

# АЛГОРИТМ И ЕГО СВОЙСТВА

Понятие алгоритма и исполнителя  
Свойства алгоритма

# Алгоритм



**Алгоритм** — это понятные и точные предписания исполнителю совершить конечное число шагов, направленных на решение поставленной задачи.

*Пример:*

- рецепт печенья;
- инструкция по ремонту утюга.

# Исполнитель

**Исполнитель** — это человек, животное или техническое устройство, которое понимает и может выполнить команды алгоритма.

**Система команд исполнителя (СКИ)** — это набор команд, которые он понимает и может исполнить.

**Среда исполнителя** — это условия, при которых становится возможным выполнение алгоритма.



# Виды исполнителей

Исполнитель

Формальный

*Не знает конечной цели алгоритма, не задумывается о результате*

Неформальный

*Знает конечную цель алгоритма*



# Задание

*Заполните таблицу:*

Вид работы	Исполнитель	Команды СКИ	Среда исполнителя
Расписание уроков	Ученик	Подготовиться к урокам, идти на урок, идти на перемену	Школа, учебники, тетради и др.
Перевозка пассажиров	Водитель	Посадить в машину, высадить из машины, отвезти пассажиров	Машина, водит. права
Рецепт врача	Пациент	Принять лекарство, помереть температуру	Таблетки, градусник и др.

# Свойства алгоритма

Дискретность

Алгоритм разбивается на конечное число элементарных действий (шагов)

Понятность

Каждое из действий (шагов) является законченным и понятным исполнителю

Однозначность

Каждое действие (шаг) пониматься в строго определенном смысле

Массовость

По данному алгоритму должна решаться не одна, а целый класс подобных задач

Результативность

Алгоритм и каждый его шаг должны приводить к определенному результату

Конечность

Алгоритм должен выполняться за конечное количество шагов

# Задание

*Какое свойство алгоритма нарушено?*

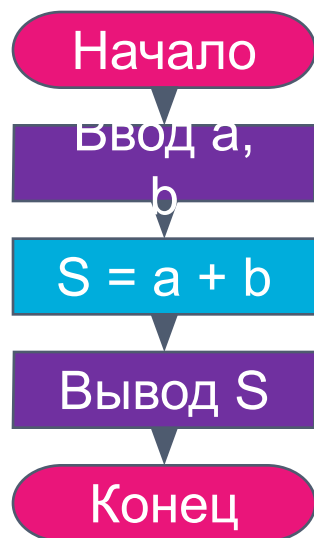
Алгоритм	Нарушенное свойство
Добавить в кофе 2-3 ложки сахара	Однозначность
Найти детерминант матрицы размером $2 \times 2$	Понятность
Найти произведение чисел 4 и 5	Массовость
Сварить суп	Дискретность
Принимать таблетки 3 раза в день после еды	Конечность
Вымыть посуду грязной губкой	Результативность

# Способы записи алгоритма

## Словесный

1. Задать  $a$  и  $b$
2. Найти сумму  $a$  и  $b$
3. Записать ответ

## Блок-схема



## Алгоритмический

```
алг СУММА
  цел a, b, S
нач
  ввод a, b
  S = a + b
  вывод S
кон
```

## Язык программирования

```
program SUM;
var a, b, S : Integer;
begin
  readln (a, b);
  S := a + b;
  writeln (S);
end.
```



# Словесная запись алгоритма

*Составьте алгоритм «Вскипятить 1 л воды».*

1. Начало.
2. Взять чайник.
3. Открыть кран.
4. Налить 1 л воды.
5. Закрыть кран.
6. Включить плиту.
7. Поставить чайник.
8. Ждать пока вода закипит.
9. Выключить плиту.
10. Конец.



*Какое свойство отсутствует у алгоритма?*

# Словесная запись алгоритма

## Составьте алгоритм решения задачи:

Преобразовать слово «БЫК» в слово «ВОЛ». Исполнитель умеет на каждом шаге менять только одну букву. При этом должно получаться существующее слово.

1. Начало.            **БЫК**
2. Ы → О            **БОК**
3. К → Р            **БОР**
4. Б → В            **ВОР**
5. Р → Л            **ВОЛ**
6. Конец.



# Словесная запись алгоритма

## Составьте алгоритм решения задачи:

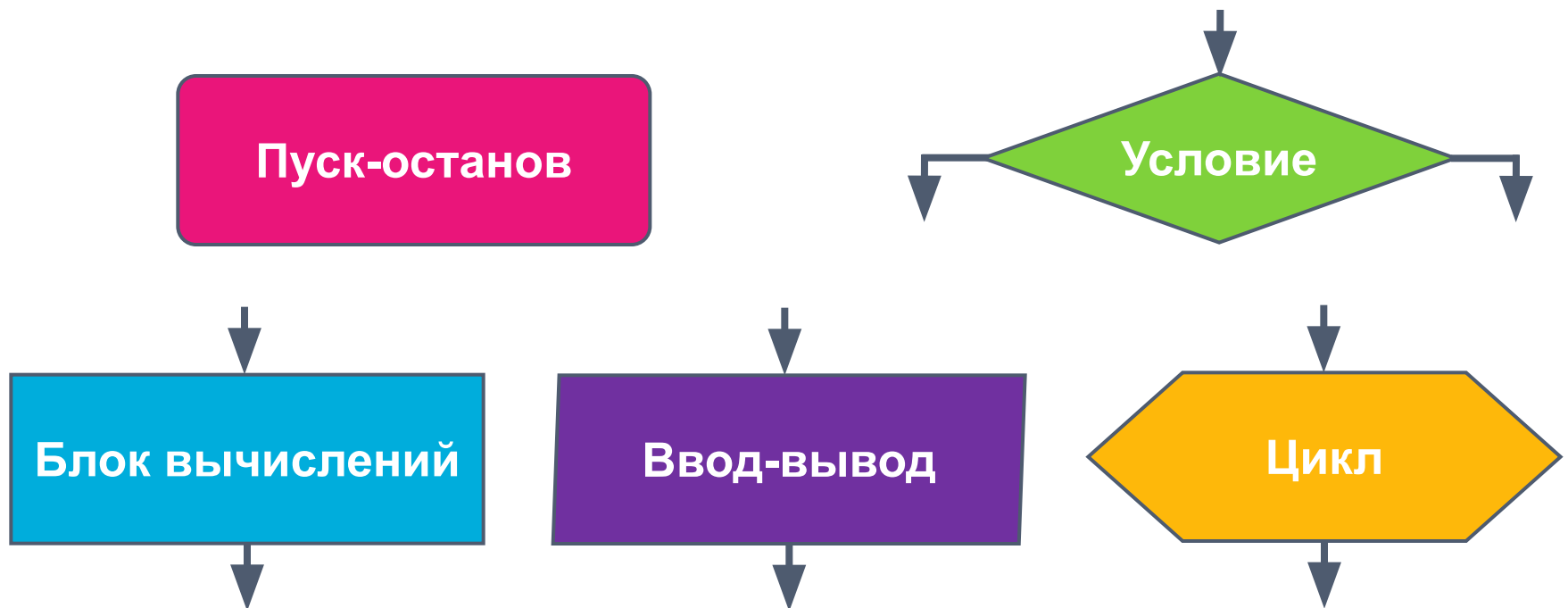
*Получить из числа «1» число «100». Исполнитель умеет на каждом шаге или умножить на 2, или прибавлять 1.*

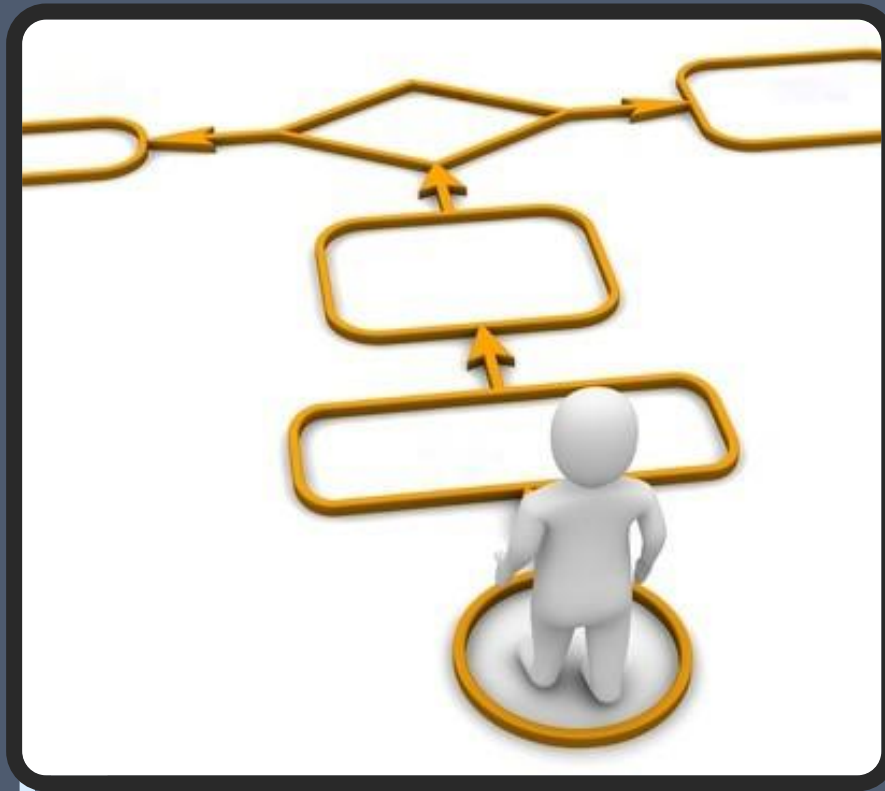
1. Начало.
2.  $1 + 1 = 2$
3.  $1 + 2 = 3$
4.  $3 \times 2 = 6$
5.  $6 \times 2 = 12$
6.  $12 \times 2 = 24$
7.  $24 + 1 = 25$
8.  $25 \times 2 = 50$
9.  $50 \times 2 = 100$
10. Конец.



# Блок-схема

**Блок-схема** — это графический способ записи алгоритма, при котором отдельные шаги изображаются в виде блоков различной формы, соединенных между собой линиями.





# БАЗОВЫЕ АЛГОРИТМИЧЕСКИЕ КОНСТРУКЦИИ

Линейный алгоритм  
Алгоритм ветвления  
Циклический алгоритм

# Линейный алгоритм



**Линейный алгоритм** — это алгоритм, который описывает последовательно выполняющиеся действия.

*Формат блока:*

**нач**

<Команда\_1>

<Команда\_2>

...

<Команда\_n>

**кон**

# Задание

## *Восстановите алгоритм «Запуск программы»*

Выбрать пункт меню «Программы»

Настроить

Нажать кнопку «Пуск»

Код

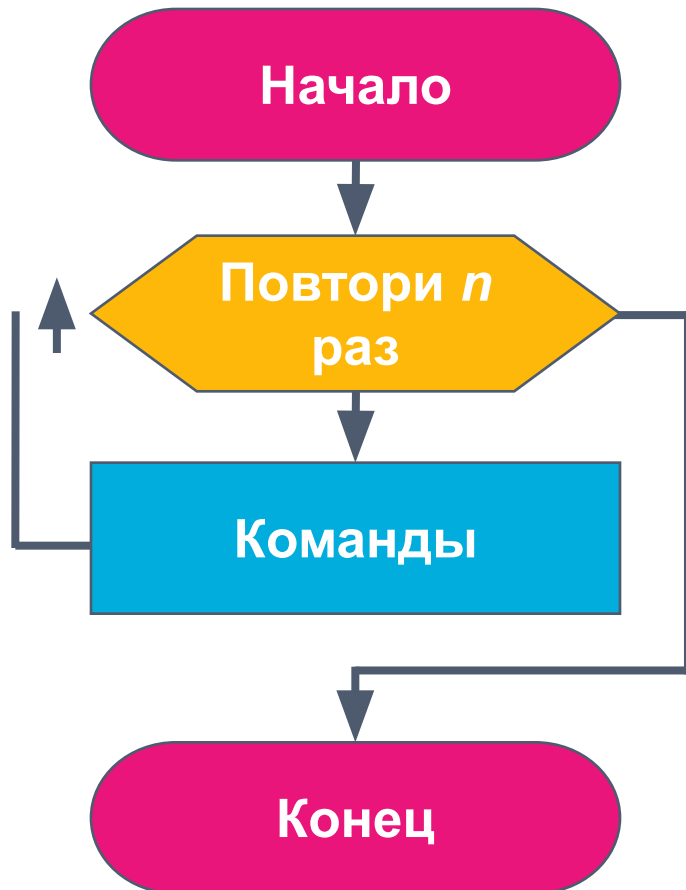
Выбрать пункт меню «Стандартные»

Дождаться загрузки ОС

Включить компьютер

Выбрать пункт меню «Калькулятор»

# Циклический алгоритм



**Циклический алгоритм** — это алгоритм, который описывает повторяющиеся заданное количество раз действия.

**Формат блока:**

Цикл n раз

нц

<Команда\_1>

<Команда\_2>

...

<Команда\_k>

кц

Тело  
цикла



# Задание

*Составьте алгоритм «Мытье посуды».*

Нач

Открыть кран

Повтори 5 раз

нц

    Взять тарелку

    Помыть тарелку

    Поставить тарелку в шкаф

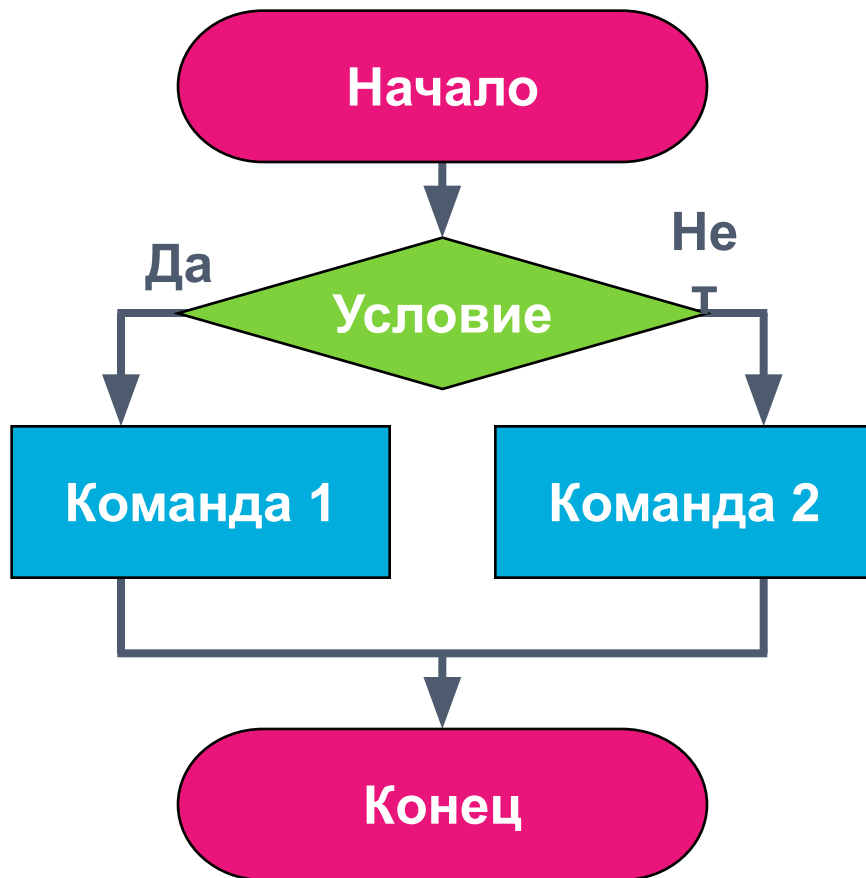
кц

Закрыть кран

Кон

*Оформите алгоритм в виде блок-схемы.*

# Разветвляющийся алгоритм



**Разветвляющийся алгоритм** — это алгоритм, в котором в зависимости от выполнения условия совершается одно или другое действие.

**Формат блока:**

```
Если <условие>  
то <команда_1>  
иначе <команда_2>
```

# Задание

*Какую задачу решает  
данный алгоритм?  
Нарисовать для него  
блок-схему.*

алг Задача

цел  $a, b$

Нач

Ввод  $a, b$

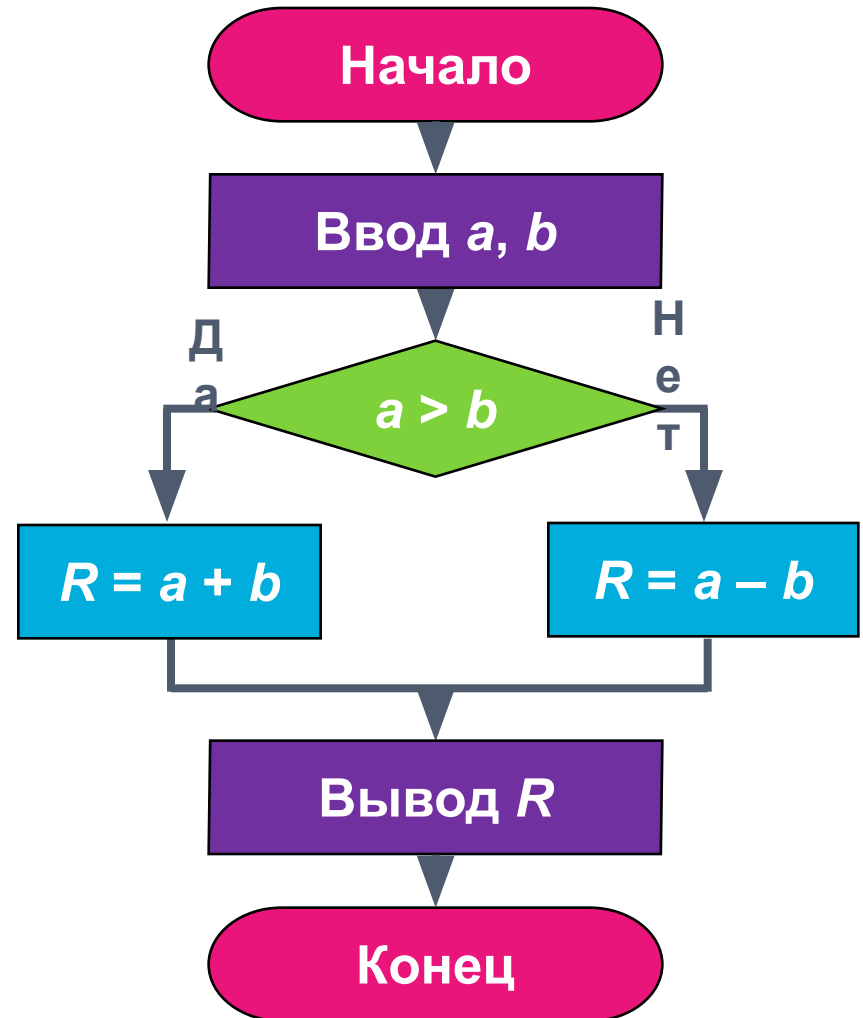
**Если**  $a > b$ , **то**

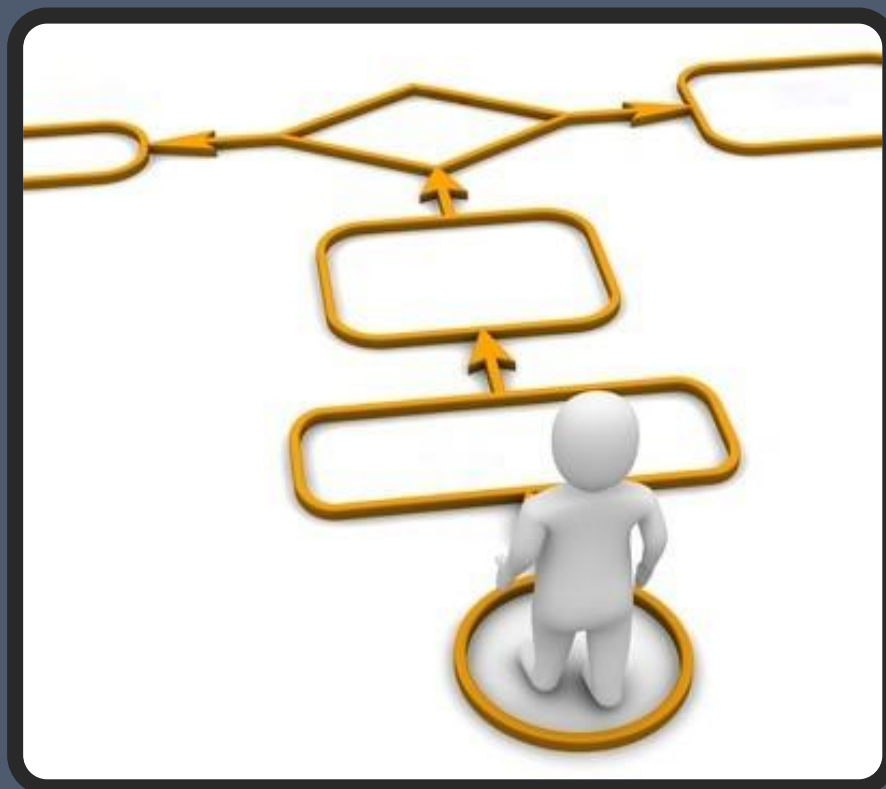
$R = a + b$

**Иначе**  $R = a - b$

Вывод  $R$

Кон





# ЭТАПЫ РЕШЕНИЯ ЗАДАЧ С ПОМОЩЬЮ ПК

# Этап 1. Постановка задачи

**Задача:** Найти сумму двух чисел.

На этапе *постановки задачи* определяются цель решения задачи, входные и выходные данные.

*Пример:*

Ввести числа  $a$  и  $b$ . Найти их сумму и вывести результат  $S$ .

## Этап 2. Моделирование

На этапе *моделирования* разрабатывается математическая или информационная модели задачи.

Для простых задач достаточно указать нужные *формулы или логические условия*.

Для более сложных задач необходимо построить *информационную модель*, т. е. модель, описывающую наиболее существенные признаки объекта, внутренние и внешние взаимосвязи и т. д.

*Пример:*

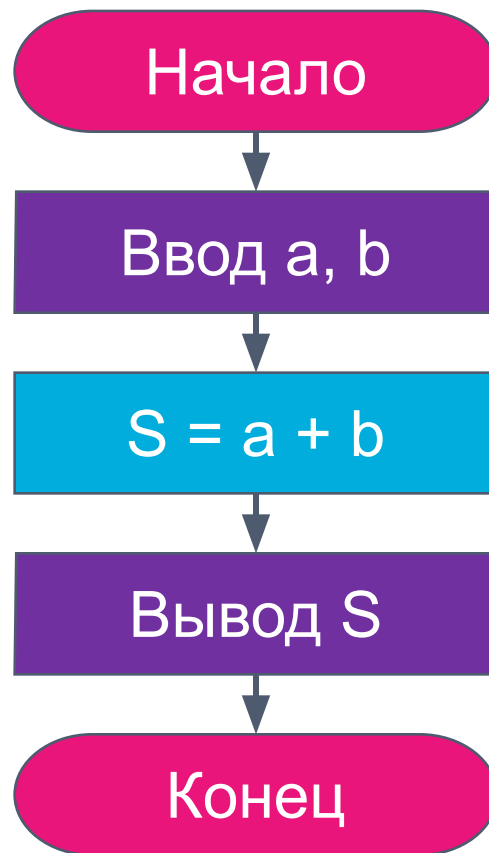
$$S = a + b, \text{ где } a, b, S \in (-\infty; +\infty)$$

# Этап 3. Алгоритмизация

На этапе *алгоритмизации* разрабатывается алгоритм решения задачи и записывается одним из способов записи.

*Пример:*

```
алг Сумма
  цел a, b, Sum
нач
  ввод a, b
  S = a + b
  вывод S
кон
```



# Этап 4. Программирование

**Программа** — это алгоритм, записанный на языке программирования и исполнителем которого является компьютер.

*Пример:*

```
Program Summa;  
  Var a, b, S : Real;  
Begin  
  Write ('Введите числа a и b');  
  ReadLn (a, b);  
  S := a + b;  
  WriteLn ('Сумма чисел S = ', S);  
End.
```



# Этап 5. Тестирование и отладка

**Тестирование** — это процесс проверки правильности работы программы на заранее подготовленных значениях.

**Отладчик** — это программа, которая позволяет проверить работу программы «по шагам» с отслеживанием промежуточных результатов.

*Пример:*

a	b	s
5	3	8
-6	-5	-11
-20	13	-7
0	0	1

Ошибка

# Этап 6. Анализ результатов

**Анализ результатов** помогает определить, достигнута ли поставленная на первом этапе цель создания программы.

Если цель **достигнута**, программа может быть использована по назначению заказчиком.



Если цель **не достигнута**, программа отправляется на переработку.



# Домашнее задание

- **Задача.** Вычислить значение функции  $Z = X^3/Y$ , где  $Y = \sin(nX) + 0.5$
- **Задача.** Вычислить множество значений функции  $Y = X^2 + b$  для всех значений  $X$  от -10 до 10 с шагом 2, при  $b = 5$ .