

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**ИНЖЕНЕРНО-ТЕХНОЛОГИЧЕСКАЯ АКАДЕМИЯ**

**Институт компьютерных технологий и информационной безопасности  
Кафедра математического обеспечения и применения ЭВМ**

## **Пояснительная записка**

К творческому проекту по дисциплине  
«Введение в инженерную деятельность»

на тему:

«Разработка программы визуализации построения и вычисления обратной  
польской записи арифметических выражений»

Выполнили:

Студент группы КТбо2-7 \_\_\_\_\_ Дуденко А.Е.

Студенты группы КТбо2-2 \_\_\_\_\_ Скороход Д.С.

Руководитель:

Ст. преподаватель каф. МОП ЭВМ \_\_\_\_\_ Балабаева И.Ю.

Таганрог 2017

## Оглавление

Введение.....	3
1. Техническое задание .....	3
1.1. Анализ ТЗ.....	4
1.2. Распределение обязанностей .....	4
1.3. План реализации .....	5
2. Описание методов .....	5
2.1. Алгоритмы .....	5
3. Реализация.....	11
3.1. Инструментарий.....	11
3.2. Интерфейс программы .....	11
3.3. Тестирование программы.....	13
4. Заключение .....	14
5. Источники .....	14

## Введение

Обратная польская запись (ОПЗ) или Обратная польская нотация (ОПН) – форма записи математических и логических выражений, в которой операнды расположены перед знаками операций. Она была предложена польским математиком Я. Лукашевичем, откуда и происходит ее название.

Традиционная (прямая) Запись	Обратная Польская Запись
$A*(B+C)-D/(F+E)$	ABC+*DFE+ /-

ОПН раньше применялась в советском инженерном калькуляторе БЗ-19М (совместная разработка с ГДР), выпущенном в 1976 году. Все выпускаемые в СССР вплоть до конца 1980-х годов программируемые микрокалькуляторы, за исключением «Электроника МК-85» и «Электроника МК-90», использовали ОПН — она проще реализовывалась и позволяла обойтись в программировании вычислений меньшим числом команд, по сравнению с обычной алгебраической нотацией, а количество программной памяти в этих моделях всегда было критическим ресурсом. Сейчас ОПН используется в современных российских программируемых калькуляторах «Электроника МК-152» и «ЭЛЕКТРОНИКА МК-161», что обеспечивает их совместимость с программами, написанными для советских калькуляторов.

Таким образом, преимуществом ОПЗ является компактность, она гораздо удобнее дерева операций, отлично подходит для вычисления на стековых машинах и сокращает количество команд для вычислений, тем самым лучше подходит для оптимизации программ.

### 1. Техническое задание

Разработать программу визуализации построения и вычисления обратной польской записи (ОПЗ) арифметических выражений.

### 1.1. Анализ ТЗ

Программ-аналогов для построения и вычисления ОПЗ можно найти в интернете достаточно большое количество. Например, существует программа, так называемая «конвертор ОПЗ» (рис.1), которая может лишь переводить из прямого выражения в формат ОПЗ и обратно. У нее отсутствует функция вычисления, она служит лишь для перевода из одного формата в другой.

Выражение  $\Rightarrow$  Обратная польская нотация

Обратная польская нотация  $\Rightarrow$  Выражение

Рисунок 1-Пример аналога

Существуют и другие аналоги, которые имеют возможность вычисления, но нет известного аналога, который бы не только выполнял все эти функции, но и демонстрировал то, как все эти преобразования происходят.

Таким образом, особенностью данного продукта будет являться то, что пользователи могут не только воспользоваться функцией перевода и вычисления, а также наглядно увидеть процесс выполнения этих функций. Это поможет преподавателям лучше объяснять материал для студентов, а также студенты смогут самостоятельно разобраться и понять процесс перевода, изучить работу новых алгоритмов и расширить знания о работе стека.

### 1.2. Распределение обязанностей

Дуденко Анастасия Евгеньевна – Разработка алгоритмов построения и вычисления, тестирование программы, документация.

Скороход Денис Сергеевич – Написание кода программы, визуализация программной части.

### **1.3. План реализации**

1 неделя – выбор алгоритма.

2 – 8 недели – консультации с руководителем, подготовка к презентации предварительных результатов.

9 неделя – презентация предварительных результатов.

10 – 15 недели – разработка и проектирование модулей программы, изучение языка C#, разработка алгоритмов.

15-20 недели –написание кода программы.

20 – 24 недели – тестирование модулей программы и программы в целом.

25 – 26 недели – оформление пояснительной записки и презентации проекта.

27 неделя – защита проекта.

## **2. Описание методов**

### **2.1. Алгоритмы**

В данной программе представлены 3 алгоритма, с помощью которых она работает: основной алгоритм, алгоритм построения и вычисления ОПЗ.

Первым алгоритмов является алгоритм всей программы(схема 1). Именно этот алгоритм связывает другие алгоритмы воедино. Запуск программы, после ввода данных в формате прямой записи, идет условие выбора способа перевода. Если выбран перевод по шагам, то программа отображает каждый шаг перевода, если нет, то сразу выводится результат перевода. Далее есть выбор, ввести другое выражение или выполнить вычисление текущего. При выборе вычисления сразу идет проверка, соответствует ли выражение для подсчета требованиям ОПЗ, если нет, то выводиться соответствующая ошибка, если да, то идет условие выбора способа вычисления. Если выбран вычисление по шагам, то программа выполняет пошаговое вычисление, если нет, то программа сразу выводит результат вычисления.

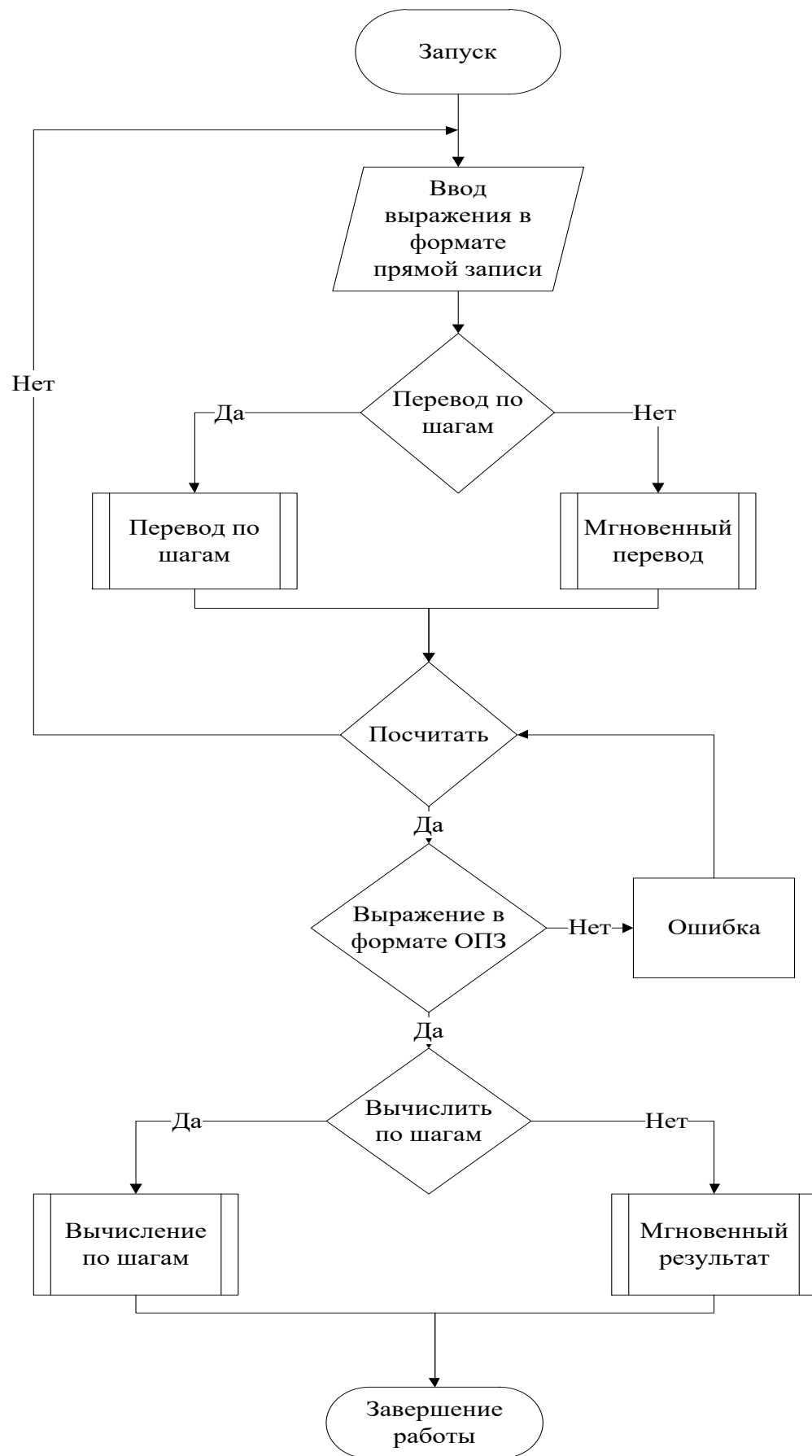


Схема 1 – общий алгоритм программы

Второй алгоритм – алгоритм построения ОПЗ. В основе этого алгоритма лежит алгоритм Дейкстры, предназначенный для трансляции арифметических и других выражений в формат ОПЗ с использованием стека.

Принцип работы следующий: используется три объекта данных: входная строка (традиционная запись выражения), выходная строка (запись в формате ОПЗ) и стек для изменения порядка следования операций. Далее стоит условие: перевести в формат ОПЗ по шагам или автоматически. При выборе перевода по шагам все изменения, происходящие в стеке, будут отображаться на экране, а при выборе обычного перевода, весь алгоритм выполниться сразу. Далее входная строка обрабатывается поэлементно слева направо. Переменные проходят из строки ввода в строку вывода. Для обработки операций и скобок используется стек и таблица приоритетов следующим образом:

- 1) Если стек пуст, то в него записывается операция.
- 2) Операция выталкивает из стека все операции с большим или равным приоритетом в выходную строку, после чего записывается в стек.
- 3) Открывающая скобка всегда записывается в стек.
- 4) Закрывающая скобка выталкивает из стека все операции до открывающей скобки и исчезает вместе с ней.
- 5) Конец входной строки выталкивает все операции из стека в выходную строку.

В этом алгоритме учитывается и таблица приоритетов для операций(табл.1).

Таблица 1

Операция	Приоритет
( )	1
+ или -	2
* Или /	3
^	4

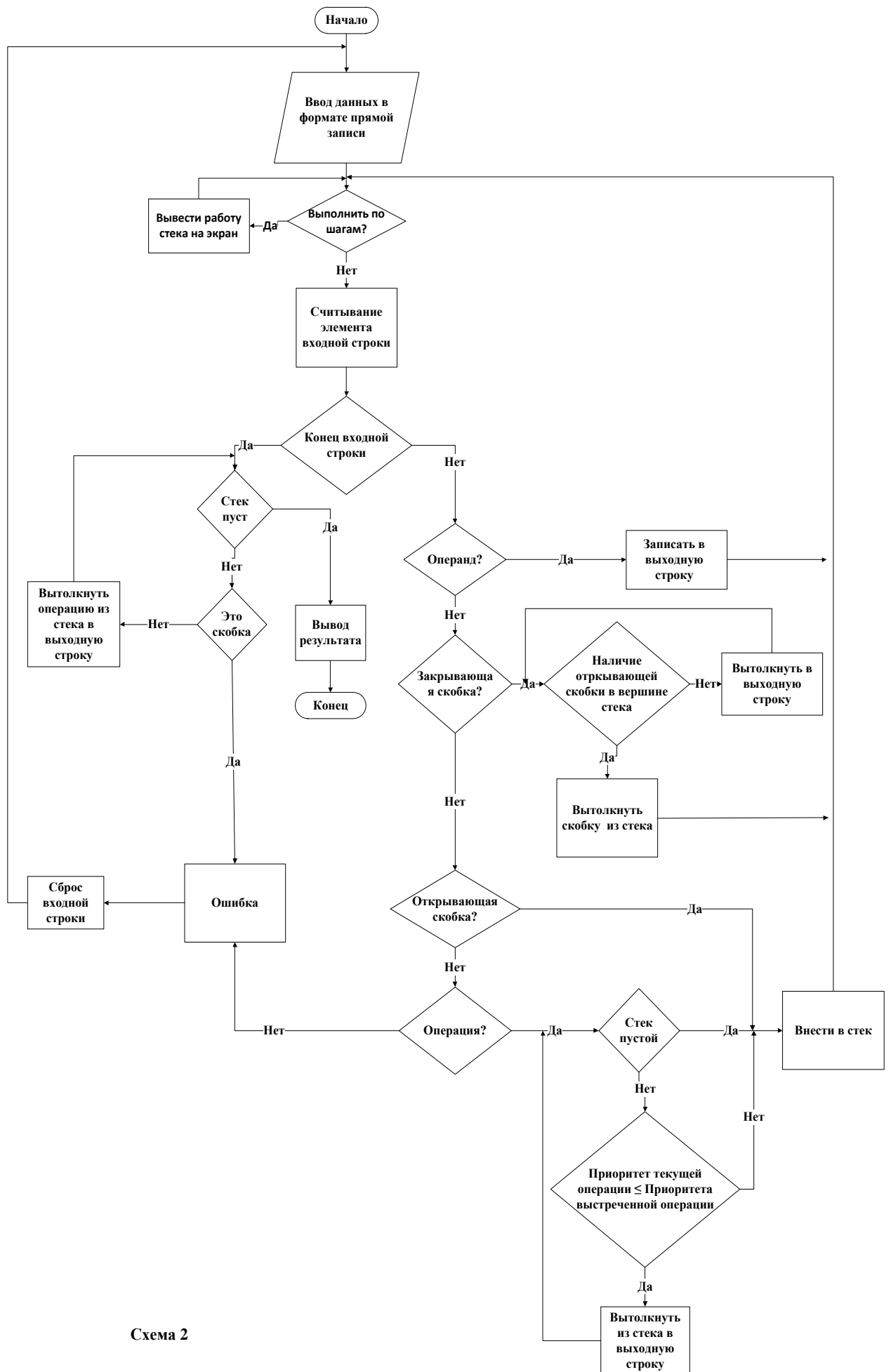


Схема 2



Третий алгоритм – это алгоритм вычисления выражения. Вычисление выражений в ОПЗ выполняется с помощью отдельного стека. Первое условие, которое запрашивает программа, это выбор мгновенного или пошаговое вычисление. В случае выбора пошагового перевода, все промежуточные результаты и содержимое стека будут выводиться на экран, в ином случае вычисление будет произведено автоматически. Далее идет считывание с входной строки. Для этого, выражение просматривается в порядке слева направо, и встречающиеся в нем элементы обрабатываются по следующим правилам:

- 1). Проверка конца строки. Если встречен конец строки, то вывод результата из стека и вычисление завершается, если нет, то продолжаем.
- 2) если встречается операнд, то он помещается в стек (на верхушку стека);
- 3) если встречается знак бинарной операции (требующей двух операндов), то сначала с верхушки стека выбирается второй операнд, затем – первый операнд, операция выполняется и результат помещается в стек (на верхушку стека).
- 4) если встречается знак унарной операции (требующей одного операнда), то программа выведет ошибку и отправит нас на входную строку для исправления.

Результат вычисления после завершения алгоритма всегда находится на верхушке стека.

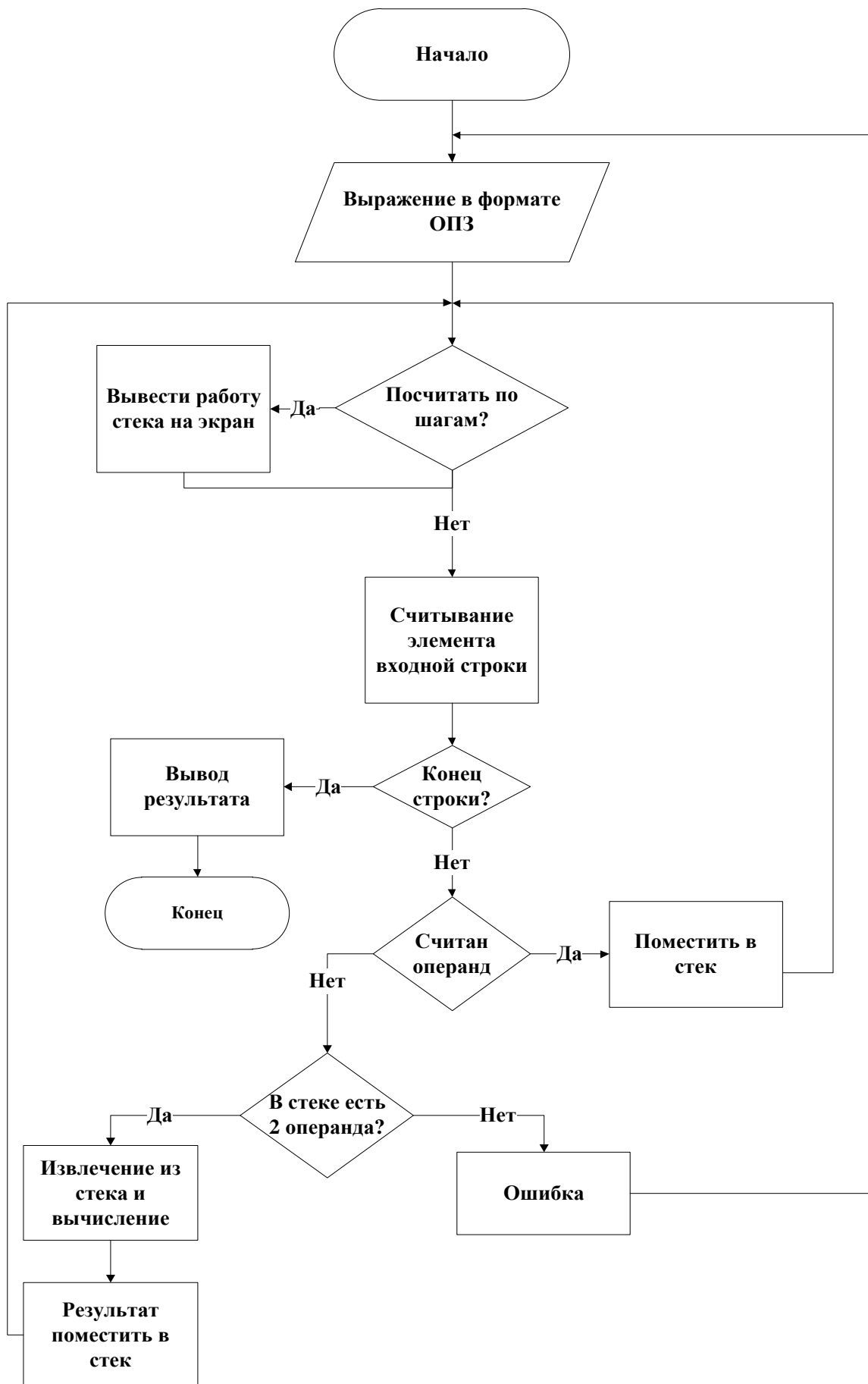


Схема 3 – алгоритм вычисления

### **3. Реализация**

#### **3.1. Инструментарий**

В качестве среды разработки программного продукта была использована программа Visual Studio версия 2015 года. Языком программирования для данного проекта послужил C#.

#### **3.2. Интерфейс программы**

Открывая программу, пользователь увидит первое окно - окно ввода прямой записи(рис.2). Требования для ввода данных следующие: вещественные числа и операции сложения «+», вычитания «-», умножения «\*», деления «/», возведения в степень «^» и скобки «()». В этом окне он должен ввести выражение и выбрать один из следующих способов перевода: перевод в формат ОПЗ или пошаговый перевод в формат ОПЗ.

При выборе кнопки «Перевести в ОПЗ» результат автоматически отобразится в формате ОПЗ и все операнды и операции в стеке. При выборе кнопки «Перевести по шагам» программа начнет с каждым нажатием кнопки отображать обработку каждого элемента прямой записи: внесение операций в стек, перенесение введенных чисел в выходную строку, извлечение операций из стека, тем самым визуализируя по шагам перевод выражения.

Кнопка «Очистить» полностью стирает все данные со всех строк и стека.

Рисунок 2- окно ввода прямой записи

После перевода можно вычислить полученное выражение. При нажатии кнопки «Посчитать» открывается второе окно пользователя (рис.3). В первой строке размещено полученное выражение в формате ОПЗ. Далее нужно снова выбрать одну из предложенных функций: посчитать или посчитать поэтапно.

При нажатии кнопки «Посчитать» результат выражения будет выведен автоматически.

При нажатии кнопки «Посчитать по шагам» программа при каждом нажатии этой кнопки будет отображать поэтапное вычисление, показывая, как извлекаются из стека операнды, происходит вычисление нужных операций и как результат помещается обратно в стек.

Кнопка «Очистить» полностью стирает все данные со всех строк и стека.

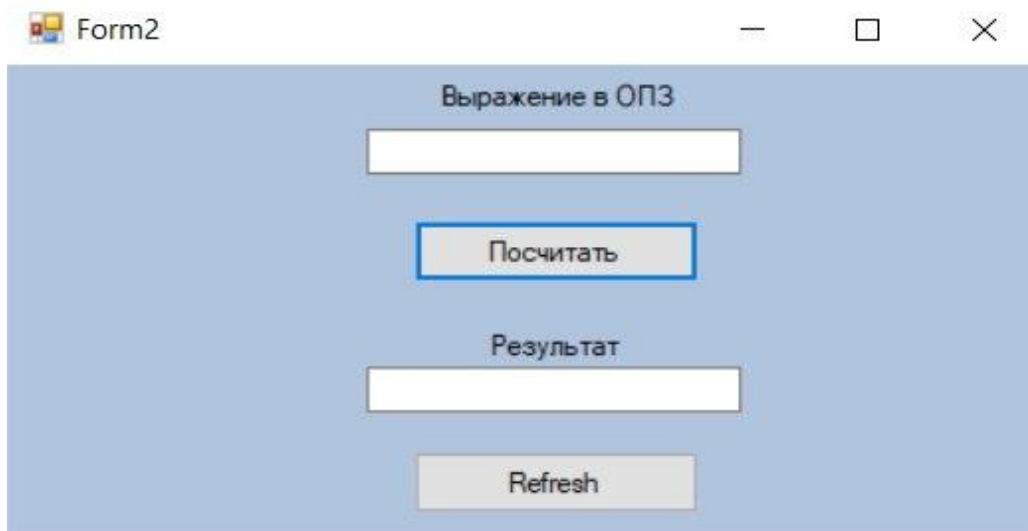


Рисунок 3 – окно вычисления

Так же в первую строку можно внести другое свое выражение в формате ОПЗ, соблюдая все требования. Если введенное выражение не будет соответствовать правилам построения ОПЗ, программа выведет соответствующую ошибку(рис.4).

### 3.3. Тестирование программы

Было выполнено тестирование конечного продукта. В ходе этого тестирования были проверены работоспособность кнопок и правильность перевода и вычисления, исключены заикливания. Итоги перевода в формат ОПЗ сравнивались с итогами на сайтах аналогов и ручного перевода. Результаты вычисления сравнивались с результатами на калькуляторе. Так же были протестирована и программная составляющая: алгоритмы мгновенного и пошагового перевода и вычисления. Тестирование позволило исключить программные ошибки.

#### **4. Заключение**

В ходе проделанной работы были рассмотрены и изучены алгоритмы Дейкстры для перевода и вычисления в формате ОПЗ. Была разработана, отлажена и протестирована общая программа, реализующая эти методы. Программа реализована на языке С#, который пришлось изучить в ходе реализации проекта. После создания проекта возникло множество идей ее усовершенствования таких, как изменение и улучшение интерфейса, добавление перевода с константами, возможное добавление более сложных операций.

В процессе выполнения проекта были улучшены навыки программирования, проектирования, изучен дополнительный язык программирования.

Планируется, что полученный продукт будет использован в практических занятиях по курсу «Теория языков программирования и вычислительных процессов», а также он может быть предложен студентам в качестве решебника для самостоятельной работы по теме «Построение и вычисление ОПЗ».

Техническое задание выполнено в полном объеме.

#### **5. Источники**

Молчанов А. Ю. Системное программное обеспечение: Учебник для вузов. 3-е изд. – СПб.: Питер, 2010. – 400 с.

Вишняков Ю.М., Балабаева И.Ю. Проектирование трансляторов: Руководство к циклу лабораторных работ по курсу «Теория языков программирования и методы трансляции». – Таганрог: Изд-во ТТИ ЮФУ, 2009. – 74 с.