Министерство образования и науки РФ
Государственное образовательное учреждение
высшего профессионального образования

Тульский государственный университет

КАФЕДРА АВТОМАТИКИ И ТЕЛЕМЕХАНИКИ

ТИП ЗАПИСЬ

Лабораторная работа № 4

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 изучение понятия «запись» и самого типа «запись».

Задание

Описать переменную "студент", содержащую: имя, фамилию, отчество студента, название учебного заведения, номер группы. Создать список студентов (N>10). Определить фамилии студентов, учащихся в одной и той же группе, в одном и том же заведении.

Теоретическая справка

Схема алгоритма

На рисунке 1 представлена схема алгоритма ввода данных, поиска одногруппников среди данных студентов и вывода информации о них.

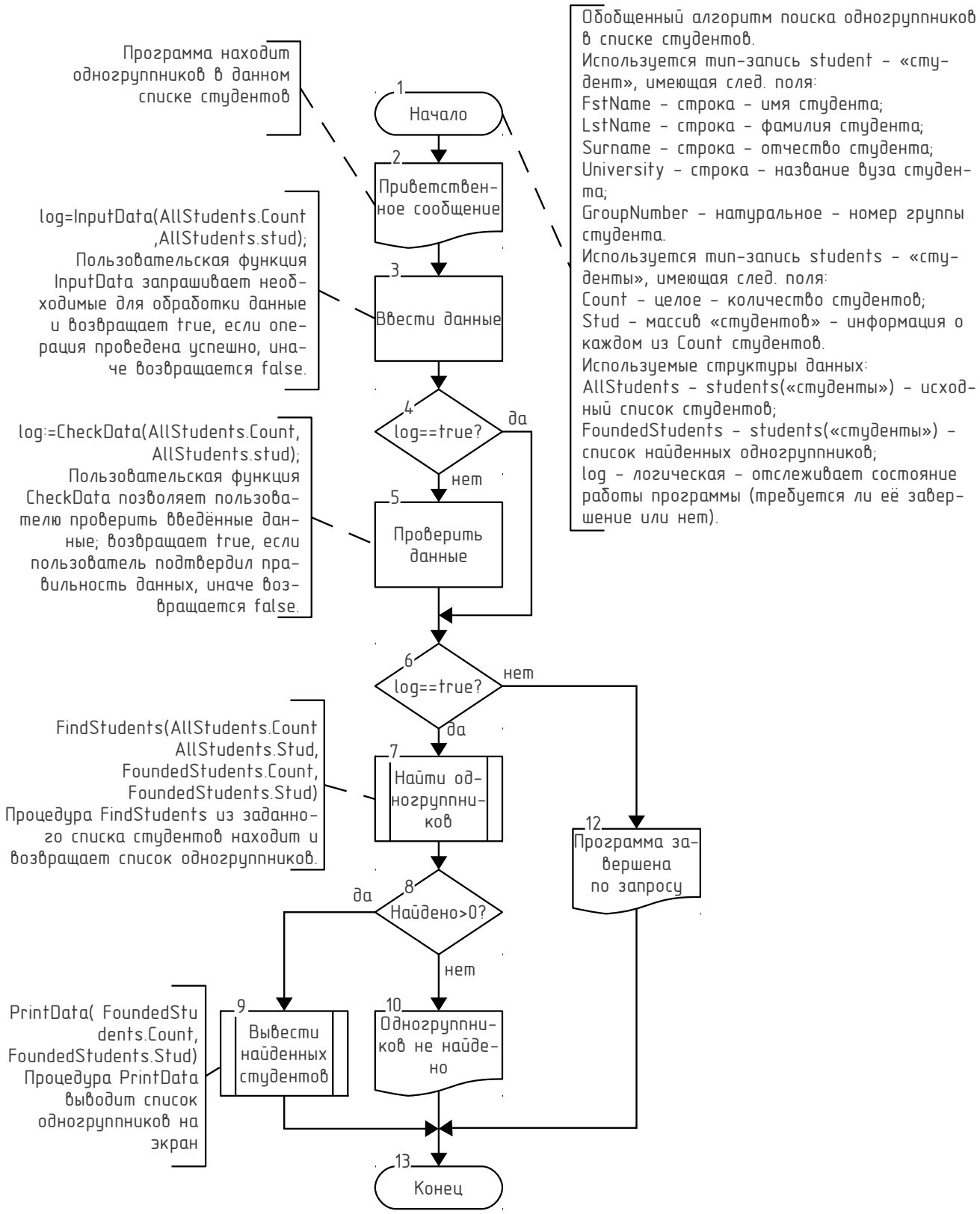

Рисунок 1 — Блок-схема обобщенного алгоритма поиска одногруппников

На рисунке 2 представлена схема алгоритма ввода списка студентов для поиска в них одногруппников.

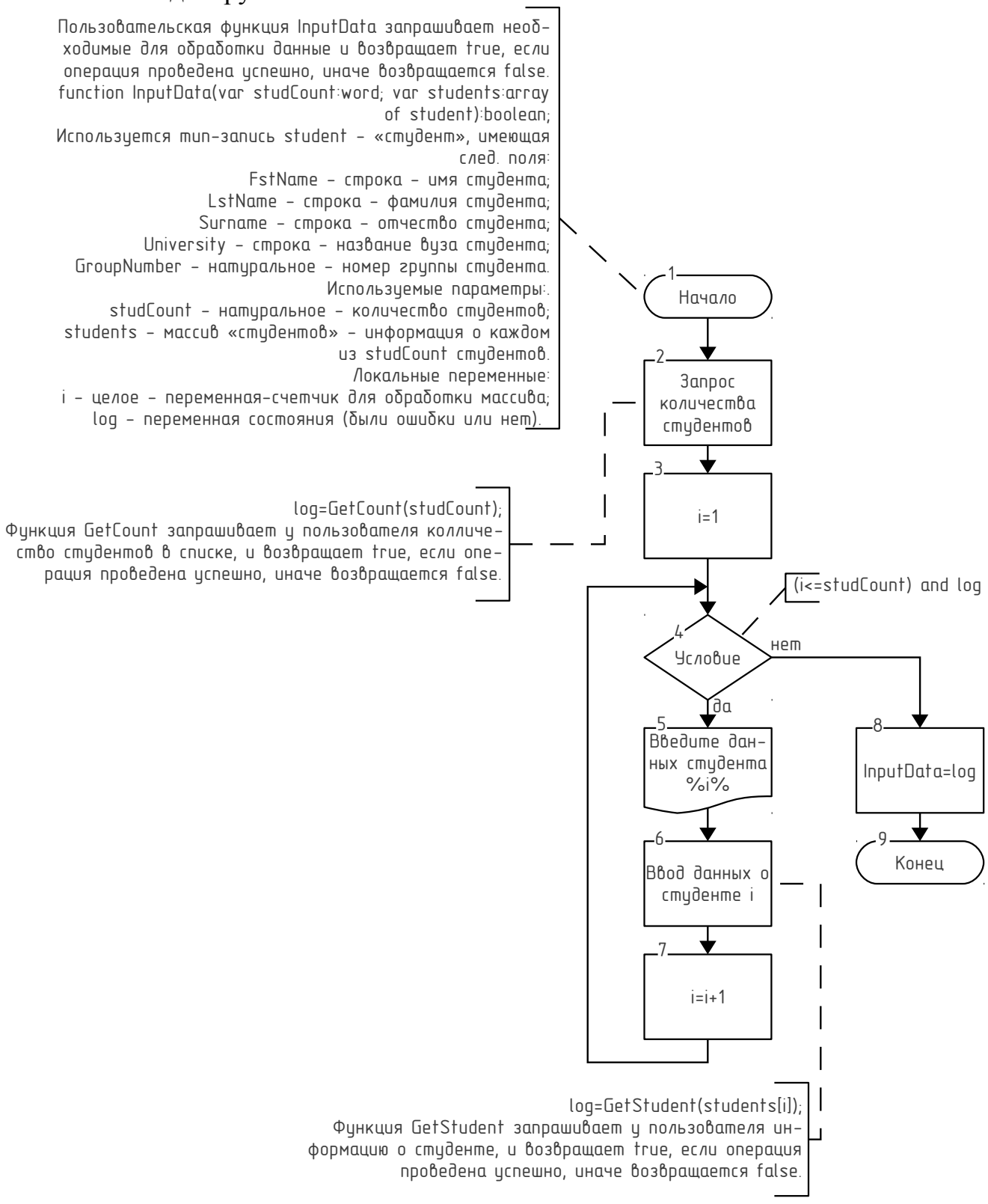

Рисунок 2 — Блок-схема алгоритма ввода списка студентов

На рисунке 3 представлена схема алгоритма ввода количества студентов в списке.

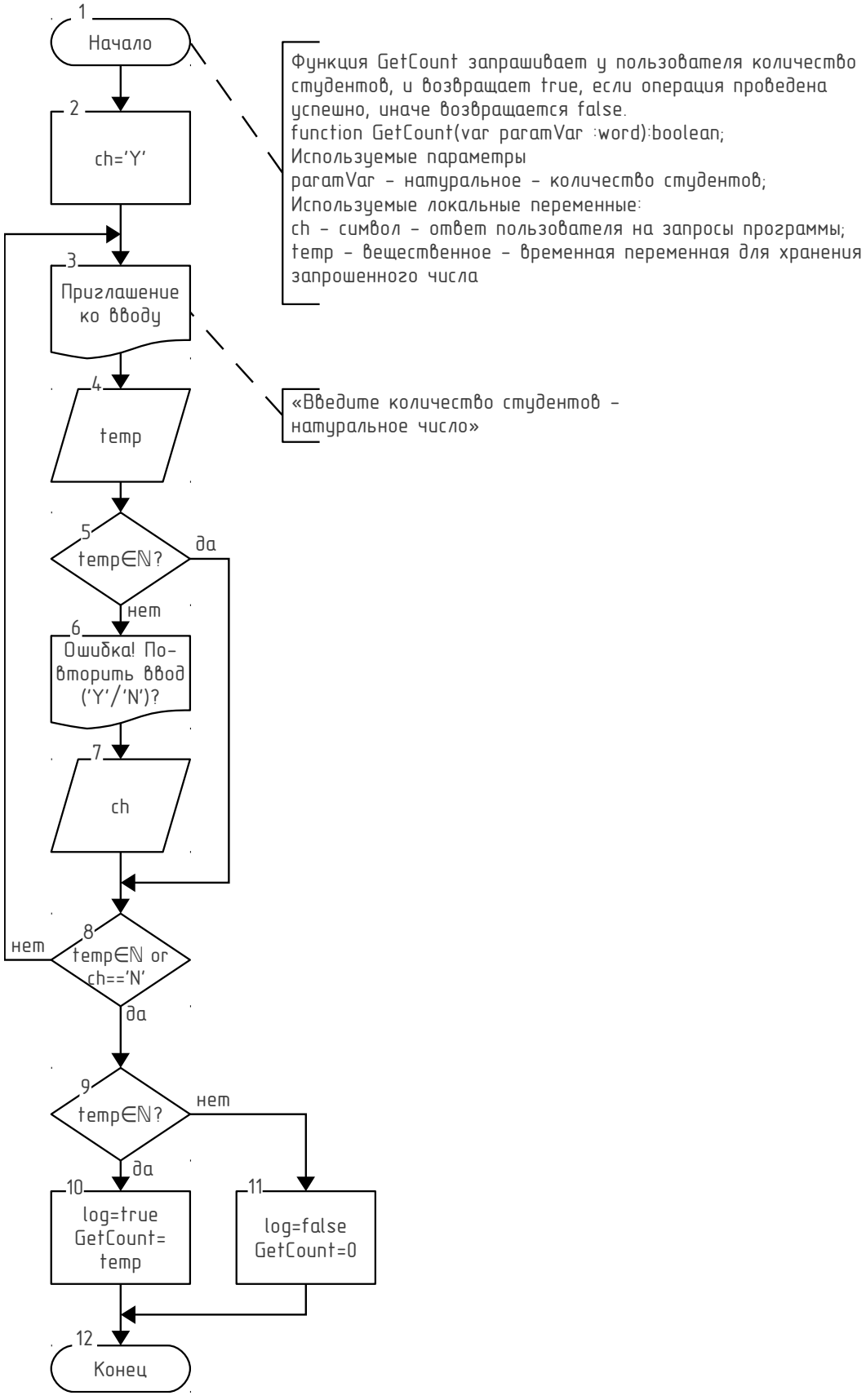

Рисунок 3 — Блок-схема алгоритма ввода количества студентов

На рисунке 4 представлена схема алгоритма ввода данных о конкретном студенте.

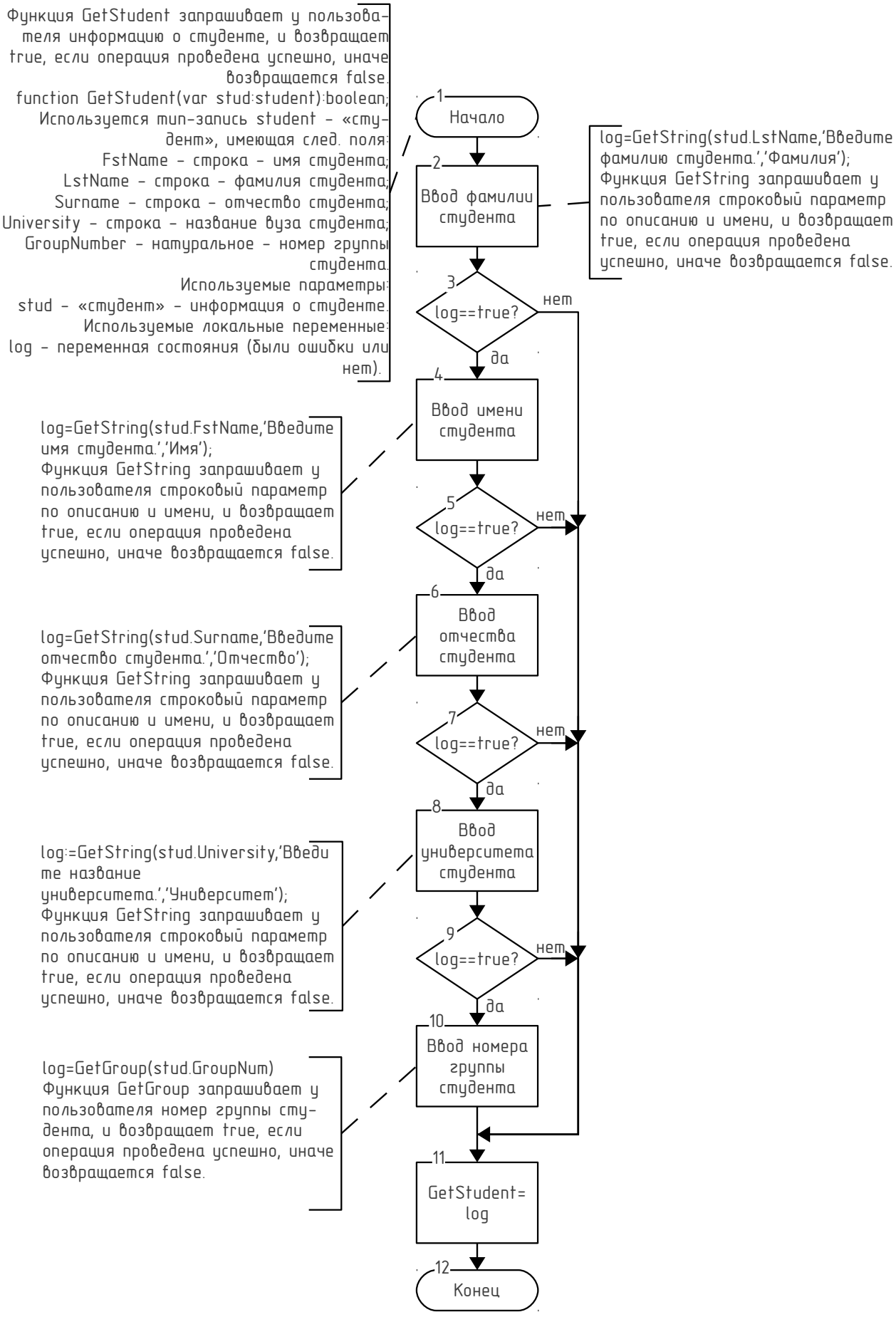

Рисунок 4 — Блок-схема алгоритма ввода данных о студенте

На рисунке 5 представлена блок-схема алгоритма запроса строкового параметра.

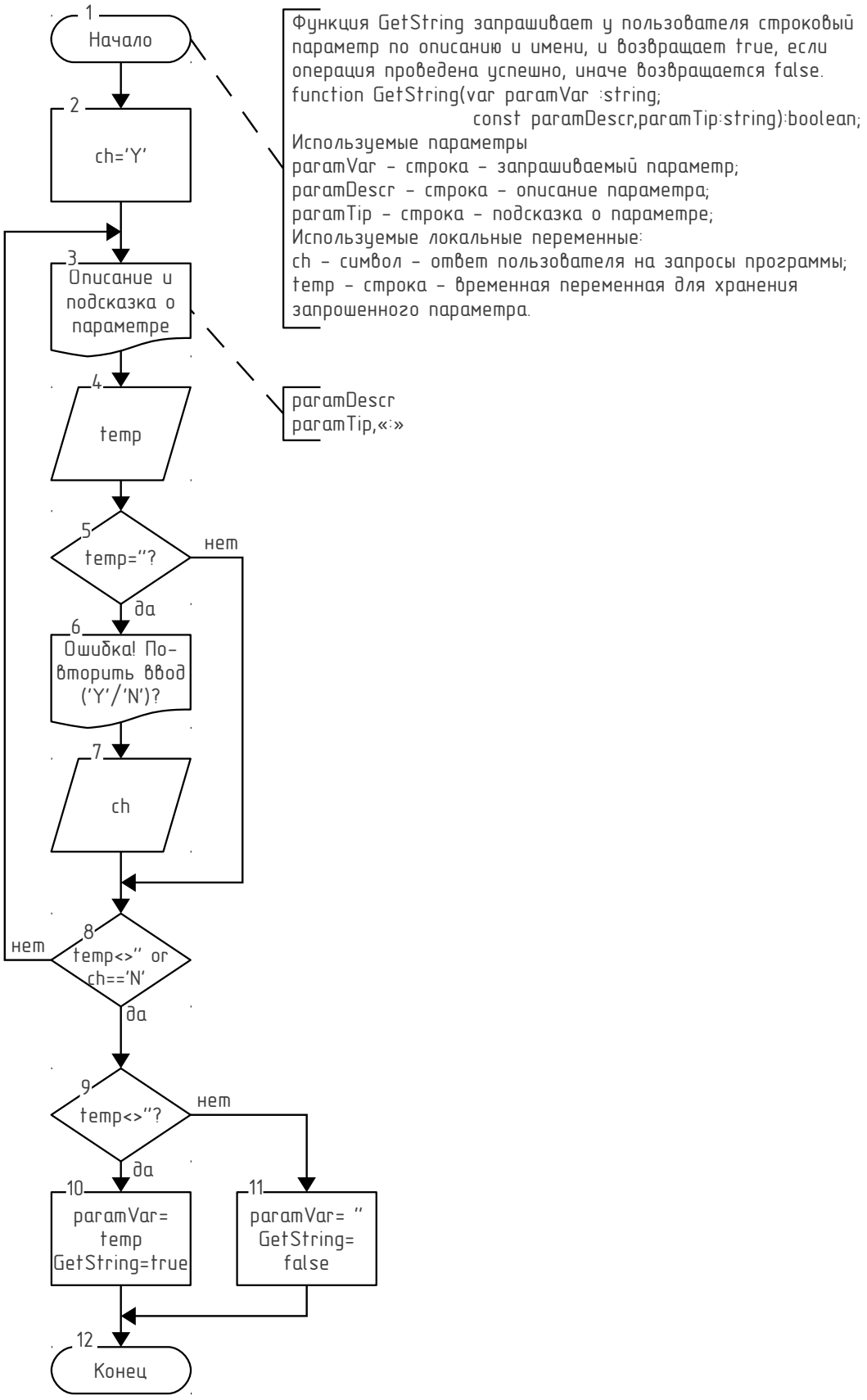

Рисунок 5 — Блок-схема алгоритма ввода строкового параметра

На рисунке 6 представлена схема алгоритма ввода номера группы студента.

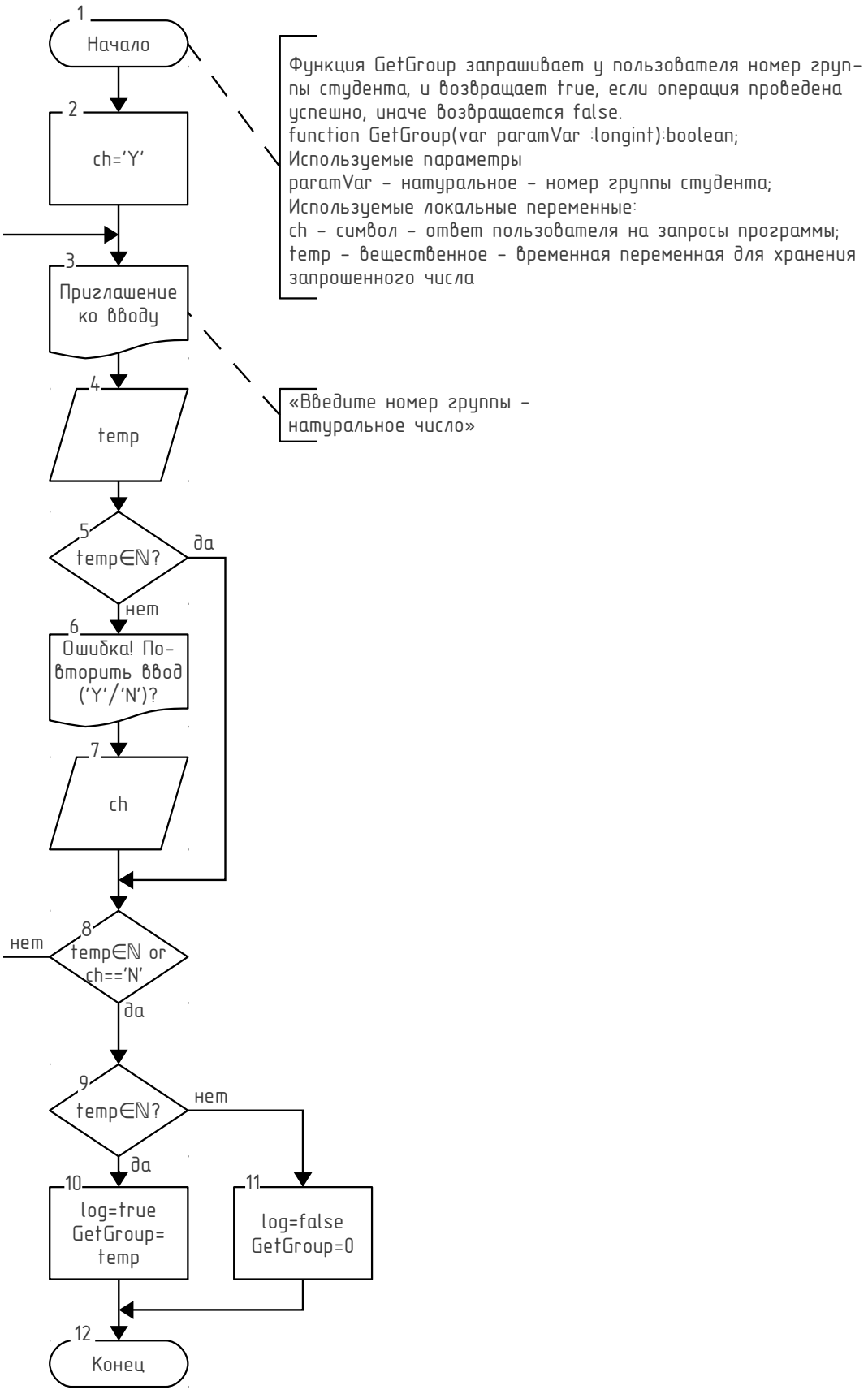

Рисунок 6 — Блок-схема алгоритма ввода номера группы

На рисунке 7 представлена схема алгоритма организации проверки пользователем введённых данных.

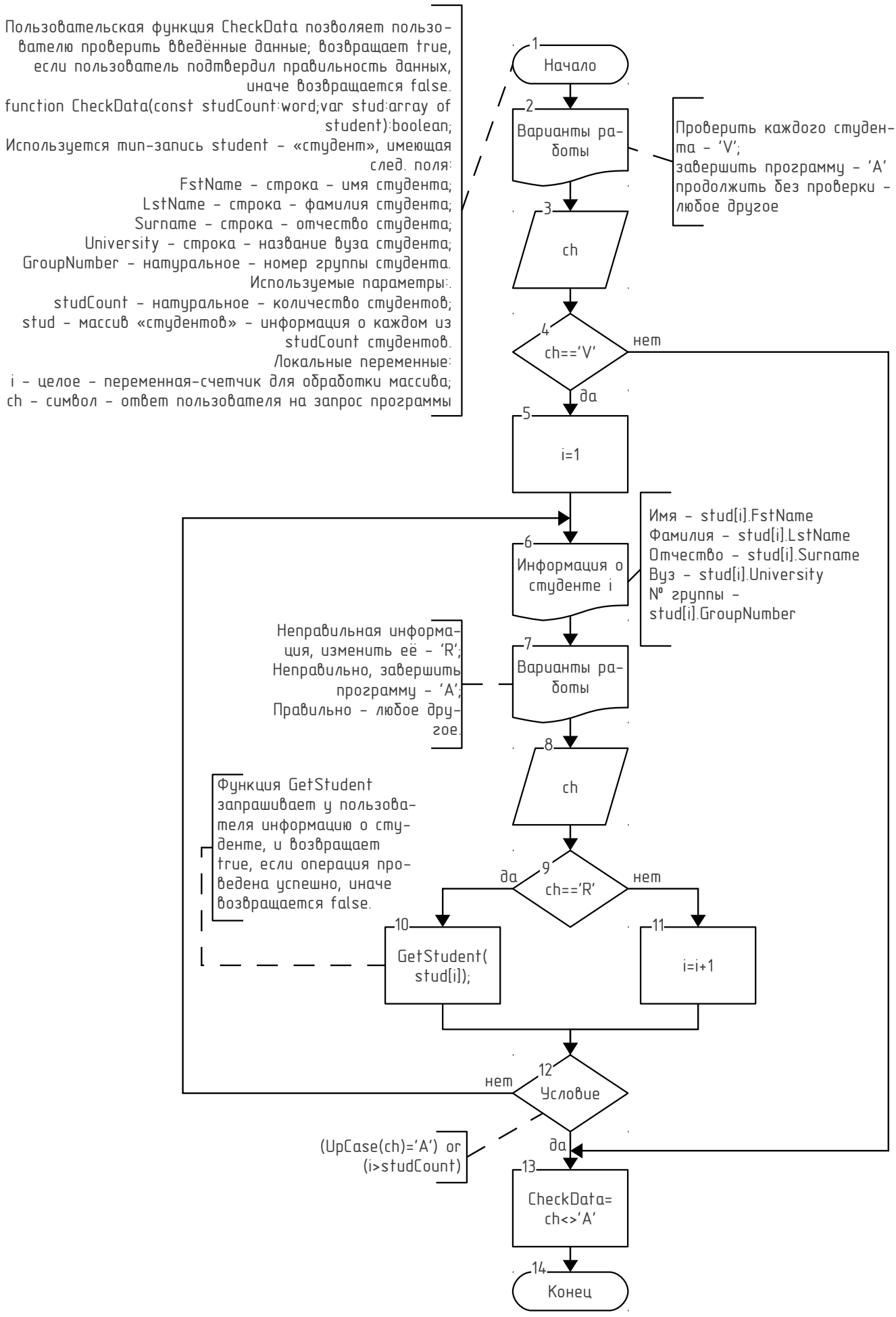

Рисунок 7 — Блок-схема алгоритма проверки пользователем введенных данных

На рисунке 8 представлена схема алгоритма поиска одногрупников среди заданного списка студентов.

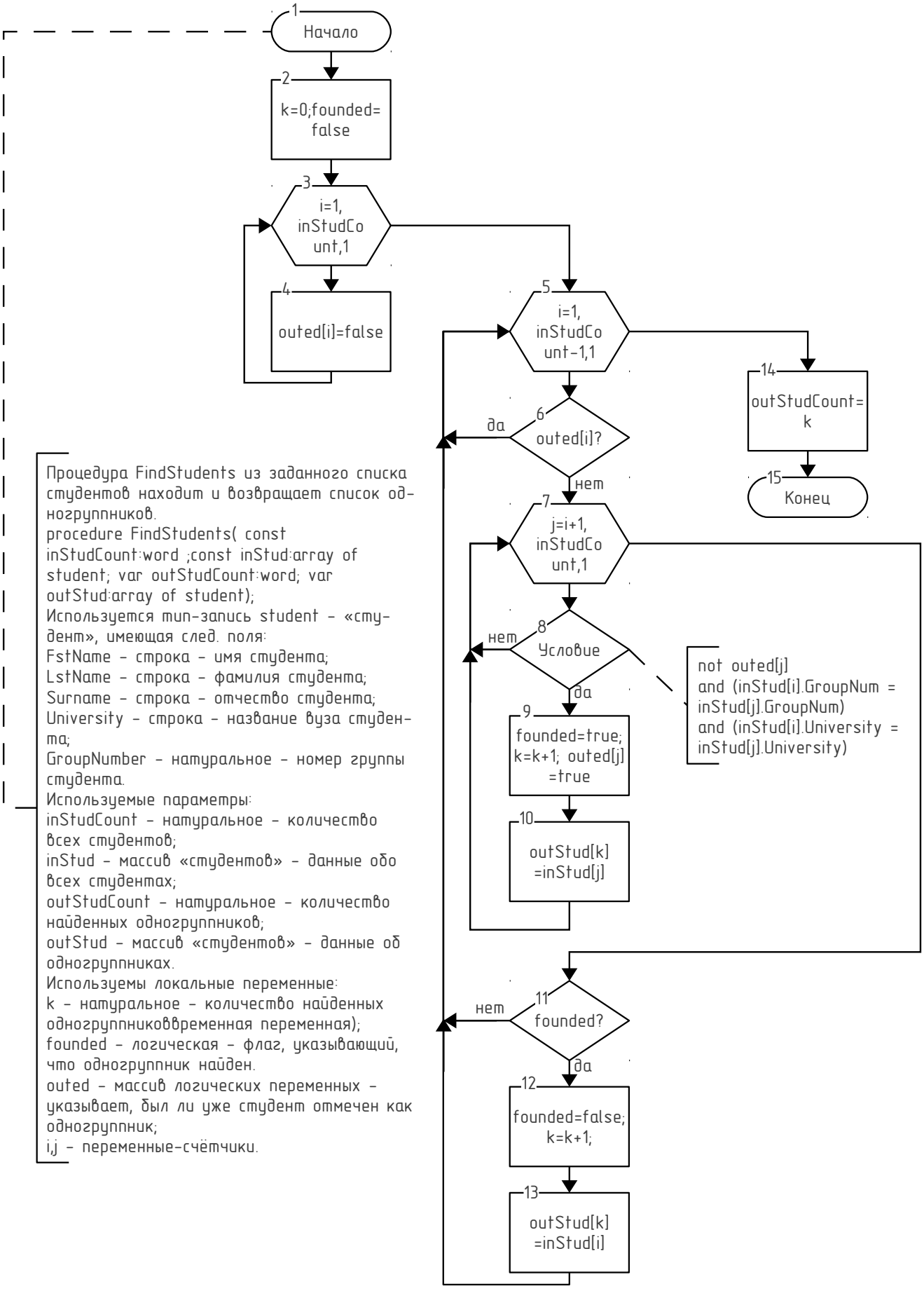

Рисунок 8 — Схема алгоритма поиска одногруппников

На рисунке 9 представлена схема алгоритма вывода данных о найденных одногруппниках на экран.

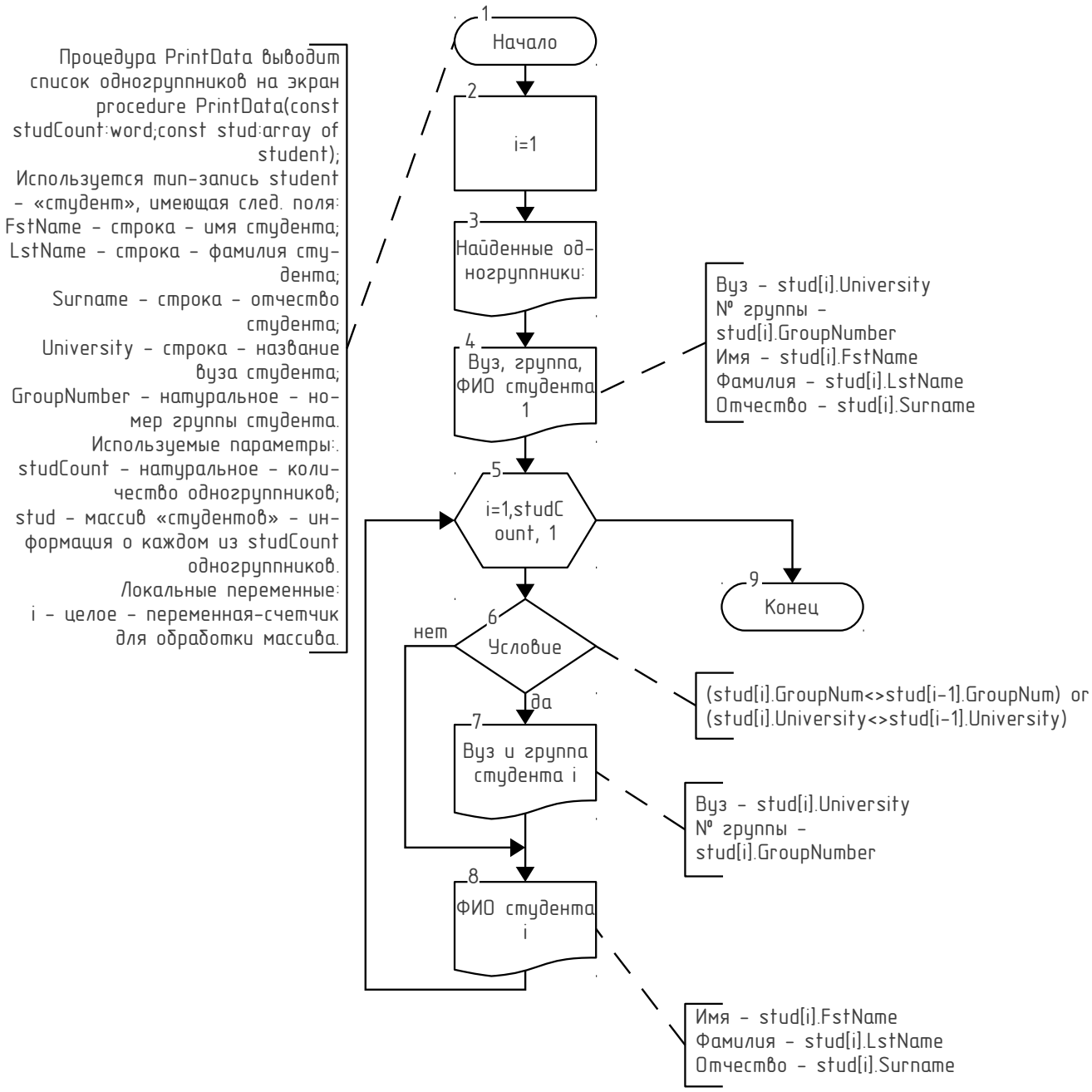

Рисунок 9 — Блок-схема алгоритма вывода данных об одногруппниках

Инструкция пользователю

Данная программа позволяет найти среди переданного ей списка студентов таких студентов, которые учатся в одном вузе в одной группе, т.е. являются одногруппниками.

Для работы программы необходимо ввести количество студентов, среди которых будет производится поиск одногруппников. Это натуральное число, большее нуля и меньшее определенного значения (определяется программой). Затем вводятся данные о каждом из студентов - его имя, фамилия, отчество, название учебного заведения и номер группы. Все параметры, кроме последнего - строковые значения, последнее - натуральное число. В случае неправильного ввода имеется возможность ввести заново данные, или отказаться от повторного воода и завершить программу. После этого предоставляется возможность проверить введённые данные для каждого студента поочередно, ответив на запрос программы 'V', завершить работу программы('A') или продолжить работу без проверки данных (любой другой ответ).

В качестве результатов работы программа выведет список одногруппников: сначала выводятся общие вуз и номер группы, затем данные обучающихся в этой группе: и так делается для каждой различной группы в списке. Если одногруппников не будет найдено, программа выведет соответствующее сообщение.

Инструкция программисту

При создании программы поиска одногруппников были предприняты следующие действия.

В основной части программы определена константа nn - максимальное количество студентов в задаваемом списке.

Определён тип запись student, описывающий некоторого студента; поля этого типа представленны в таблице 1.

Таблица 1 - Поля типа-записи, описывающего студента

имя	тип	предназначение
FstName, LstName, Surname	string[20]	Имя, фамилия, отчество студента соответственно.
University	string[50]	Название вуза студента.
GroupNum	longint	Номер группы студента.

Локально определен тип запись students, описывающий список студентов; поля этого типа представленны в таблице 2.

Таблица 2 - Поля типа-записи, описывающего список студентов

имя	тип	предназначение
Count	word	Количество студентов в списке.
Stud	array [1..nn] of student	Данные о каждом из Count студентов (естественно, Count<=nn)

Были введены структуры данных, описание которых представлено в таблице 3.

Таблица 3 - Структуры данных, используемые в в основной части программы поиска одногруппников

имя	тип	предназначение
AllStudents	students	Задаваемый пользователем список студентов
FoundedStudents	students	Список найденных одногруппников.
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 GetCount - запрашивает у пользователя количество студентов в передаваемом списке. Возвращает true, если пользователь передал корректное значение, иначе возвращается false. Для работы подпрограммы необходима константа nn.

function GetCount(var paramVar :word):boolean;

В теле функции в цикле с постусловием происходит запрос у пользователя целого числа. Запрошенное значение сначала считывается в буфер, потом производится попытка его извлечения (функцией val). Если попытка была успешной, а также это число лежит в промежутке [0,nn], функция завершает работу и передает указанное значение и значение log, равное true . Если нет, то производится запрос пользователя о продолжении работы. Если пользователь ответил 'N', функция завершает работу и возвращает значение log, равное false и значение параметра - 0. Иначе операция повторяется до достижения двух вышеуказанных условий.

Используемые функцией параметры-переменные представленны - в таблице 4, локальные переменные - в таблице 5.

Таблица 4 - Параметры-переменные функции ввода количества студентов

имя	**тип**	**предназначение**
paramVar	word	Количество студентов в списке

Таблица 5 - Локальные переменные функции ввода ввода количества студентов

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.
code	integer	Код ошибки извлечения значения из буфера;код 0 - нет ошибки.
temp	word	Временная переменная для хранения запрошенного параметра.

2. Функция GetString запрашивает у пользователя строковый параметр - фамилию студента, имя и т.д. Описание параметра и его название(или подсказка)

передается через параметры функции. Возвращает true, если пользователь передал корректное значение, иначе возвращается false.

function GetString(var paramVar :string; const paramDescr, paramTip:string):boolean;

В теле функции в цикле с постусловием происходит запрос у пользователя вещественного числа, причем сначала выводится переданная информация о параметре. Если введенная строка не пустая, функция завершает работу и передает указанное значение и значение log, равное true . Если нет, то производится запрос пользователя о продолжении работы. Если пользователь ответил 'N', функция завершает работу и возвращает значение log, равное false и пустую строку. Иначе операция повторяется до достижения двух вышеуказанных условий.

Используемые функцией параметры-константы приведены в таблице 6; параметры-переменные - в таблице 7, локальные переменные - в таблице 8.

Таблица 6 - Параметры-константы функции ввода строкового параметра

имя	**тип**	**предназначение**
paramDescr	string	Описание запрашиваемого параметра.
paramTip	string	Краткая подсказка о параметре.

Таблица 7 - Параметры-переменные функции ввода строкового параметра

имя	**тип**	**предназначение**
paramVar	string	Запрашиваемый параметр.

Таблица 8 - Локальные переменные функции ввода строкового параметра

имя	**тип**	**предназначение**
ch	char	Содержит ответ пользователя на запросы программы о повторении ввода данных.
temp	string	Временная переменная для хранения запрошенного параметра.

3. Функция GetGroup запрашивает у пользователя номер группы студента. Возвращает true, если пользователь передал корректное значение, иначе возвращается false.

function GetGroup(var paramVar :longint):boolean;

В теле функции в цикле с постусловием происходит запрос у пользователя целого числа. Запрошенное значение сначала считывается в буфер, потом производится попытка его извлечения (функцией val). Если попытка была успешной, а также это число больше 0, функция завершает работу и передает указанное значение и значение log, равное true . Если нет, то производится запрос пользователя о продолжении работы. Если пользователь ответил 'N', функция завершает работу и возвращает значение log, равное false и значение параметра - 0. Иначе операция повторяется до достижения двух вышеуказанных условий.

Используемые функцией параметры-переменные приведены в таблице 9, локальные переменные - в таблице 10.

Таблица 9 - Параметры-переменные функции ввода номера группы

имя	**тип**	**предназначение**
paramVar	longint	Номер группы студента.

Таблица 10 - Локальные переменные функции ввода номера группы

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.
code	integer	Код ошибки извлечения значения из буфера;код 0 - нет ошибки.
temp	longint	Временная переменная для хранения запрошенного параметра.

4. Функция GetStudent запрашивает у пользователя информацию о конкретном студенте. Возвращает true, если пользователь передал корректное значение, иначе возвращается false. Должен быть доступен тип “student”.

function GetStudent(var stud:student):boolean;

В теле функции через GetString запрашиваются имя, фамилия, отчество, вуз студента и через GetGroup - номер группы студента. Если на каком-то этапе возникла ошибка, то последующие этапы не выполняются и возвращается false, иначе возвращается true.

Используемые функцией параметры-переменные приведены в таблице 11, локальные переменные - в таблице 12.

Таблица 11 - Параметры-переменные функции ввода произвольного параметра расчета корня

имя	тип	предназначение
stud	student	Данные о студенте.

Таблица 12 - Локальные переменные функции ввода произвольного параметра расчета корня

имя	тип	предназначение
log	boolean	Указывает на отсутствие ошибок (значение true) ввода данных.

5. Функция InputData вводит необходимые данные - количество студентов и данные об этих студентах; возвращает true, если не было ошибки ввода, иначе возвращается false. Функция InputData использует функции GetStudent и GetCount, поэтому они должны быть глобальными по отношению к InputData. Должен быть доступен тип “student”.

function InputData(var studCount:word; var students:array of student):boolean;

Сначала запрашивается количество студентов с помощью GetCount. Если не возникло ошибки, то в цикде с предусловием запрашивается указанное количество студентов через GetStudent, пока не будет введено нужное количество студентов или не возникнет ошибки. Если ошибок не было, возвращается true, иначе - false.

Используемые функцией параметры-переменные приведены в таблице 13; локальные переменные - в таблице 14.

Таблица 13 - Параметры-переменные функции ввода данных для поиска одногруппников

имя	тип	предназначение
studCount	word	Количество студентов в списке.
stud	array of student	Данные о студентах.

Таблица 14 - Локальные переменные функции ввода данных для поиска одногруппников

имя	тип	предназначение
i	word	Переменная-счетчик для обработки массива.
log	boolean	Указывает на отсутствие ошибок (значение true) ввода данных.

6. Функция CheckData позволяет организовать проверку пользователем введенных им ранее значений. Возвращает true, если пользователь подтвердил правильность данных. Должен быть доступен тип “student”. Функция использует GetStudent , поэтому она должны быть глобальными по отношению к CheckData.

function CheckData(const studCount:word;var stud:array of student):boolean;

Функция сначала спрашивает пользователя, желает ли он продолжить работу без проверки, проверить данные, или завершить программу. Если он ответил 'A', подпрограмма завершается, если он ответил 'V', то перебираются в цикле с постусловием студенты из массива stud, где выводятся данные о текущем студенте и предлагаются варианты: считать данные правильными, завершить программу или изменить данные. Если пользователь ответил 'R', то данные вводятся заново с помощью функции GetStudent. Это продолжается, пока студенты не закончатся, либо пользователь не ответит 'A', означающее завершение работы. В конце возвращается true, если пользователь не отвечал 'A', иначе возвращается false.

Используемые функцией параметры-константы приведены в таблице 15; локальные переменные - в таблице 16.

Таблица 15 - Параметры-константы функции проверки пользователем введенных значений

имя	**тип**	**предназначение**
studCount	word	Количество студентов в списке.
stud	array of student	Данные о студентах.

Таблица 16 - Локальные переменные функции проверки пользователем введенных значений

имя	**тип**	**предназначение**
ch	char	Содержит ответ пользователя на запрос программы о правильности данных.
i	word	Переменная-счетчик для обработки массива.

7. Процедура FindStudents находит одногруппников в переданном списке студентов. Должен быть доступен тип “student”. Для работы подпрограммы необходима константа nn.

procedure FindStudents(const inStudCount:word;const inStud:array of student;

var outStudCount:word;var outStud:array of student);

Сначала все элементы массива outed устанавливаются в false, такое же значение устанавливается и для founded. k устанавливается в 0. Затем в цикле для i от 1 до InStudCount-1 проверяется, не “выбыл” ли этот студент - не был ли он уже найден как одногрупник, т. е. не равно ли outed[i] true. Если нет то, то проходим j от i+1 до inStudCount, опять-таки проверяем не равно ли outed[j] true, и если нет, то если имя вуза и номер группы студентов i и j, то устанавливается founded в true, увеличивается на 1 k, inStud[j] сохраняется в outStud[k]. После завершения цикла j, если founded равно true, то founded сбрасывается в false, k инкрементируется и inStud[i] сохраняется в outStud[k]. После окончания цикла i k созраняется в outStudCount.

Используемые процедурой параметры-константы приведены в таблице 17, параметры-переменные - в таблице 18, локальные переменные - в таблице 19 .

Таблица 17 - Параметры-константы процедуры поиска одногруппников

имя	**тип**	**предназначение**
inStudCount	word	Количество студентов в общем списке.
inStud	array of student	Данные о студентах общего списка.

Таблица 18 - Параметры-переменные процедуры поиска одногруппников

имя	**тип**	**предназначение**
outStudCount	word	Количество найденных одногруппников.
outStud	array of student	Данные о найденных одногруппниках.

Таблица 19 - Локальные переменные процедуры поиска одногруппников

имя	**тип**	**предназначение**
k	word	Временная переменная, для хранения количества найденных одногруппников.
i,j	word	Переменные-счетчики для обработки массива.
founded	boolean	Указывает на то, что одногруппник был найден.
outed	array [1..nn] of boolean	Указывает, был ли уже студент отмечен как одногруппник.

8. Процедура PrintData выводит список одногруппников на экран. Должен быть доступен тип “student”.

procedure PrintData(const studCount:word;const stud:array of student);

Сначала процедура выводит все данные о первом студенте, потом в цикле перебираются все студенты, и если его вуз и группы отличны от предыдушего, то они выводятся на экран, иначе выводятся только фамилия, имя и отчество студента.

Используемые процедурой параметры-константы приведены в таблице 20, локальные переменные - в таблице 21 .

Таблица 20 - Параметры-константы процедуры вывода списка одногруппников

имя	**тип**	**предназначение**
StudCount	word	Количество найденных одногруппников.
Stud	array of student	Данные о найденных одногруппниках.

Таблица 21 - Локальные переменные процедуры вывода списка одногруппников

имя	**тип**	**предназначение**
i	word	Переменная-счетчик для обработки массива.

Текст программы

Ниже представлен текст программы, написанной на языке Turbo Pascal 7, которая находит среди данных студентов одногруппников.

```
{-----ПРОГРАММА ПОИСКА ОДНОГРУПНИКОВ-----}
program FindStudents;
const nn=10;
type
    student=record
        FstName,LstName,Surname:string[20];
        University:string[50];
        GroupNum:longint;
    end;

{-----ВЫВОД СТРОКОВОГО ПАРАМЕТРА-----}
function GetString(var paramVar :string; const paramDescr,paramTip:string
                                                                    ):boolean;
var ch:char; temp:string;
begin
    ch:='y';
    repeat
        WriteLn(paramDescr);
        Write(paramTip,':');
        ReadLn(temp);
        if temp='' then begin
            WriteLn('Ошибка! Повторить ввод?(Y/N)');
            ReadLn(ch);
        end
    until (UpCase(ch)='N') or (temp<>'');
    if temp<>'' then begin
        GetString:=true;
        paramVar:=temp;
    end else begin
        GetString:=false;
        paramVar:='';
    end;
end;
{-----ВВОД НОМЕРА ГРУППЫ-----}
function GetGroup(var paramVar :longint):boolean;
var ch:char; buf:string; code:integer; temp:longint;
begin
    ch:='y';
    repeat
        WriteLn('Введите номер группы.');
```

```
        Write('Номер группы:');
        ReadLn(buf); val(buf,temp,code);
        if code<>0 then begin
            WriteLn('Ошибка! Повторить ввод?(Y/N)');
            ReadLn(ch);
        end else
            if temp<=0 then begin
                code:=-1;
                WriteLn('Номер группы<=0! Повторить ввод?(Y/N)');
                ReadLn(ch);
            end;
    until (UpCase(ch)='N') or (code=0);
    if code=0 then begin
        GetGroup:=true;
        paramVar:=temp;
    end else begin
        GetGroup:=false;
        paramVar:=0;
    end;
end;
{-----ВВОД КОЛИЧЕСТВА СТУДЕНТОВ-----}
function GetCount(var paramVar :word):boolean;
var ch:char; buf:string; code:integer; temp:word;
begin
    ch:='y';
    repeat
        WriteLn('Введите общее количество студентов.(>=1 и <=',nn,')');
        Write('Студентов:');
        ReadLn(buf); val(buf,temp,code);
        if code<>0 then begin
            WriteLn('Ошибка! Повторить ввод?(<Y>es/<N>o)');
            ReadLn(ch);
        end else
            if (temp<1) or (temp>nn) then begin
                code:=-1;
                WriteLn('Количество студентов неправильно!
                          Повторить ввод?(<Y>es/<N>o)');
                ReadLn(ch);
            end;
    until (UpCase(ch)='N') or (code=0);
    if code=0 then begin
        GetCount:=true;
        paramVar:=temp;
    end else begin
```

```
        GetCount:=false;
        paramVar:=0;
    end;
end;
{-----ВВОД ПАРАМЕТРОВ СТУДЕНТА-----}
function GetStudent(var stud:student):boolean;
var log:boolean;
begin
    log:=GetString(stud.LstName,'Введите фамилию студента.','Фамилия');
    if log then
        log:=GetString(stud.FstName,'Введите имя студента.','Имя');
    if log then
        log:=GetString(stud.Surname,'Введите отчество студента.','Отчество');
    if log then
        log:=GetString(stud.University,'Введите название университета.',
                        'Университет');
    if log then
        log:=GetGroup(stud.GroupNum);
    GetStudent:=log;
end;
{-----ВВОД ДВННЫХ-----}
function InputData(var studCount:word; var students:array of student):boolean;
var i:word; log:boolean;
begin
    log:=GetCount(studCount);
    i:=1;
    while (i<=studCount) and log do begin
        WriteLn('****************************');
        WriteLn('Ввод данных о студенте №',i);
        WriteLn('****************************');
        log:=GetStudent(students[i]);
        inc(i);
    end;
    InputData:=log;

end;
{-----ПРОВЕРКА ДАННЫХ-----}
function CheckData(const studCount:word;var stud:array of student):boolean;
var i:word; ch:char;
begin
    WriteLn('Хотите проверить данные о каждом студенте по порядку?
                                                (ответьте <V>erify);');
    WriteLn('завершить работу?<A>bort;');
    WriteLn('продолжить работу?<C>ontinue или любое другое.');
```

```
    ReadLn(ch);
    if UpCase(ch)='V' then begin
        i:=1;
        repeat
            WriteLn('***************************');
            WriteLn('Карточка студента №',i);
            WriteLn('***************************');
            WriteLn('***Университет: ',stud[i].University,' Группа:',
                        stud[i].GroupNum);
            WriteLn(stud[i].LstName,' ', stud[i].FstName,' ', stud[i].SurName);
            WriteLn('Правильно? Да - <Y>es (по умолчанию);');
            WriteLn('Нет, изменить данные - <R>');
            WriteLn('Нет, завершить программу - <A>');
            ReadLn(ch);
            if UpCase(ch)='R' then begin
                WriteLn('***************************');
                WriteLn('Ввод данных о студенте №',i);
                WriteLn('***************************');
                GetStudent(stud[i]);
                {if not GetStudent(students[i]) then UpCase(ch)='A';}
            end else inc(i);
        until (UpCase(ch)='A') or (i>studCount);
    end;
    CheckData:=UpCase(ch)<>'A';
end;
{-----ПОИСК ОДНОГРУППНИКОВ-----}
procedure FindStudents(const inStudCount:word;const inStud:array of student;var
outStudCount:word;var outStud:array of student);
var k,i,j:word; founded:boolean; outed:array [1..nn] of boolean;
begin

    k:=0;
    for i:=1 to nn do
        outed[i]:=false;
    founded:=false;

        for i:=1 to inStudCount-1 do begin
            if not outed[i] then begin
                for j:=i+1 to inStudCount do
                    if not outed[j]
                       and (inStud[i].GroupNum=inStud[j].GroupNum)
                       and (inStud[i].University=inStud[j].University) then begin
                          founded:=true;inc(k);
                          outStud[k]:=inStud[j];
```

```
                        outed[j]:=true;
                    end;
                if founded then begin
                    inc(k);
                    outStud[k]:=inStud[i];
                    founded:=false;
                end;
            end;
        end;

    outStudCount:=k;

end;
{-----ВЫВОД НА ЭКРАН-----}
procedure PrintData(const studCount:word;const stud:array of student);
var i:word;
begin
    WriteLn('Найденные одногрупники:');
    i:=1;

        WriteLn('***Университет: ',stud[i].University,' Группа:',
                    stud[i].GroupNum);
        WriteLn(stud[i].LstName,' ', stud[i].FstName,' ', stud[i].SurName);
        for i:=2 to studCount do begin
            if (stud[i].GroupNum<>stud[i-1].GroupNum)
               or (stud[i].University<>stud[i-1].University) then
                  WriteLn(#10#13,'***Университет: ',stud[i].University,
                              ' Группа:',stud[i].GroupNum);
            WriteLn(stud[i].LstName,' ', stud[i].FstName,' ', stud[i].SurName);
        end;
end;
{тип "студенты"}
type students=record
        Count:word;
        Stud:array[1..nn] of student;
    end;
{-----ОСНОВНАЯ ЧАСТЬ-----}
var AllStudents, FoundedStudents:students; {все студенты, найденные студенты}
    log:boolean; {переменная состояния}
BEGIN
    WriteLn('Программа находит одногрупников срди данных студентов');
    log:=InputData(AllStudents.Count,AllStudents.stud);
    if log then
        log:=CheckData(AllStudents.Count,AllStudents.stud);
```

```
    if log then begin
        FindStudents(AllStudents.Count,AllStudents.Stud,FoundedStudents.Count,Foun
dedStudents.Stud);
        if FoundedStudents.Count<>0 then
            PrintData(FoundedStudents.Count,FoundedStudents.Stud)
        else
            WriteLn('Одногрупников не найдено');
    end else
        Writeln('Программа завершена пользователем');
    Readln;
END.
```

Тестовые примеры

На рисунках 10,11,12 представлен пример работы программы нахождения одногруппников для 10 студентов.

```
Введите общее количество студентов.(>=1 и <=10)
Студентов:10
**************************
Ввод данных о студенте №1
**************************
Введите фамилию студента.
Фамилия:Белым
Введите имя студента.
Имя:Андрей
Введите отчество студента.
Отчество:Андреевич
Введите название университета.
Университет:ТулГУ
Введите номер группы.
Номер группы:220601
**************************
Ввод данных о студенте №2
**************************
Введите фамилию студента.
Фамилия:Богун
Введите имя студента.
Имя:Алексей
Введите отчество студента.
Отчество:Валерьевич
Введите название университета.
Университет:ТулГУ
Введите номер группы.
Номер группы:220601
**************************
Ввод данных о студенте №3
**************************
Введите фамилию студента.
Фамилия:Матвеев
Введите имя студента.
Имя:Артур
Введите отчество студента.
Отчество:Янович
Введите название университета.
Университет:ТулГУ
Введите номер группы.
Номер группы:720201
**************************
Ввод данных о студенте №4
**************************
Введите фамилию студента.
Фамилия:Глаголев
Введите имя студента.
Имя:Владислав
Введите отчество студента.
Отчество:Сергеевич
Введите название университета.
Университет:ТулГУ
Введите номер группы.
Номер группы:720201
```

Рисунок 10— Пример работы программы поиска одногруппников:ввод студентов 1-4

```
**************************
Ввод данных о студенте №5
**************************
Введите фамилию студента.
Фамилия:Кошелев
Введите имя студента.
Имя:Кирилл
Введите отчество студента.
Отчество:Игоревич
Введите название университета.
Университет:МГТУ им. Баумана
Введите номер группы.
Номер группы:220601
**************************
Ввод данных о студенте №6
**************************
Введите фамилию студента.
Фамилия:Пушкин
Введите имя студента.
Имя:Александр
Введите отчество студента.
Отчество:Сергеевич
Введите название университета.
Университет:Лицей
Введите номер группы.
Номер группы:720201
**************************
Ввод данных о студенте №7
**************************
Введите фамилию студента.
Фамилия:Семенов
Введите имя студента.
Имя:Семен
Введите отчество студента.
Отчество:Семенович
Введите название университета.
Университет:НИИИИ
Введите номер группы.
Номер группы:1
**************************
Ввод данных о студенте №8
**************************
Введите фамилию студента.
Фамилия:Антонов
Введите имя студента.
Имя:Антон
Введите отчество студента.
Отчество:Антонович
Введите название университета.
Университет:НИИИИ
Введите номер группы.
Номер группы:1
```

Рисунок 11— Пример работы программы поиска одногруппников:ввод студентов 5-8

```
**************************
Ввод данных о студенте №9
**************************
Введите фамилию студента.
Фамилия:Семенов
Введите имя студента.
Имя:Семен
Введите отчество студента.
Отчество:Семенович
Введите название университета.
Университет:НИИИИ
Введите номер группы.
Номер группы:1
**************************
Ввод данных о студенте №10
**************************
Введите фамилию студента.
Фамилия:Иванов
Введите имя студента.
Имя:Иван
Введите отчество студента.
Отчество:Иванович
Введите название университета.
Университет:НИИИИ
Введите номер группы.
Номер группы:1
Хотите проверить данные о каждом студенте по порядку?(ответьте <V>erify);
завершить работу?<A>bort;
продолжить работу?<C>ontinue или любое другое.
c
Найденные одногрупники:
***Университет: ТулГУ Группа:220601
Богун Алексей Валерьевич
Белым Андрей Андреевич

***Университет: ТулГУ Группа:720201
Глаголев Владислав Сергеевич
Матвеев Артур Янович

***Университет: НИИИИ Группа:1
Антонов Антон Антонович
Семенов Семен Семенович
Иванов Иван Иванович
Семенов Семен Семенович
```

Рисунок 12— Результат работы программы поиска одногруппников:ввод студентов 9-10 и результат

Вывод

В ходе выполнения данной лабораторной работы я научился использовать записи при написании программ. Записи помогают скомпоновать разнородные данные, что помогает при построении моделей объектов реального мира; кроме того т. н. пространства имен позволяют обеспечить более высокую структуризацию программы. Можно считать, что кроме небезопасного преобразования вариантной части в записях с вариантами у указанного вида структур данных практически нет недостатков.