

---

# **ТРАССИРОВКА**

Трассировка — пошаговое выполнение программы

---

---

# ТРАССИРОВКА

## Трассировочная таблица

Для трассировки удобно использовать трассировочную таблицу, в которой по вертикали откладывается шаг программы, а по горизонтали значение переменных и показания экрана

### Пример

```
Program chetnost;  
var i, x: integer;  
begin  
  for i:=1 to 4 do  
    begin  
      x:= i mod 2;  
      if x = 0  
        then writeln(i, ' – четное')  
        else writeln(i, ' – нечетное');  
    end;  
end.  
end.
```

Трассировочная таблица

<b>i</b>	<b>x</b>	<b>Экран</b>
1	1	1 – нечетное
2	0	2 – четное
3	1	3 – нечетное
4	0	4 – четное

# ТРАССИРОВКА

## Трассировочная таблица

Для поиска ошибок удобно визуализировать переменные на каждом шаге работы программы, Для этого в тело программы вставляют операторы вывода переменных на экран.

### Пример

```
Program sravnenie;  
var x, y: integer;  
begin  
  x:=2;  
  y:=10;  
  while x<y do  
    begin  
      x:=sqr(x);  
      y:=2*y;  
    end;  
  write(x:4,y:4);  
end.
```

Добавим в цикл  
оператор вывода



```
Program sravnenie;  
var x, y: integer;  
begin  
  x:=2;  
  y:=10;  
  while x<y do  
    begin  
      x:=sqr(x);  
      y:=2*y;  
      writeln('x=', x, ' y=', y);  
    end;  
  write(x:4,y:4);  
end.
```

### Трассировочная таблица

x	y	Экран
2	10	
4	20	x=4 y=20
16	40	x=16 y=40
256	80	x=256 y=80
		256 80

Таким образом мы получим аналог трассировочной таблицы на экране.

После того, как программа отлажена вставленный оператор вывода можно закомментировать или удалить.

# ТРАССИРОВКА

## Трассировочная таблица

Если значение какой-то переменной в текущем шаге программы не изменилось, то соответствующую клетку таблицы оставляют пустой.

### Пример

Трассировочная таблица

i	x	y	Экран
	2	11	
1	4	22	<i>i=1 4 22</i>
2	16		<i>i=2 16</i>
3	256		<i>i=3 256</i>
			256 22

```
Program uslovie;  
var i, x, y: integer;  
begin  
  x:=2;  
  y:=11;  
  for i:=1 to 3 do  
    if x mod 4 = 0  
    then  
      x:=sqr(x);  
    else  
      begin  
        x:=2*x;  
        y:=2*y;  
      end;  
      write(x,' ',y);  
    end.  
end.
```

Для удобства трассировки мы добавили вспомогательные операторы вывода, Но из-за этого нам пришлось добавить и дополнительные begin и end, Так как в теле операторов for и if стало больше одной команды.

```
Program uslovie;  
var i, x, y: integer;  
begin  
  x:=2;  
  y:=11;  
  for i:=1 to 3 do  
    begin  
      write('i=',i,' ');  
      if x mod 4 = 0 then  
        begin  
          x:=sqr(x);  
          writeln( x, ' ');  
        end  
      else  
        begin  
          x:=2*x;  
          y:=2*y;  
          writeln( x, ' ', y);  
        end;  
      end;  
      write(x,' ',y);  
    end.  
end.
```