Министерство образования Республики Беларусь

Учреждение образования

Белорусский государственный университет информатики и

Радиоэлектроники

Факультет информационных технологий и управления

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №1

по курсу “Логические основы интеллектуальных систем”

Вариант 4

Выполнил:

Студент гр. 221701 Телица И. Д.

Проверил: Ивашенко В. П.

Минск

2024

**Тема:**

Представление и синтаксическая проверка формул языка логики высказываний.

**Цель:**

Приобрести навыки программирования алгоритмов синтаксического разбора формул языка логики высказываний.

**Задание:**

Проверить, является ли формула сокращенного языка логики высказываний нейтральной.

**Дополнительно:**

Предусмотреть работу системы в режиме тестирования знаний пользователя.

**Грамматика языка логики высказываний:**

<константа> ::=1|0

<символ> ::=A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z

<отрицание> ::= !

<конъюнкция> ::= /\

<дизъюнкция> ::= \/

<импликация> ::= ->

<эквиваленция> ::= ~

<открывающая скобка> ::= (

<закрывающая скобка> ::= )

<бинарная связка> ::= <конъюнкция>|<дизъюнкция>|<импликация>|<эквиваленция>

<атомарная формула> ::= <латинская заглавная буква>

<унарная сложная формула> ::= <открывающая скобка><отрицание><формула><закрывающая скобка>

<бинарная сложная формула> ::= <открывающая скобка><формула><бинарная связка><формула><закрывающая скобка>

<формула> ::=<логическая константа> |<атомарная формула>|<сложная формула>

Невыполнимая формула -- формула, ложная на всех её интерпретациях.

**Блок-схемы функций программы:**

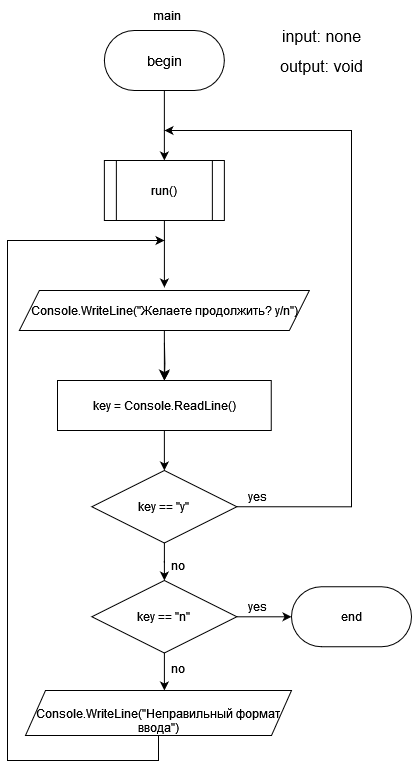


Рис 1. Функция main().

Рис 2. Функция run().

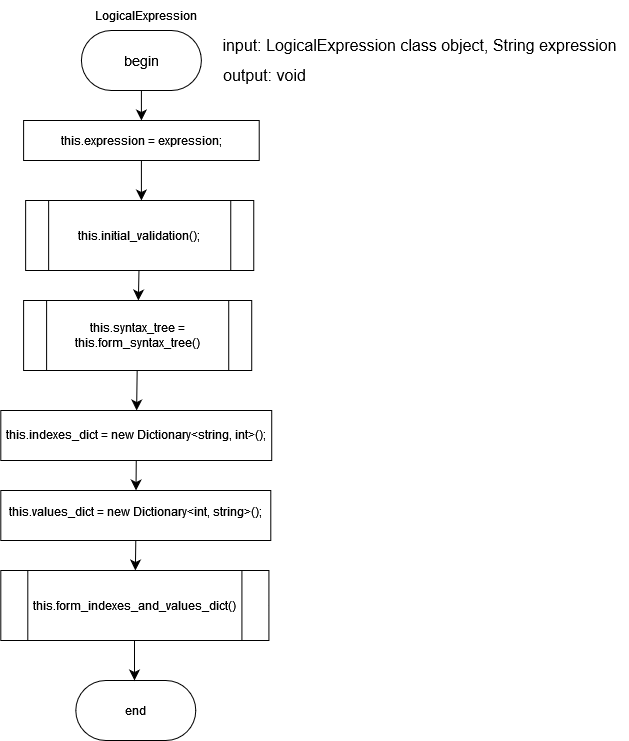
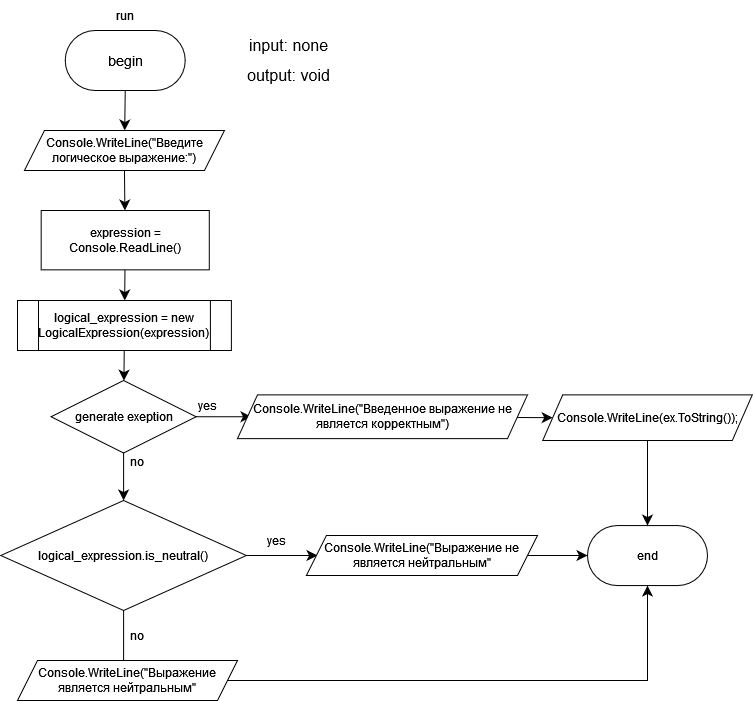


Рис 3. Метод LogicalExpression() класса LogicalExpression.

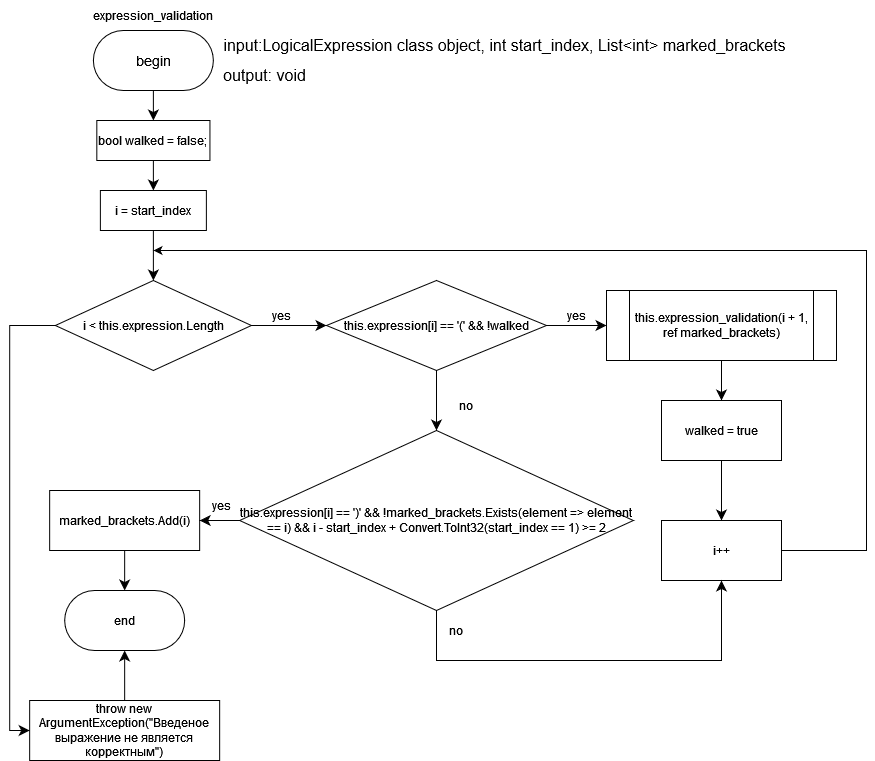


Рис 4. Метод expression\_validation класса LogicalExpression.

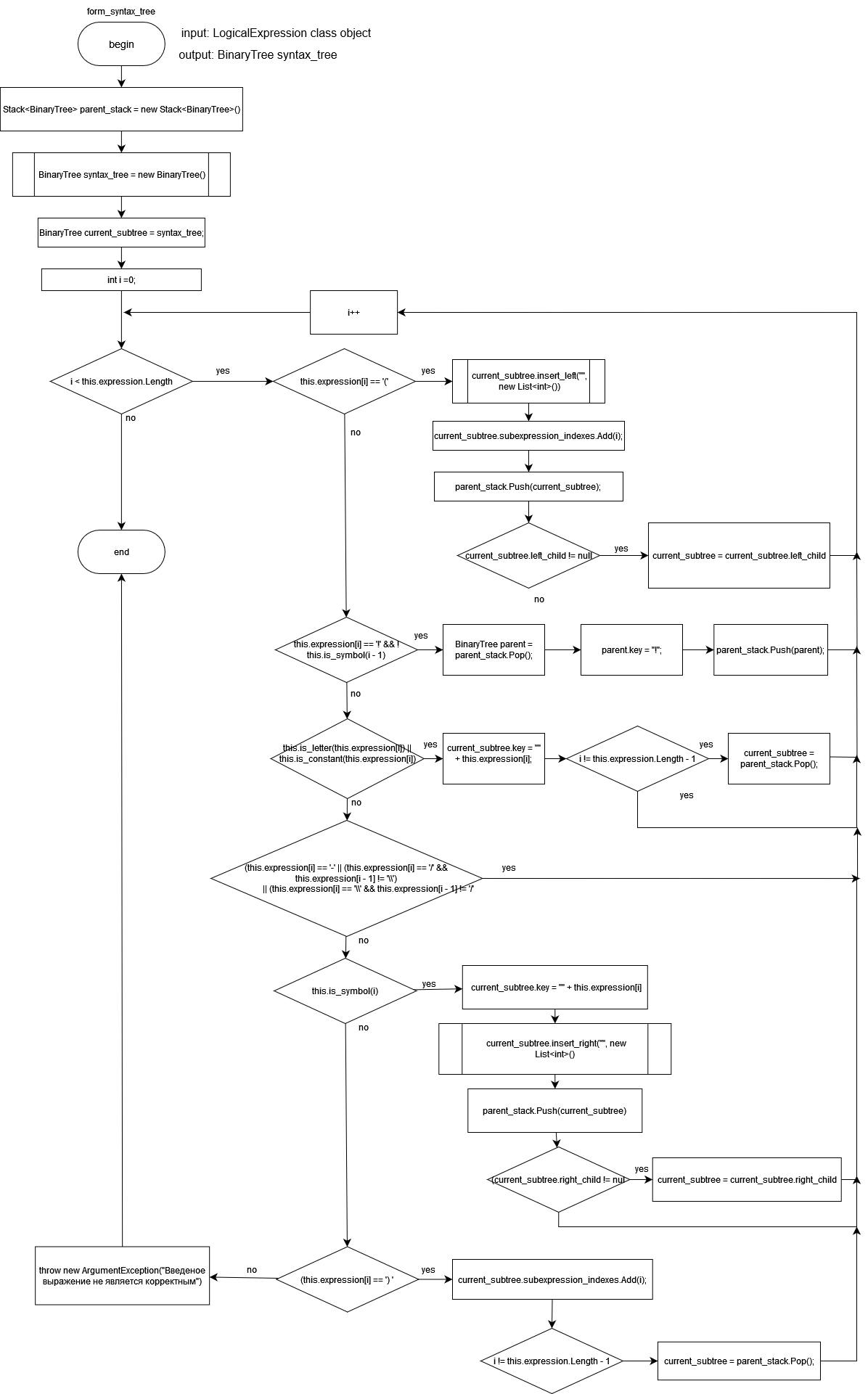


Рис 5. Метод form\_syntax\_tree класса LogicalExpression.

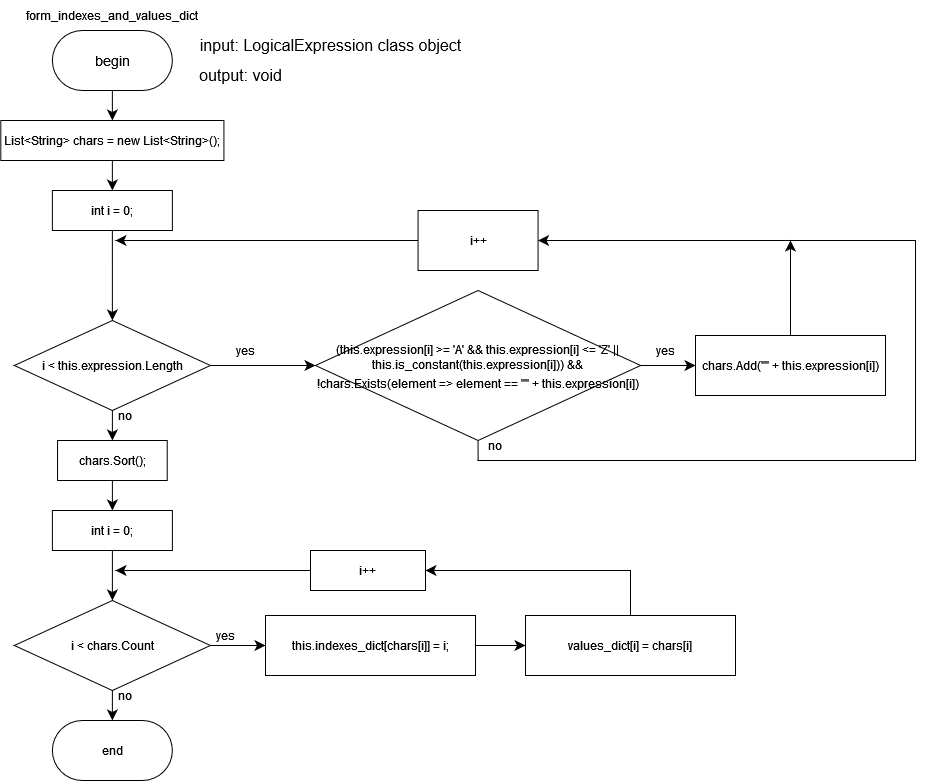


Рис 6. Метод form\_indexes\_and\_values\_dict класса LogicalExpression.

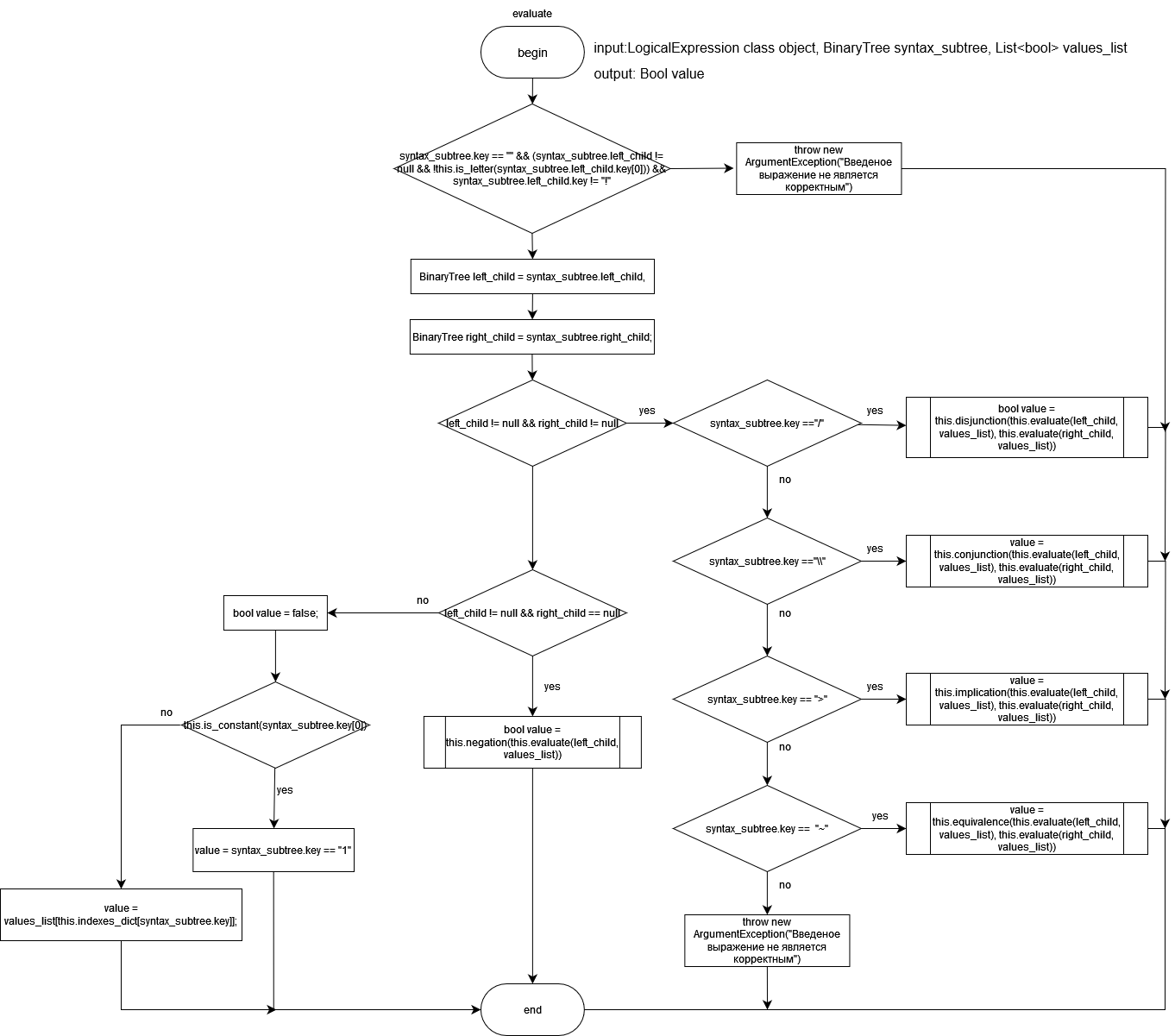


Рис 7. Метод evaluate класса LogicalExpression.

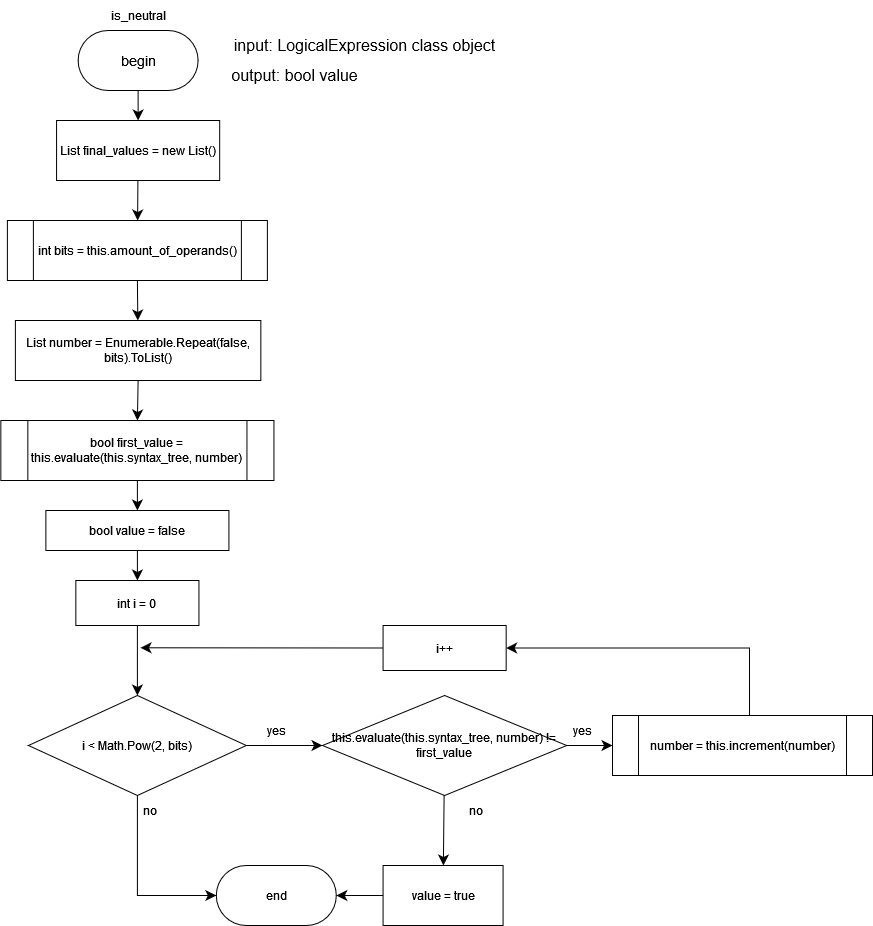


Рис 8. Метод is\_neutral() класса LogicalExpression.

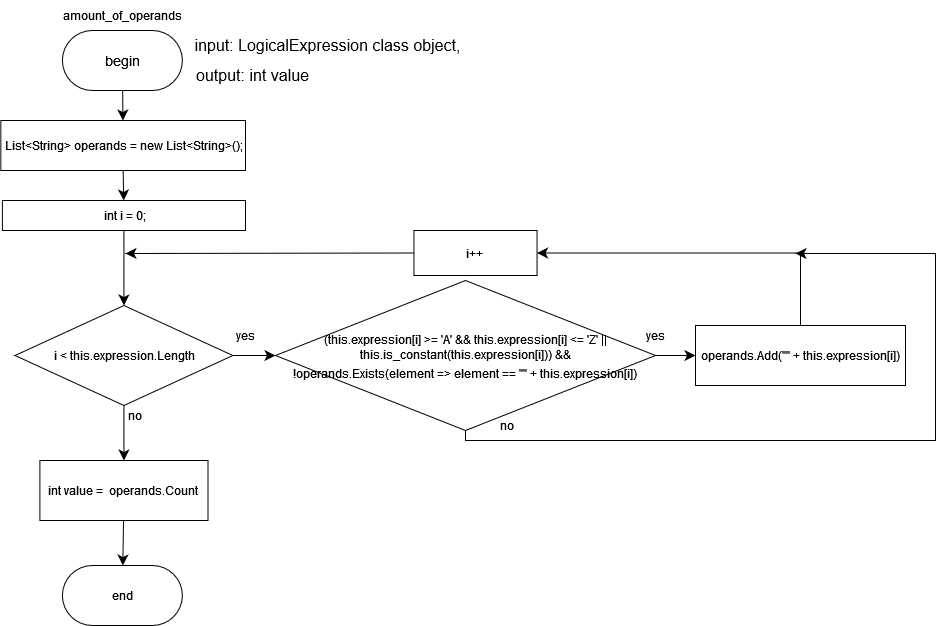


Рис 9. Метод amount\_of\_operands() класса LogicalExpression.

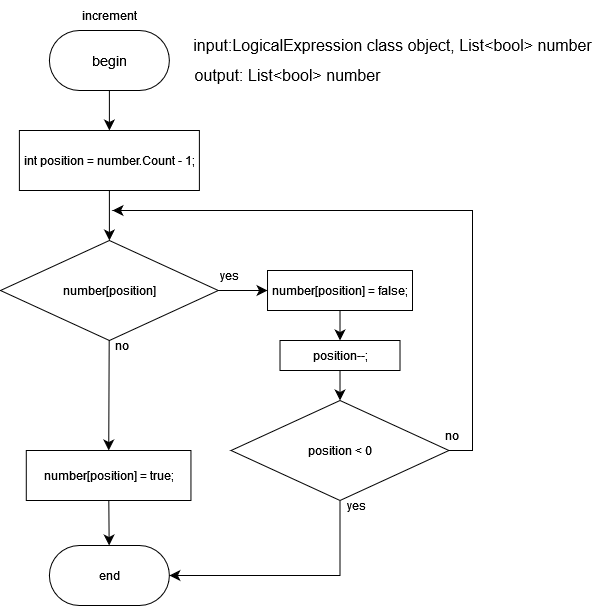


Рис 10. Метод increment() класса LogicalExpression.

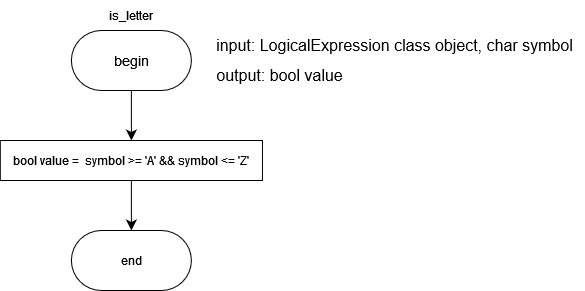


Рис 11. Метод is\_letter() класса LogicalExpression

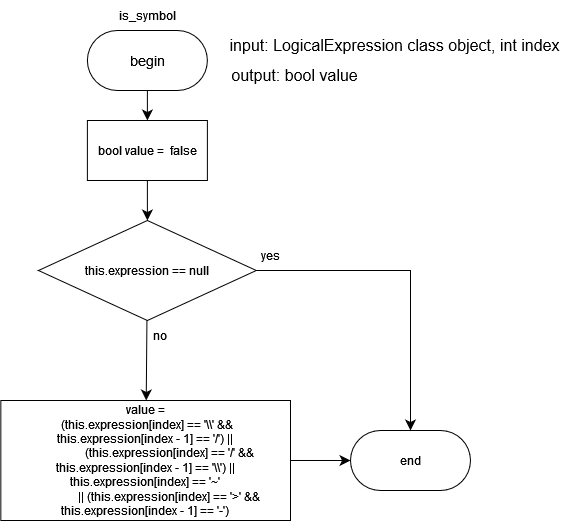


Рис 12. Метод is\_symbol() класса LogicalExpression.

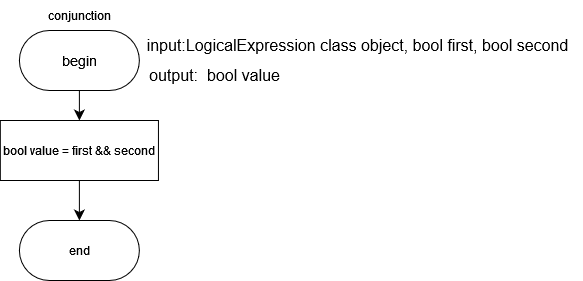


Рис 13. Метод conjunction() класса LogicalExpression.

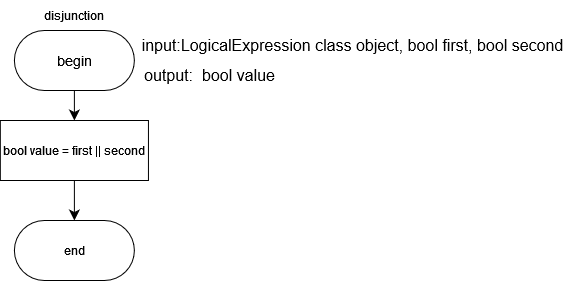


Рис 14. Метод disjunction() класса LogicalExpression.

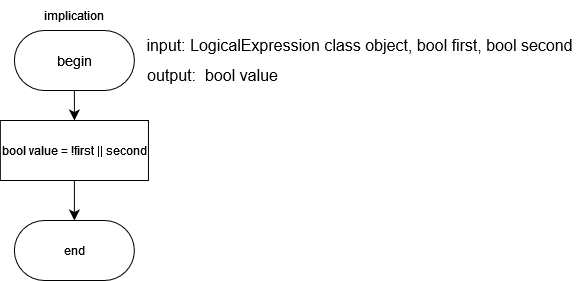


Рис 15. Метод implication() класса LogicalExpression.

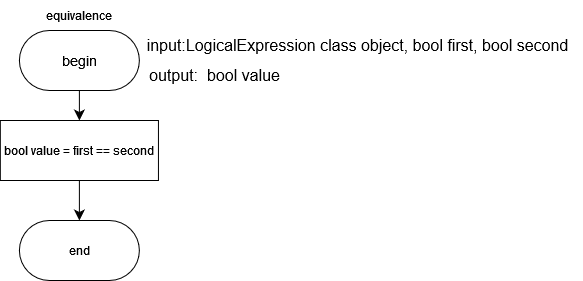


Рис 16. Метод equivalence() класса LogicalExpression.



Рис 17. Метод negation() класса LogicalExpression.

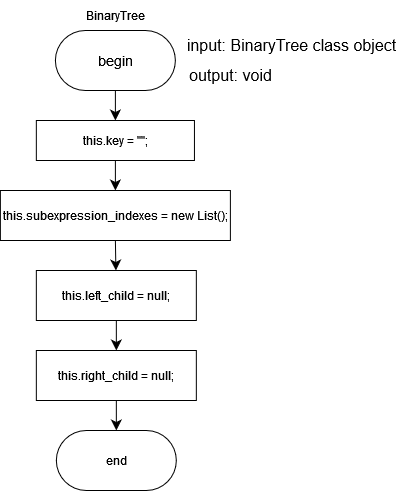


Рис 18. Метод BinaryTree() класса BinaryTree.

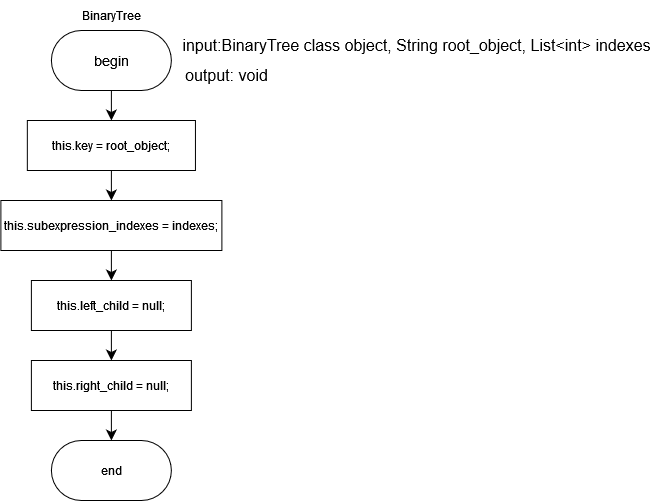


Рис 19. Метод BinaryTree() (перегруженный) класса BinaryTree.

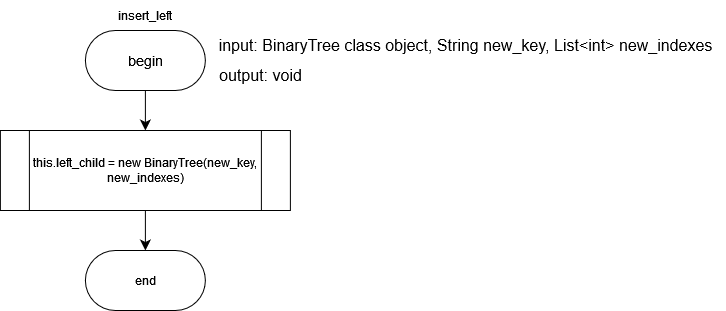


Рис 20. Метод insert\_left() класса BinaryTree.



Рис 21. Метод insert\_right() класса BinaryTree.

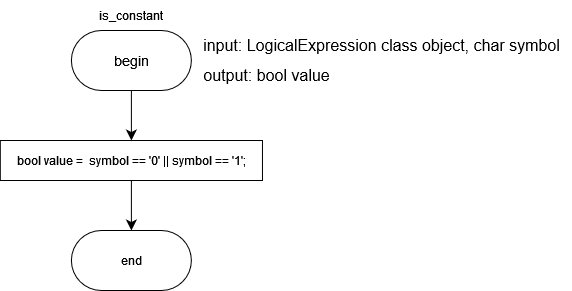


Рис 22. Метод is\_constant() класса LogicalExpression.

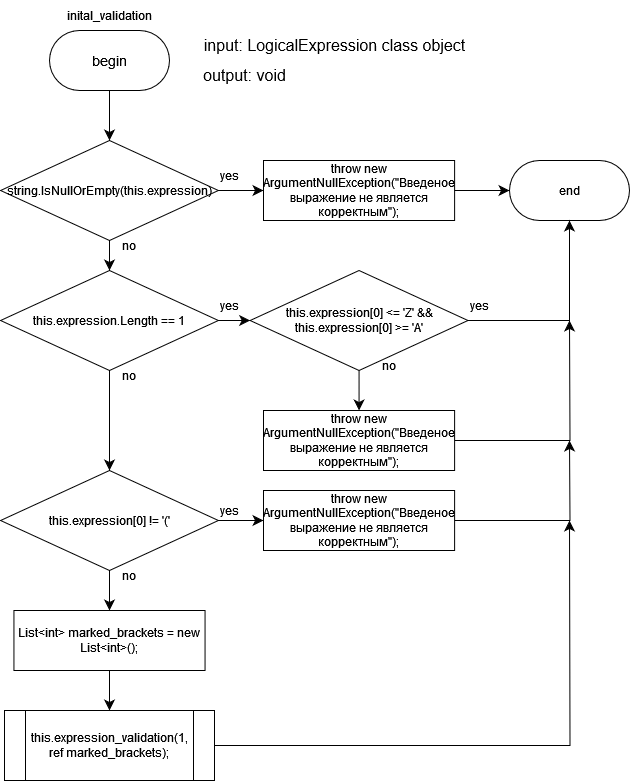


Рис 22. Метод inital\_validation() класса LogicalExpression.

**Выполнение:**

Данная лабораторная работа выполнялась в бригаде, состоящей из двух человек: Телица И.Д. (221701) и Карпук М.В. (221701).

Для выполнения данной лабораторной работы был использован язык программирования C#. Основой кода является синтаксическое бинарное дерево, где каждый узел представляет собой логическую связку. В методах использовались структуры данных список, строка, словарь, стэк.

После запуска программы программа запрашивает пользователя ввести логическую формулу.



Рис 23. Ввод формулы.

Если пользователь ввел некорректную формулу, то программа выведет сообщение об ошибке, после чего спросит у пользователя, желает ли он продолжить работу:



Рис 24. Вывод сообщения об ошибке.



Рис 25. Предложение продолжить.

Если пользователь ввел корректную формулу, то программа выведет сообщение, является ли логическая формула нейтральной или нет, после которого спросит желает ли пользователь продолжить работу:

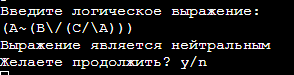


Рис 26. Вывод сообщения результата проверки.

**Примеры:**



Рис 27. Пример 1



Рис 28. Пример 2



Рис 29. Пример 3

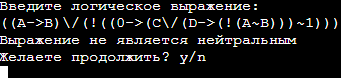


Рис 30. Пример 4

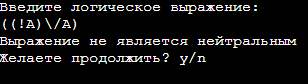
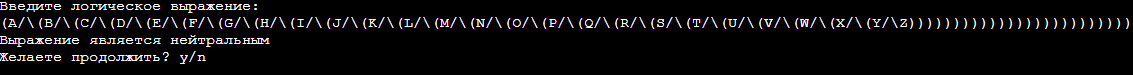


Рис 31. Пример 5



Рис 32. Пример 6

Рис 33. Пример 7

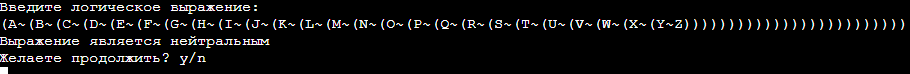


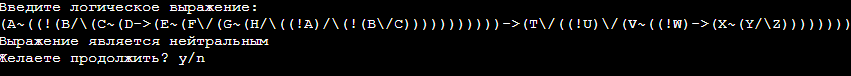
Рис 34. Пример 8

Рис 35. Пример 9

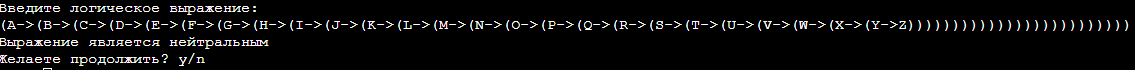


Рис 36. Пример 10

**Вывод:**

В ходе работы были приобретены навыки программирования алгоритмов синтаксического разбора формул языка логики высказываний.

Также была реализована процедурная программа, синтаксического разбора формул языка логики высказываний, проверяющая, является ли формула сокращенного языка логики высказываний нейтральной.

**Теоретические сведения были взяты из следующих источников:**

1. Логические основы интеллектуальных систем. Практикум: учебно-методическое пособие / В. В. Голенков, В. П. Ивашенко, Д. Г. Колб, К. А. Уваров. – Минск: БГУИР, 2011.