|  |  |
| --- | --- |
| Министерство образования Республики Беларусь | |
| Учреждение образования | |
| БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ | |
| ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ | |
|  | |
|  | |
| Факультет компьютерных систем и сетей | |
| Кафедра программного обеспечения информационных технологий | |
| Дисциплина: **Название дисциплины (АББРЕВИАТУРА)** | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
| **ОТЧЁТ** | |
| по лабораторной работе № **X** | |
|  | |
| Тема работы: **Название темы** | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
| Выполнил: | Фамилия И.О. |
|  | гр. **XXXXXX** |
|  | Вариант **X** |
|  |  |
| Проверил: | Фамилия И.О. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| Минск **202X** | |

содержание

[1 Постановка задачи 3](#_Toc105841147)

[2 Методика решения задачи 4](#_Toc105841148)

[2.1 Использование рекурсии 4](#_Toc105841149)

[2.2 Применение свойств булевых операций 4](#_Toc105841150)

[3 Описание алгоритмов решения задачи 5](#_Toc105841151)

[4 Структура данных 6](#_Toc105841152)

[4.1 Структура данных программы 6](#_Toc105841153)

[4.2 Структура данных алгоритма UnoOp 6](#_Toc105841154)

[4.3 Структура данных алгоритма DuoOp 6](#_Toc105841155)

[5 Схема алгоритма решения задачи по ГОСТ 19.701-90 7](#_Toc105841156)

[5.1 Схема алгоритма решения задачи 7](#_Toc105841157)

[5.2 Схема алгоритма DuoOp 8](#_Toc105841158)

[5.3 Схема алгоритма UnoOp 9](#_Toc105841159)

[6 Результаты расчетов 10](#_Toc105841160)

[Приложение А 11](#_Toc105841161)

[Приложение Б 13](#_Toc105841162)

# Постановка задачи

**Постановка 1 (изначальная)**

Во входном файле записано без ошибок логическое выражение следующего вида:

<Логическое\_выражение>::= True | False | (<Операция>(<Операнды>)).

<Операция>::= Not | And | Or.

<Операнды>::= <Операнд> {, <Операнд>}.

<Операнд>::= <Логическое\_выражение>.

Ввести это выражение и вычислить его значение. Например, результатом вычисления выражения And(Or(False,Not(False)),True,Not(True)) будет False.

**Постановка 2 (реализованная)**

Во входном файле записано без ошибок логическое выражение следующего вида:

<Логическое\_выражение>::= True | False | (<Операция>(<Операнды>)).

<Операция>::= Not | And | Or.

<Операнды>::= <Операнд> {, <Операнд>}.

<Операнд>::= <Логическое\_выражение>.

Ввести это выражение и вычислить его значение. Например, результатом вычисления выражения And(Or(False,Not(False)),True,Not(True)) будет False. При реализации учесть:

* пробелы, запятые или их отсутствие не должны влиять на выполнение программы;
* скобки должны быть обязательно прописаны;
* программа должна выполняться в том числе и со вторым уровнем вложенности, то есть операция внутри операции.

# Методика решения задачи

## Использование рекурсии

Согласно постановке, необходимо использовать рекурсию – вызов процедурами самих себя.

В лабораторной работе будет иметь место замена сложных выражений более простыми до тех пор, пока все сложные выражения конкретного вида не будут упрощены. При этом процедура замены сложного выражения будет вызывать саму себя, пока позиция сложного выражения в строке будет более нуля.

## Применение свойств булевых операций

Известно, что And(False, …) всегда будет равняться False, поскольку это доминирующий операнд данной операции.

Известно, что Or(True, …) всегда будет равняться True, поскольку это доминирующий операнд данной операции.

# Описание алгоритмов решения задачи

Таблица 1 – Описание алгоритмов решения задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  п.п | Наименование алгоритма | Назначение алгоритма | Формальные параметры | Предполагаемый тип реализации |
| 1. | Основной алгоритм | Вызывает следующие подпрограммы:  UnoOp, DuoOp |  |  |
| 2. | UnoOp(Sus) | Осуществляет замену всех унарных выражений эквивалентными им операндами, внося изменения в строку Sus | Sus – получает значение от фактического параметра. | Процедура |
| 3. | DuoOp(Sus) | Осуществляет замену всех бинарных выражений эквивалентными им операндами, внося изменения в строку Sus | Sus – получает значение от фактического параметра. | Процедура |

# Структура данных

## Структура данных программы

Таблица 2 – Структура данных

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| Sus | String | Строка |

## Структура данных алгоритма UnoOp

Таблица 3 – Структура данных алгоритма UnoOp(Sus)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| Sus | String | Выражение | Формальный |
| OpPos | Integer | Позиция операции | Локальный |

## Структура данных алгоритма DuoOp

Таблица 4 – Структура данных алгоритма DuoOp(Sus, Op, Value)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| Sus | String | Выражение | Формальный |
| Op, Value | String | Операция и её доминирующий операнд | Формальный |
| OpPos | Integer | Позиция операции | Локальный |
| LB, RB | Integer | Позиции скобок | Локальный |
| Part1, Part2 | String | Сохранённые части строки для определения позиции скобок | Локальный |

# Схема алгоритма решения задачи по ГОСТ 19.701-90

## Схема алгоритма решения задачи

|  |
| --- |
|  |
| Рисунок – Схема алгоритма решения задачи |

## Схема алгоритма DuoOp

|  |
| --- |
|  |
| Рисунок 2 – Схема алгоритма DuoOp |

## Схема алгоритма UnoOp

|  |
| --- |
|  |
| Рисунок 3 – Схема алгоритма UnoOp |

# Результаты расчетов

Вследствие результатов программы мы получаем следующие результаты:

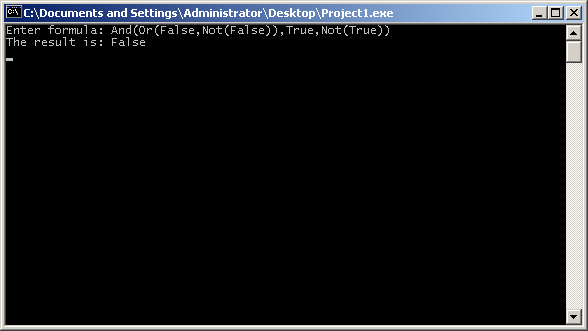


Рисунок 4 – Результаты

Приложение А

(обязательное)

Исходный код программы

Program Lab11;

{Program calcs boolean exprs}

{$APPTYPE CONSOLE}

Uses

SysUtils;

Var

Sus: String;

//This procedure calcs unar ops

//Sus - our string

Procedure UnoOp(Var Sus: String);

Var

OpPos: Integer;

Begin

//Find op pos

OpPos:= Pos('Not', Sus);

If (OpPos > 0) Then

Begin

If (Sus[OpPos + 4] = 'T') Then

Begin

//Replace first type

Delete(Sus, OpPos, Length('Not(True)'));

Insert('False', Sus, OpPos);

End

Else

Begin

//Replace second type

Delete(Sus, OpPos, Length('Not(False)'));

Insert('True', Sus, OpPos);

End;

UnoOp(Sus);

End;

End;

//This procedure calcs binar ops

//Sus - our string

Procedure DuoOp(Var Sus: String; Const Op, Value: String);

Var

OpPos, LB, RB: Integer;

Part1, Part2: String;

Begin

OpPos:= Pos(Op, Sus);

If (OpPos > 0) Then

Begin

//Define content between brackets

LB:= OpPos + Length(Op);

Part1:= Copy(Sus, LB, Length(Sus) - LB + 1);

RB:= Pos(')', Part1);

Part2:= Copy(Part1, 1, RB);

//Replace with equal operand

If Pos(Value, Part2) > 0 Then

Begin

Delete(Sus, OpPos, Length(Part2) + Length(Op));

Insert(Value, Sus, OpPos)

End;

DuoOp(Sus, Op, Value);

End;

End;

Begin

Write('Enter formula: ');

Readln(Sus);

UnoOp(Sus);

//Define priority

If Pos('And', Sus) > Pos('Or', Sus) Then

Begin

DuoOp(Sus, 'And', 'False');

DuoOp(Sus, 'Or', 'True');

End

Else

Begin

DuoOp(Sus, 'Or', 'True');

DuoOp(Sus, 'And', 'False');

End;

WriteLn('The result is: ', Sus);

ReadLn;

End.

Приложение Б

(обязательное)

Тестовые наборы

Ситуация 1 (только операнд)

Введено: True

Ожидание: True

Реальность: True



Ситуация 2 (только унарная операция)

Введено: Not(False)

Ожидание: True

Реальность: True



Ситуация 3 (только бинарная операция)

Введено: And(True, False)

Ожидание: True

Реальность: True



Ситуация 4 (бинарные и унарные операции)

Введено: Or(False, Not(False))

Ожидание: True

Реальность: True



Ситуация 5 (вторая вложенность, нестрогая простановка запятых и пробелов)

Введено: Or(False,Not(False)And(True,False))

Ожидание: True

Реальность: True

