоков) с линиями–связями, показывающими порядок выполнения отдельных операций. Внутри блоков указывается информация об операциях, подлежащих выполнению.

Графическая форма записи является удобным средством изображения алгоритмов и получила широкое распространение в научной и учебной литературе.

- **Программа** – запись алгоритма на алгоритмическом языке. Среди программистов, пишущих программы для ПК, наибольшей популярностью пользуются языки Си, Паскаль и Бейсик.

Базовые структуры алгоритмов

В зависимости от последовательности выполнения действий выделяют алгоритмы *линейной, разветвляющейся и циклической* структуры.

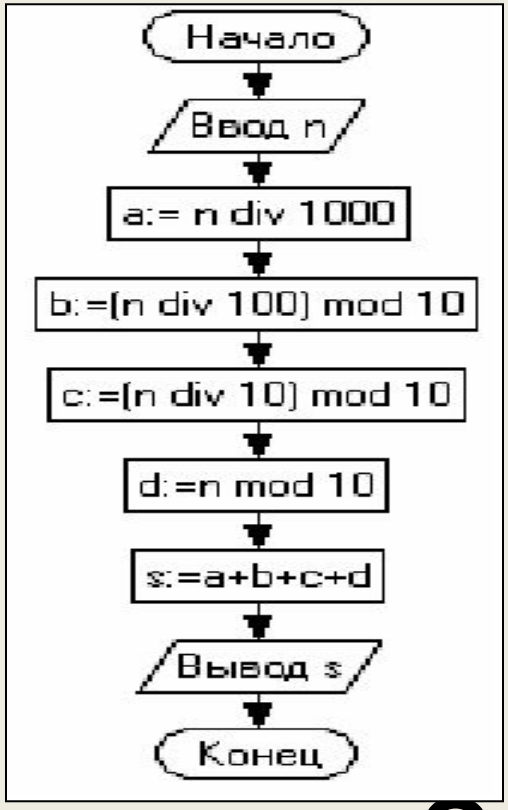
- **Линейный алгоритм** – это алгоритм, в котором действия выполняются последовательно одно за другим от начала до конца.

Как правило, основу линейного алгоритма составляют три операции: операция ввода исходных данных, операция определения значения переменной и операция вывода результата.

Пример линейного алгоритма

Для заданного алгоритма при исходном значении **n=5874** значение переменной **S** будет равно. **?**

Блок-схема



Псевдокод

Начало
Ввод n
  a := n div 1000
  b := (n div 100) mod 10
  c := (n div 10) mod 10
  d := n mod 10
  S := a+b+c+d
Вывод S
Конец

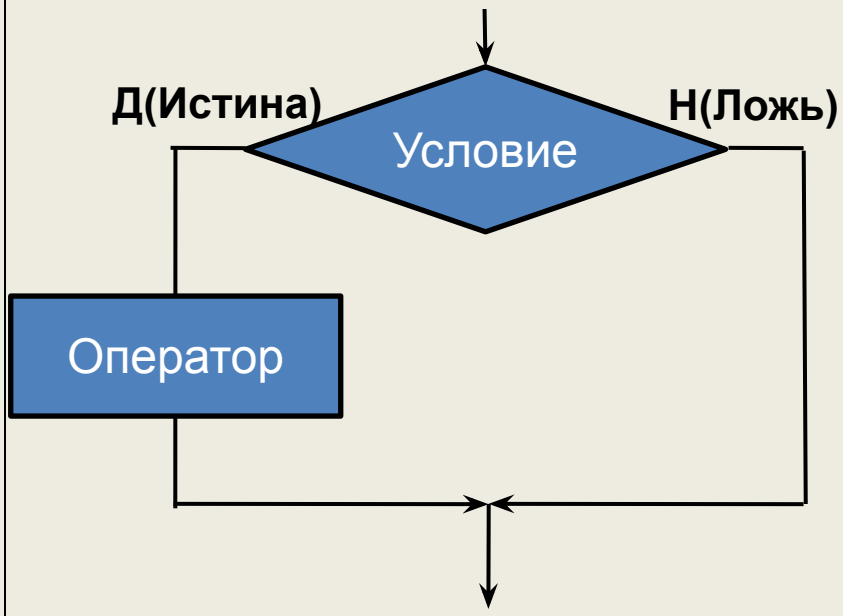
Турбо-Паскаль

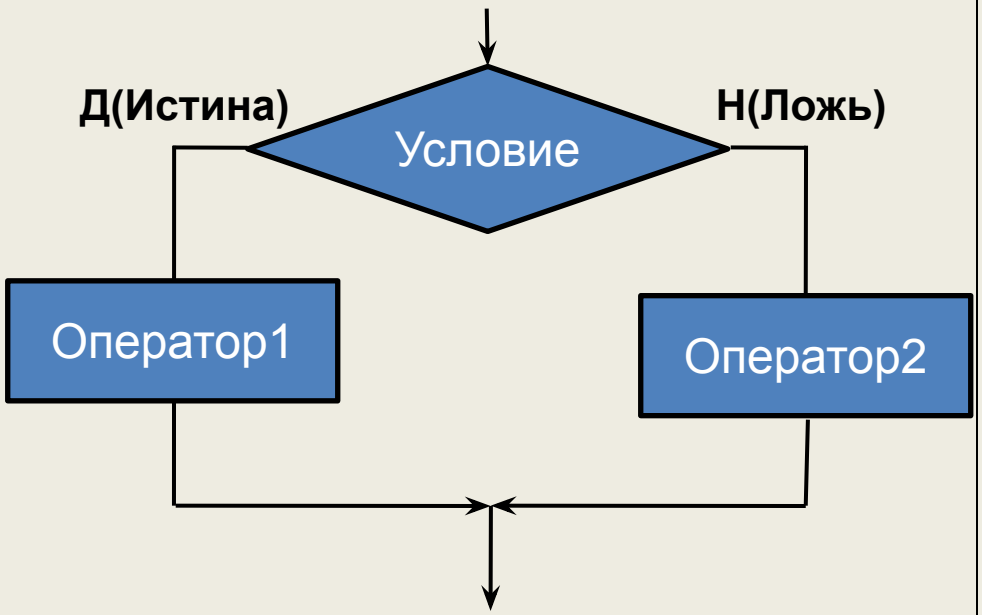
```
Program LVP;
Var
   a,b,c,d,S: integer;
Begin
  writeln (‘Введите n’);
  read(n);
  a := n div 1000;
  b := (n div 100) mod 10;
  c := (n div 10) mod 10;
  d := n mod 10;
  S := a+b+c+d;
  writeln(‘S=‘, S);
End.
```

Ответ: **S=24**

- **Разветвляющийся алгоритм** в зависимости от проверяемого условия позволяет реализовать вычисление по одному из нескольких направлений (*ветвей*). При разработке алгоритма необходимо учитывать все возможные ветви вычислений.

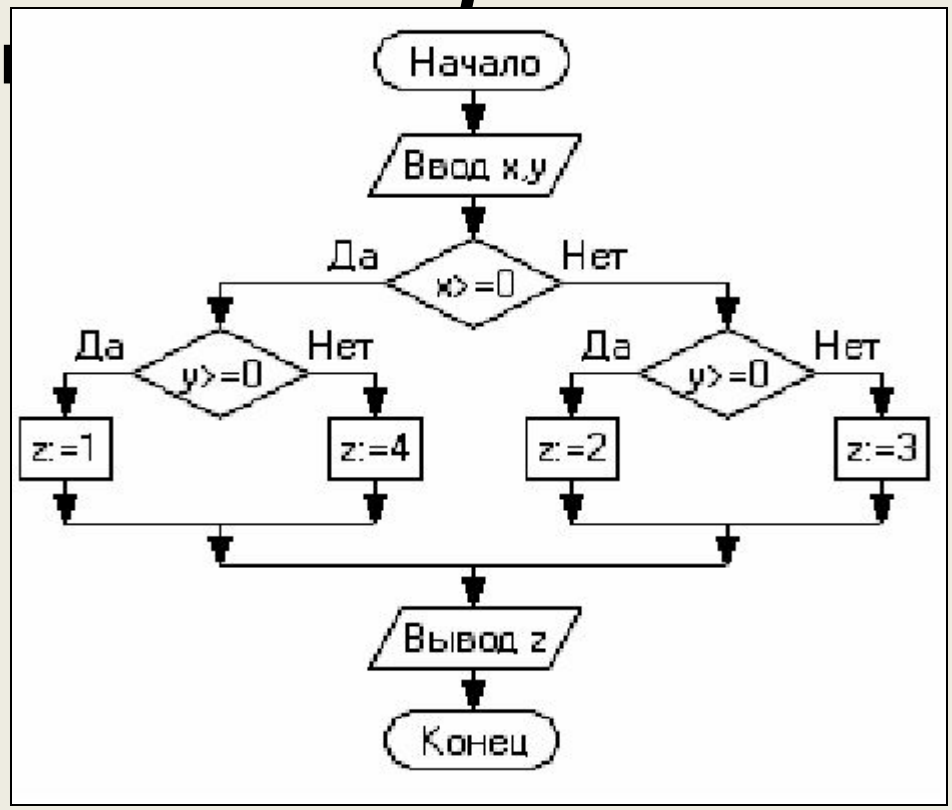
Различают неполную (ЕСЛИ-ТО) и полную (ЕСЛИ-ТО-ИНАЧЕ) структуру ветвления.





Пример разветвляющегося алгоритма

Алгорит

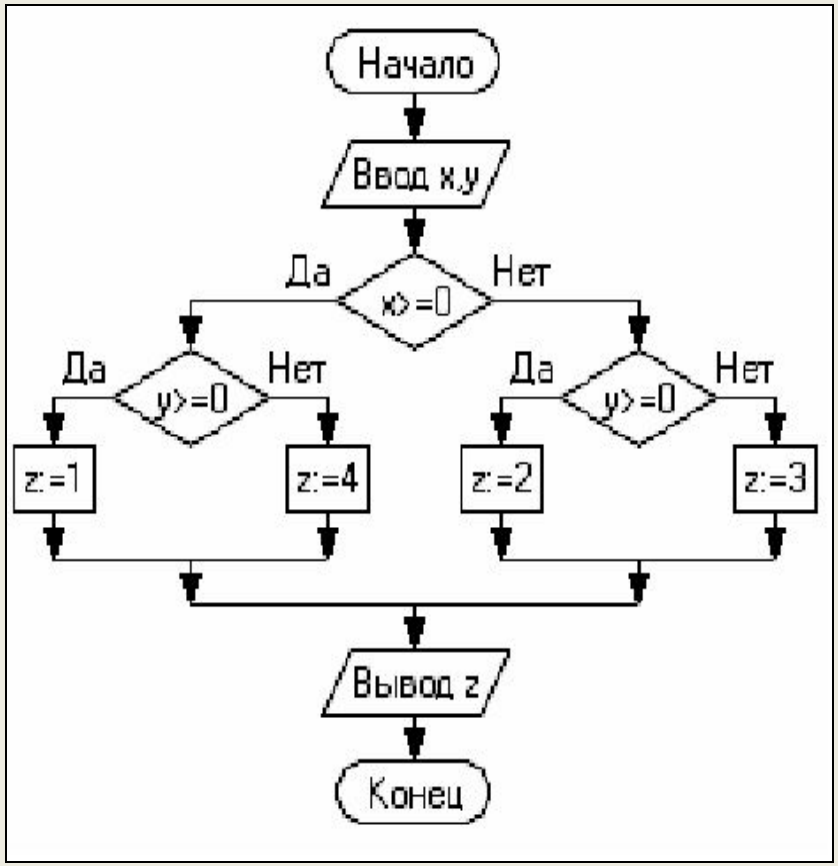


В результате выполнения алгоритма при исходных да? ых **x=-25**, **y=** **Z=2** ение переменной

Задание 1

Записать алгоритм в псевдокодах .

Блок-схема



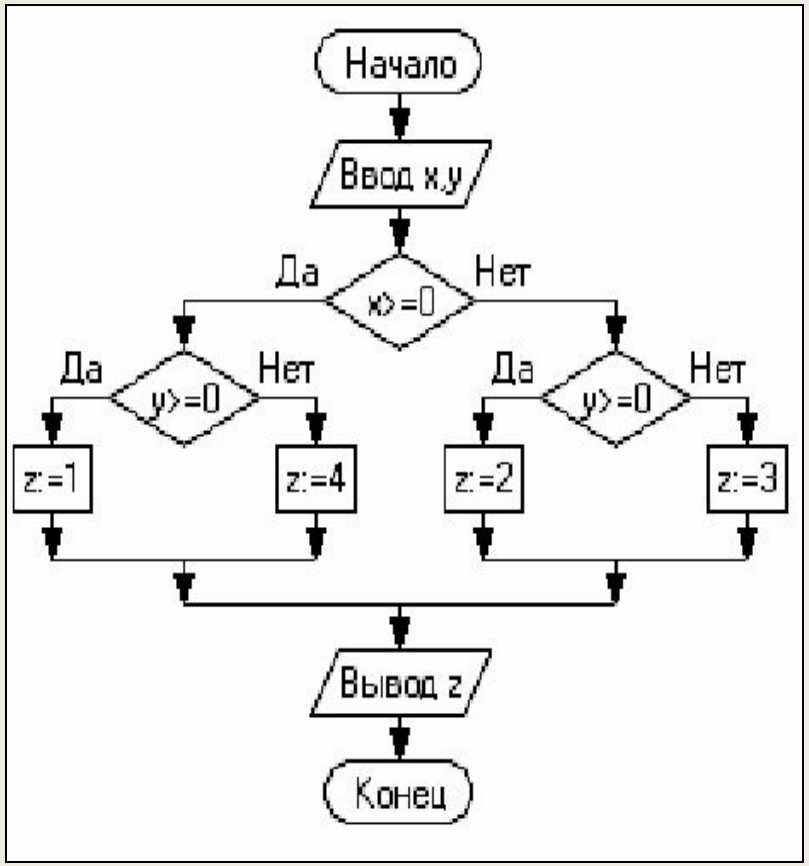
Псевдокод

```
Начало
Ввод  x, y
если x ≥ 0 то
   если y ≥ 0
   то z:=1
   иначе Z:=4
   все
иначе
   если y ≥ 0
   то z:=2
   иначе Z:=3
   все
все
Вывод  z
Конец
```

Задание 2 (*самостоятельно*)

Записать алгоритм на языке Турбо Паскаль.

Блок-схема



Турбо Паскаль

```
Program RVP;
Var
   x, y, z:real;
Begin
  writeln ('Введите числа x, y');
  read (x, y);
  if x>=0 then
     if y>=0 then z:=1 else z:=4
  else
      if y>=0 then z:=2 tlse z:=3;
  writeln ('z=',  z:6:2);
End.
```

- **Циклический алгоритм** – это алгоритм, который характеризуется многократным выполнением одних и тех же действий. Группа действий, повторяющихся в цикле, называется *телом цикла*. Числом повторений тела цикла управляет специальная переменная, называемая либо *управляющей* переменной, либо *параметром* цикла.

Широкое применение получили три типа циклов:

- цикл с предусловием;
- цикл с постусловием;
- цикл с параметром.

В блок-схемах каждому типу цикла соответствует определенная структура:

Цикл с предусловием	Цикл с постусловием	Цикл с параметром

Ответ

В цикле с предусловием условие проверяется **перед** выполнением тела цикла. Это означает, что тело цикла может не выполниться ни разу, если условие не выполняется сразу на первом шаге.

В цикле с постусловием условие записывается **после** выполнения тела цикла. Это означает, что тело цикла обязательно выполнится хотя бы один раз на первом шаге.

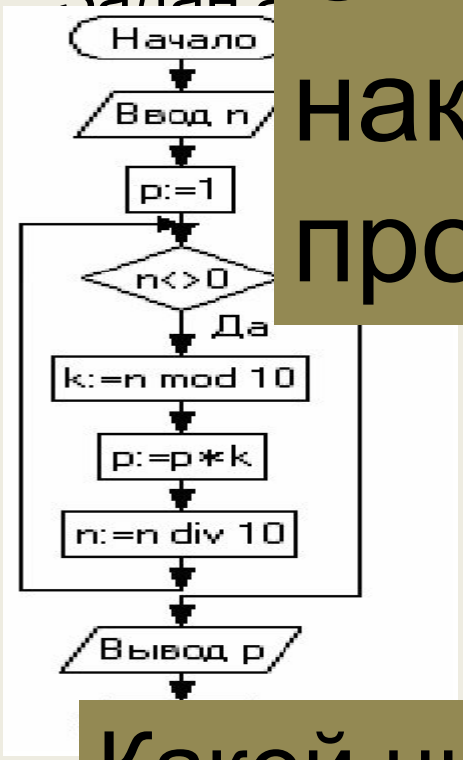
i – параметр цикла;

n1 – начальное значение параметра;

n2 – конечное значение

Пример циклического алгоритма с

Ответ: алгоритм накопления произведения




```
   p := p*k
   n := n div 10
кц
Вывод p
Конец
```

```
p:=1;
while n<>0 do
   begin
      k := n mod 10;
      p := p*k;
      n := n div 10;
   end;
```

Какой циклический алгоритм для переменной P реализуется в данной задаче ?

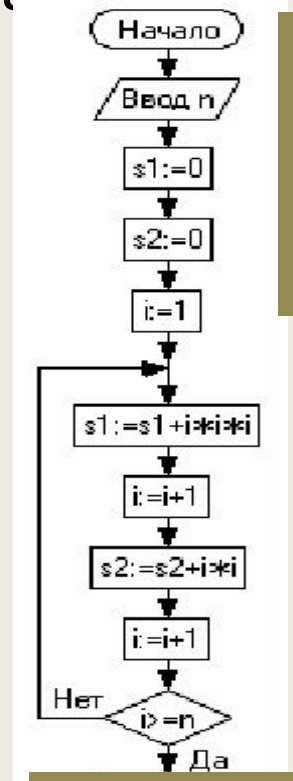
?

знач … о …

Ответ:

Пример циклического алгоритма с постусловием

Блок-схема



Псевдокод

```
нц
   S1 := S1 + i*i*i
   i := i + 1
   S2 := S2 + i*i
    i := i + 1
кц до i≥n
S := S1 + S2
Вывод S
```

Турбо-Паскаль

```
st;

ger;

e n');

S1 := 0;
S2 := 0;
i := 1;
repeat
   S1 := S1 + i*i*i;
   i := i +1;
   S2 := S2 + i*i;
   i := i +1;
until i>=n;
```

Задан алгоритм:

Ответ: алгоритм накопления суммы

Какой циклический алгоритм для переменных S1 и S1 реализуется в данной задаче ?

Пр s

Ответ

Пример циклического алгоритма с параметром

Ответ

Для переменной T - алгоритм накопления суммы.

Для переменной P - алгоритм накопления произведения.

Какие циклические алгоритмы для переменных T и P реализуются в данной задаче ?


```
Begin
        P := P * T;
    end;
writeln('P=', P);
End.
```

Чему равно значение **P** при **k=3**?

Произведени	**P = 162**	**кратных 3**