Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Основы алгоритмизации и программирования (ОАиП)

ОТЧЕТ

по лабораторной работе №7

Тема работы: Множества

Выполнил

студент: гр. 151003 Барановский Р.А.

Проверил: Фадеева Е.П.

Минск 2022

Содержание

[1 Постановка задачи 4](#_Toc104204240)

[2 Описание алгоритмов решения задачи 5](#_Toc104204241)

[3 Структура данных 6](#_Toc104204242)

[3.1 Структура типов программы 6](#_Toc104204243)

[3.2 Структура данных основной программы 6](#_Toc104204244)

[3.3 Структура данных алгоритма IsValid 6](#_Toc104204245)

[4 Схема алгоритма решения задачи по ГОСТ 19.701-90 8](#_Toc104204246)

[4.1 Схема алгоритма подпрограммы IsValid 9](#_Toc104204247)

[5 Результаты расчетов программы 14](#_Toc104204248)

[Приложение А 15](#_Toc104204249)

[Приложение Б 19](#_Toc104204250)

# Постановка задачи

Дан некоторый текст, за которым следует точка (в сам текст точка не входит). Определить, является ли этот текст правильной записью Формулы: <Формула> ::= <Терм> | ((<Формула><Знак><Формула>)).

<Знак> ::= + | - | \*.

<Терм> ::= <Имя> | <Целое>.

<Имя> ::= <Буква> | <Имя><Буква> | <Имя><Цифра>.

<Целое> ::= <Цифра> | <Целое><Цифра>.

<Буква> ::= a | b | c | d | e | f | g.

<Цифра> ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9.

# Описание алгоритмов решения задачи

Таблица 1 – Описание алгоритмов решения задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  п.п. | Наименование алгоритма | Назначение  алгоритма | Формальные параметры | Предпологаемый тип реализации |
|  | Основной алгоритм | Ввод строки и проверка правильности. Вызывает следующие подпрограммы:  IsValid(str, res) |  |  |
|  | IsValid(str, res) | Проверяет, является ли строка правильной записью Формулы | str – получает от фактического параметра адрес с защитой;  res – возвращаемый функцией параметр | Функция.  res – возвращаемый параметр |

# Структура данных

## Структура типов программы

Таблица 2 – Структура типов программы

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| TNums | set of '1' .. '9' | Пользовательский тип-множество, содержащий возможные значения цифр |
| TSigns | set of '\*' .. '-' | Пользовательский тип-множество, определяющий возможные значения знаков |
| TLetters | set of 'a' .. 'g' | Пользовательский тип-множество, определяющий возможные значения букв |
| TCondition | (error, none, formula, formulaPlus, sign, name, int) | Пользовательский тип-перечисление, содержащий информацию о типе выражения в определенный момент |

## Структура данных основной программы

Таблица 3 – Структура данных основной программы

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| inputStr | string | Вводимая пользователем строка |
| isCorrect | boolean | Результат проверки выражения на соответствие |

## Структура данных алгоритма IsValid

Таблица 4 – Структура данных алгоритма IsValid(str, res)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| str | string | Проверяемая строка | Формальный |
| res | boolean | Результат проверки | Формальный |
| condition | TCondition | Тип выражения в определенный момент | Локальный |
| i | boolean | Счетчик цикла | Локальный |
| signs | TSigns | Множество знаков | Локальный |

Продолжение таблицы 4

|  |  |  |  |
| --- | --- | --- | --- |
| nums | TNums | Множество цифр | Локальный |
| letters | TLetters | Множество букв | Локальный |

# Схема алгоритма решения задачи по ГОСТ 19.701-90

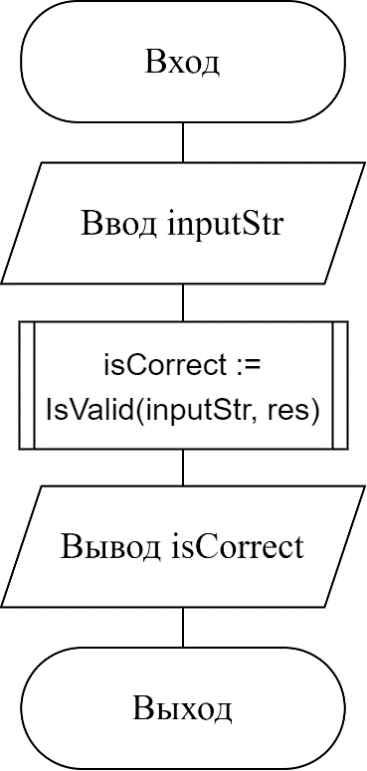


Рисунок 1 – Схема алгоритма решения задачи по ГОСТ 19.701-90

## Схема алгоритма подпрограммы IsValid

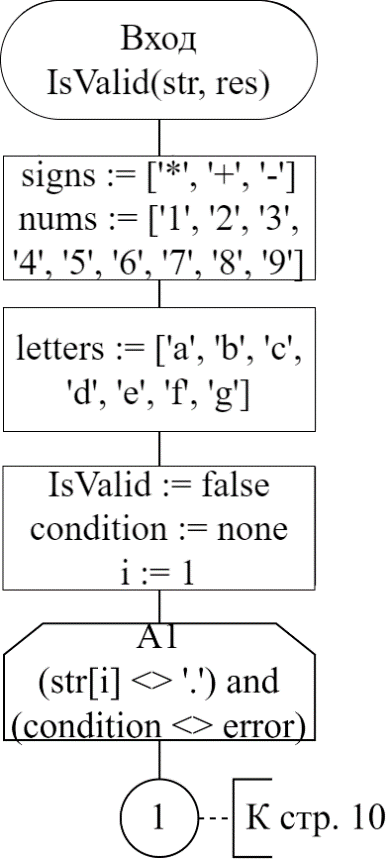


Рисунок 2 – Схема алгоритма IsValid (часть 1)

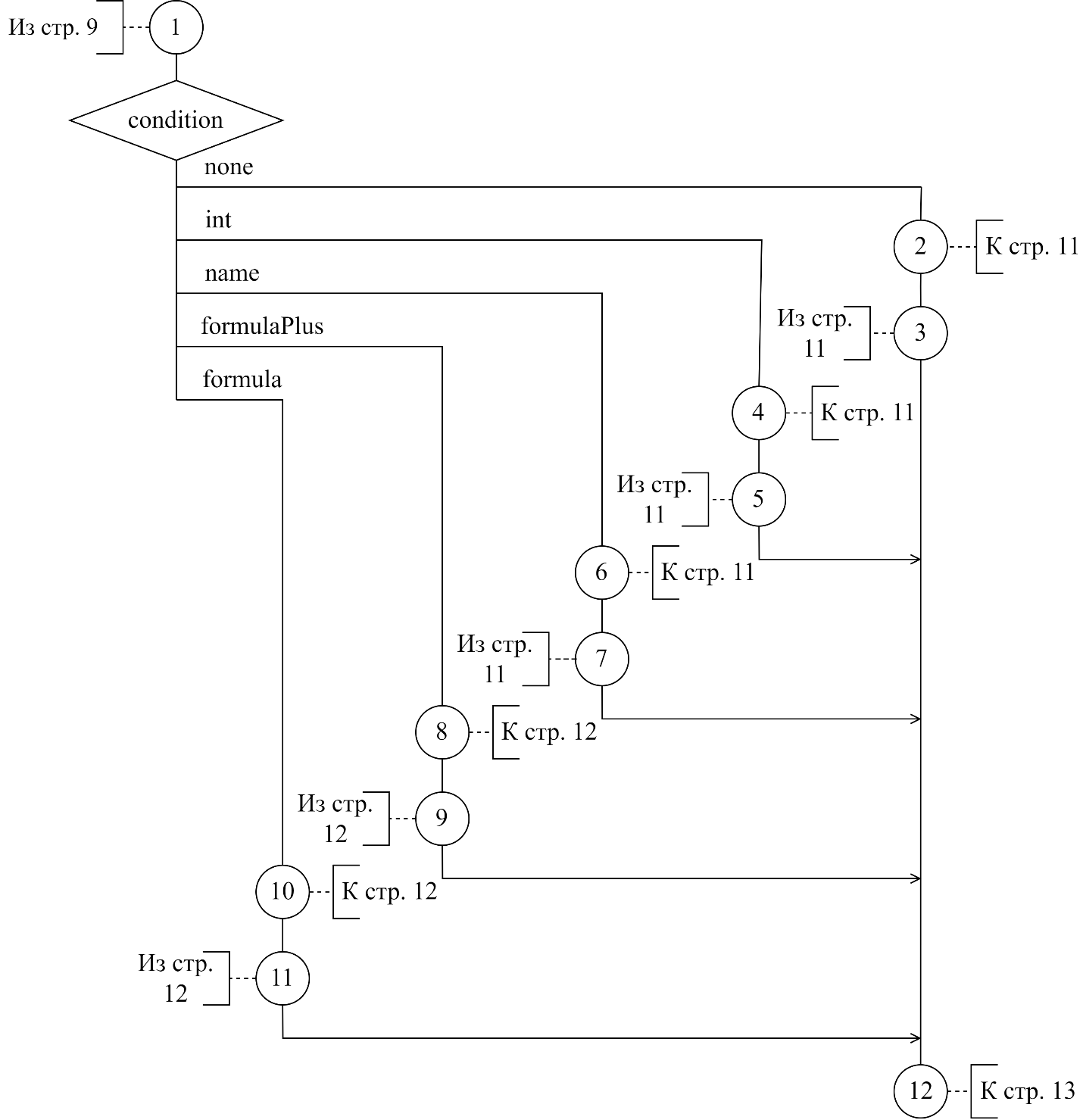


Рисунок 3 – Схема алгоритма IsValid (часть 2)

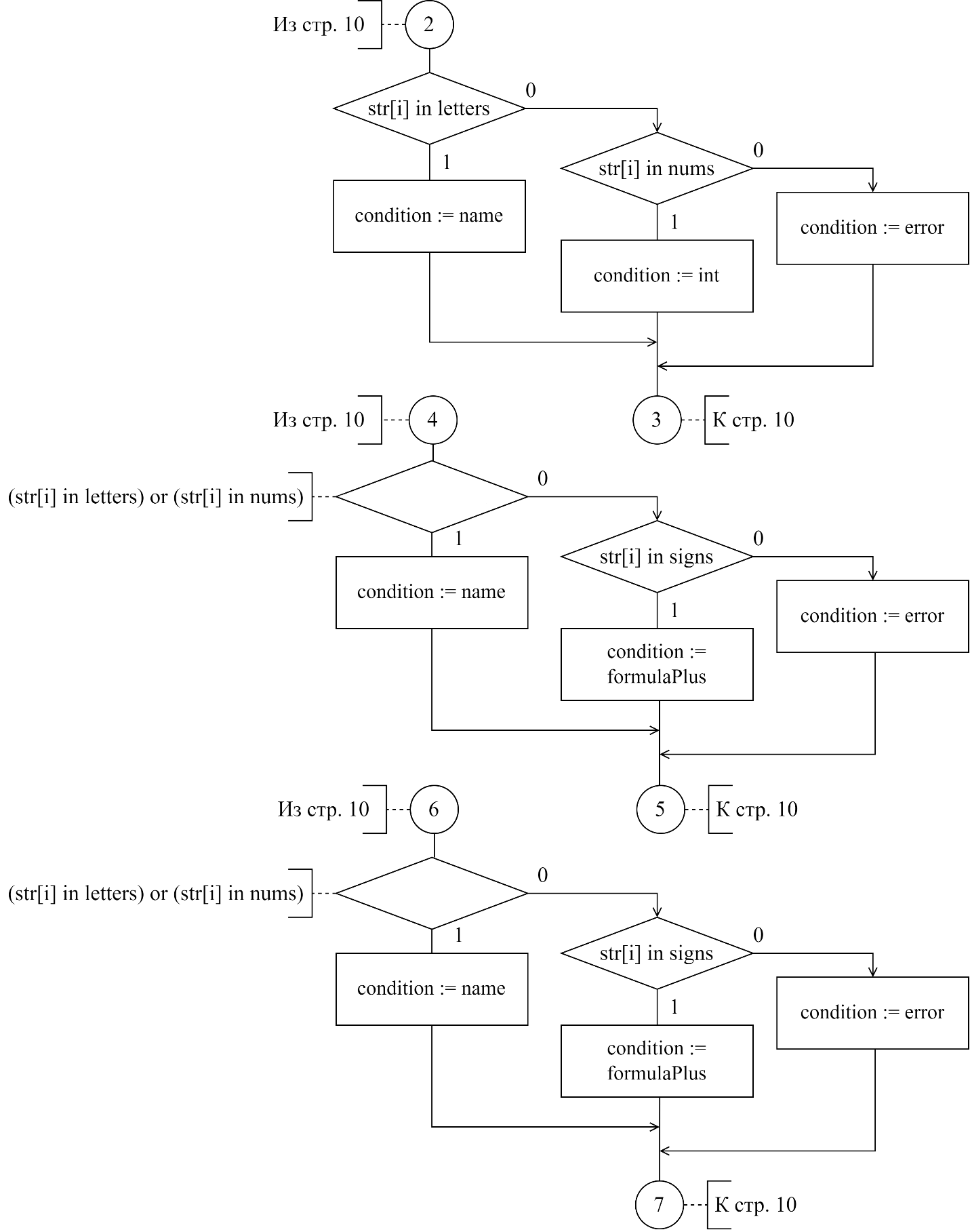


Рисунок 4 – Схема алгоритма IsValid (часть 3)

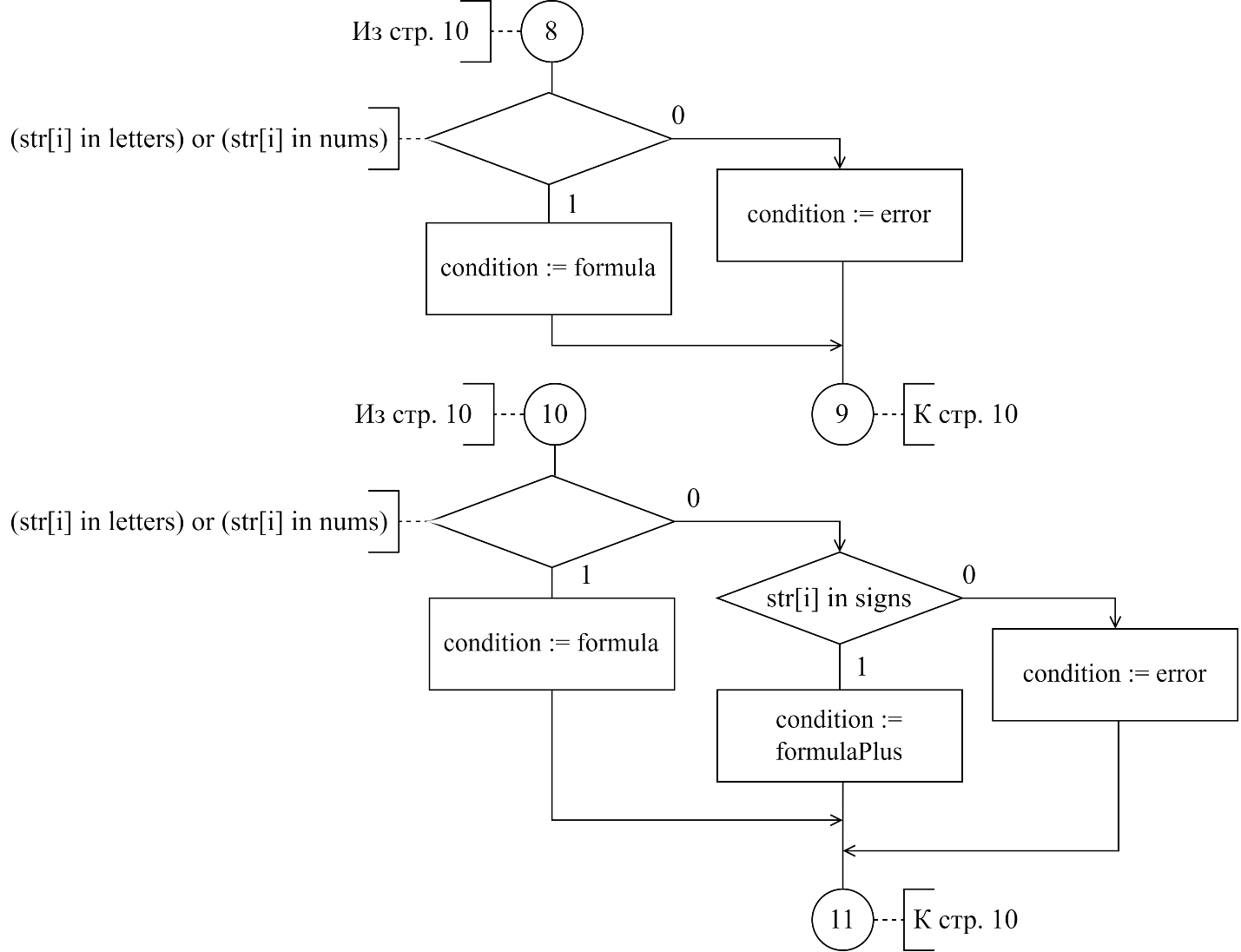


Рисунок 5 – Схема алгоритма IsValid (часть 4)

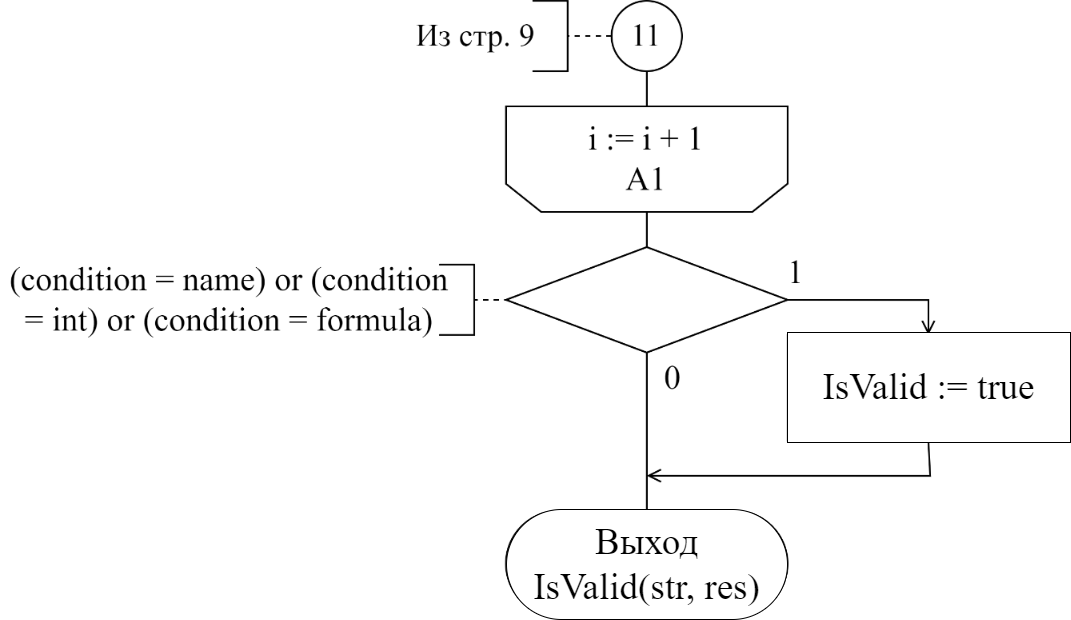


Рисунок 6 – Схема алгоритма IsValid (часть 5)

# Результаты расчетов программы



Рисунок 7 – Результат выполнения программы

Приложение А

(обязательное)

Исходный код программы

program Lab7;

// A program that determines if this text is a

// valid formula

// Console app

{$APPTYPE CONSOLE}

// Modules declaration

uses

System.SysUtils;

// A function which verifies text

function IsValid(const str: string): boolean;

// Types declaration

type

TNums = set of '1' .. '9';

TSigns = set of '\*' .. '-';

TLetters = set of 'a' .. 'g';

TCondition = (error, none, formula, formulaPlus,

sign, name, int);

// Const declaration

const

signs: TSigns = ['\*', '+', '-'];

nums: TNums = ['1', '2', '3', '4', '5', '6', '7',

'8', '9'];

letters: TLetters = ['a', 'b', 'c', 'd', 'e',

'f', 'g'];

// Vars declaration

var

i: integer;

condition: TCondition;

begin

// Setting variables to the original values

IsValid := false;

condition := none;

i := 1;

// Start of the cycle A1, checking every element

// and changing condition type

while (str[i] <> '.') and (condition <> error) do

begin

case condition of

// To start determining

none:

begin

if str[i] in letters then

condition := name

else if str[i] in nums then

condition := int

else

condition := error

end;

// If the condition only contains numbers

int:

begin

if str[i] in nums then

condition := int

else if str[i] in signs then

condition := formulaPlus

else

condition := error

end;

// If the condition is a name

name:

begin

if (str[i] in letters) or (str[i] in nums)

then

else

if str[i] in signs then

condition := formulaPlus

else

condition := error

end;

// If the condition is a formula with something

formulaPlus:

begin

if str[i] in letters then

condition := formula

else

if str[i] in nums then

condition := formula

else

condition := error;

end;

// If the condition is a finished formula

formula:

begin

if (str[i] in letters) or (str[i] in nums)

then

condition := formula

else

if str[i] in signs then

condition := formulaPlus

else

condition := error

end

end;

// Finish of the cycle A1

inc(i);

end;

// The condition is a formula if it only contains

//numbers or its a name

if (condition = name) or (condition = int)

or (condition = formula) then

IsValid := true;

end;

// Variables declaration

var

inputStr: string;

isCorrect: boolean;

{ inputStr - string the program verifies

isCorrect - is inputted expression correct}

begin

//Input string

write('Input condition: ');

readln(inputStr);

//Check string

isCorrect := IsValid(inputStr);

// Outputting the result of the verification

write('Is correct: ');

writeln(isCorrect);

//Final operations

readln;

end.

Приложение Б

(обязательное)

Тестовые наборы

Таблица 5 – Тестовые наборы

|  |  |  |
| --- | --- | --- |
| Тест | Исходные данные и ожидаемый результат | Полученный результат |
| 1. | Ожидается: TRUE |  |
| 2. | Ожидается: TRUE |  |
| 3. | Ожидается: TRUE |  |
| 4. | Ожидается: TRUE |  |
| 5. | Ожидается: TRUE |  |
| 6. | Ожидается: FALSE |  |
| 7. | Ожидается: FALSE |  |
| 8. | Ожидается: FALSE |  |
| 9. | Ожидается: FALSE |  |
| 10. | Ожидается: TRUE |  |
| 11. | Ожидается: FALSE |  |
| 12. | Ожидается: FALSE |  |