The background is a solid blue color with various abstract elements. In the top right, there are three concentric white circles with a central white dot. A white line with four white circular nodes curves from the left side towards the center. The background is filled with faint, light blue binary code (0s and 1s). In the bottom right corner, there is a stylized globe with a grid pattern. In the bottom left corner, there is a faint map of the world. The text 'deposit photos' is repeated several times in a light blue, sans-serif font across the background.

# Решение задач с использованием условного оператора

# ТЕСТ

## на тему: «Программирование линейных алгоритмов»

1. По определению стандартного Паскаля, структура программы состоит из:

- а) заголовка программы;
- б) разделов;
- в) тела программы (блока);
- г) операторов.



# ТЕСТ

## на тему: «Программирование линейных алгоритмов»

2. Раздел переменных в блоке программы обозначается словом:

- а) Type ;
- б) Program ;
- в) Var;
- г) Begin.

# ТЕСТ

## на тему: «Программирование линейных алгоритмов»

3. Раздел операторов в блоке программы обозначается словом:

- а) Type ;
- б) Program;
- в) Var;
- г) Begin.



# ТЕСТ

## на тему: «Программирование линейных алгоритмов»

4. Формат оператора ввода имеет следующий вид:

- а)  $\langle \text{переменная} \rangle := \langle \text{выражение} \rangle;$
- б) Read ( $\langle \text{список ввода} \rangle$ );
- в) Write ( $\langle \text{список ввода} \rangle$ ).

# ТЕСТ

## на тему: «Программирование линейных алгоритмов»

5. Формат оператора присваивания имеет следующий вид:

- а)  $\langle \text{переменная} \rangle := \langle \text{выражение} \rangle;$
- б) Read ( $\langle \text{список ввода} \rangle$ );
- в) Write ( $\langle \text{список ввода} \rangle$ ).



# ТЕСТ

## на тему: «Программирование линейных алгоритмов»

6. Формат оператора вывода имеет следующий вид:

- а)  $\langle \text{переменная} \rangle := \langle \text{выражение} \rangle;$
- б) Read ( $\langle \text{список ввода} \rangle$ );
- в) Write ( $\langle \text{список ввода} \rangle$ ).

# ТЕСТ

## на тему: «Программирование линейных алгоритмов»

7. Знак пунктуации обозначающий конец программы:

- а) точка;
- б) точка с запятой;
- в) двоеточие и равно.



# ТЕСТ

## на тему: «Программирование линейных алгоритмов»

8. Точку с запятой можно не ставить:

- а) после оператора Begin ;
- б) после Var;
- в) перед End .

# Разветвляющийся алгоритм

**Разветвляющимся** называется алгоритм, в котором выбирается одна из нескольких возможных серий команд. Каждый подобный путь называется ветвью алгоритма.

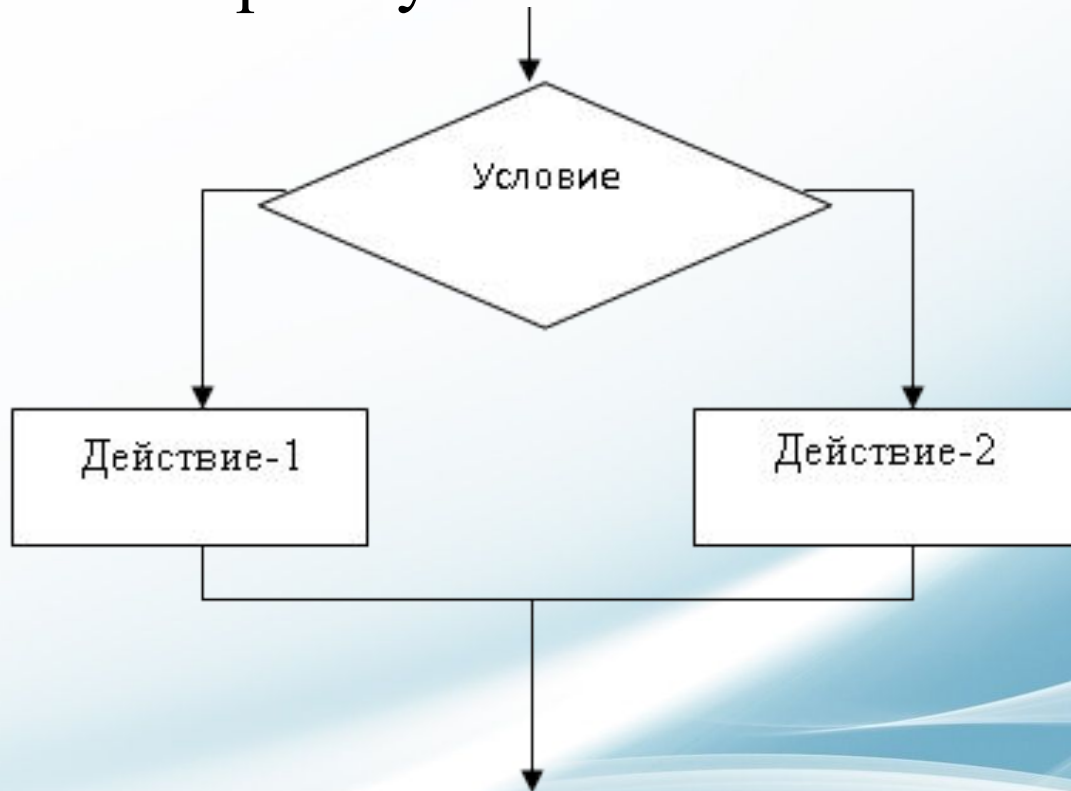
Разветвляющийся алгоритм используется для реализации процесса ветвления. А обязательным атрибутом ветвления является **условный оператор**.

**Условие** – это логическое выражение, которое может быть записано в операторе явно или вычислено в программе.



# Условный оператор

**Условный оператор if** предназначен для выбора к исполнению одного из возможных действий (операторов) в зависимости от некоторого условия. Его **схема** выглядит так:



Если некоторое условие выполняется, то нужно перейти к действию-1. Если это условие не выполняется, то переходим к действию-2.

# Форма условного оператора

Полная форма условного оператора выглядит так:

**if** <логическое выражение>    **then** <оператор 1>    **else** <оператор 2>  
(если)                                    (то)                                    (иначе)



# Правила выполнения оператора условия if

- 1.** Вычисляется логическое выражение (булевый тип: true (истина), false (ложь)). Название «булевые» произошло от имени математика Джорджа Буля, разработавшего в 19 веке булеву логику и алгебру логики. Переменная, которая может принимать одно из двух значений (истина) или (ложь), называется булевой переменной. (на слайде портрет Д.Буля).
- 2.** Если значение логического выражения true (истина), выполняется оператор-1, указанный после then.
- 3.** Если значение логического выражения false (ложь), выполняется оператор-2, указанный после else.

# Используемые знаки

В конструкции условного оператора используются знаки сравнения, которые в Turbo Pascal обозначаются так:

>	больше
<	меньше
>=	больше или равно
<=	меньше или равно
<>	не равно



# Поэтапное решение задачи

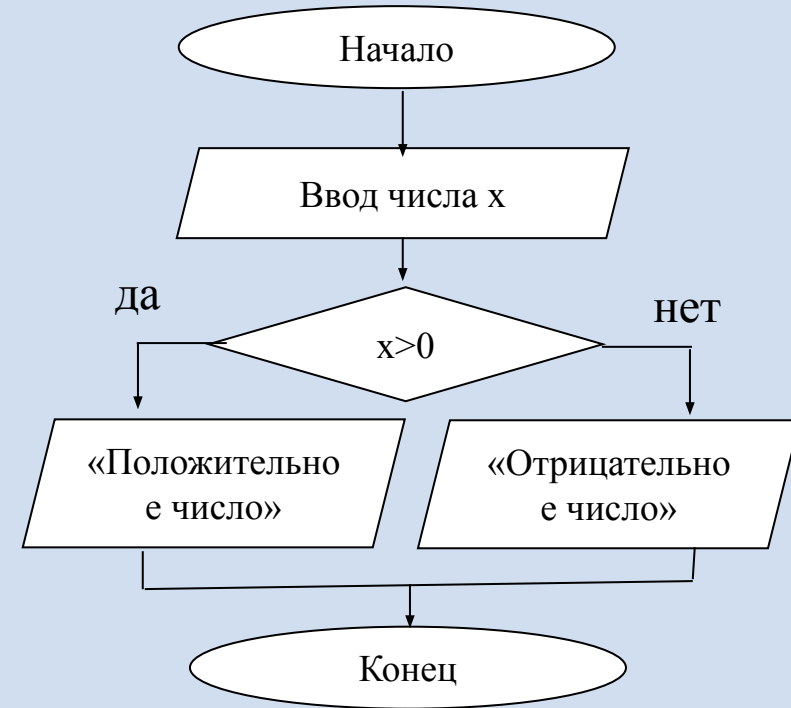
**Задача 1.** Ввести число. Если это число положительное, то на экран вывести сообщение «Положительное число». Если это число отрицательное, то на экран вывести сообщение «Отрицательное число».

Этап решения	Описание
<b>1. Постановка задачи</b> – <i>определение исходных данных и искомым результатов</i>	Дано $x$ – вводимое число, которое является целочисленным типом данных (integer). Определить значение вводимого числа и вывести соответствующее сообщение.
<b>2. Формализация</b> – <i>переход к задаче обработки некоторый знаковой системы (математ. задаче)</i>	$x > 0$

## Этап решения

**3. Построение алгоритма** – *определение структуры алгоритма, последовательность команд*

## Описание





Этап решения	Описание
<p><b>4. Составление программы</b> – запись и отладка программы на языке программирования</p>	<pre><b>Program</b> Trial1; <b>Var</b>   x: integer; <b>Begin</b>     write ('введите число'); readln (x);     if x&gt;0   then   writeln ('ПОЛОЖИТЕЛЬНОЕ ЧИСЛО')         else   writeln ('ОТРИЦАТЕЛЬНОЕ ЧИСЛО') <b>End.</b></pre>
<p><b>5. Тестирование</b> – экспериментальное доказательство правильности алгоритма и работоспособности программы</p>	<p>Протестируйте программу на Турбо Паскале 7.0</p>

# Домашнее задание

**Семакин И. Г. Информатика. Базовый уровень: учебник для 10 класса. С. 116-131.**

**Выполнить самостоятельную работу студента №3-4.**