Министерство образования и науки РФ
Государственное образовательное учреждение
высшего профессионального образования

Тульский государственный университет

КАФЕДРА АВТОМАТИКИ И ТЕЛЕМЕХАНИКИ

# ПОДПРОГРАММА-ФУНКЦИЯ

Лабораторная работа № 2

по курсу «Программирование на ЯВУ»

Вариант № 4

Выполнил: студент группы 220601 _____ Белым А.А.
(подпись)

Проверил: _____ Сулимова В.В.
(подпись)

Тула 2011

## Цель работы

Целью работы является изучить работу с подпрограмм-функций и реализация программы, использующей подпрограмму-функцию,

## Задание

Написать программу, вычисляющую значения переменных x и y в системе уравнений $a_1x+b_1y=c_1$ и $a_2x+b_2y=c_2$, где $a_1$, $b_1$, $c_1$, $a_2$, $b_2$, $c_2$ — некоторые числа, вводимые пользователем с клавиатуры.

## Теоретическая справка

## Схема алгоритма

На рисунке 1 представлена схема алгоритма ввода параметров системы 2-х уравнений с 2-мя неизвестными, её решения и вывода корней системы.

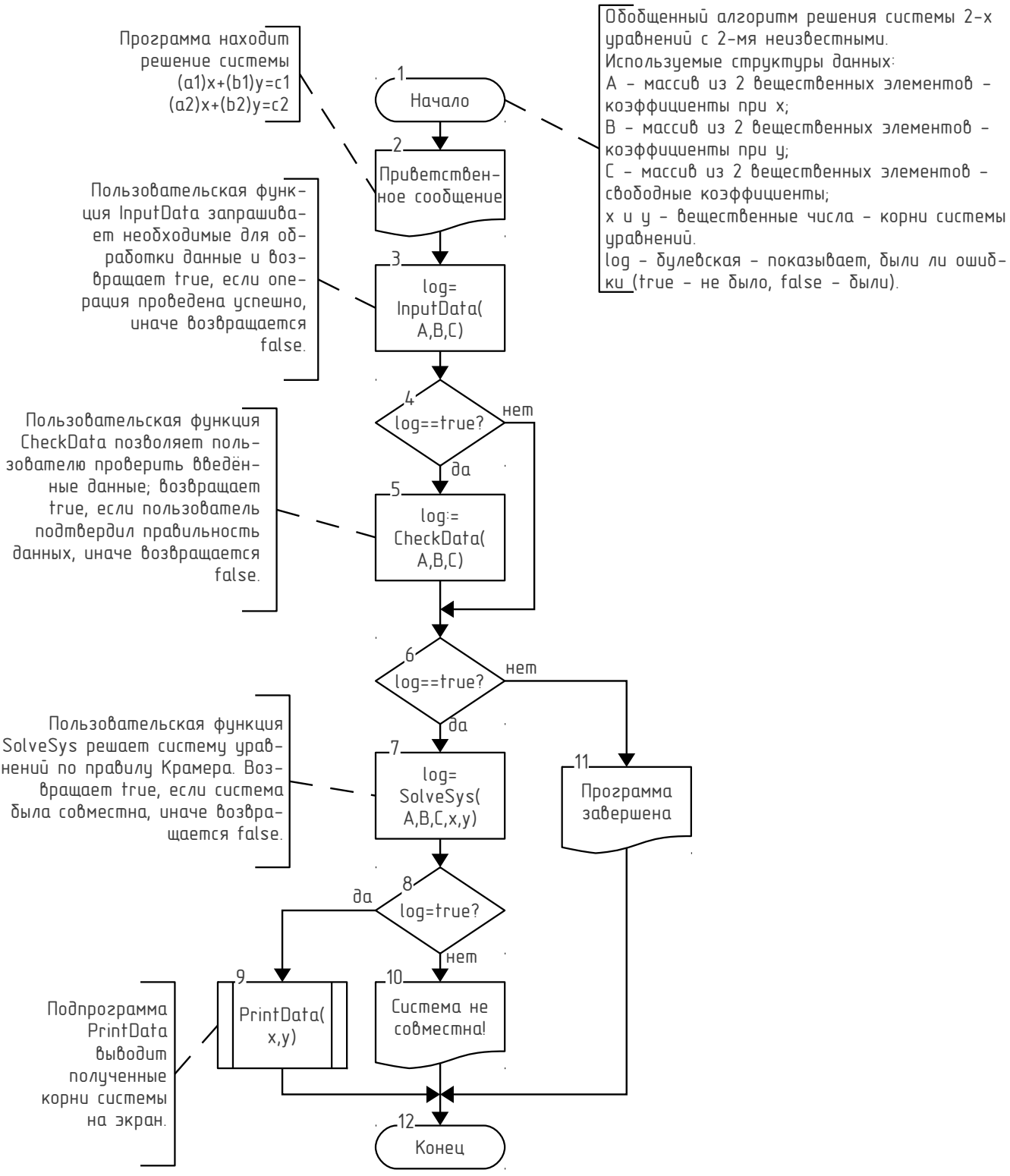


Рисунок 1 — Блок-схема обобщенного алгоритма решения системы уравнений

На рисунке 2 представлена схема алгоритма ввода параметров системы 2-х уравнений.

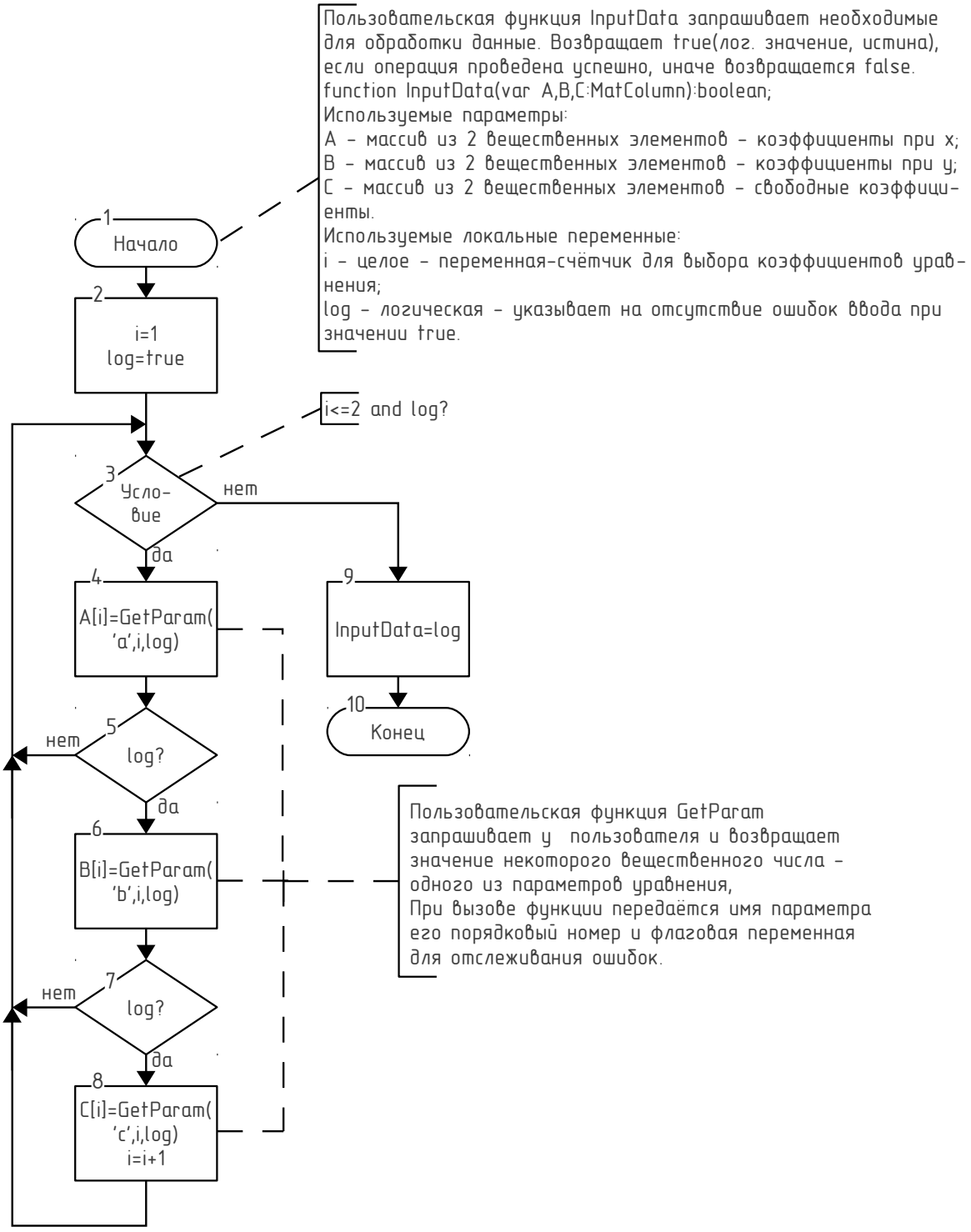


Рисунок 2 — Блок-схема алгоритма ввода параметров системы уравнений

На рисунке 3 представлена блок-схема алгоритма запроса одного из параметров системы.

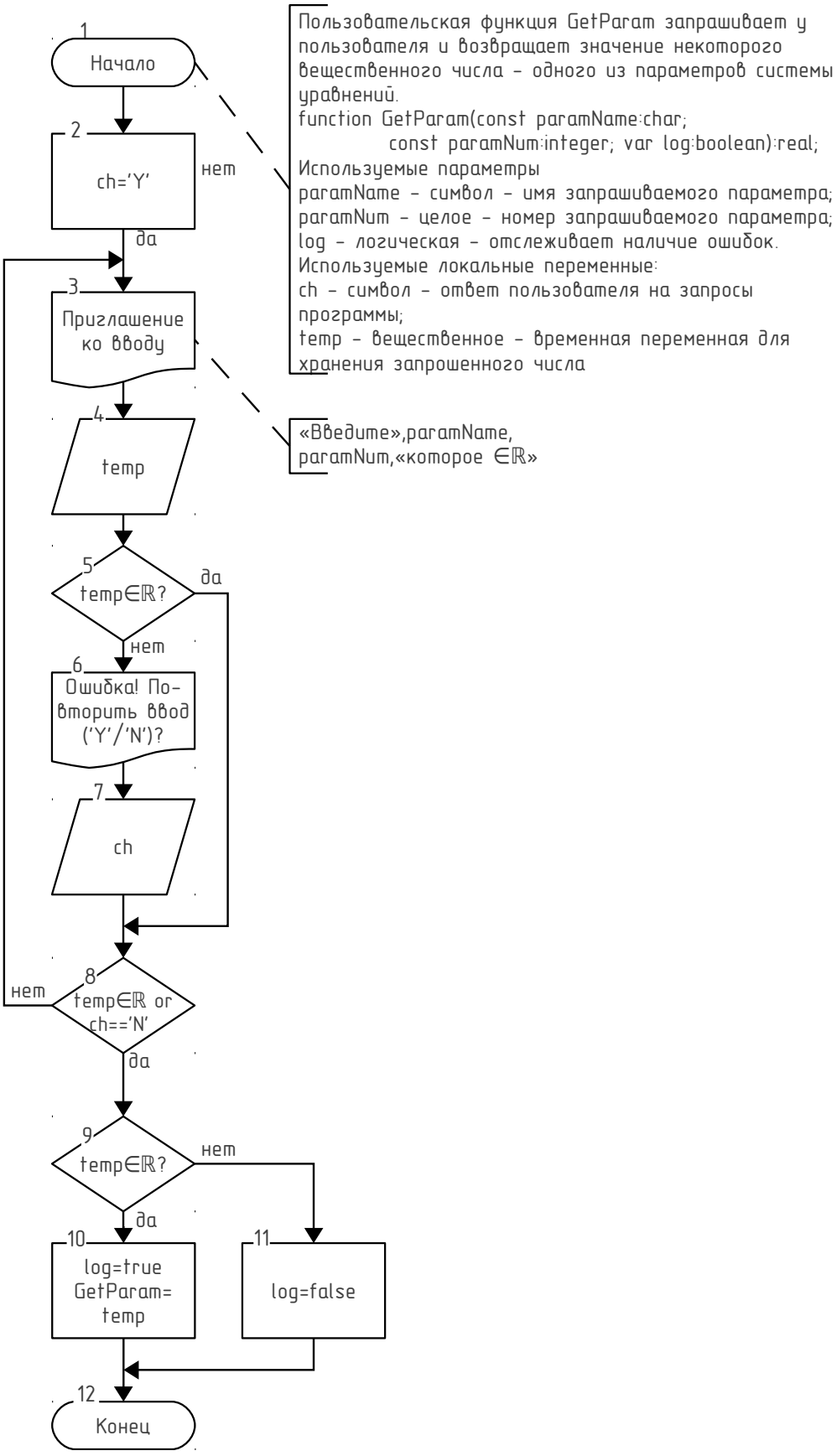


Рисунок 3 — Блок-схема алгоритма ввода произвольного параметра

На рисунке 4 представлена схема алгоритма организации проверки пользователем введённых данных.

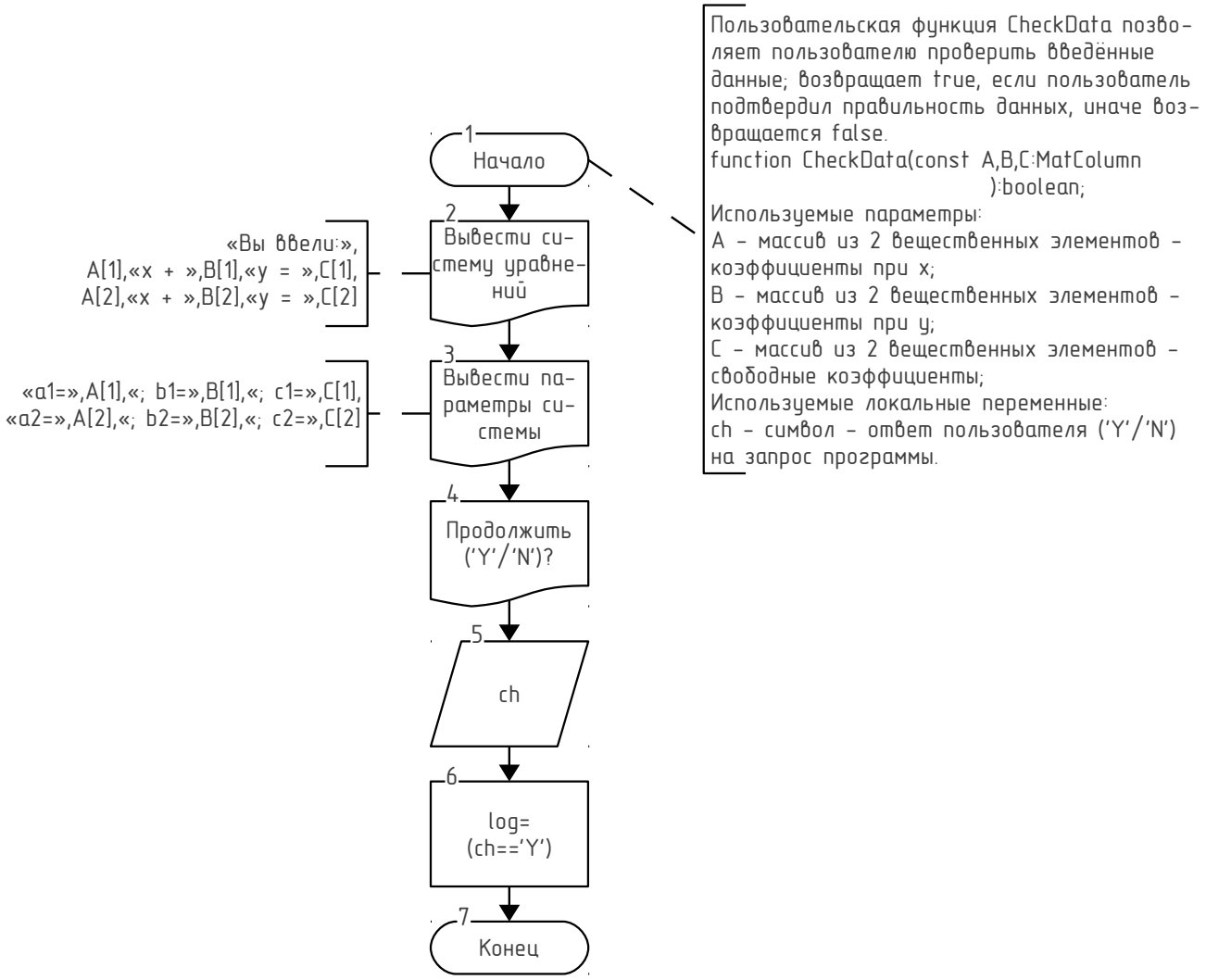


Рисунок 4 — Блок-схема алгоритма организации проверки пользователем введённых данных

На рисунке 5 представлена схема алгоритма решения системы 2-х уравнений с 2-мя неизвестными.

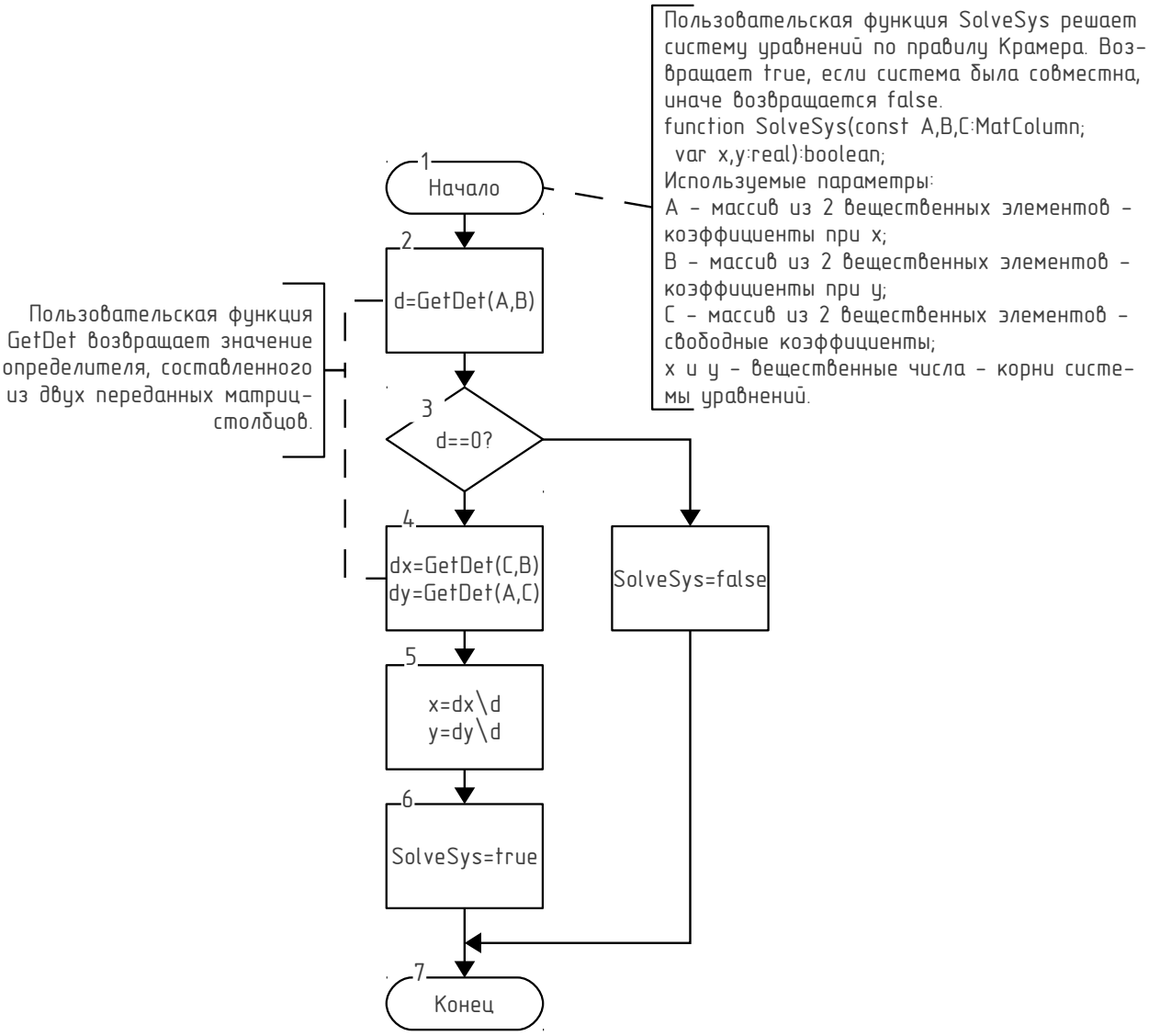


Рисунок 5 — Блок-схема алгоритма решения системы уравнений

На рисунке 6 представлена схема алгоритма расчета определителя 2-ого порядка.

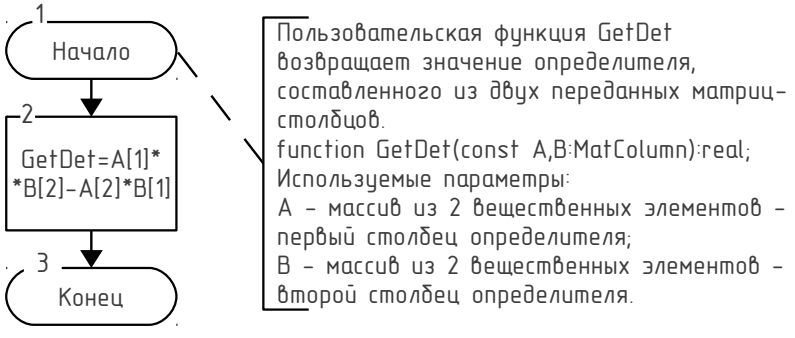


Рисунок 6 — Схема алгоритма  расчета определителя 2-ого порядка

На рисунке 7 представлена схема алгоритма вывода корней системы 2-х уравнений с 2-мя неизвестными.

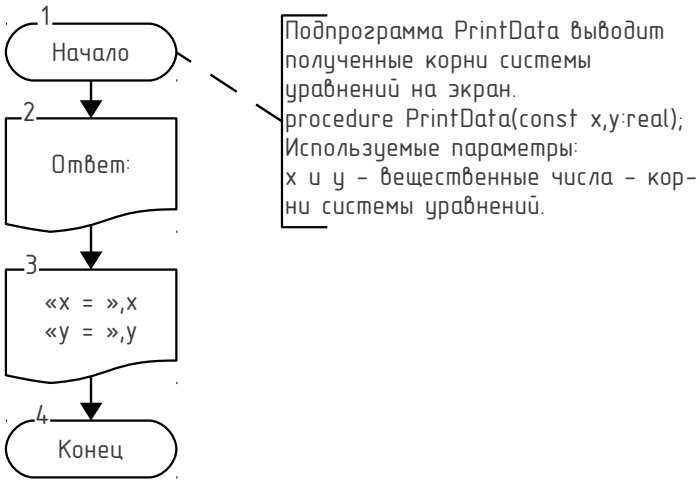


Рисунок 7 — Блок-схема алгоритма вывода корней системы уравнений

## Инструкция пользователю

Данная программа позволяет решать системы из двух линейных уравнений.

Для работы программы необходимо ввести с клавиатуры коэффициенты уравнений — сначала коэффициент при x, затем при y и свободный коэффициент первого уравнения, затем коэффеценты второго уравнения в том же порядке. Все параметры — вещественные числа. После этого предоставляется возможность проверить введённые данные; если они корректны, для продолжения работы нужно ввести 'y', ввод любого другого значения приведёт к завершению работы программы с соответствующим сообщением.

В качестве результатов работы программа либо выведет 2 числа - x и y, удовлетворяющие системе, если система совместна; либо если система не имеет решений или имеет бесконечно много решений (т. е. если определитель системы равен нулю), то будет сообщено, что система не совместна.

## Инструкция программисту

При создании программы решения системы 2-х уравнений с 2-мя неизвестными были сделаны следующие действия.

В основной части программы определена костанта maxvars, равная 2, которая определяет количество неизвестных системы.

Определён тип MatColumn как array[1..maxvars] of real, для представления коэффициентов системы в форме столбцов (в отдельном массиве коэффициенты при x, в другом при y, и т.д.).

Были введены структуры данных, описание которых представлено в таблице 1.

Таблица 1 - Структуры данных, используемые в в основной части программы оценки элементов массива

| имя | тип | предназначение |
|---|---|---|
| A,B,C | MatColumn | Столбцы расширенной матрицы системы: коэффициенты уравнений при x, коэффициенты системы при y и свободные коэффициенты соответственно. |
| x,y | real | Корни системы уравнений. |
| log | boolean | Указывает на отсутствие ошибок (значение true) ввода данных или желания пользователя прервать программу. |

Кроме того, в процессе создания вышеуказанной программы были определены следующие подпрограммы:

1. Функция GetParam запрашивает у пользователя произвольный (по имени и номеру) параметр системы - вещественное число. Возвращает запрошенный параметр; кроме того, в случае ошибки пользователя уведомляет об этом основную программу с помощью переменной log.

```
function GetParam(const paramName:char;const paramNum:integer;
                  var log:boolean):real;
```

Используемые функцией параметры-константы приведены в таблице 2; параметры-переменные - в таблице 3, локальные переменные - в таблице 4.

Таблица 2 - Параметры-константы функции ввода произвольного параметра системы

| имя | тип | предназначение |
|---|---|---|
| paramName | char | Имя запрашиваемого параметра. |
| paramNum | integer | Порядковый номер запрашиваемого параметра. |

Таблица 3 - Параметры-переменные функции ввода произвольного параметра системы

| имя | тип | предназначение |
|---|---|---|
| log | boolean | Указывает на желание пользователя продолжить работу программы(при значении true, false - завершение программы) |

Таблица 4 - Локальные переменные функции ввода произвольного параметра системы

| имя | тип | предназначение |
|---|---|---|
| ch | char | Содержит ответ пользователя на запросы программы о повторении ввода данных. |
| buf | string | Буфер, который содержит значения в строковом формате на случай ошибочного ввода параметра. |

2. Функция InputData вводит коэффициенты системы уравнений. Возвращает true, если ввод завершился без ошибок. Так как функция использует тип MatColumn и константу maxvars, то они должны быть определены глобально в основной программе. Кроме того, функция InputData использует функцию GetParam, поэтому она также должна быть глобальной по отношению к InputData.

function InputData(var A,B,C:MatColumn):boolean;

Используемые функцией параметры-переменные приведены в таблице 5.

Таблица 5 - Параметры-переменные функции ввода параметров уравнений системы

| имя | тип | предназначение |
|---|---|---|
| A,B,C | MatColumn | Столбцы расширенной матрицы системы: коэффициенты уравнений при x, коэффициенты системы при y и свободные коэффициенты соответственно. |

3. Функция CheckData позволяет организовать проверку пользователем введенных им ранее значений. Возвращает true, если пользователь подтвердил правильность данных. Так как функция использует тип MatColumn, то он должен быть определен глобально в основной программе.

function CheckData(const A,B,C:MatColumn):boolean;

Используемые функцией параметры-константы приведены в таблице 6; локальные переменные - в таблице 7.

Таблица 6 - Параметры-константы функции проверки пользователем введенных значений

| имя | тип | предназначение |
|---|---|---|
| A,B,C | MatColumn | Столбцы расширенной матрицы системы: коэффициенты уравнений при x, коэффициенты системы при y и свободные коэффициенты соответственно. |

Таблица 7 - Локальные переменные функции проверки пользователем введенных значений

| имя | тип | предназначение |
|---|---|---|
| ch | char | Содержит ответ пользователя на запрос программы о правильности данных. |

4. Функция GetDet рассчитывает определитель 2-ого порядка, составленный из 2-х матриц-столбцов. Возвращает искомый определитель. Так как функция использует тип MatColumn, то он должен быть определен глобально в основной программе.

function GetDet(const A,B:MatColumn):real;

Используемые функцией параметры-константы приведены в таблице 8.

Таблица 8 - Параметры-константы функции подсчета определителя

| имя | тип | предназначение |
|---|---|---|
| A,B | MatColumn | Первый и второй столбец рассчитываемого определителя |

5. Функция SolveSys решает систему 2-х уравнений с 2-мя неизвестными. Возвращает true, если переданная система была совместна. Так как функция использует тип MatColumn, то он должен быть определен глобально в основной программе.

function SolveSys(const A,B,C:MatColumn; var x,y:real):boolean;

Используемые функцией параметры-константы приведены в таблице 9; параметры-переменные - в таблице 10; локальные переменные - в таблице 11.

Таблица 9 - Параметры-константы функции решения системы уравнений

| имя | тип | предназначение |
|---|---|---|
| A,B,C | MatColumn | Столбцы расширенной матрицы системы: коэффициенты уравнений при x, коэффициенты системы при y и свободные коэффициенты соответственно. |

Таблица 10 - Параметры-переменные функции решения системы уравнений

| имя | тип | предназначение |
|---|---|---|
| x,y | real | Корни системы уравнений. |

Таблица 11 - Локальные переменные функции решения системы уравнений

| имя | тип | предназначение |
|---|---|---|
| d | real | Определитель данной системы. |
| dx | real | Определитель из свободных членов и коэффициентов при y. |
| dy | real | Определитель из коэффициентов при x и свободных членов. |

6. Процедура PrintData выводит корни системы уравнений.

procedure PrintData(const x,y:real);

Используемые процедурой параметры-константы приведены в таблице 12.

Таблица 12 - Параметры-константы процедуры вывода корней системы уравнений

| имя | тип | предназначение |
|---|---|---|
| x,y | real | Корни системы уравнений. |

## Текст программы

Ниже представлен текст программы, написанной на языке Turbo Pascal 7, которая решает систему 2-х уравнений с 2-мя неизвестными.

```
{-----ПРОГРАММА РЕШЕНИЯ СИСТЕМЫ 2-Х УРАВНЕНИЙ С 2-МЯ НЕИЗВЕСТНЫМИ-----}
Program SysSolver;
const maxvars=2;{кол-во переменных системы}
type MatColumn=array[1..maxvars] of real; {тип "матрица-столбец"}

{------ФУНКЦИЯ ВВОДА ВЕЩЕСТВЕННОГО ПАРАМЕТРА-----}
{Параметры: имя запрашиваемого параметра; номер параметра; переменная состояния}
function GetParam(const paramName:char;const paramNum:integer;
                                              var log:boolean):real;
var temp:real; buf:string; code:integer; ch:char;
begin
    ch:='Y'; temp:=0; {инициализация переменных}
    repeat
        WriteLn('Введите ',paramName,paramNum,' - вещественное число.');
        Write(paramName,paramNum,':');ReadLn(buf); Val(buf,temp,code);
        if code<>0 then begin {введено не вещественное число?}
            WriteLn('Недопустимое значение ',paramName,paramNum,'.');
            WriteLn('Повторить ввод?(y/n)');
            ReadLn(ch);
        end;
    until (Upcase(ch)='N') or (code=0);
    {пока пользователь не отказался или число некорректное}
    log:=code=0;{число введено корректно?}
    if log then
        GetParam:=temp;{вернём значение}
end;

{------ФУНКЦИЯ ВВОДА ДАННЫХ-----}
{Параметры:коэффициенты при x; коэффициенты при y; свободные коэффициенты}
function InputData(var A,B,C:MatColumn):boolean;
var i:byte; log:boolean;
begin

    log:=true;i:=1; {инициализация переменной}
    while (i<=maxvars) and log do begin {пока не было ошибки или не введены
                                                 уравнения}
        A[i]:=GetParam('a',i,log);
        if log then
            B[i]:=GetParam('b',i,log);
        if log then
```

```
            C[i]:=GetParam('c',i,log);
        inc(i);
    end;
    InputData:=log;
end;

{------ФУНКЦИЯ ПРОВЕРКИ ЗНАЧЕНИЙ-----}
{Параметры:коэффициенты при x; коэффициенты при y; свободные коэффициенты}
function CheckData(const A,B,C:MatColumn):boolean;
var ch:char;
begin
      WriteLn('Вы ввели:');
      WriteLn(A[1]:1:4,'x + ',B[1]:1:4,'y = ',C[1]:1:4);
      WriteLn(A[2]:1:4,'x + ',B[2]:1:4,'y = ',C[2]:1:4);
      WriteLn('a1=',A[1]:1:4,'; b1=',B[1]:1:4,'; c1=',C[1]:1:4);
      WriteLn('a2=',A[2]:1:4,'; b2=',B[2]:1:4,'; c2=',C[2]:1:4);
      WriteLn('OK?(y/n)');
      Readln(ch);
      CheckData:=UpCase(ch)='Y';
end;

{------ФУНКЦИЯ ПОДСЧЕТА ОПРЕДЕЛИТЕЛЯ-----}
{Параметры: первый столбец определителя; второй столбец определителя}
function GetDet(const A,B:MatColumn):real;
begin
    GetDet:=A[1]*B[2]-B[1]*A[2];
end;

{------ФУНКЦИЯ РЕШЕНИЯ СИСТЕМЫ-----}
{Параметры:коэффициенты при x; коэффициенты при y; свободные коэффициенты
                        найденное значение x; значение y}
function SolveSys(const A,B,C:MatColumn; var x,y:real):boolean;
var d,dx,dy:real;
begin
    d:=GetDet(A,B);
    if d<>0 then begin
        dx:=GetDet(C,B);
          dy:=GetDet(A,C);
          x:=dx/d;
          y:=dy/d;
          SolveSys:=true;
      end else SolveSys:=false;
end;
```

```
{------ПРОЦЕДУРА ВЫВОДА КОРНЕЙ СИСТЕМЫ-----}
{Параметры: значение x; значение y}
procedure PrintData(const x,y:real);
begin
    WriteLn('Ответ:');
    WriteLn('x =',x);
    WriteLn('y =',y);
end;

{------ОСНОВНАЯ ПРОГРАММА-----}
var A,B,C:MatColumn;
{коэффеценты при x; при y; свободные коэффеценты}
      x,y:real;
{значение x; значение y}
    log:boolean;
{переменная состояния}
BEGIN
      WriteLn('Данная программа находит корни системы уравнений');
      WriteLn('(a1)x+(b1)y=(c1);');
      WriteLn('(a2)x+(b2)y=(c2);');
      log:=InputData(A,B,C);
      if log then
          log:=CheckData(A,B,C);
      if log then begin
          log:=SolveSys(A,B,C,x,y);
          if log then
              PrintData(x,y)
          else
              WriteLn('Система не совместна!');
      end else
          WriteLn('Программа завершена.');
      WriteLn('Нажмите <Enter>...');
      ReadLn;
END.
```

## Тестовые примеры

На рисунке 8 представлен пример работы программы решения системы уравнений для системы  x+2y=7 и 2x+y=5.

```
Данная программа находит корни системы уравнений
(a1)x+(b1)y=(c1);
(a2)x+(b2)y=(c2);
Введите a1 - вещественное число.
a1:1
Введите b1 - вещественное число.
b1:2
Введите c1 - вещественное число.
c1:7
Введите a2 - вещественное число.
a2:2
Введите b2 - вещественное число.
b2:1
Введите c2 - вещественное число.
c2:5
Вы ввели:
1.0000x + 2.0000y = 7.0000
2.0000x + 1.0000y = 5.0000
a1=1.0000; b1=2.0000; c1=7.0000
a2=2.0000; b2=1.0000; c2=5.0000
OK?(y/n)
y
Ответ:
x = 1.00000000000000E+000
y = 3.00000000000000E+000
Нажмите <Enter>...
```

Рисунок 8— Пример работы программы решения системы уравнений

## Вывод

В ходе выполнения данной лабораторной работы я научился использовать подпрограммы-функции при написании программ. Функции возвращают через свое имя некоторое значение, что и отличает их от процедур. Эта особенность позволяет функциям участвовать в качестве операнда в выражениях, что бывает полезно при подсчете различных значений в математических, физических, финансовых и многих других областях.