**Учреждение образования**

**«Белорусский государственный университет**

**информатики и радиоэлектроники»**

**Факультет компьютерных систем и сетей**

**Кафедра информатики**

Республика Беларусь, Минск

220013, ул. Платонова, 39, тел/факс +375 (17) 293-84-66

**Автор:**

Резунов Максим Вячеславович, студент 853501 учебной группы, 1 курс

**Визуализатор пространственно-временных функций**

**Назначение. Область применения:**

Интерактивная программа предназначена для отображения графиков пространственных функций, постоянных или изменяющихся во времени. Применяется для визуализации графических статических и динамических данных в пространстве, построении графиков, диаграмм. Она позволяет:

* рассчитывать значения функций z=f(x,y,t);
* интерактивно отображать графики функций в удобном для восприятия виде;
* задавать набор начальных данных для построения графиков;
* отображать динамические функции в любой момент времени из заданного диапазона времени;
* сохранять и загружать построенные графики;
* использовать заранее построенные «Демо» графики.

**Технологии, используемые при разработке**

При разработке программы использовались язык программирования C++ и графическая библиотека OpenGL.

**Программное обеспечение, необходимое для использования продукта**

Программа требует наличия на компьютере операционной системы

Windows 7-10, стандартных библиотек Windows для работы с файлами, стандартной библиотеки C++, программной платформы NET.framework v4.

**СОЖЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc531640757)

[ОПИСАНИЕ РАБОТЫ 5](#_Toc531640758)

[1 ОБЩИЕ СВЕДЕНИЯ 5](#_Toc531640759)

[1.1 Обозначение и наименование программы 5](#_Toc531640760)

[1.2 Программное обеспечение, необходимое для функционирования программы 5](#_Toc531640761)

[1.3 Языки программирования, на которых написана программа 5](#_Toc531640762)

[2 ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ 6](#_Toc531640763)

[2.1 Назначение программы 6](#_Toc531640764)

[2.2 Сведения о функциональных ограничениях на применение 6](#_Toc531640765)

[3 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 7](#_Toc531640766)

[3.1 Описание формы Windows Forms «Параметры функций z = f(x,y,t)». 7](#_Toc531640767)

[3.2 Описание окна OpenGL. 15](#_Toc531640768)

[4. ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ 20](#_Toc531640769)

[4.1 Общий алгоритм программы 20](#_Toc531640770)

[4.2. Используемые методы 20](#_Toc531640771)

[ЗАКЛЮЧЕНИЕ 22](#_Toc531640772)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 24](#_Toc531640773)

[ДОПОЛНИТЕЛЬНЫЕ МАТЕРИАЛЫ 24](#_Toc531640774)

[Приложение А 25](#_Toc531640775)

[Приложение Б 29](#_Toc531640776)

# ВВЕДЕНИЕ

Физиологически, восприятие визуальной информации является основой для взаимодействия человека и окружающего мира. Есть многочисленные исследования, подтверждающие, что 90% информации человек воспринимает через зрение. Визуализационные инструменты всегда были и остаются актуальными и востребованными на рынке разработки программного обеспечения.

Визуализация — это наглядное представление различной информации. Визуальная информация лучше воспринимается и позволяет быстрее и более эффективно донести до пользователя мысли и идеи программного продукта.

В этой исследовательской работе представлена программа «Визуализатор пространственно-временных функций», предназначенная для визуализации функций 1 или 2 переменных координат и времени z=f(x,y,t).

Основная цель исследовательской работы — разработать программное приложение, которое даёт наглядное представление о введенных функциях двух переменных пространственных координат x и y, координаты времени t.

Определены задачи данного исследования:

1 Исследовать, изучить, разработать

2

3

Для решения поставленных задач были использованы следующие методы

КАК ДОСТИГНУТЬ ЗАДАЧИ

1

2

Данная работа состоит из введения, трех глав, заключения и двух приложений

Область применения

Основная задача программы — построить постоянные (далее - статические) или изменяющиеся (далее - динамические) во времени графики введенных функций в удобном для восприятия виде. Для этих целей реализованы возможности вращения, цветового отображения графиков и изменения масштаба.

В программе реализованы следующие возможности:

* Ввод нескольких функций в инфиксной нотации (например: 2 + 2 \* 2);
* Ввод параметров, необходимых для построения графиков;
* Преобразование функций в обратную польскую запись (например: 2+2\*2 => 2 2 2 \* +);
* Расчет матриц значений функций в форме обратной польской записи (для каждого момента времени, если функция динамическая);
* Отображение графиков функций z = f(x,y,t) в разных цветовых схемах в отдельном окне;

Отрисовка осей, вывод минимальных и максимальных значений координат и времени.

Параметры, необходимые для построения функций и поле ввода самих функций представлены на форме Windows Forms. Также на форме располагается функция в обратной польской записи, поля, управляющие вращением графиков, отображением осей, минимальных и максимальных значений координат и времени.

# ОПИСАНИЕ РАБОТЫ

## ОБЩИЕ СВЕДЕНИЯ

### Обозначение и наименование программы

Программа визуализации функций имеет следующие атрибуты:

|  |  |
| --- | --- |
| Наименование исполняемого файла | 3D Vizualizer v5.ехе |
| Размер исполняемого файла | 249 Кб??? |
| Версия продукта | 5.0 |
| Разработчик | Резунов Максим |
| Языки | Русский, Английский |

### Программное обеспечение, необходимое для функционирования программы

Системные программные средства, используемые программой «Визуализатор пространственно-временных функций», должны быть представлены локализованной версией операционной системы Windows 7-10.

Для установки данной программы достаточно скопировать папку «Визуализатор пространственно-временных функций» на компьютер пользователя. Каких-либо настроек после копирования программы не требуется.

Содержимое папки:

|  |  |
| --- | --- |
| data | ??? |
| 3D Vizualizer v5.exe | 249 кб |
| 3D Vizualizer v5.pdb | 915 кб |

### Языки программирования, на которых написана программа

Исходным языком программирования для приложения «Визуализатор пространственно-временных функций» является С++. Среда разработки, компилятор – Microsoft Visual Studio 2017 (локализованная, русская версия). В программе используется графическая библиотека OpenGL, внедренная в проект.

## ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

### Назначение программы

Программа предназначена для построения графиков функций z=f(x,y,t) с использованием скобок, операций сложения, вычитания, умножения, деления, возведения в степень, модуля, тригонометрических функций синуса, косинуса, тангенса, котангенса, обратных им тригонометрических функций, функций округления до ближайшего целого, до наименьшего и наибольшего целого значения, функций логарифмирования.

### Сведения о функциональных ограничениях на применение

Вводимые числа не должны по модулю превышать 1010. Количество точек на оси не должно превышать 100, количество кадров в секунду – 100. Данное ограничение сделано для того, чтобы уменьшить потребляемые программой ресурсы процессора и оперативной памяти, также нецелесообразно делать количество кадров в секунду больше 100, а при качестве 100 точек на оси и правильном выборе диапазона значений координат и времени график близок к точной кривой.

## РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

### Описание формы Windows Forms «Параметры функций z = f(x,y,t)».

Программа включает в себя формы:

* Windows Forms «Параметры функций z=f(x,y,t)»
* «Расчет»
* «Демо»
* «Справка»

Окно OpenGL «Визуализатор функций z=f(x,y,t)».

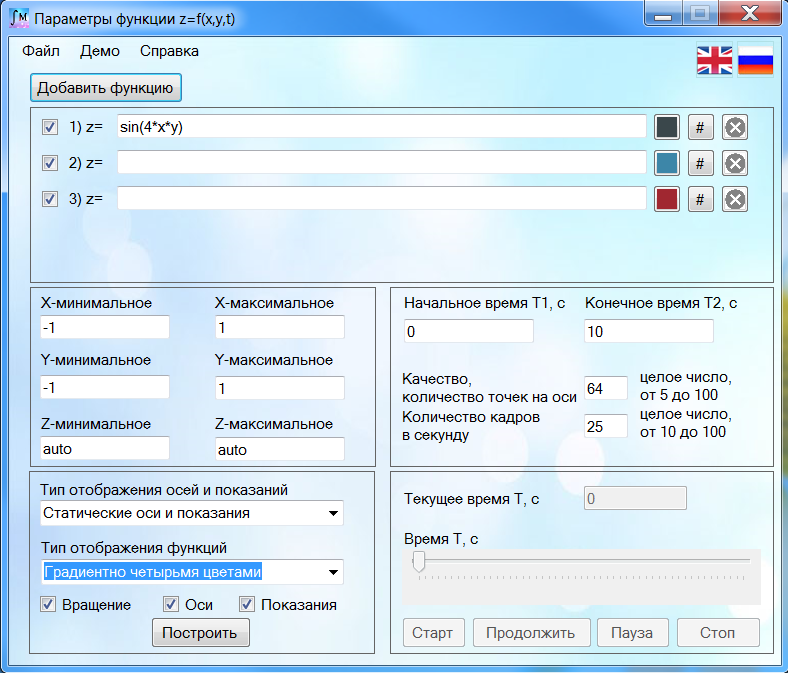


Рисунок 1 – форма Windows Forms «Параметры функций z = f (x,y,t)».

При нажатии на кнопки смены языка (английский, русский) меняется языковой интерфейс приложения (Рисунок 2)



Рисунок 2 – кнопка смены языка

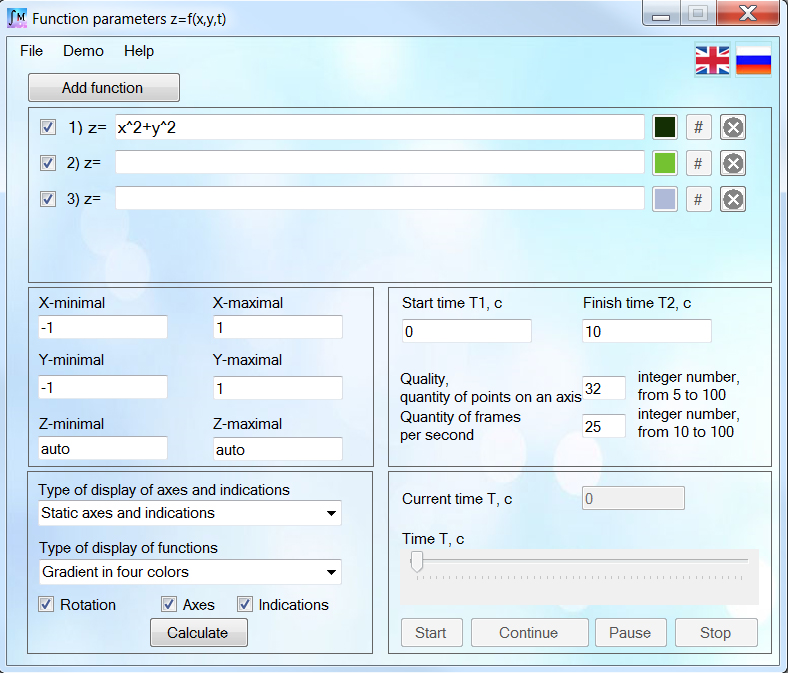


Рисунок 3 – английский интерфейс

На форме расположена кнопка «Добавить функцию», поля ввода для функций, кнопки (Рисунок 4):

* выбора цвета графиков
* смены цвета на случайный
* удаления графиков.

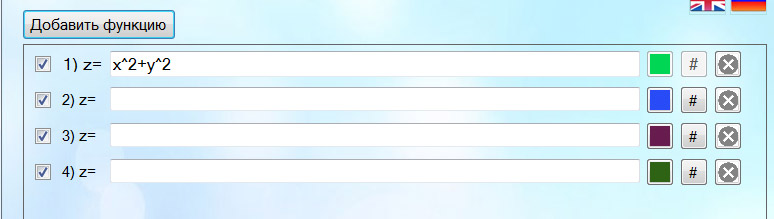


Рисунок 4 – кнопка «Добавить функцию», поля ввода для функций

Расположены также поля ввода минимальных и максимальных значений координат x, y, z и времени t (Рисунок 5), начального времени Т1 и конечного времени Т2 (в секундах) (Рисунок 6).

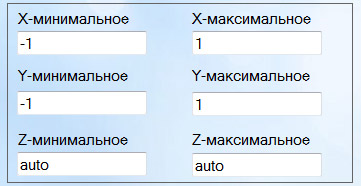


Рисунок 5 – поля ввода минимальных и максимальных значений

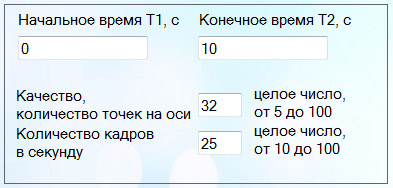


Рисунок 6 – поля ввода начального времени Т1 и конечного времени Т2

Z-минимальное и Z-максимальное могут принимать значения auto, в этом случае программа автоматически определит минимальное и максимальное значение z.

На форме расположены поля ввода качества графиков (количества точек на оси) и количества кадров в секунду (Рисунок 7).

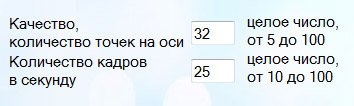


Рисунок 7 – поля ввода качества графиков

Элементы управления флажки отвечают за вращение графиков, отображение осей, минимальных и максимальных значений на графике (показаний) (Рисунок 8).



Рисунок 8 – Элементы управления флажки

Поле со списком «Тип отображения осей и показаний» (Рисунок 9) позволяет выбрать наиболее удобный режим отображения осей и показаний:

* Статические оси и показания. В этом режиме местоположение осей и показаний остается на своих исходных позициях, вне зависимости от того, как повернут график.
* Статические оси и динамические показания. В этом режиме местоположение осей статично, а показания отображаются таким образом, что видны всегда, вне зависимости от того, как повернут график.
* Динамические оси и показания. В этом режиме оси и показания отображаются таким образом, что видны всегда, вне зависимости от того, как повернут график.

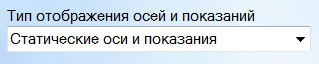


Рисунок 9 – поле со списком «Тип отображения осей и показаний»

Поле со списком «Тип отображения функций» реализует 4 варианта цветовой схемы графиков (Рисунок 10):

* Монотонно одним цветом (Рисунок 11)
* Градиентно одним цветом (Рисунок 12)
* Градиентно четырьмя цветами (Рисунок 13)
* Цветовая схема (Рисунок 14)

По-умолчанию выбрано значение «Градиентно четырьмя цветами»

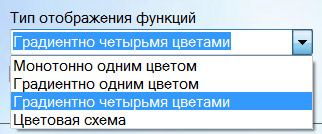


Рисунок 10 – поле со списком «Тип отображения функций»

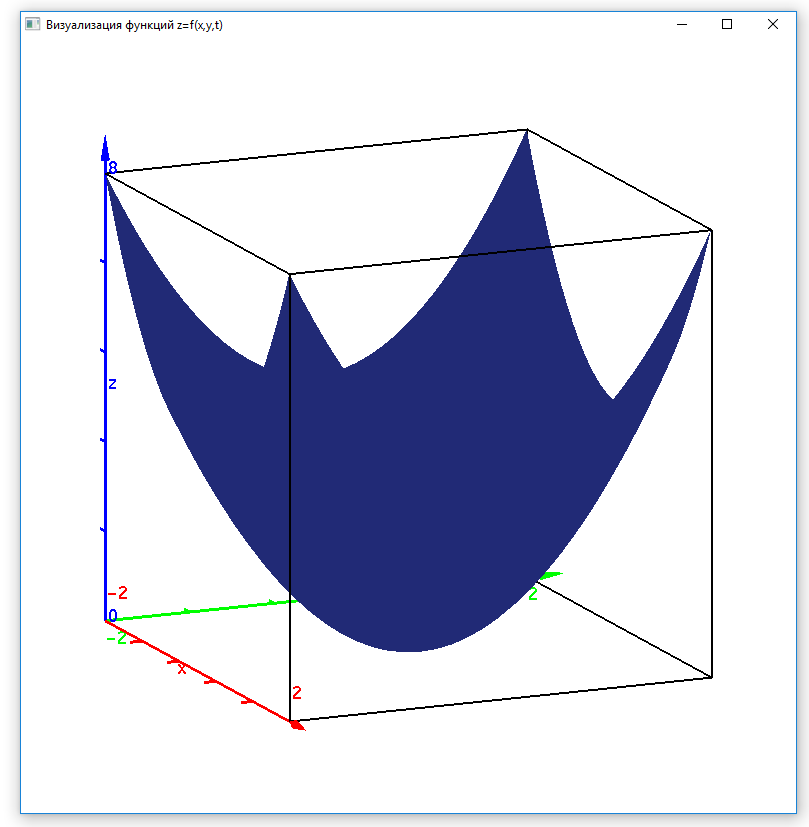


Рисунок 11 – тип отображения функций «Монотонно одним цветом»

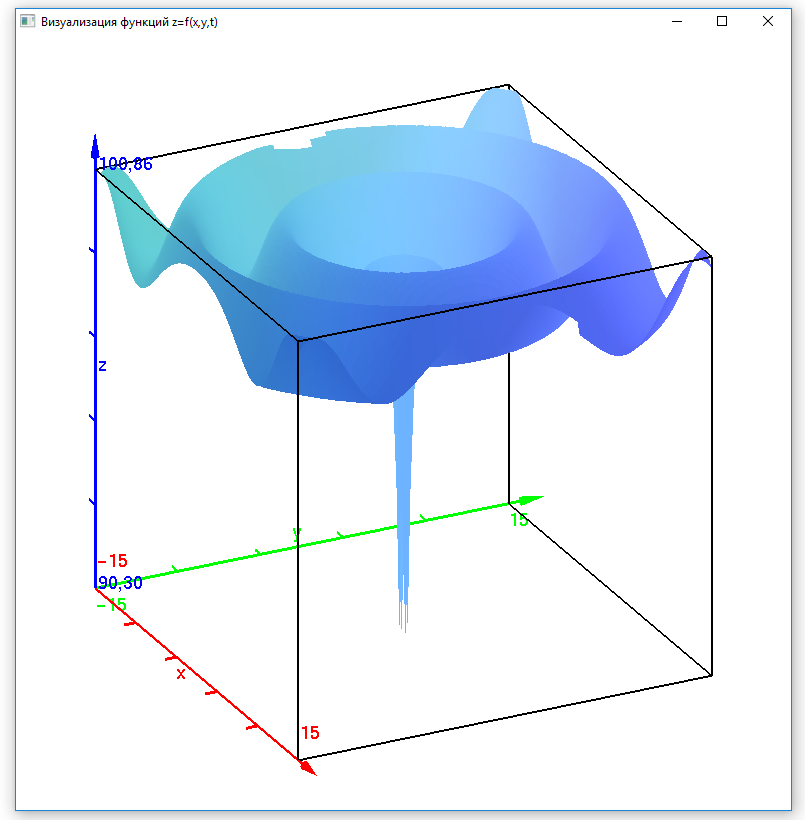


Рисунок 12 – тип отображения функций «Градиентно одним цветом»

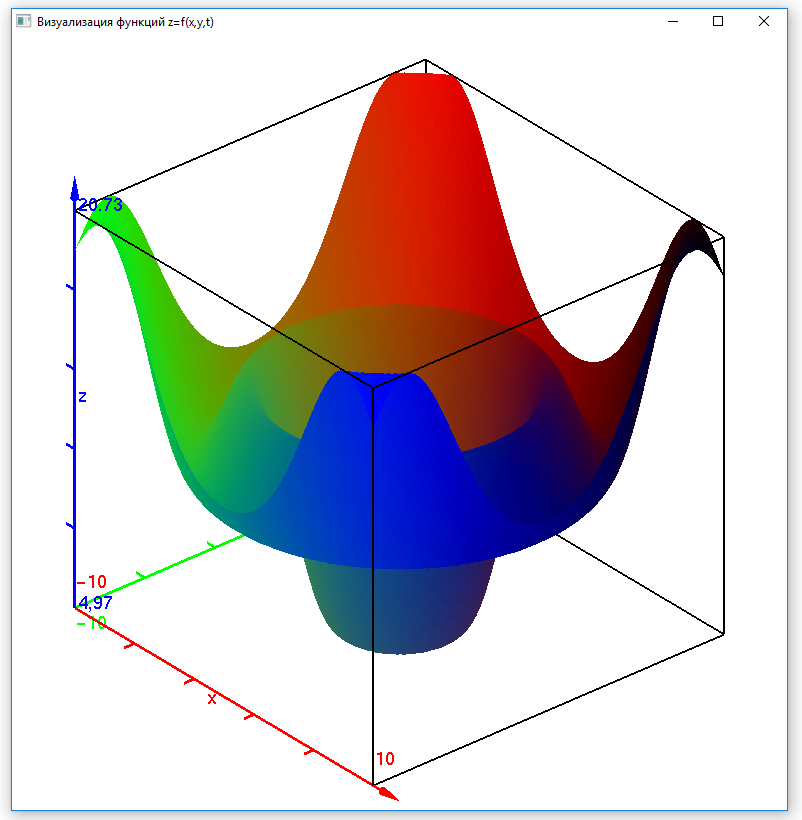


Рисунок 13 – тип отображения функций «Градиентно четырьмя цве-тами»

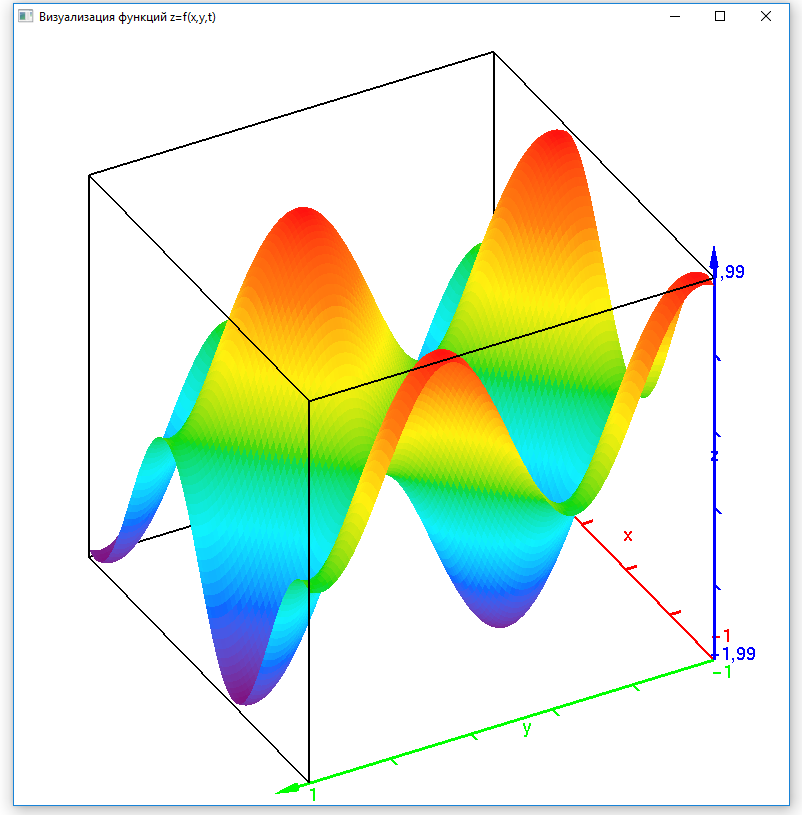


Рисунок 14 – тип отображения функций «Цветовая схема»

Инициализация функций, минимальных и максимальных значений координат x, y, z, времени t, качества графиков и количества кадров в секунду происходит по нажатию кнопки «Построить» (Рисунок 15).



Рисунок 15 – кнопка «Построить»

После нажатия кнопки происходит расчет значений новых функций. Прогресс расчета показывается на форме «Расчёт». По окончанию расчета происходит отображение графиков.

Программа автоматически определяет, являются ли функции статическими или динамическими после нажатия на кнопку «Построить». При этом, если функция динамическая, становятся доступны элементы управления временем (Рисунок 16).

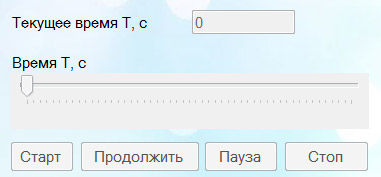


Рисунок 16 – элементы управления временем

Для управления ходом отображения графиков функций во времени предназначены кнопки «Старт», «Продолжить», «Пауза», «Стоп», поле ввода текущего момента времени (в секундах) и бегунок.

Изменение режима отображения осей и показаний, а также вращения графиков, отображения осей и показаний происходит при изменении соответствующих полей (флажков и поля со списком) т.е. не требует нажатия кнопки «Построить».

### Описание окна OpenGL.

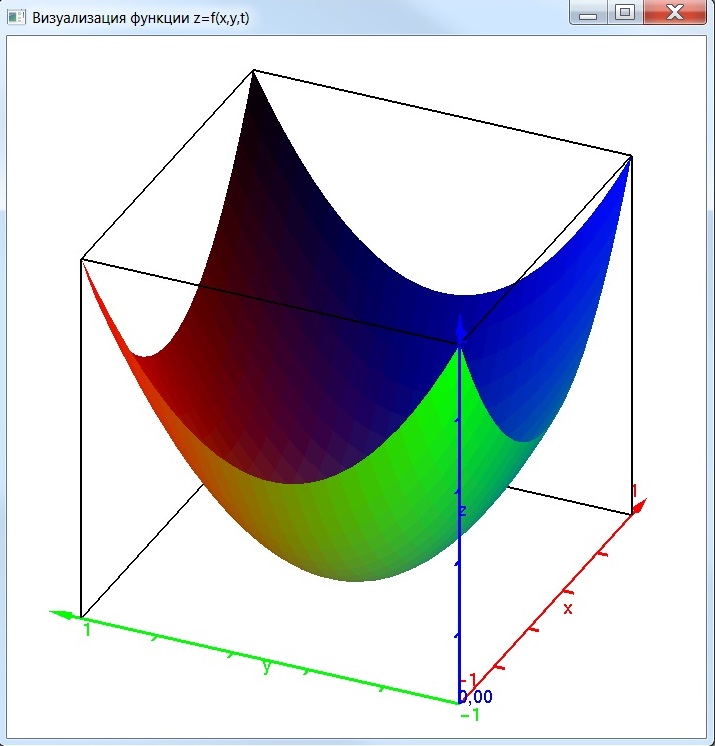


Рисунок 22 – окно OpenGL

В окне OpenGL (Рисунок 22) располагается цветной график функций, ограничивающий ее прямоугольный параллелепипед, оси x, y и z, минимальные и максимальные значения осей.

Нажав левую клавишу мыши и двигая ей, можно поворачивать графики, колесом мыши можно масштабировать функции. Размеры окна OpenGL можно изменять и раскрывать окно во весь экран.

Меню

На форме «Параметры функций z = f (x,y,t)» в верхней части слева расположено меню (Рисунок 17).

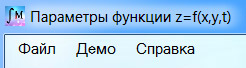


Рисунок 17 – меню

Пункты меню:

* Файл
* Демо
* Справка

Пункт меню «Файл» состоит из пунктов:

* Очистить функции
* Сохранить функции
* Загрузить функции

При вызове пункта меню «Демо» открывается форма «Демо» (Рисунок 18):

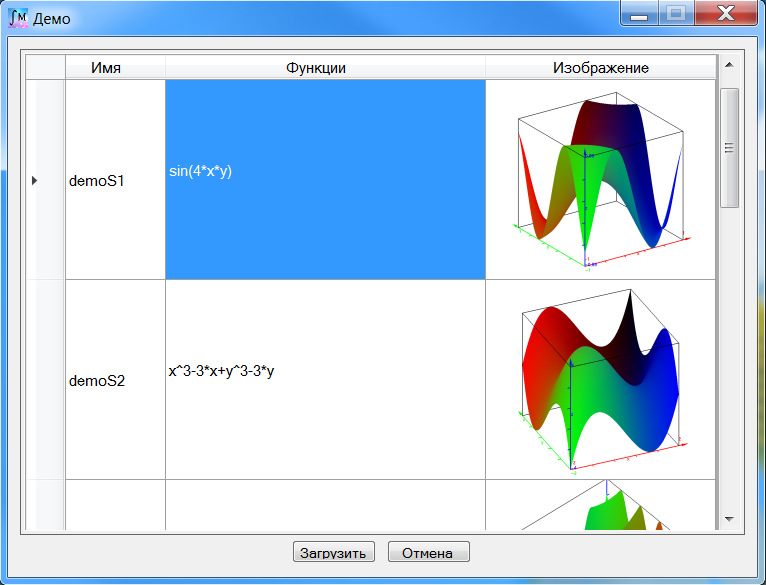


Рисунок 18 – форма «Демо»

После выбора функций и нажатия на кнопку «Загрузить» происходит загрузка заранее рассчитанных функций «Параметры функций z = f (x,y,t)».

Форма «расчёт»

Форма «**Расчёт**» (Рисунок 19) отображает процесс расчёта графиков функций, количество рассчитанных функций и сообщения о возможных ошибках.

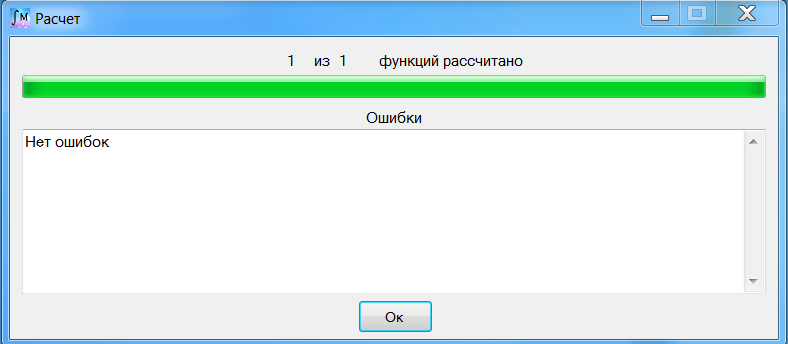


Рисунок 19 – форма «Расчёт»

Демонстрация и загрузка заранее рассчитанных графиков функций

Форма «Демо».

Демонстрация и загрузка выбранных графиков функций представлена на Рисунке 20

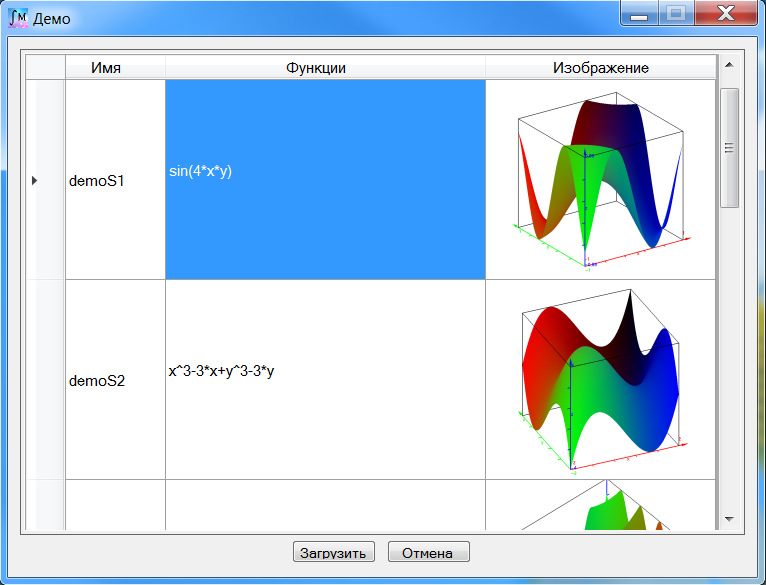


Рисунок 20 – форма «Демо»

**Справка**

По нажатию на пункт меню «Справка» отображается форма справки в формате chm (Рисунок 21):

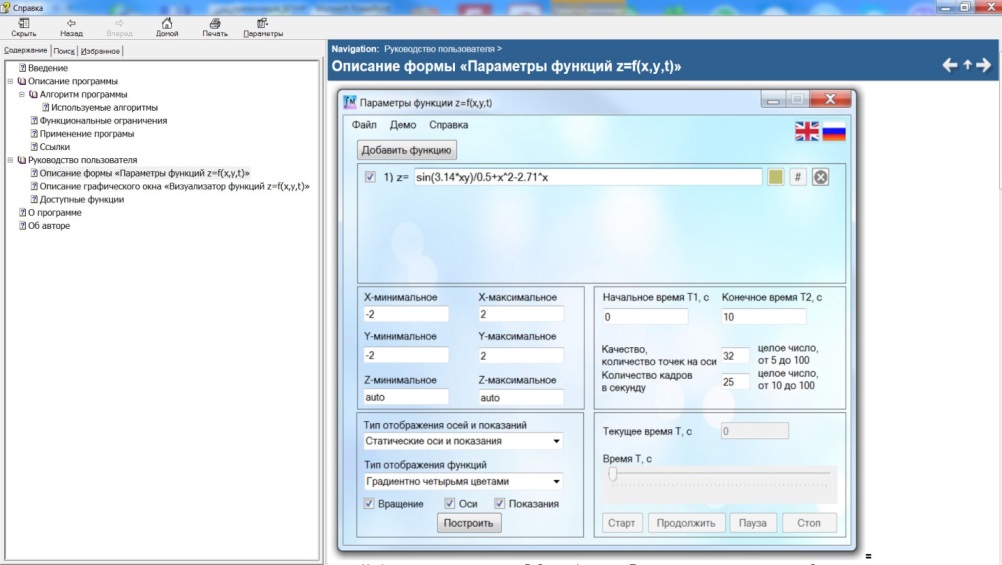


Рисунок 21 – форма «Справка»

Описание раздела «Справка» приведено в [приложении Б.](#_Приложение_Б)

## 4. ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

### 4.1 Общий алгоритм программы

Общий алгоритм программы представлен в [приложении Б](#_Приложение_Б)

Рисунок 21 - Общий алгоритм программы

### 4.2 Используемые методы

1. Преобразование из инфиксной нотации в обратную польскую (алгоритм приведен без обработки ошибок и исключений)

* Пока есть ещё символы для чтения:
* Читаем очередной символ.
* Если символ является числом, добавляем его к выходной строке.
* Если символ является символом функций, помещаем его в стек.
* Если символ является открывающей скобкой, помещаем его в стек.
* Если символ является закрывающей скобкой:
* До тех пор, пока верхним элементом стека не станет открывающая скобка, выталкиваем элементы из стека в выходную строку. При этом открывающая скобка удаляется из стека, но в выходную строку не добавляется. Если стек закончился раньше, чем мы встретили открывающую скобку, это означает, что в выражении либо неверно поставлен разделитель, либо не согласованы скобки.
* Если символ является оператором о1, тогда:

1) пока…

… (если оператор ***o1*** право-ассоциированный) приоритет ***o1*** меньше приоритета оператора, находящегося на вершине стека…

… (если оператор ***o1*** ассоциированный, либо лево-ассоциированный) приоритет ***o1*** меньше либо равен приоритету оператора, находящегося на вершине стека…

… выталкиваем верхний элемент стека в выходную строку;

2) помещаем оператор ***o1*** в стек.

* Когда входная строка закончилась, выталкиваем все символы из стека в выходную строку. В стеке должны были остаться только символы операторов, если это не так, то в выражении не согласованы скобки.

1. Расчет матрицы значений функций в обратной польской записи (алгоритм приведен без обработки ошибок и исключений)

* Для каждого элемента матрицы:
  + Считываем очередной символ
  + Если операция сложения:

Считываем два числа из стека. Вычисляем сумму этих двух чисел. Помещаем результат в стек.

* + Если операция вычитания:

Считываем два числа из стека. Вычисляем разность второго и первого числа. Помещаем результат в стек.

* + Если операция умножения:

Считываем два числа из стека. Вычисляем произведение двух чисел. Помещаем результат в стек.

* + Если операция деления:

Считываем два числа из стека. Вычисляем частное двух чисел. Помещаем результат в стек.

* + Если операция возведения в степень:

Считываем два числа из стека. Возводим второе число в степень первого числа. Помещаем результат в стек.

* + Если символ «x», «y» или «t»:

Помещаем текущее значение переменной (определяется в главном цикле) в стек.

* + Если число:

Помещаем число в стек.

# ЗАКЛЮЧЕНИЕ

В ходе поисково-исследовательского этапа данной работы был проведен поиск аналогов программы в открытых источниках в сети интернет. Среди них не найдено программ, способных визуализировать графики динамических функций. В результате научного проекта была разработана программа «Визуализатор пространственно-временных функций».

ОТВЕТИТЬ НА ЦЕЛЬ.

Отличительной особенностью программы является возможность построения графиков пространственных функций, изменяющихся во времени.

Было изучено, насколько в настоящее время актуальна визуализация статических и динамических данных. По результатам исследования можно сделать вывод о том, что это очень востребовано при восприятии человеком информации в визуальной форме.

Практическое применение программы «Визуализатор пространственно-временных функций»:

* в различных технических науках, где требуется визуализировать пространственно-временные функции, например, в математике, физике, информатике и др.
* она может быть полезна на различных предприятиях и учреждениях, где потребуется визуализировать функции 1 или 2 переменных координат и времени;
* школьникам для лучшего понимания вида различных функций. Это поможет им при решении поставленных задач;
* учителям математики, физики, информатики, географии, биологии и др. при графическом представлении информации;
* преподавателям и студентам высших учебных заведений для наглядности представляемой информации;

Работа была представлена на научно-исследовательской конференции «Поиск» 2017 года.

Перспективы проекта:

* При дальнейшей разработке планируется добавить возможность отображения нескольких функций в одном окне.
* Найти множество общих точек пересечения этих функций.
* Найти локальных экстремумов функций.
* Реализовать решения всевозможных уравнений.
* Данные могут поступать в виде потока информации и интерактивно отображаться (потоковое видео). Прием информации может осуществляться напрямую из некоторого устройства в поставленных экспериментах (служить программой-драйвером).
* В программу можно будет загружать готовые массивы данных и выводить их в графическое окно. Примером такого решения может быть ландшафт некоторой местности.

Эти возможности говорят о том, что программа имеет практическое применение в различных областях науки, производства, народного хозяйства.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] Материал сайта [Электронный ресурс]:<https://ru.wikipedia.org/>

[2] Материал сайта [Электронный ресурс] <http://cppstudio.com/>

[3] Материал сайта [Электронный ресурс] <https://msdn.microsoft.com/>

[4] Информация статьи сайта [Электронный ресурс] <https://habrahabr.ru/company/devexpress/blog/240325/>

# ДОПОЛНИТЕЛЬНЫЕ МАТЕРИАЛЫ

В следующей ссылке находится архив программы, скриншоты и видео демонстрации примеров графиков. Для запуска программы скачайте архив и разархивируйте его.

[5] [Электронный ресурс] <https://drive.google.com/open?id=1Qf9MUojnf4UK9hh3LI7_VHf1yVf6MpN->

# Приложение А

Примеры функций

* 1. Статическая функция

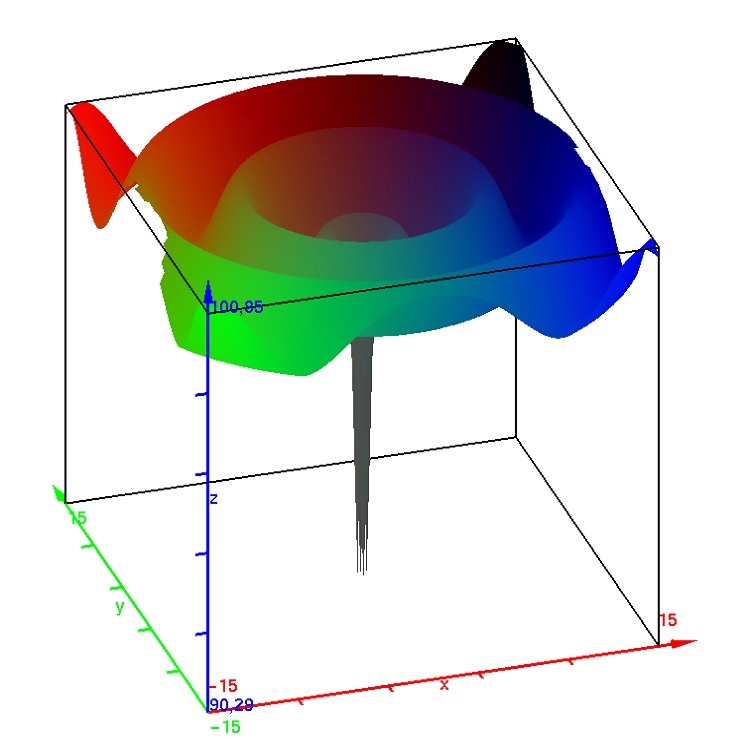


Рисунок А.1 – график функции

* 1. Статическая функция представлена на рисунке А.2

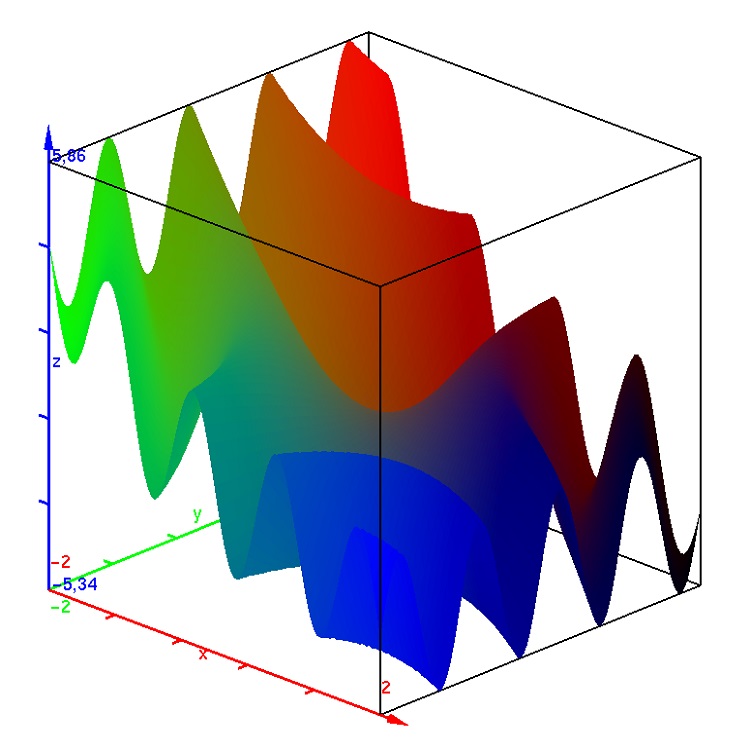


Рисунок А.2 – график функции

* 1. Динамическая функция

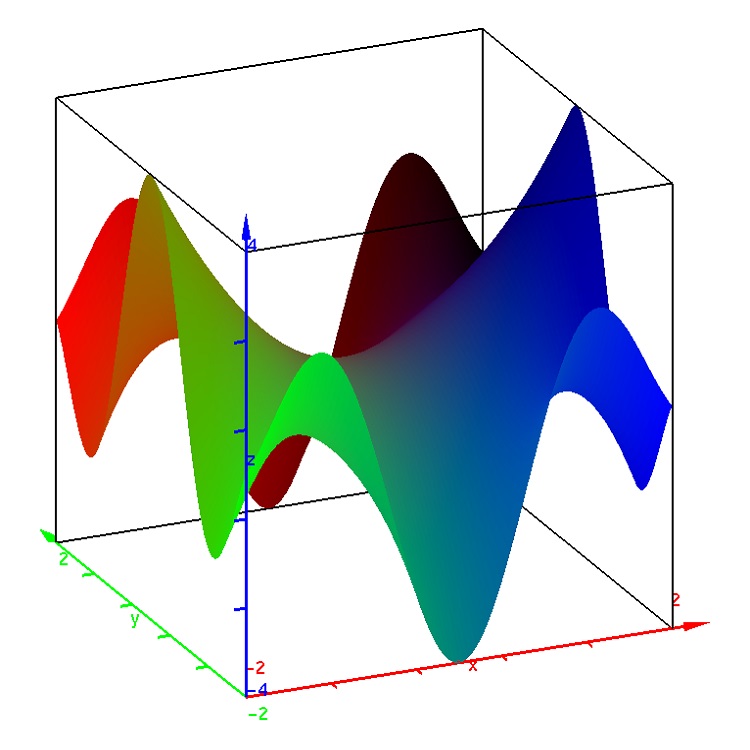


Рисунок А.3 – график функции

* 1. Динамическая функция

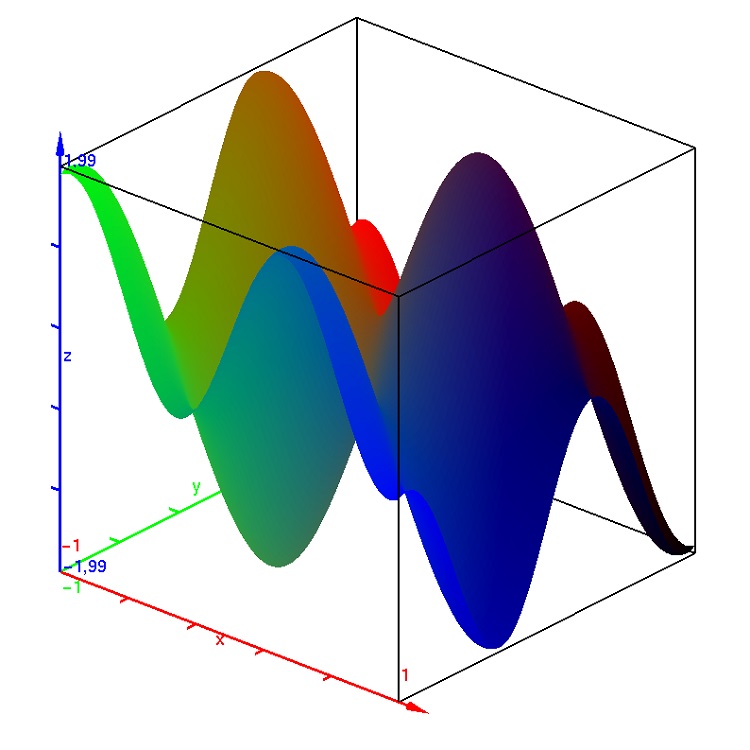


Рисунок А.4 – график функции

# Приложение Б

Содержание справки:

* Введение
* Описание программы
* Применение программы
* Алгоритм программы
  + - Используемые алгоритмы
* Функциональные ограничения
* Ссылки
* Руководство пользователя
* Описание формы «Параметры функций z=f(x,y,t)»
* Описание графического окна «Визуализация функций z=f(x,y,t)»
* Доступные функции
* О программе
* Об авторе

**Введение**

Визуализационные инструменты всегда были и остаются востребованными на рынке разработки программного обеспечения.

Визуализация — это наглядное представление различной информации. Визуальная информация лучше воспринимается и позволяет быстро и эффективно донести до зрителя собственные мысли и идеи. Физиологически, восприятие визуальной информации является основной для взаимодействия человека и окружающего мира. Есть многочисленные исследования, подтверждающие, что 90% информации человек воспринимает через зрение.

Программа «Визуализатор пространственно-временных функций» предназначена для визуализации функций 1 или 2 переменных координат и времени z = f (x,y,t).

Приложение позволяет построить постоянные (далее - статические) или изменяющиеся во времени (далее - динамические) графики введенных функций в удобном для восприятия виде. Для этих целей реализованы возможности вращения, цветового отображения графиков и изменения их масштаба.

**Описание программы**

В программе реализованы следующие возможности:

* Ввод нескольких функций в инфиксной нотации (например: 2 + 2 \* 2);
* Ввод параметров, необходимых для построения графиков;
* Преобразование функций в обратную польскую запись (например: 2+2\*2 => 2 2 2 \* +);
* Расчет матриц значений функций в форме обратной польской записи (для каждого момента времени, если функция динамическая);
* Отображение графиков функций z=f(x,y,t) в разных цветовых схемах в отдельном окне;
* Отрисовка осей, вывод минимальных и максимальных значений координат и времени.

**Применение программы**

Практическое применение этой программы:

* в различных технических науках, где требуется визуализировать пространственно-временные функции, например, в математике, физике, информатике и др.
* она может быть полезна на различных предприятиях и учреждениях, где потребуется визуализировать функции 1 или 2 переменных координат и времени;
* школьникам для лучшего понимания вида различных функций. Это поможет им при решении поставленных задач;
* учителям математики, физики, информатики, географии, биологии и др. при графическом представлении информации;
* преподавателям и студентам высших учебных заведений для наглядности представляемой информации.

**Алгоритм программы**

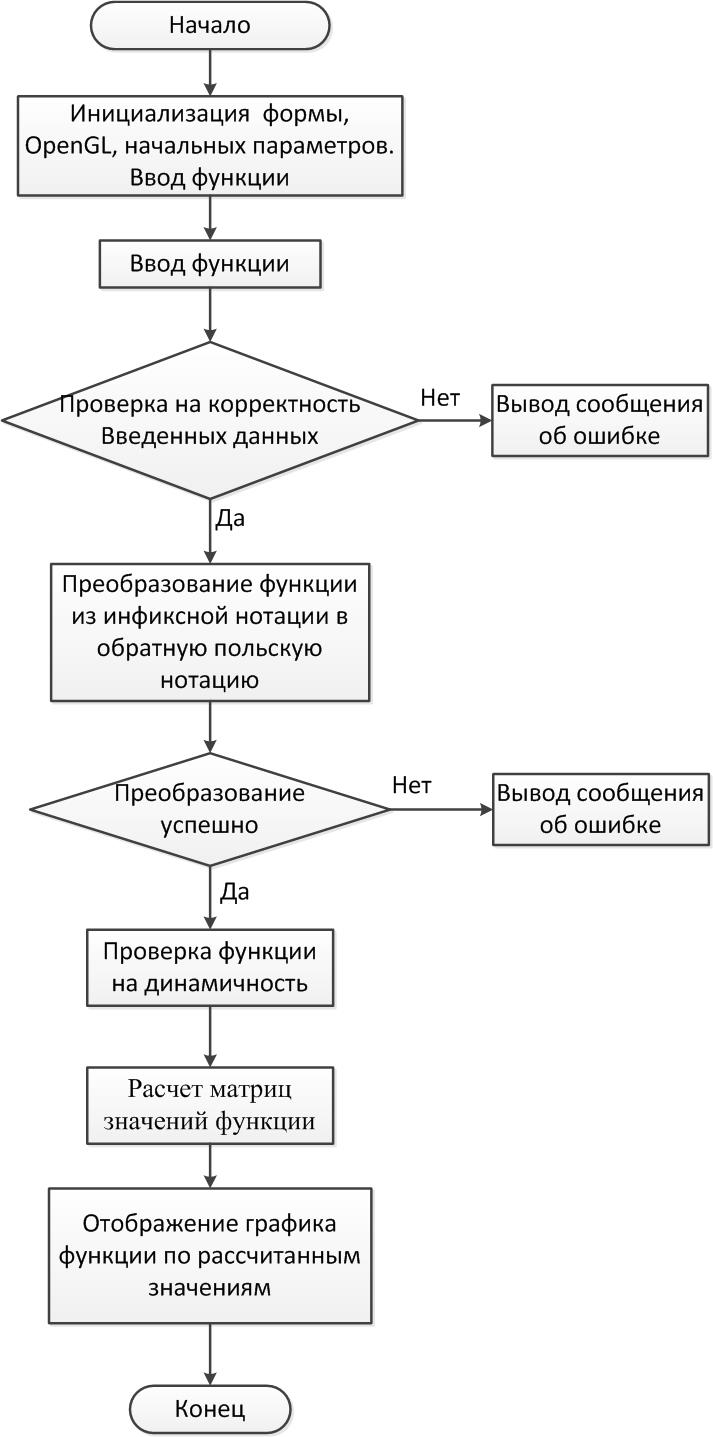


Рисунок Б.1 – Алгоритм программы

**Используемые алгоритмы**

1. [Преобразование из инфиксной нотации в обратную польскую](mk:@MSITStore:E:\MAX\C++new\3d%20visualizator_v5\Release\data\3D_Visualizer_v5.chm::/-5.html) (алгоритм приведен без обработки ошибок и исключений)

* Пока есть ещё символы для чтения:
* Читаем очередной символ.
* Если символ является числом, добавляем его к выходной строке.
* Если символ является символом функции, помещаем его в стек.
* Если символ является открывающей скобкой, помещаем его в стек.
* Если символ является закрывающей скобкой:

До тех пор, пока верхним элементом стека не станет открывающая скобка, выталкиваем элементы из стека в выходную строку. При этом открывающая скобка удаляется из стека, но в выходную строку не добавляется. Если стек закончился раньше, чем мы встретили открывающую скобку, это означает, что в выражении либо неверно поставлен разделитель, либо не согласованы скобки.

* Если символ является оператором о1, тогда:
  1. пока…

… (если оператор o1 право-ассоциированный) приоритет o1 меньше приоритета оператора, находящегося на вершине стека…

… (если оператор o1 ассоциированный, либо лево-ассоциированный) приоритет o1 меньше либо равен приоритету оператора, находящегося на вершине стека…

… выталкиваем верхний элемент стека в выходную строку;

* 1. помещаем оператор o1 в стек.
* Когда входная строка закончилась, выталкиваем все символы из стека в выходную строку. В стеке должны были остаться только символы операторов, если это не так, то в выражении не согласованы скобки.

1. Расчет матрицы значений функций в обратной польской записи (алгоритм приведен без обработки ошибок и исключений)

* Для каждого элемента матрицы:
* Считываем очередной символ
* Если операция сложения:

Считываем два числа из стека. Вычисляем сумму этих двух чисел. Помещаем результат в стек.

* Если операция вычитания: считываем два числа из стека. Вычисляем разность второго и первого числа. Помещаем результат в стек.
* Если операция умножения: считываем два числа из стека. Вычисляем произведение двух чисел. Помещаем результат в стек.
* Если операция деления: считываем два числа из стека. Вычисляем частное двух чисел. Помещаем результат в стек.
* Если операция возведения в степень: считываем два числа из стека. Возводим второе число в степень первого числа. Помещаем результат в стек.
* Если символ «x», «y» или «t»: помещаем текущее значение переменной (определяется в главном цикле) в стек.
* Если число: помещаем число в стек.

**Функциональные ограничения**

Вводимые числа не должны по модулю превышать 1010. Количество точек на оси не должно превышать 100, количество кадров в секунду – 100.

Данное ограничение сделано для того, чтобы уменьшить потребляемые программой ресурсы процессора и оперативной памяти, также нецелесообразно делать количество кадров в секунду больше 100, а при качестве 100 точек на оси и правильном выборе диапазона значений координат и времени график близок к точной кривой.

**Перспективы проекта:**

* Найти множество общих точек пересечения этих функций.
* Найти локальных экстремумов функций.
* Реализовать решения всевозможных уравнений.
* Данные могут поступать в виде потока информации и интерактивно отображаться (потоковое видео). Прием информации может осуществляться напрямую из некоторого устройства в поставленных экспериментах (служить программой-драйвером).
* В программу можно будет загружать готовые массивы данных и выводить их в графическое окно. Примером такого решения может быть ландшафт некоторой местности.

Эти возможности говорят о том, что программа имеет практическое применение в различных областях науки, производства, народного хозяйства.

**Список использованных источников**

[1] Материал сайта [Электронный ресурс]: <https://ru.wikipedia.org/>

[2] Материал сайта [Электронный ресурс] <http://cppstudio.com/>

[3] Материал сайта [Электронный ресурс] <https://msdn.microsoft.com/>

[4] Информация статьи сайта [Электронный ресурс] <https://habrahabr.ru/company/devexpress/blog/240325/>

**Доступные функции**

Функции могут содержать:

* переменные x, y, t;
* действительные числа;
* операции сложения (+), вычитания (-), умножения (\*), деления (/), возведения в степень (^);
* скобки ();
* математические функций, описанные ниже.

**Описание доступных функций:**

* ^ возведение в степень, например: 55^x
* возведение в степень имеет правую ассоциативность, т.е. 2^2^3=2^(2^3)
* sqrt, квадратный корень, например: sqrt(5y/3)
* abs, модуль числа, например: abs(x\*y)
* round, округление числа до ближайшего целого, например: round(2.67)
* ceil, округление числа в большую сторону, например: ceil(2.1)
* floor, округление числа в меньшую сторону, например: floor(-x^2)

Аргументы тригонометрических функций указываются в радианах. Значения обратных тригонометрических функций рассчитаны в радианах.

* sin, синус угла, например: sin(x)
* cos, косинус угла, например: cos(5x-1)
* tg, тангенс угла, например: tg(3.14)
* ctg, котангенс угла, например: ctg(1/x^y)
* asin, арксинус числа, например: asin(-0.5)
* acos, арккосинус числа, например: acos(0.6)
* atg, арктангенс числа, например: atg(10^x)
* actg, арккотангенс числа, например: actg(1)
* log, десятичный логарифм числа, например: log(10^(-1))
* ln, натуральный логарифм числа, например: ln(2.718^4)

**О программе**

Программа  «Визуализатор пространственно-временных функций» была разработана самостоятельно. Была представлена на научно-исследовательской конференции «Поиск» в конце 2017 года, где была отмечена дипломом 2 степени.

Был проведен поиск аналогов программы в открытых источниках в сети интернет. Среди них не найдено программ, способных визуализировать графики динамических функций. Разработанная программа уникальна тем, что может строить графики пространственных функций, изменяющиеся во времени.

Было изучено, насколько в настоящее время актуальна визуализация статических и динамических данных. По результатам исследования можно сделать вывод о том, что это очень востребовано при восприятии человеком информации в визуальной форме.

Исходным языком программирования для приложения «Визуализатор пространственно-временных функций» является С++.

Среда разработки - Microsoft Visual Studio 2017. В программе используется графическая библиотека OpenGL.

**Об авторе**

Email: [max.rez@list.ru](mailto:max.rez@list.ru)