

Учреждение образования  
“БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ”

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №2  
по курсу «ЛОИС»  
на тему: «Преобразование и интерпретация формул языка логики высказы-  
вания»  
Вариант 6

Выполнил студент группы  
821703:

Шкут Р.В.

Проверил:

Ивашенко В.П.

МИНСК  
2021

## Тема

Преобразование и интерпретация формул языка логики высказывания.

## Цель

Приобрести навыки программирования алгоритмов синтаксического разбора формул языка логики высказываний.

## Вариант 6

Проверить являются ли формулы равносильными.

## Дополнительные теоретические сведения

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> ::=  $\wedge$

<дизъюнкция> ::=  $\vee$

<импликация> ::=  $\rightarrow$

<эквиваленция> ::=  $\sim$

<открывающая скобка> ::= (

<закрывающая скобка> ::= )

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

<атом> ::= <символ>

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

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

<бинарная связка><формула><закрывающая скобка>

<формула> ::= <константа> | <атом> | <унарная сложная формула> |

<бинарная сложная формула>

## Программная реализация

В рамках лабораторной работы стандартными средствами языка Java был реализован алгоритм, позволяющий проверить являются ли формулы равносильными. Суть алгоритма заключается в первичной проверке формул и дальнейшей проверке равносильности выбранных формул через получение таблиц истинности множеств противоречивых формул, формируемых и вычисляемых в соответствии с правилами проверки обоюдного следования формул друг из друга.

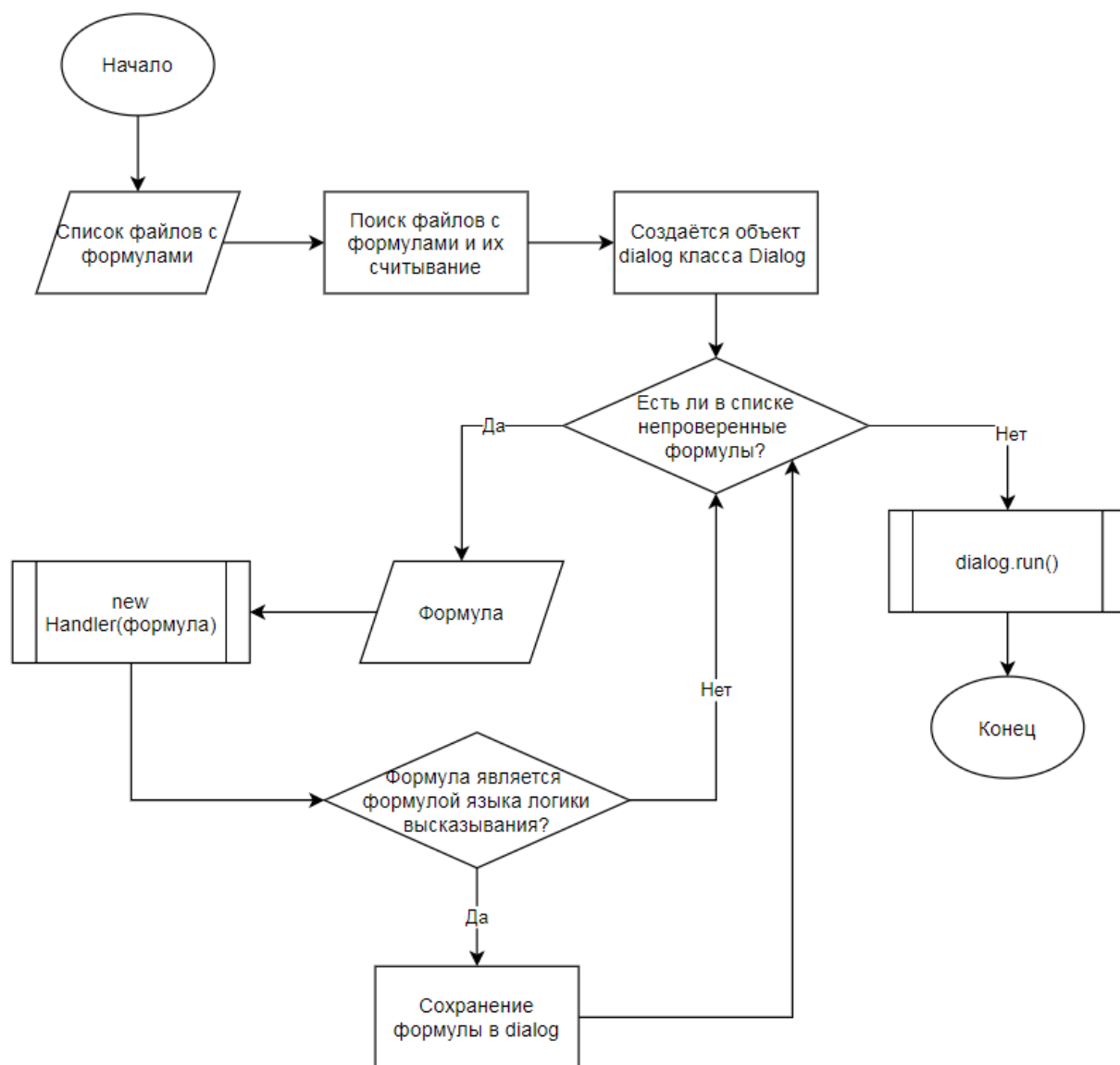


Рис. 1 – Блок-схема функции Main

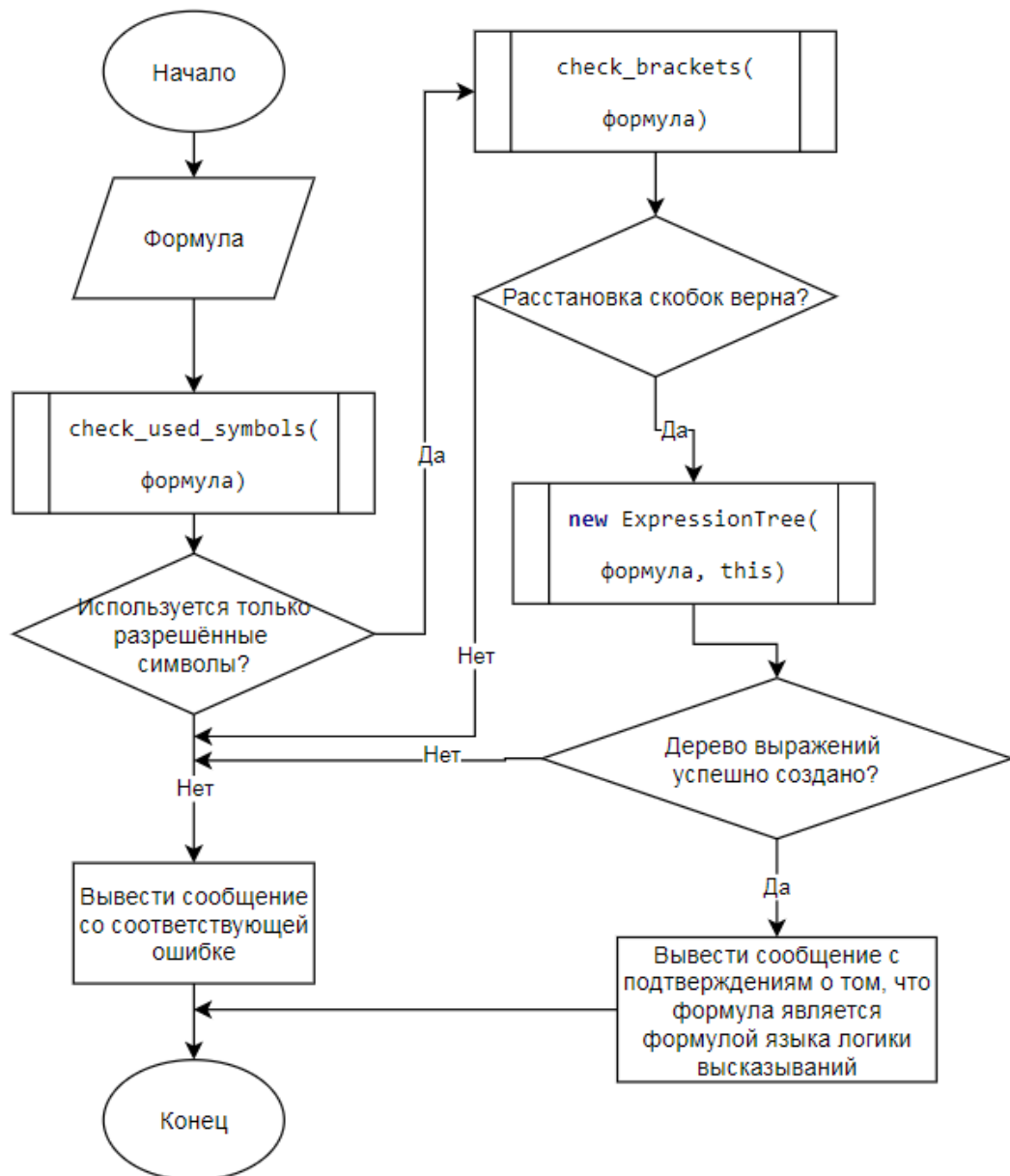


Рис. 2 – Блок-схема конструктора класса Handler

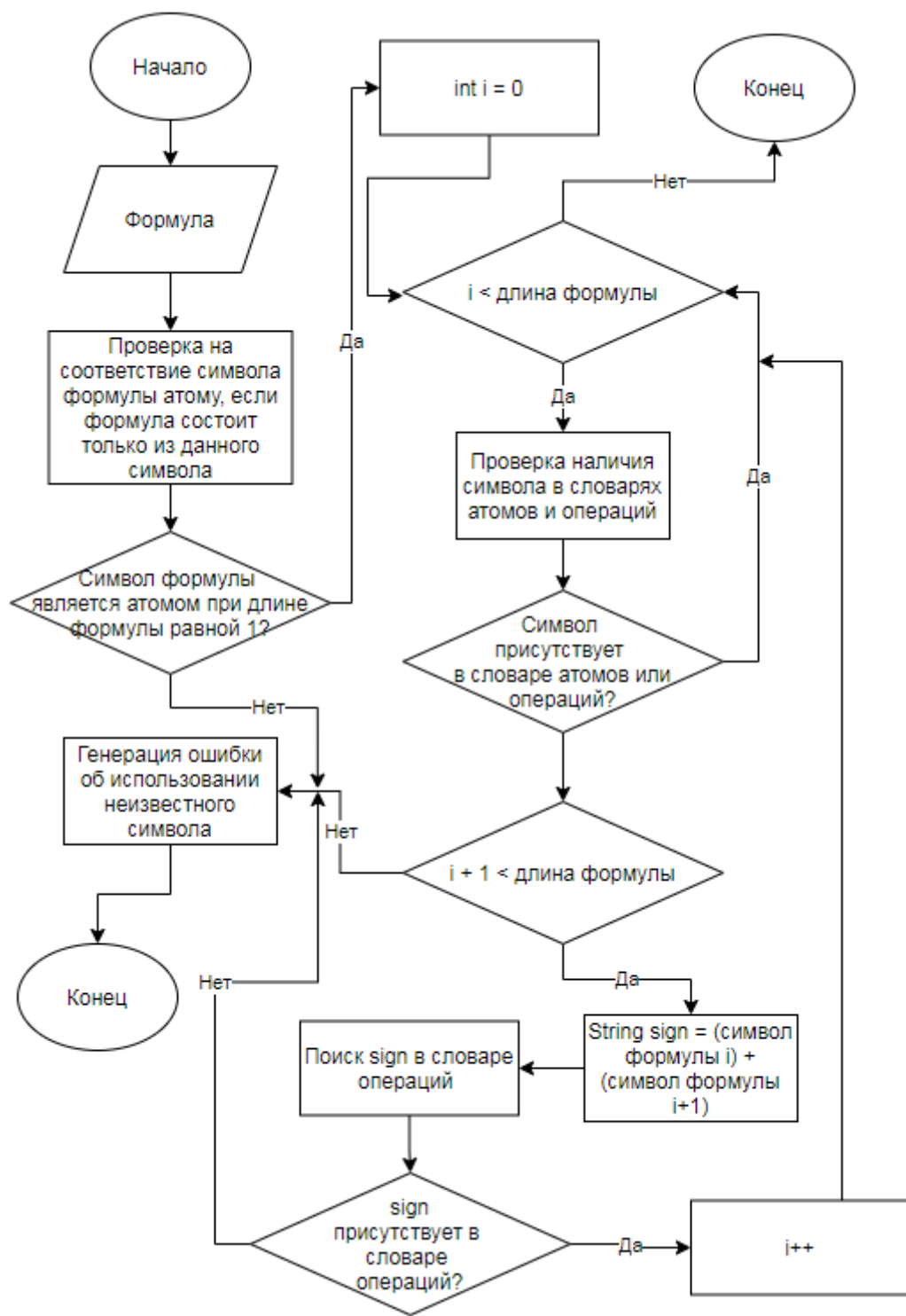


Рис. 3 – Блок-схема функции check\_used\_symbols

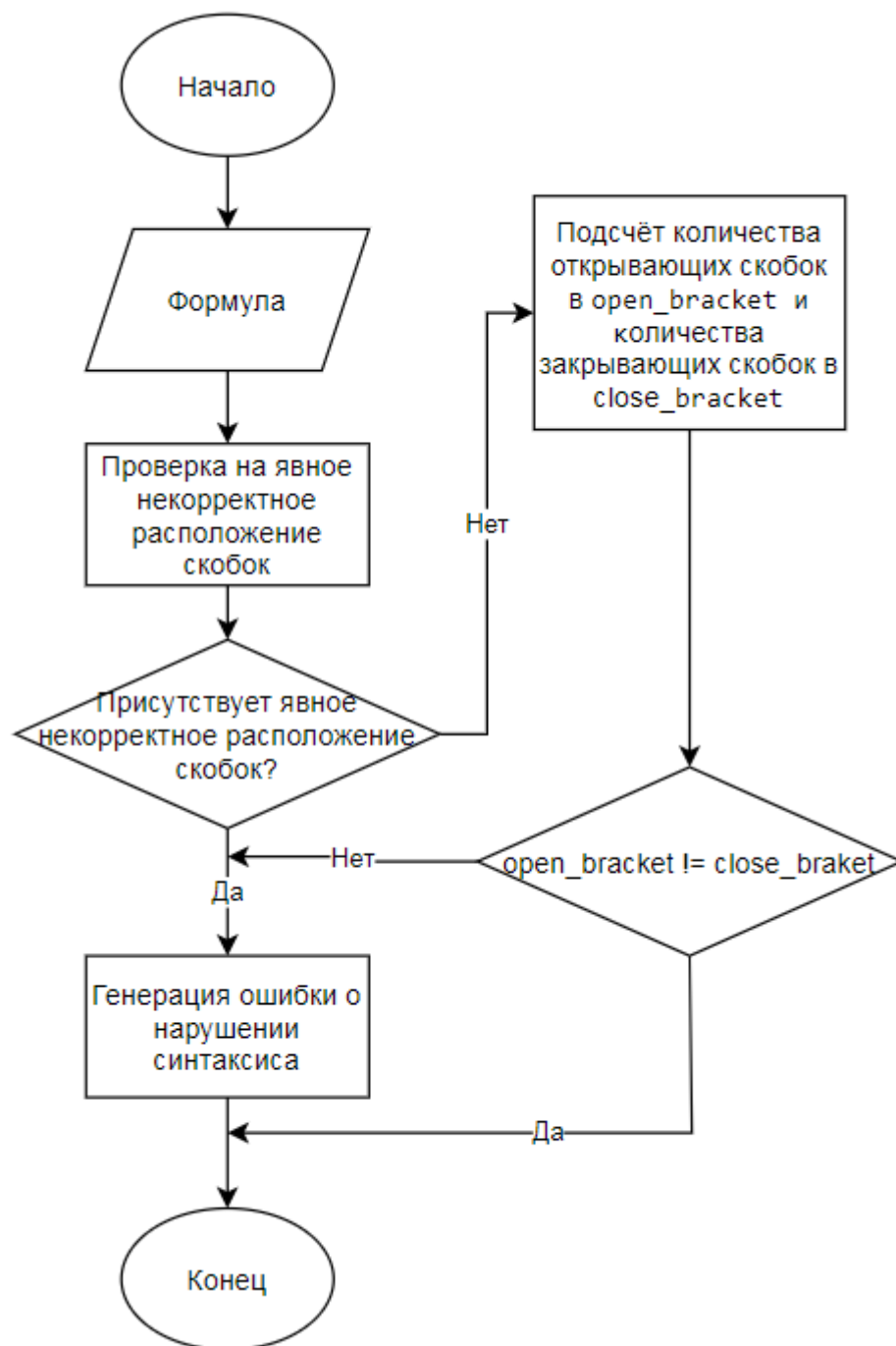


Рис. 4 – Блок-схема функции check\_brackets

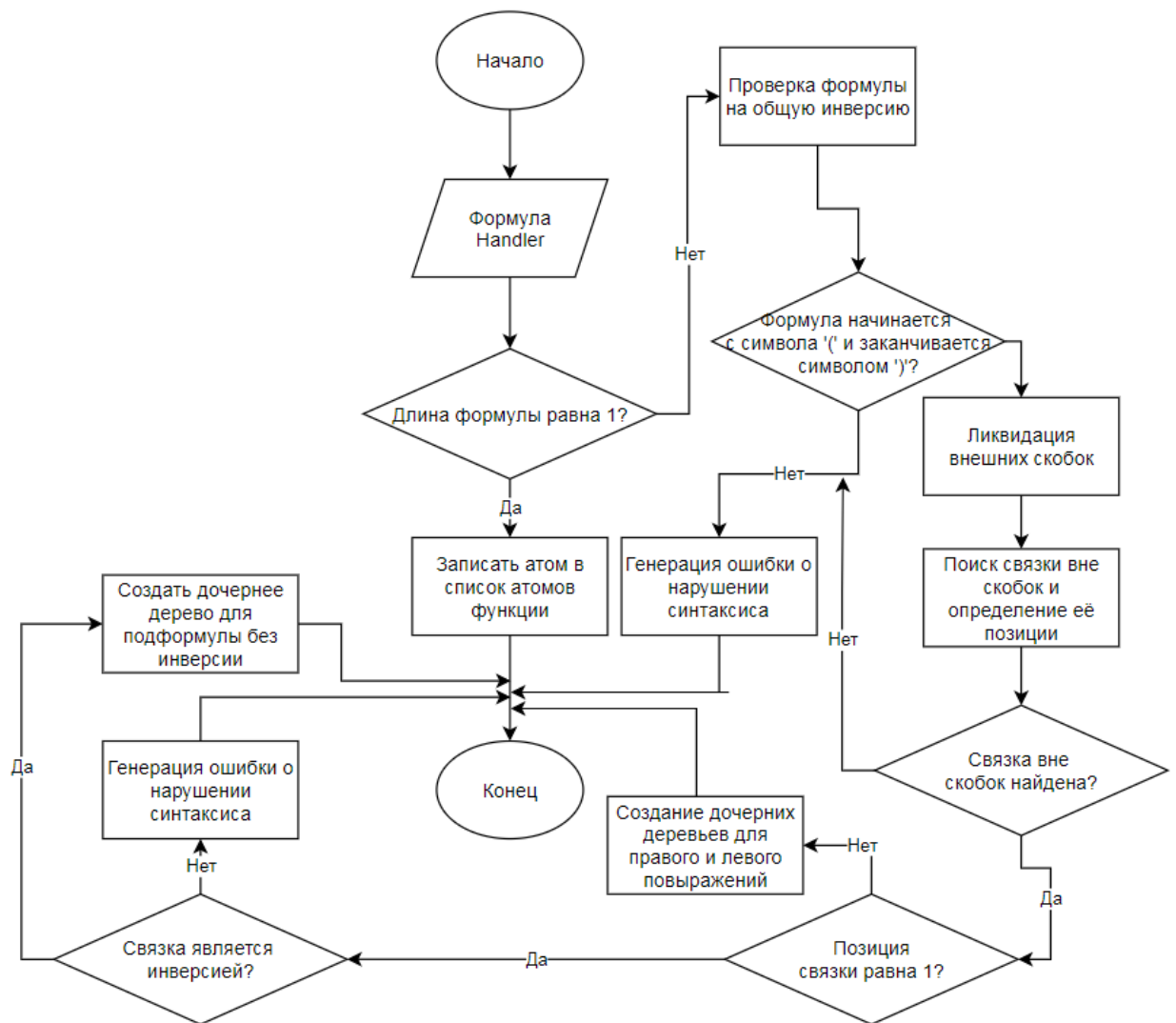


Рис. 5 – Блок-схема конструктора класса ExpressionTree

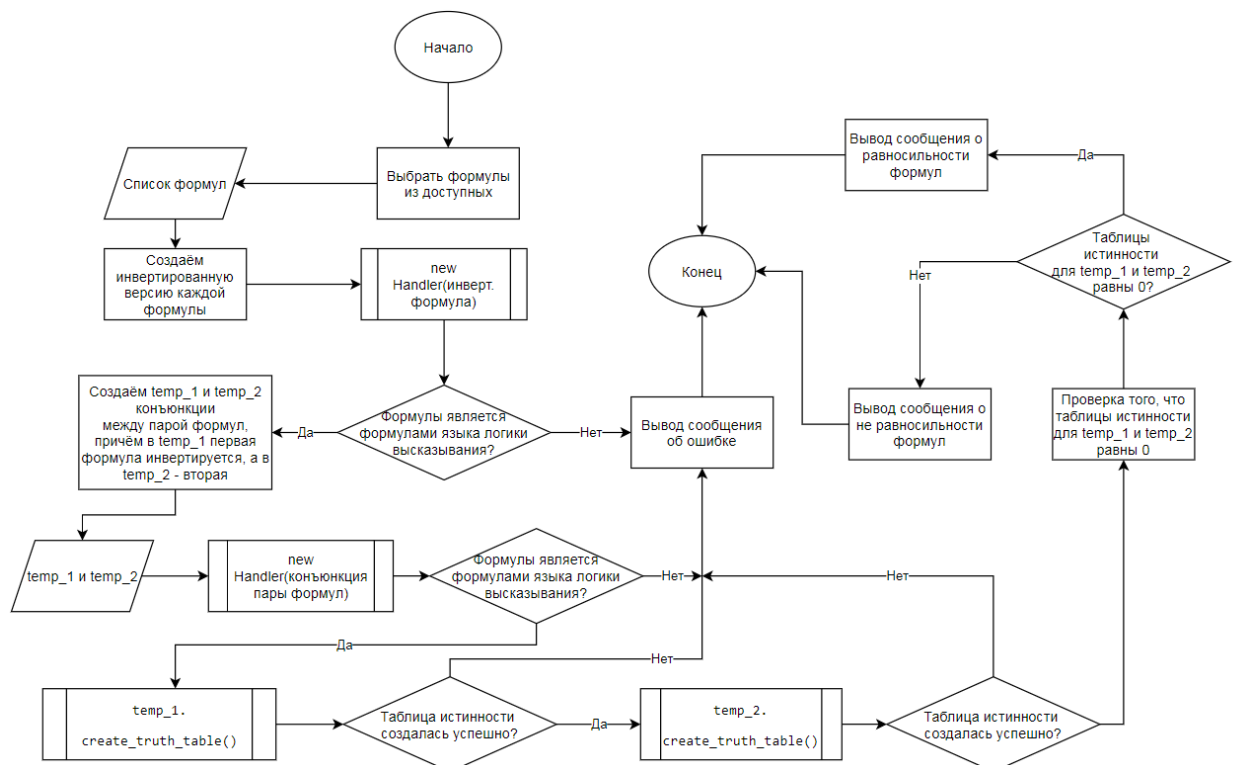


Рис. 6 – Блок-схема функции run класса Dialog

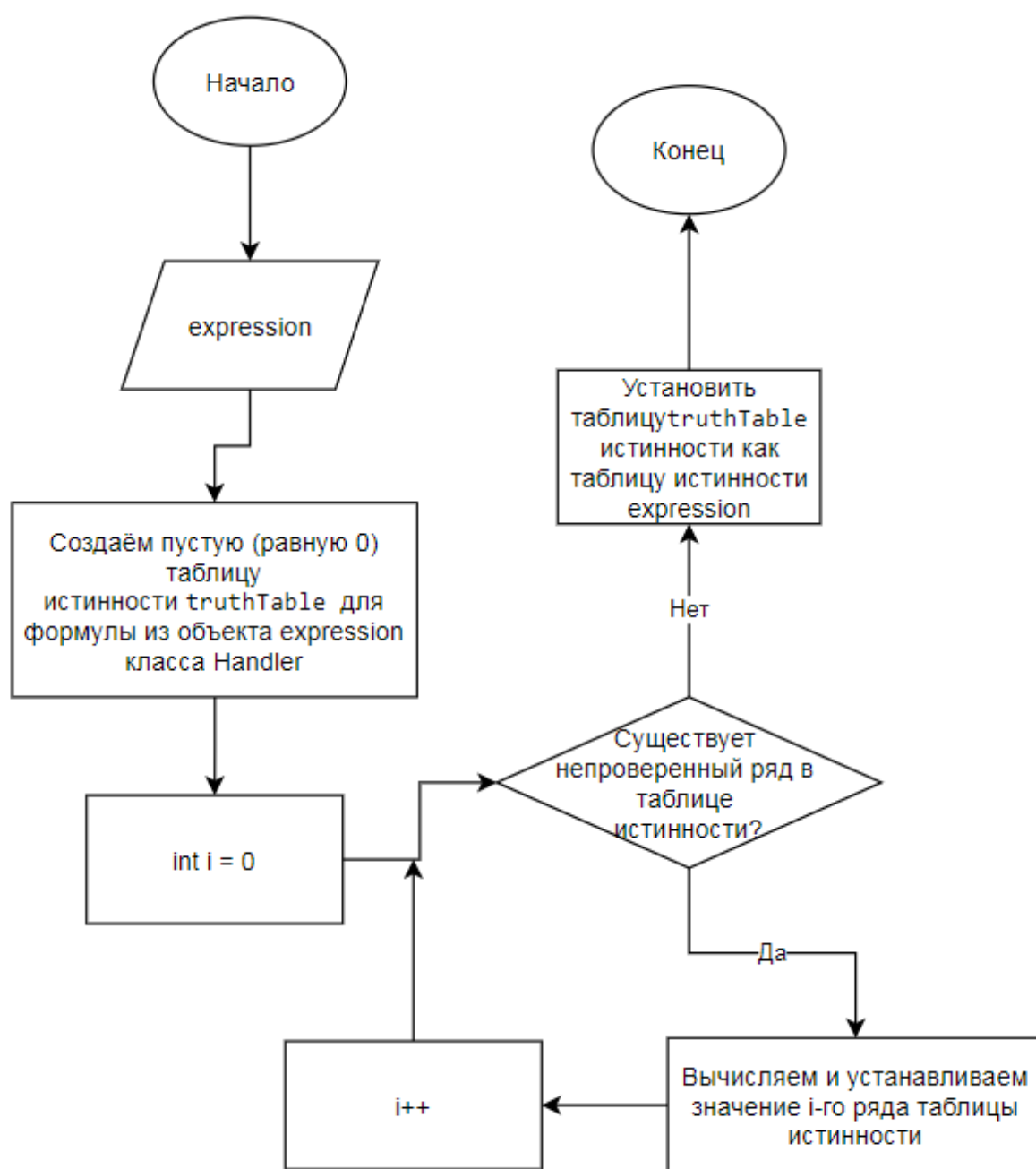


Рис. 7 – Блок-схема функции create\_truth\_table класса Handler



## Примеры выполнения

$((A \sim B) \wedge (!A)) \vee ((!A) \wedge B)$

$((A \rightarrow B) \wedge (!A \wedge B))$

Результат - формулы равносильны

$((A \sim B) \wedge (!A)) \vee ((!A) \wedge B)$

$(!A \wedge B)$

Результат - формулы не равносильны

$((A \sim B) \wedge (!A)) \vee ((!A) \wedge B)$

$((!A) \wedge (!B))$

Результат - формулы не равносильны

$((A \rightarrow B) \wedge (!A \wedge B))$

$(!A \wedge B)$

Результат - формулы не равносильны

$((A \rightarrow B) \wedge (!A \wedge B))$

$((!A) \wedge (!B))$

Результат - формулы не равносильны

$(!A \wedge B)$

$((!A) \wedge (!B))$

Результат - формулы равносильны

Рисунок 8 - Пример работы алгоритма для формул, из которых  $((A \sim B) \wedge (!A)) \vee ((!A) \wedge B)$  равносильна  $((A \rightarrow B) \wedge (!A \wedge B))$ , а  $(!A \wedge B)$  равносильна  $((!A) \wedge (!B))$

$(!A \wedge B)$

$((!A) \wedge (!B))$

Результат - формулы равносильны

Рисунок 9 - Пример работы алгоритма для равносильных формул по правилу де Моргана

```
(0→A)  
(0→B)  
Результат - формулы равносильны
```

Рисунок 10 - Пример работы алгоритма для равносильных формул с 0

```
((1\0)/\A)  
(1/\A)  
Результат - формулы равносильны
```

Рисунок 10 - Пример работы алгоритма для равносильных формул с 1 и 0

### Вывод

В ходе выполнения лабораторной работы были приобретены навыки программирования алгоритмов интерпретации и преобразования формул языка логики высказываний; была разработана программа, позволяющая проверить равносильность для любых заданных формул языка логики высказываний.