

Анищенко Юлия Михайловна  
учитель информатики  
Государственное бюджетное образовательное учреждение  
Лицей №64  
г. Санкт-Петербург

## ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ ВЕТВЛЕНИЯ НА ПАСКАЛЕ. ЛОГИЧЕСКИЕ ВЫРАЖЕНИЯ

План урока

**Используемый УМК:** Семакин И.Г.

**Класс:** 10

**Используется:** урок - получение нового знания с применением технологий сетевого взаимодействия учителяиучащихся.

**Тема курса:** «Программирование»

**Цели урока:**

*Обучающие:*

- овладение учащимися инструментарием “логические операции и выражения” для решения типовых задач программирования ветвлений на Паскале;

*Развивающие:*

- развитие навыков групповой самоорганизации, умения вести диалог;
- развитие абстрактного и логического мышления;
- развитие ответственного отношения к учебе, аккуратность;
- умение выделять из класса задач задачи, решаемые с использованием конкретного алгоритма.

*Воспитательные:*

- воспитание ценностей личного отношения к изучаемым знаниям;
- воспитание в учениках уверенности в своих силах средствами

урока информатики

**Задачи урока:**

*Образовательные:*

- повторить: структура программы на языке Паскаль, оператор ветвления;
- изучить способы решения типовой задачи (вхождение точки с указанными координатами в заштрихованную область) по теме “Организация ветвления на языке TurboPascal”;
- формировать умения и навыки по работе с логическими операциями в разветвляющихся конструкциях в программах, написанных на языке TurboPascal;

*Развивающие:*

- развивать умение вести учебный диалог, отстаивать свою точку зрения при решении задач;
- развивать логическое мышление через решение типовых задач на использование оператора ветвления;
- развивать умение выделять из класса задач задачи, решаемые с использованием алгоритма с ветвлением.

*Воспитательные:*

- продемонстрировать ценность личного отношения к изучаемым знаниям при решении целого ряда задач;
- воспитывать в учениках уверенность в своих силах при решении задач.

**Продолжительность урока:** 45 минут.

**Техническое и программное обеспечение:**

- проектор;
- правильно и полностью инсталлированные программы Windows;
- язык программирования Pascal;
- презентация урока;

- локальная сеть, Интернет;

**Необходимые знания и умения учащихся к этому моменту:**

- понимание структуры программы на языке Pascal, знание простых типов данных в языке программирования Pascal,
- способность использовать начальные знания программирования для составления линейных алгоритмов;
- знание основ математической логики и умение составлять логические выражения (логические операции, приоритет логических операций, логические выражения);
- умение пользоваться локальной сетью в компьютерном классе.

**Информационная карта урока:**

Этапы урока	Вре мя мин.	Деятельность учителя	Деятельность учащихся	ЦОР и ЭОР
Организационны й момент.	2	Приветствие учащихся, психологический настрой	Учащиеся готовятся к уроку. Настраиваются на работу и приветствуют учителя	
Объявление темы занятия. Стадия вызова на основе приема решения поставленной задачи.	7	Ставится задача. Требуется по введенным координатам определить попадает ли точка в указанную область. Активизирует деятельность учащихся, предлагает заполнить тестовую форму.	Обсуждают условие и пытаются решить задачу известными способами. Воспроизводят алгоритм решения известными им способами. Возможно при помощи сервиса <a href="https://www.lucidchart.com">https://www.lucidchart.co</a> <a href="https://www.lucidchart.com">m</a> они строят блок-схему алгоритма (но можно и в тетрадах)	1.Сервис гугл рисования диаграмм блок- схем <a href="https://www.lucidchart.com">https://www.lucidchart.com</a> 2.Тренировоч ные упражнения (задание №1).
Ознакомление учащихся с новыми понятиями Активное слушание	7	Вводятся новые понятия урока. Используется прием эффективного слушания. По материалам учащимся предлагается составить ментальную карту (конспект урока) средствами <a href="http://www.mindmeister.com/ru">http://www.mindmeister.co</a>	На основе известных им логических операций учащиеся составляют интеллект-карту по объяснениям учителя средствами <a href="http://www.mindmeister.com/ru">http://www.mindmeister.c</a> <a href="http://www.mindmeister.com/ru">om/ru</a>	Презентация к уроку <a href="#">Сервис</a> <a href="#">создания</a> <a href="#">интеллект-карт</a>

		<a href="#">m/ru</a>		
Актуализация знаний, формирование новых умений и навыков. Решение типовых задач	8	Учитель выдает карточки, на которых написаны задачи. Предлагается решить вторую задачу на выявление принадлежности точки заданной области (интел. карта процесса и справка по функции)	Решают задачу, на практике применяя полученные знания, активизируя уже имеющиеся.	Тренировочные упражнения (Задание №2)
Формирование новых умений и навыков. Выявление проблем, подведение предварительных итогов	8	Разбирает решение. Организует мозговой штурм по выявлению проблемных вопросов, фиксирует все выводы на доске, поддерживает доброжелательную атмосферу в классе (пограничные точки, точки лежащие на границе области). Смотрит все ли учащиеся смогли решить задачу	Обсуждают с учителями проблемы, которые могли возникнуть в процессе решения задачи (пограничные точки, вывод ровно одной буквы области и т.д.) Первичный контроль умений и навыков полученных в ходе урока.	<a href="#">Сервис создания интеллект-карт</a>
Рефлексия	4	Подведение итогов мозгового штурма. Выявление наиболее ценных и важных результатов. Выявление метапредметных знаний (алгебра и начала анализа, информатика)	Фиксируются результаты в интеллект-карте (конспекте).	
Закрепление, систематизация	5	Предлагает решить задачу 3, используя полученные ранее знания. Для способных учеников предлагается задача 4,5, в которой не просто требуется использовать новые знания, но и придумать собственную схему построения решения.	Решают задачу, на практике применяя полученные знания, активизируя уже имеющиеся.	Тренировочные упражнения (задачи 3,4)
Задание на дом	2	Учащимся предлагается решить последние две задачи из тренировочного листа	Самостоятельные упражнения	Тренировочные упражнения

**Планируемые образовательные результаты:**

Сформулированные цели реализуются через достижение образовательных результатов

***Предметные:***

- умение использовать логические значения, операции и выражения с ними;
- умение формально выполнять алгоритмы, описанные с использованием конструкций ветвления (условные операторы);
- умение создавать и выполнять программы для решения несложных алгоритмических задач в выбранной среде программирования с использованием оператора ветвления;
- умение использовать готовые прикладные компьютерные программы и сервисы, умение работать с описаниями программ и сервисами;
- навыки выбора способа представления данных в зависимости от поставленной задачи.

***Личностные:***

- развитие логического и абстрактного мышления.
- формирование ответственного отношения к учению, готовности и способности обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию;
- развитие осознанного и ответственного отношения к собственным поступкам;
- формирование коммуникативной компетентности в процессе образовательной, учебно-исследовательской деятельности.

***Метапредметные результаты:***

- умение самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в познавательной деятельности;

- владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;
- умение определять понятия, создавать обобщения, устанавливать аналогии, устанавливать причинно-следственные связи, строить логическое рассуждение, умозаключение и делать выводы;
- умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебных и познавательных задач;
- формирование и развитие компетентности в области использования информационно-коммуникационных технологий (далее ИКТ-компетенции).

## Тренировочные задания

**Задача 1** Требовалось написать программу, которая определяет, лежит ли точка  $A(x_0, y_0)$  внутри треугольной области, ограниченной осями координат и прямой  $y=2-x$  («внутри» понимается в строгом смысле, т.е. случай, когда точка  $A$  лежит на границе области, недопустим). В результате программа должна выводить соответствующее текстовое сообщение. Программист сделал в программе ошибки.

```

var x0, y0, y: real;
begin
  readln (x0, y0);
  if (x0 < 2)then begin
    if (x0 > 0)then begin
      y = 2 - x0;
      if (y0 < y) then
        writeln ('точка лежит внутри области')
      elsewritein ('точка не лежит внутри области');
    end
    elsewriteln ('точка не лежит внутри области');
  end
  elsewriteln ('точка не лежит внутри области');
end.

```

Последовательно выполните задания:

1. Приведите пример таких чисел  $x_0$  и  $y_0$ , при которых программа неверно решает поставленную задачу.
2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы (можно указать любой способ доработки исходной программы).
3. Укажите, как можно доработать программу, чтобы вместо вложенных операторов **IF** она содержала логическую операцию **AND**.

**Задача 2** Требовалось написать программу, которая определяет, лежит ли точка  $A(x,y)$  внутри некоторого кольца. Центр кольца находится в начале координат. Для кольца заданы внутренний и внешний радиусы  $r1, r2$ ; известно, что  $r1$  отлично от  $r2$ , но неизвестно,  $r1 > r2$  или  $r2 > r1$ . В том случае, когда точка  $A$  лежит внутри кольца, программа должна выводить соответствующее сообщение, в противном случае никакой выходной информации не выдается. Программист сделал в программе ошибки.

```

var x, y, r1,r2:real;
h: real;
begin
readln (x, y, r1, r2);
h := sqrt (x*x + y*y);
if (h < r1) AND (h > r2) then
writeln ('точка A лежит внутри кольца');
end.

```

Последовательно выполните задания:

1.Приведите пример таких чисел  $x,y, r1, r2$ , при которых программа неверно решает поставленную задачу.

2.Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).

3.Укажите, как можно доработать программу, чтобы она не содержала логических операций **AND** или **OR**.

Примечание: для обозначения расстояния от точки  $A$  до начала координат используется вспомогательная переменная  $h$ .

**Задача 3** Требовалось написать программу, которая определяет, имеется ли среди введенных с клавиатуры положительных целых чисел  $a$  и  $b$

хотя бы одно четное. Была написана следующая программа:

```

var a, b: integer;
begin
  readln(a, b);
  a := a mod 2;
  if a > 0 then b := b mod 2;
  if b > 0 then
    writeln ('четныхчиселнет')
  elsewriteln ('четное число есть');
end.

```

Известно, что программа написана с ошибками. Последовательно выполните три задания:

1. приведите пример таких чисел **a**, **b**, при которых программа неверно решает поставленную задачу;
2. укажите, как, по вашему мнению, нужно доработать программу;
3. укажите, как можно доработать программу, чтобы она вместо вложенных операторов **IF** содержала логическую операцию **OR**.

**Задача 4** Требовалось написать программу, которая определяет, можно ли построить треугольник из отрезков с длинами **x**, **y**, **z**. Программист сделал в программе ошибки.

```

var x, y, z: real;
begin
  readln (x, y, z);
  if (x + y > z) then
    begin
      if (x + z > y) then
        if (y + z > x) then

```

```
writeln('треугольник построить можно');
end
elsewriteln('треугольник построить нельзя');
end
```

Последовательно выполните задания:

1. Приведите пример таких чисел  $x$ ,  $y$ ,  $z$ , при которых программа неверно решает поставленную задачу.
2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы.
3. Укажите, как можно доработать программу, чтобы она вместо вложенных операторов **IF** содержала логическую операцию **AND**.

**Задача 5** Требовалось написать программу, которая определяет, лежит ли точка  $A(x,y)$  внутри некоторого кольца. Центр кольца находится в начале координат. Для кольца заданы внутренний и внешний радиусы  $r_1$ ,  $r_2$ ; известно, что  $r_1$  отлично от  $r_2$ , но неизвестно,  $r_1 > r_2$  или  $r_2 > r_1$ . В том случае, когда точка  $A$  лежит внутри кольца, программа должна выводить соответствующее сообщение, в противном случае никакой выходной информации не выдается. Программист сделал в программе ошибки.

```
var x, y, r1, r2: real;
h: real;
begin
readln (x, y, r1, r2);
h := sqrt (x*x + y*y);
if (h < r1) AND (h > r2) then
writeln ('точка A лежит внутри кольца');
end.
```

Последовательно выполните задания:

1. Приведите пример таких чисел  $x, y, r_1, r_2$ , при которых

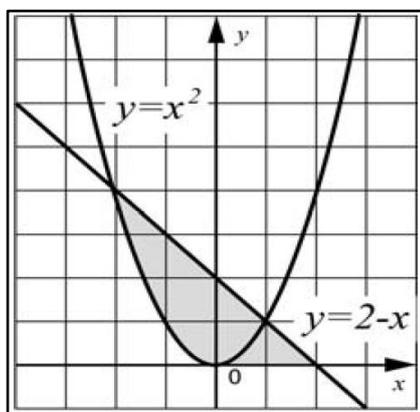
программа неверно решает поставленную задачу.

2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).

3. Укажите, как можно доработать программу, чтобы она не содержала логических операций **AND** или **OR**.

Примечание: для обозначения расстояния от точки **A** до начала координат используется вспомогательная переменная **h**.

**Задача 6** Требовалось написать программу, при выполнении которой с клавиатуры считываются координаты точки на плоскости ( $x, y$  – действительные числа) и определяется принадлежность этой точки заданной закрашенной области (включая границы). Программист торопился и написал программу неправильно.



```

var x,y: real;
begin
  readln(x,y);
  if y<=2-x then
    if y>=0 then
      if y>=x*x then
        write('принадлежит')
      else
        write('не принадлежит')
    end.

```

Последовательно выполните следующее.

1. Перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (A, B, C, D, E, F, G и H). Точки, лежащие на границах областей, отдельно не рассматривать.

Область	$y \leq 2-x$ ?	$y >= 0$ ?	$y >= x*x$ ?	вывод	верно?
A					
B					
C					
D					
E					
F					
G					
H					

В столбцах условий укажите "да", если условие выполнится, "нет" если условие не выполнится, "—" (прочерк), если условие не будет проверяться, «не изв.», если программа ведет себя по-разному для разных значений, принадлежащих данной области. В столбце "Программа выведет" укажите, что программа выведет на экран. Если программа ничего не выводит, напишите "—" (прочерк). Если для разных значений, принадлежащих области, будут выведены разные тексты, напишите «не изв.». В последнем столбце укажите "да" или "нет".

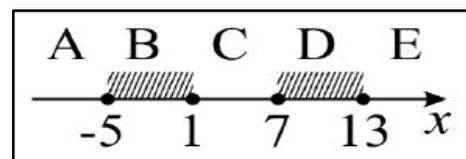
2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы.

**Задача 7** Требовалось написать программу, при выполнении которой с клавиатуры считывается координата точки на прямой ( $x$  – действительное число) и определяется принадлежность этой точки одному из выделенных отрезков B и D (включая границы). Программист торопился и написал программу неправильно.

```

var x: real;
begin
  readln(x) ;
  if x>1 then

```



```

if x>=7 then
if x>13 then
  write('непринадлежит')
else
  write('принадлежит')
end.

```

Последовательно выполните следующее.

1. Перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (А, В, С, D и E). Границы (точки -5, 1, 7 и 13) принадлежат заштрихованным областям.

Область	Условие 1 (x > 1)	Условие 2 (x >= 7)	Условие 3 (x > 13)	Программа выведет	Область обрабатывается верно
А					
В					
С					
D					
E					

В столбцах условий укажите «да», если условие выполнится, «нет», если условие не выполнится, «—» (прочерк), если условие не будет проверяться, «не изв.», если программа ведет себя по-разному для разных значений, принадлежащих данной области. В столбце «Программа выведет» укажите, что программа выведет на экран. Если программа ничего не выводит, напишите «—» (прочерк). Если для разных значений, принадлежащих области, будут выведены разные тексты, напишите «не изв.». В последнем столбце укажите «Да» или «Нет».

2. Укажите, как нужно доработать программу, чтобы не было случаев её неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)