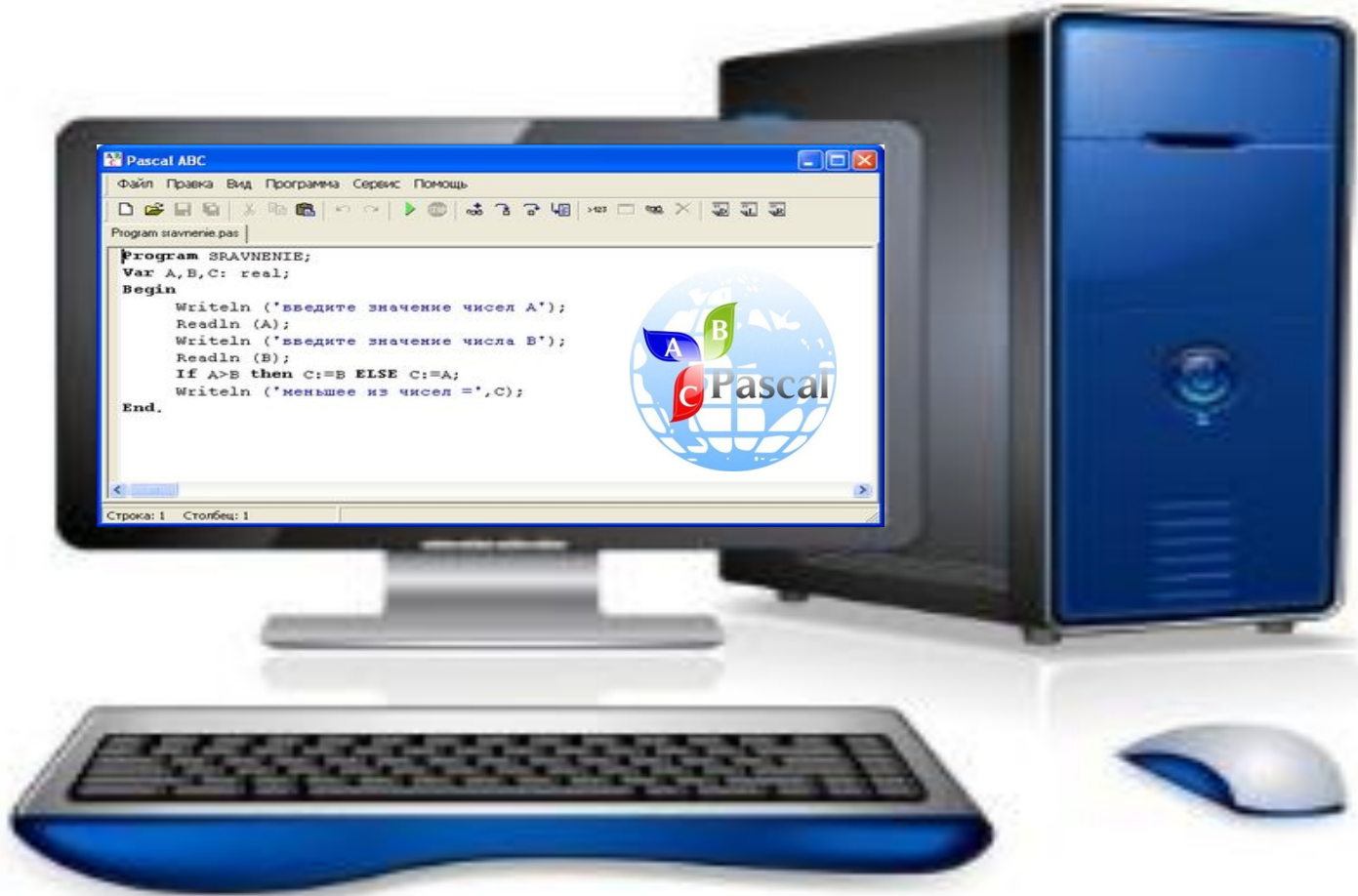


Тема урока:

# Программирование ветвлений





# Оператор ветвления на Паскале

---

**Ветвление** – это алгоритмическая конструкция, в которой в зависимости от условия выполняется та или иная последовательность действий.

В языке Паскаль имеется **оператор ветвления**.

Другое его название - **условный**

# Формат полного оператора ветвления следующий:

---

if <логическое выражение> then  
    <оператор1>  
else <оператор2>



Здесь:  
if - "если",  
then - "то",  
else - "иначе".

# Программирование полного или неполного ветвления

Сравните запись алгоритма БИД1 на алгоритмическом языке с соответствующей программой на языке программирования Паскаль.

```
алг БИД1
вещ А, В, С
нач ввод А, В
  если А>В
  то С:=А
  иначе С:=В
кв
вывод С
кон
```

Полное ветвление

```
Program BID1;
var A, B, C : real;
begin readln (A, B);
  if A>B
  then C:=A
  else C:=B;
  writeln (C)
end.
```

Простой формой логического выражения является **оператор отношения**.

**В Паскале допускаются все виды отношений (ниже указаны их знаки):**

**< (меньше);**

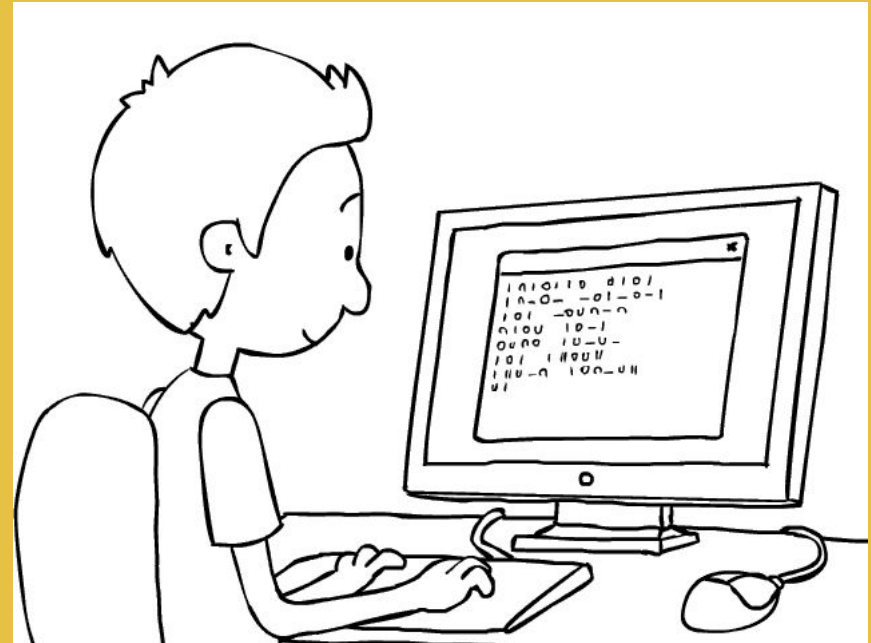
**> (больше);**

**= (равно);**

**<> (не равно);**

**<= (больше или равно);**

**>= (меньше или равно).**



# А теперь запрограммируем на Паскале алгоритм БИД, в котором использовано ветвление.

Программирование на алгоритмическом языке: неполное ветвление

```
алг БИД2
вещ A, B, C
нач ввод A, B
  C:=A
  если B > A
  то C:=B
  кв
  вывод C
кон
```

Неполное ветвление

```
Program BID2;
var A, B, C : real;
begin readln(A, B);
      C:=A;
      if B>A
      then C:=B;
      write(C)
end.
```

Опять все очень похоже. Ветвь else в операторе ветвления может

# Программирование вложенных ветвлений

Запишем на Паскале программу определения большего из трех чисел, используя блок-схему.

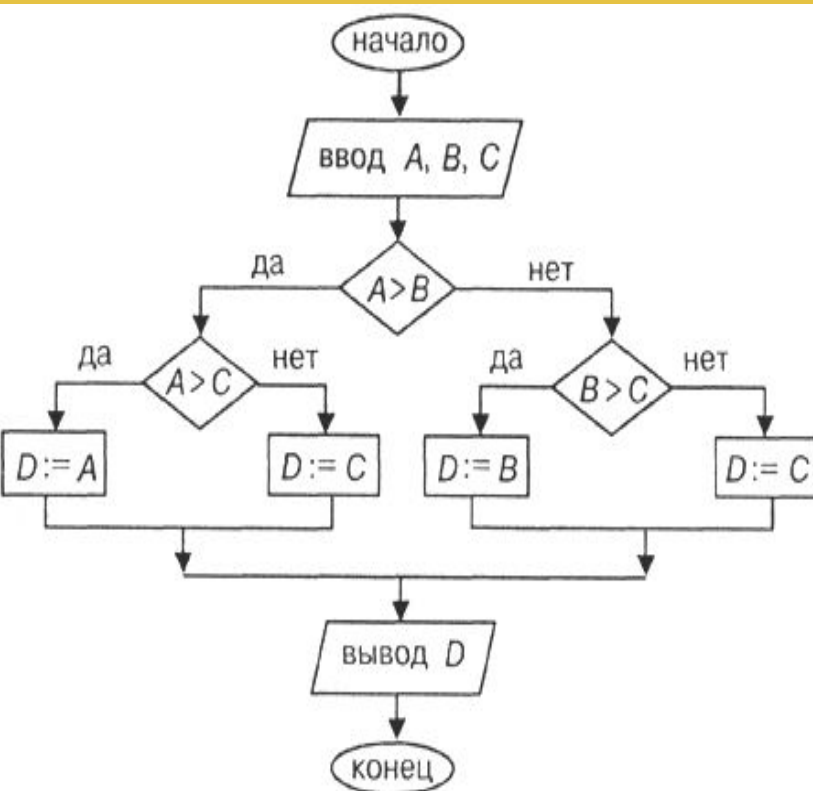
**ветвлений**

Составим программу на языке программирования Паскаль:

```
Program BIT2;  
var A, B, C, D: real;  
begin readln(A, B, C);  
  if A>B  
    then if A>C then D:=A else D:=B  
    else if B>C then D:=B else D:=C;  
  writeln(D)  
end.
```

2 вариант решения задачи

Перед else не ставится знак



Структура алгоритма – выбор  
большего из 3 чисел

# Составим программу упорядочения значений двух переменных.

**алг СОРТИРОВКА**

**вещ X, Y, C**

**нач ввод X, Y**

**если  $X > Y$**

**то  $C := X$**

**$X := Y$**

**$Y := C$**

**кв**

**вывод X, Y**

**кон**

**Program SORTING;**

**var X, Y, C : real;**

**begin readln (X, Y) ;**

**if  $X > Y$**

**then begin C := X;**

**$X := Y$ ;**

**$Y := C$**

**end;**

**write (X,Y)**

**end.**

Этот пример иллюстрирует следующее правило Паскаля: *если на какой-то из ветвей оператора ветвления находится несколько последовательных операторов, то их нужно записывать между служебными словами `begin` и `end`.*

Конструкция такого вида:

```
begin <Последовательность операторов>  
      end
```

называется **составным оператором**.

Следовательно, в описанной выше общей форме ветвления <оператор1> и <оператор2> могут быть

Вспомним **понятия алгебры логики**, прежде чем приступить к рассмотрению третьего варианта решения задачи (найти большее из трех чисел):

**КОНЬЮНКЦИЯ** — логическая операция, по своему применению максимально приближенная к союзу «и».

Синонимы: **логическое «И»**, **логическое умножение**, иногда просто «И»

**ДИЗЪЮНКЦИЯ** — логическая информация, по своему применению максимально приближённая к союзу «или» в смысле «или то, или это, или оба сразу»

Синонимы: **логическое сложение**, **логическое ИЛИ**, **включающее ИЛИ**; иногда просто «ИЛИ»

**ИМПЛИКАЦИЯ** — бинарная логическая связка, по своему применению приближенная к союзам «**если... то...**».

**ОТРИЦАНИЕ** - операция над суждениями, результатом которой является суждение (в известном смысле) «противоположное» исходному.

Синоним: **логическое "НЕ"**

# Логические

**операции**  
Составить еще один, третий вариант программы определения большего числа из трех.

---

В Паскале  
присутствуют все три  
основные логические  
операции:

**and** - И (конъюнкция),  
**or** - ИЛИ (дизъюнкция),  
**not** - НЕ (отрицание).

```
Program BIT3;  
var A,B,C,D: real;  
begin readln(A,B,C);  
    if (A>=B) and (A>=C) then D:=A;  
    if (B>=A) and (B>=C) then D:=B;  
    if (C>=A) and (C>=B) then D:=C;  
    writeln(D)  
end.
```

Операция **and** – это **логическое умножение** или конъюнкция.  
Ее результат - "истина", если значения обоих операндов - "истина".  
Очевидно, что если  $A \geq B$  и  $A \geq C$ , то A имеет наибольшее значение

и т. д.

# Сложные логические

**Выражения**  
Связываемые логическими операциями, заключаются в скобки (Так надо делать всегда!).

---

Пример, требуется определить, есть ли среди чисел A, B, C хотя бы одно отрицательное.

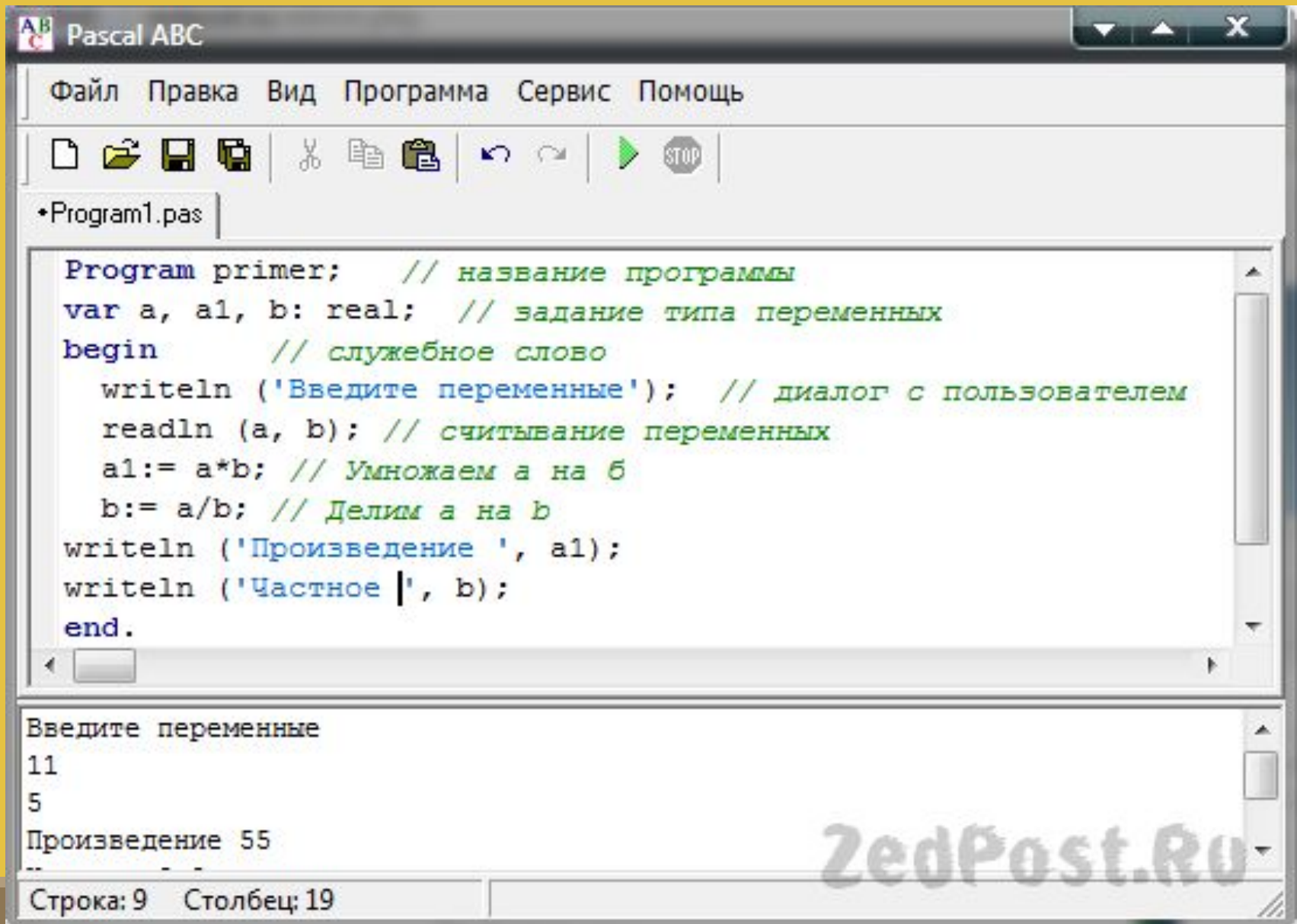
Эту задачу решает следующий оператор ветвления:

```
if (A < 0)or(B < 0)or(C < 0)  
  then write ( 'YES ' ) else write( 'NO ' );
```

Выражение, истинное для отрицательного числа, может быть записано еще и так:

```
not ( A >= 0)
```

# ПРАКТИЧЕСКАЯ РАБОТА



The screenshot shows the Pascal ABC IDE interface. The title bar reads "Pascal ABC". The menu bar includes "Файл", "Правка", "Вид", "Программа", "Сервис", and "Помощь". The toolbar contains icons for file operations (new, open, save, save as), editing (cut, copy, paste), and execution (run, stop). The file name "Program1.pas" is shown in the editor's title. The code in the editor is as follows:

```
Program primer;    // название программы
var a, a1, b: real; // задание типа переменных
begin             // служебное слово
  writeln ('Введите переменные'); // диалог с пользователем
  readln (a, b); // считывание переменных
  a1:= a*b; // Умножаем a на b
  b:= a/b; // Делим a на b
  writeln ('Произведение ', a1);
  writeln ('Частное |', b);
end.
```

The output window at the bottom shows the following text:

```
Введите переменные
11
5
Произведение 55
..
```

The status bar at the bottom indicates "Строка: 9" and "Столбец: 19". A watermark "ZedPost.Ru" is visible in the bottom right corner.

# ПРАКТИЧЕСКАЯ РАБОТА

Скриншот программы PascalABC.NET, демонстрирующей код программы-калькулятора и ее выполнение.

**Код программы (Program1.pas):**

```
var a, b, sum, raz, pro, c: integer;
del: real;

begin
  readln (a, b); // считываем два числа
  sum:=a+b; // складываем числа
  raz:= a-b; // вычитаем
  pro:=a*b; // умножаем
  del:=a/b; // выполняем простое деление
  c:=a div b; //делаем целочисленное деление
  a:= a mod b; // находим остаток, при этом меняем значение a
  writeln ('сумма - ', sum); // выводим сумму
  writeln ('разность - ', raz); // выводим разность
  writeln ('произведение - ', pro); // выводим произведение
  writeln ('простое деление - ', del); // выводим обычное деление
  writeln ('целочисленное деление - ', c); // выводим результат целочисленного деления
  writeln ('остаток - ', a); // выводим остаток
end.
```

**Выходные данные (Окно вывода):**

```
5
6
сумма - 11
разность - -1
произведение - 30
простое деление - 0.8333333333333333
целочисленное деление - 0
остаток - 5
```

**Текст задания:**

Написать программу - простейший калькулятор, который выполняет с любыми введенными двумя числами все действия сразу.

**Логотип PascalABC.NET:**

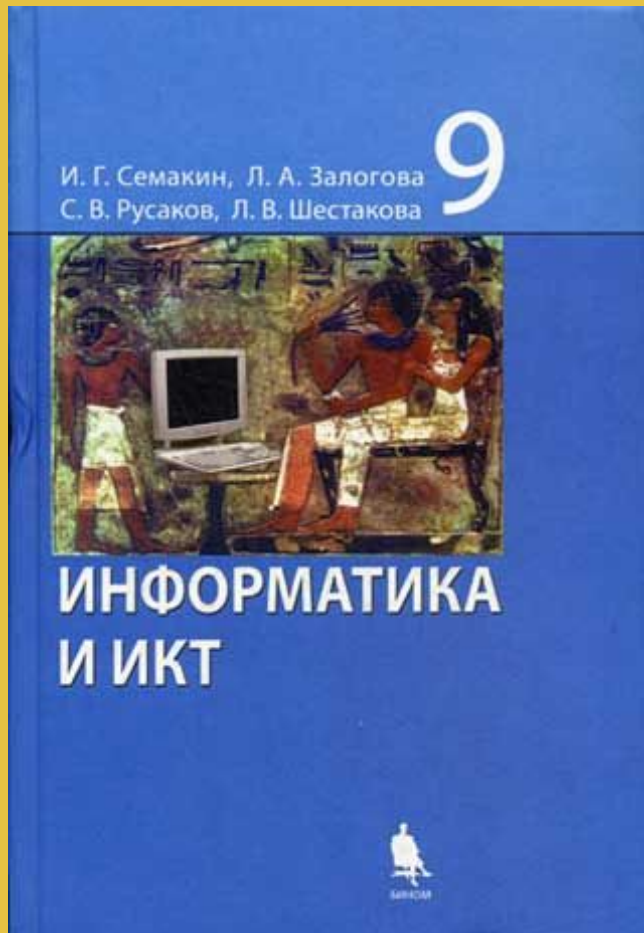


Строка 18    Столбец 5

Компиляция прошла успешно (18 строк)

# Домашнее задание:

---



- \* По учебнику § 37 + записи по тетради;
- \* Ответить на вопросы устно с. 207 №1-4.
- \* В тетради с. 207 задание №5, 6;
- \* Выучить по тетради.

# Использованные источники информации:

---

- <http://zedpost.ru/26/>
- [http://russkie-programmy.ru/load/obrazovanie\\_i\\_nauka/programma\\_pascal\\_abc\\_na\\_russkom\\_paskal\\_abc/7-1-0-118](http://russkie-programmy.ru/load/obrazovanie_i_nauka/programma_pascal_abc_na_russkom_paskal_abc/7-1-0-118)
- <http://www.5byte.ru/9/0017.php>
- [http://ru.wikipedia.org/wiki/%D0%9B%D0%BE%D0%B3%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F\\_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D1%8F](http://ru.wikipedia.org/wiki/%D0%9B%D0%BE%D0%B3%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D1%8F)
- **Семакин И.Г. Информатика и ИКТ: учебник для 9 класса / И.Г. Семакин, Л.А. Залогова, С.В. Русаков, Л.В. Шестакова. – 5-е изд. – М.: БИНОМ. Лаборатория знаний, 2012.**

**СПАСИБО ЗА ВНИМАНИЕ!**