Использование алгоритма ветвления при решении физико-математических задач на Паскале

Повторение пройденного материала

- что называется алгоритмом?
- какие базовые типы алгоритмов?
- в чем отличие условных алгоритмов с полным и неполным ветвлением?
- какова структура программы на Паскале?
- какие типы переменных используются в языке Паскаль?
- какова структура ветвления на Паскале?
- сколько операторов разрешает синтаксис IF ставить после THEN и ELSE?
- как преодолеть эту трудность? Как записывается составной оператор?
- что такое форматируемый вывод и для чего он нужен?

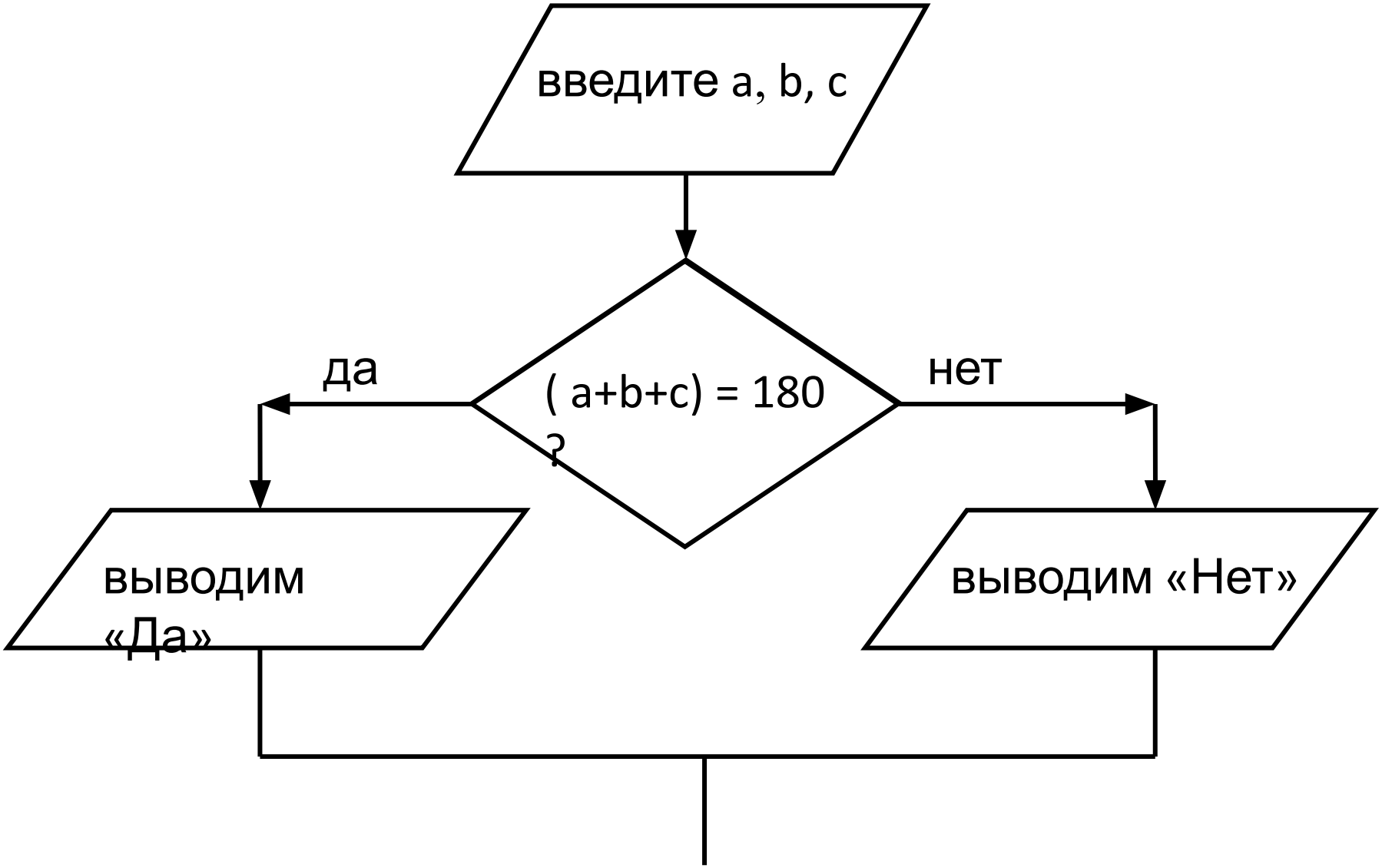
Задача №1

Требуется запросить значения трех углов и по введенным значениям определить, можно ли по этим углам построить треугольник?

Алгоритм решения:

- запрашиваем значения углов и вводим их с клавиатуры;
- анализируем сумму углов: если она равна 180^0, то сообщаем «треугольник существует», в противном случае выводим «треугольник не существует».

Блок-схема решения задачи №1



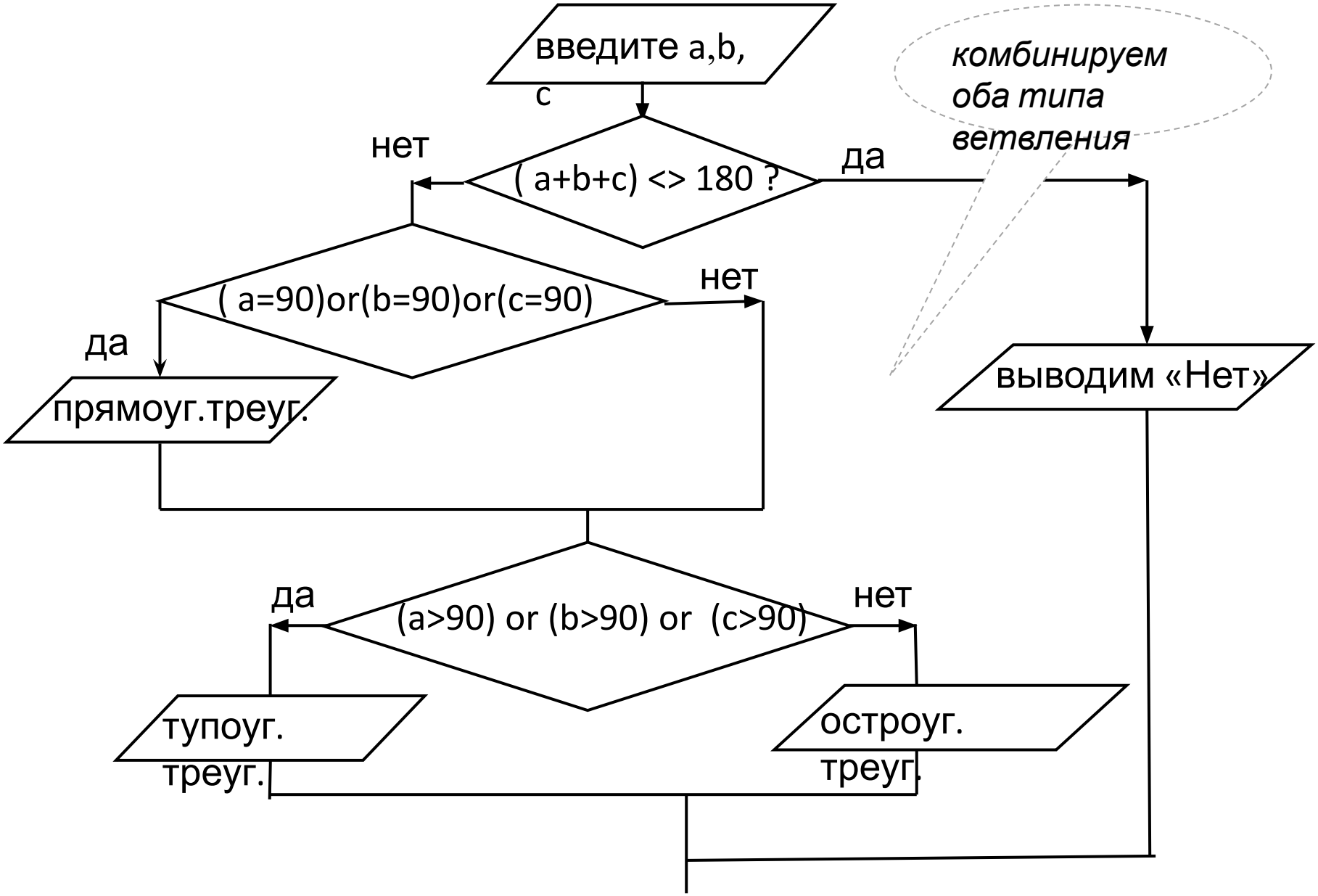
```
program treug-1;
var a,b,c :integer;
begin
writeln('введите три угла');
read (a,b,c);
if (a+b+c)=180 then write ('треугольник
существует')
    else write ('треугольник не существует');
end.
```

Задача №2

В задаче №1 в случае, если треугольник по введенным углам построить можно, уточнить, а какой именно треугольник будет построен – прямоугольный, остроугольный или тупоугольный?

При решении задачи будем использовать вложенные условия, а также сложные условия, связанные логическим «ИЛИ» (OR).

Блок-схема решения задачи №2



```
program treug_2;
var a,b,c :integer;
begin
writeln('введите три угла');
readln (a,b,c);
if (a+b+c)<>180 then write ('Нет')
    else
        if (a=90) or (b=90) or (c=90) then write
                ('прямоуг.треуг');
        if (a>90) or(b>90) or (c>90) then write
                ('тупоуг.треуг')
            else write ('остроуг.треуг');
end.
```

Задача №3

Запросить значения двух сопротивлений цепи R_1 и R_2,
а также вид соединения (1-последовательное или 2-параллельное). Посчитать и вывести значение общего сопротивления цепи R_0.

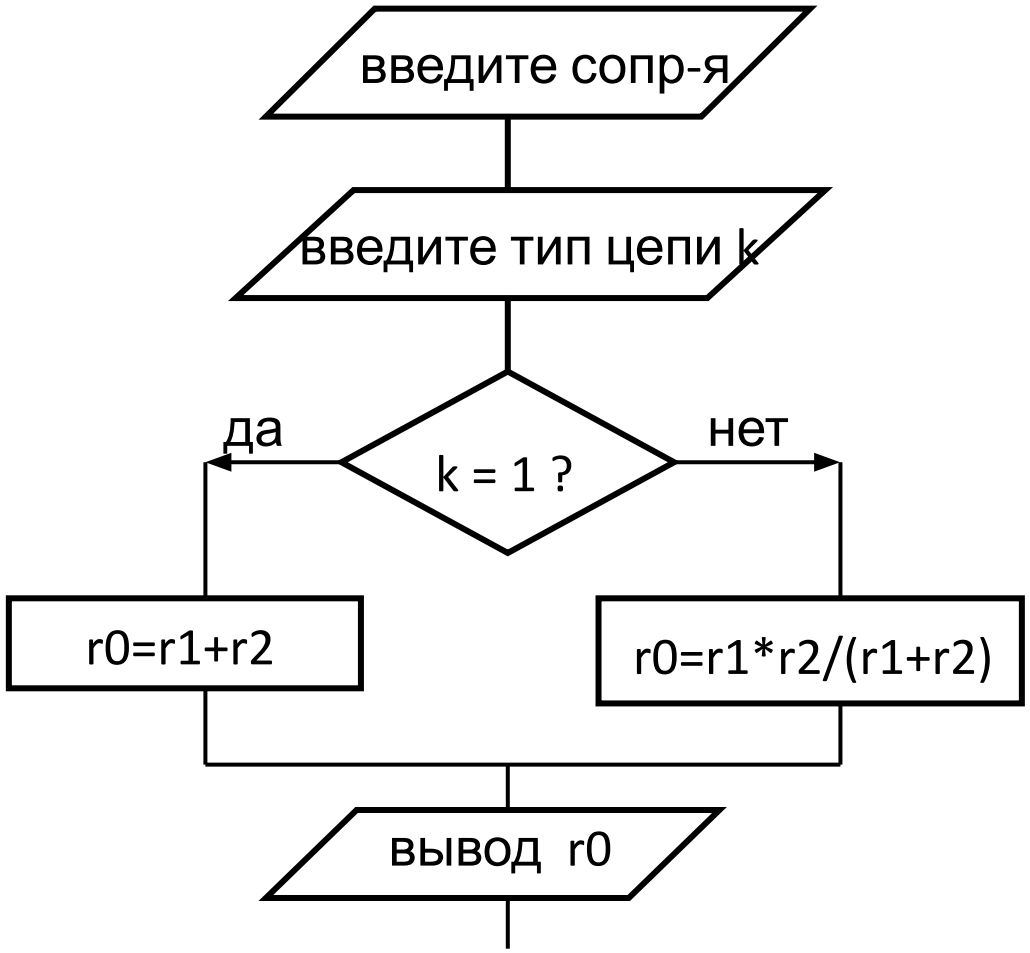
При последовательном соединении:

$$R_0 = R_1 + R_2$$

При параллельном соединении:

$$R_0 = R_1 * R_2 / (R_1 + R_2)$$

Блок-схема решения задачи №3



```
program   rezistor;
var   r1,r2,k  :integer;  r0 :real;
begin
writeln(‘введите два сопротивления);
readln (r1,r2);
writeln(‘введите тип соединения:
            1- последов., 2 – паралл.');
readln (k);
if k=1 then r0 = r1 + r2
    else
        r0 = r1 * r2 / (r1 + r2);
write (‘общее сопротивление цепи = ‘, r0 :4 :1);
end.
```

Задача №4

Программа запрашивает ввод двух координат некоей точки, анализирует введенные числа и выводит сообщение – какой четверти координатной плоскости принадлежит эта точка.

Алгоритм решения:

- запросить и ввести координаты точки;
- проанализировать четыре различных комбинации чисел;
- для каждого случая сделать вывод соответствующего сообщения.

Решение задачи будет более простым, если использовать неполное ветвление.

```
program    koordinat_plosk;
var a,b :integer;
begin
writeln('введите две координаты точки');
readln (a,b);
if (a > 0) and (b > 0) then write ('1 четверть') ;
if (a < 0) and (b > 0) then write ('2 четверть') ;
if (a < 0) and (b < 0) then write ('3 четверть') ;
if (a > 0) and (b < 0) then write ('4 четверть') ;
end.
```

Домашнее задание:

От станции к даче едет велосипедист. Проезжая мимо лодочной станции, он имел скорость $\mathbf{v_0}$ км/час. До дачи ему оставалось проехать **s** км. На это он потратил времени **t** час. Нужно ввести значения **s, t,** $\mathbf{v_0}$ и ответить, каким было движение велосипедиста – равномерным, равноускоренным или равно-замедленным.