**Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение высшего образования  
«Московский государственный технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)**

**Факультет Информатика и системы Управления (ИУ)**

**Кафедра «Информационная безопасность» (ИУ8)**

**Отчет**

**по курсовой работе**

на тему Программа для построения 2D графиков различных функций

ФИО студента: Андронов Сергей Игоревич

Группа: ИУ8-32 Курс: 2

Специальность: 10.05.03 «Информационная безопасность автоматизированных систем»

Руководитель курсового проекта Колесников А. В.

Работа выполнена С.И.Андронов

дата подпись студента И.О.Фамилия студента

Допуск к защите А. В. Колесников

дата подпись руководителя И.О.Фамилия руководителя

Дата защиты КР \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Результаты защиты \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва

2018

# СОДЕРЖАНИЕ

[ЦЕЛЬ КУРСОВОГО ПРОЕКТА 5](#_Toc532499488)

[ЗАДАЧИ 5](#_Toc532499489)

[Арифметический парсер 5](#_Toc532499490)

[Графический интерфейс 5](#_Toc532499491)

[Сохранение 5](#_Toc532499492)

[Документация 5](#_Toc532499493)

[ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ 5](#_Toc532499494)

[ВВЕДЕНИЕ 6](#_Toc532499495)

[АРХИТЕКТУРА 6](#_Toc532499496)

[РАЗРАБОТКА 6](#_Toc532499497)

[Арифметический парсер 6](#_Toc532499498)

[Графический интерфейс 8](#_Toc532499499)

[Сохранение графика 10](#_Toc532499500)

[ПРИМЕР РАБОТЫ 10](#_Toc532499501)

[РЕЗУЛЬТАТЫ 12](#_Toc532499502)

[ДОКУМЕНТАЦИЯ 12](#_Toc532499503)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 15](#_Toc532499504)

# ЦЕЛЬ КУРСОВОГО ПРОЕКТА

Целью курсового проекта является изучение на практике теоретических основ разработки программ, написания документации.

# ЗАДАЧИ

Разработать программу для построения графиков функций с графическим интерфейсом и возможностью сохранения графиков формате BMP.

## Арифметический парсер

Реализовать парсер арифметических выражений способный вычислять значения выражения в любых точках.

## Графический интерфейс

Реализовать графический интерфейс и отрисовку графика.

## Сохранение

Реализовать сохранение клиентской области окна в файл в формате BMP.

## Документация

Создать документацию поясняющую основные возможности и особенности программы.

# ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Парсер или синтаксический анализатор — часть программы, преобразующей входные данные (как правило, текст) в структурированный формат.

WinAPI- общее наименование набора базовых функций интерфейсов программирования приложений операционных систем семейств Microsoft Windows корпорации «Майкрософт».

GDI (Graphics Device Interface, Graphical Device Interface) — один из трёх основных компонентов или «подсистем», вместе с ядром и Windows API, составляющих пользовательский интерфейс (оконный менеджер GDI) Microsoft Windows.

Формат файла BMP (сокращенно от BitMaP) - это формат растровой графики для Windows.

# ВВЕДЕНИЕ

Программа представляет собой графопостроитель для различных функций одной переменной, способный затем эти графики сохранять.

# АРХИТЕКТУРА

Программа написана на языке С++, при помощи WinAPI под операционную систему Windows. Программа реализована при помощи класса ariphmetic\_node, представляющего собой древовидную структуру содержащую арифметическое выражение в удобной для вычисления значений форме, и главного модуля программы в котором реализуется графический интерфейс. Главный модуль состоит из точки входа, функции WinMain, а также функций обработки сообщений для двух главных окон приложений и две функции реализующие сохранение и отрисовку графика.

# РАЗРАБОТКА

## Арифметический парсер

Парсер был реализован с помощью класса ariphmetic\_node находящегося в файлах ariphmetic\_node.h с описанием обьекта и обьявлением его функций и ariphmetic\_node.cpp с определениями соответствующих функций. Сам класс представляет собой узел древовидной структуры со следующими полями:

std::string expression-выражение переданное конструктору узла, без внешних скобок и пробелов;

char var-содержит имя переменной, если узел представляет собой символ латинского алфавита, в противном случае равно NULL;

intnumber-содержит положительное целое число, или -1 в ином случае;

charoperation-символ представляющий собой идентификатор операции выполняемой в узле (например ‘с’ функции косинуса и ‘+’ для сложения) и NULL в противном случае;

ariphmetic\_node \*leftChild-указатель на левый дочерний узел, представляющий собой левый операнд операции выполняемой в узле;

ariphmetic\_node \*rightChild-указатель на правый дочерний узел, представляющий собой правый операнд операции выполняемой в узле;

и следующими функциями:

ariphmetic\_node(std::string expression\_string)-конструктор, создающий дерево арифметического выражения из строки expression\_string, и возвращает его корневой узел ;

double calc(double x)-функция вычисляющая значение функции в точке x;

void check\_except()-проверка выражения на корректность, выкидывает исключения «неверное выражение», «слишком большие числа в выражении» и «слишком маленькие числа в выражении»;

std::string calc\_str(double x)-преобразование значения функции в точке x, в строку;

~ariphmetic\_node()-деструктор освобождающий память.

Так же в файле ariphmetic\_node.h присутствует константная

глобальная переменная std::map<char, double> priorities, содержащая приоритеты операций в виде действительных чисел между 0 и 1.

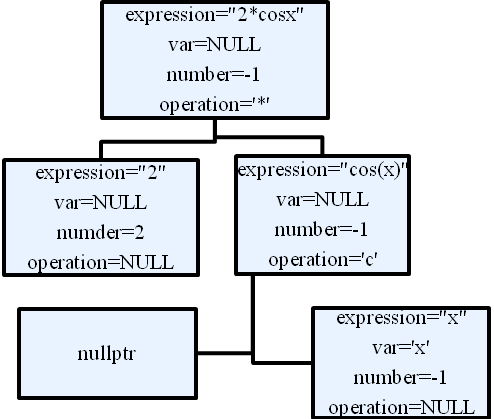


Рис1:дерево объектов ariphmetic\_node,

для выражения "2\*cos(x)"

## Графический интерфейс

Графический интерфейс представляет собой два окна: главное окно программы, где происходит построение графика и реализовано системное меню с пунктом для сохранения графика, и диалоговое «окно построения», в котором пользователь вводит функцию и пределы в которых необходимо построить график в формы ввода. В окне «окно построения» также расположена кнопка «Построить».

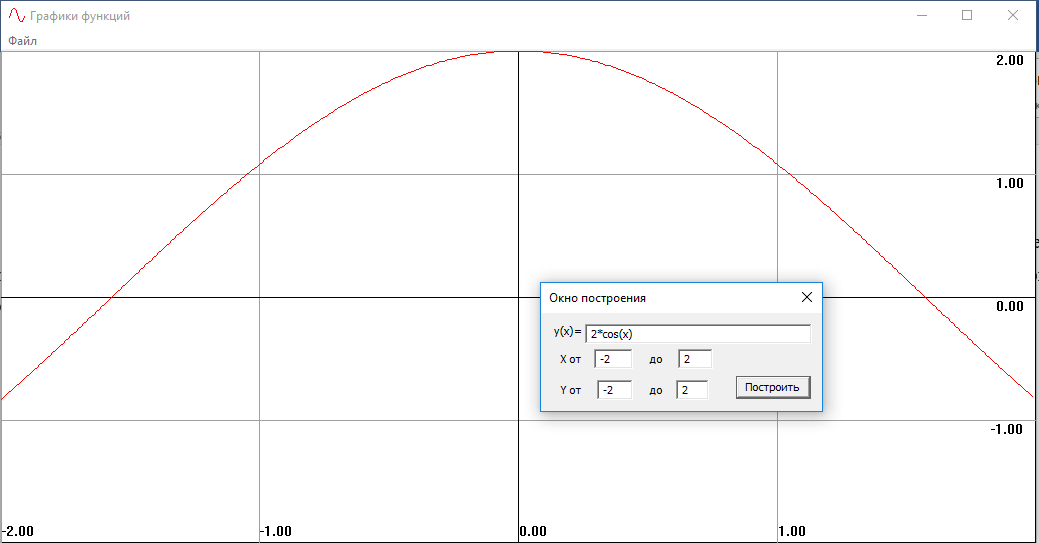


Рис2:графический интерфейс программы

Функция обработки сообщений диалогового окна DlgProc() при получении сообщения от кнопки «Построить» извлекает при помощи функции GetWindowText() информацию из форм и создает на ее основе класс ariphmetic\_node, который сохраняется как глобальная переменная nodefunc.

Для отрисовки графика используется функция DrawFunc(), в которую передается глобальная переменная nodefunc, а также пределы в которых необходимо построить график. Далее DrawFunc при помощи функций GDI: MoveToEx(),LineTo() и TextOut(); рисует координатную сетку и оси, а затем и саму функцию вычисляя ее значения в последовательно расположенных по оси x точках. Для рисования разным цветом используются объекты HPEN, которые создаются функцией CreatePen(), устанавливаются в контекст экрана функцией SelectObject() и удаляются функцией DeleteObject().

Вызов DrawFunc() происходить при нажатии кнопки «Построить» в окне построения, и при получении функцией обработки сообщений главного окна сообщения WM\_PAINT.

## Сохранение графика

Сохранение происходит в функции save(), которая производит захват клиентской области главного окна с помощью функции CreateDIBSection() с переданным в нее дескриптором дисплейного контекста устройства главного окна. Перед этим создаются структуры BITMAPFILEHEADER,BITMAPINFO и BITMAPINFOHEADER, необходимые для корректной записи рисунка в файл. Также объявляется указатель на байтовый массив BYTE\* куда функция CreateDIBSection() загружает сам точечный рисунок. Далее используется структура OPENFILENAME и функция GetSaveFileName(), которые реализуют вызов стандартного диалога сохранения windows, для получения имени создаваемого файла. Далее с помощью функций CreateFile() и WriteFile(),программа создает новый файл и записывает туда все данные связанные с полученным точечным рисунком.

# ПРИМЕР РАБОТЫ

Cначала соберем проект в интегрированной среде разработки Microsoft Visual Studio, откроем наше решение, затем выберем в системном меню Сборка->Собрать решение. Теперь для запуска программы перейдем в папку Release в папке проекта, где лежит exe-файл собранный программы. Теперь мы можем запустить программу и продемонстрировать ее работу.

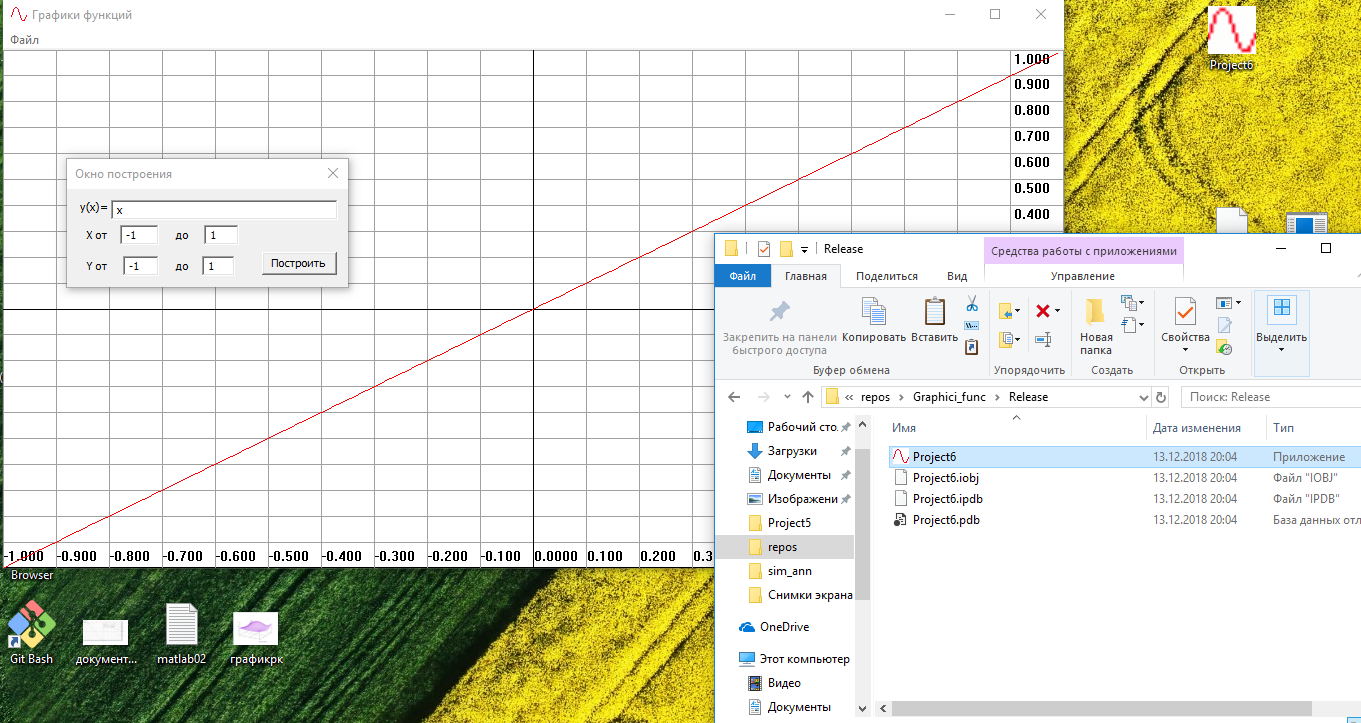


Рис3:демонстрация работы программы

Теперь перейдем в пункт системного меню Файл->Cохранить, и проверим, правильно ли сохраняется файл

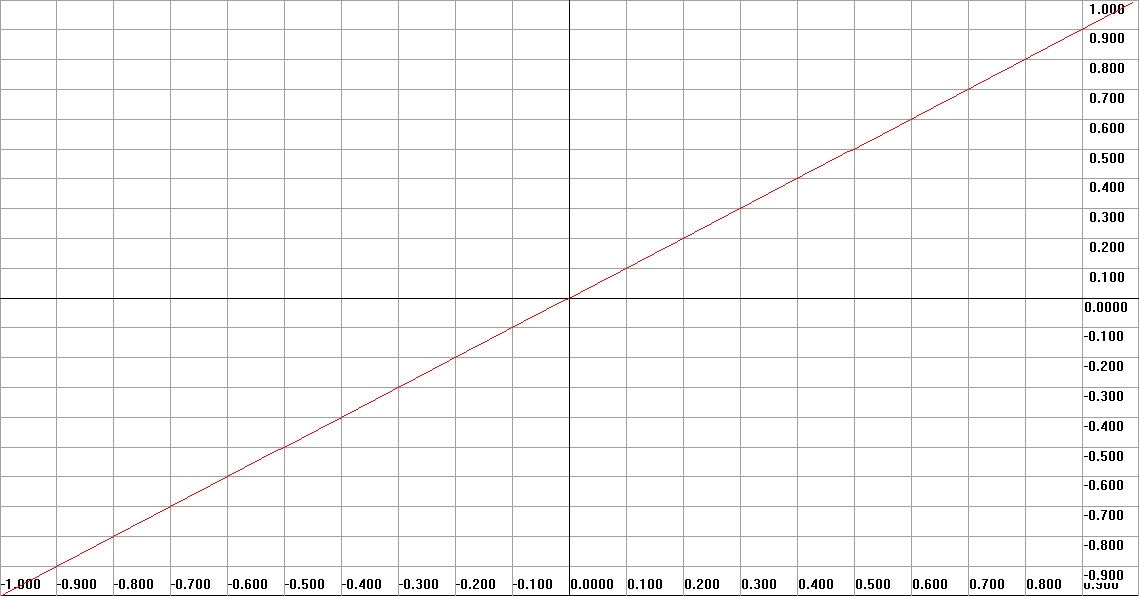


Рис4:Файл полученный в результате сохранения графика программой

# РЕЗУЛЬТАТЫ

В результате работы все задачи были выполнены, а цель достигнута. В будущем планируется дальнейшее совершенствование в разработке программ.

# ДОКУМЕНТАЦИЯ

**Программа для построения графиков функций «Графики функций».**

Программа предназначена для построения графиков, задаваемых пользователем в виде выражений от одной переменной. Графический интерфейс программы представляет собой два основных окна: основное окно программы и окно «Окно построения». В первом из них рисуется непосредственно график, второе служит для ввода пользовательских данных.

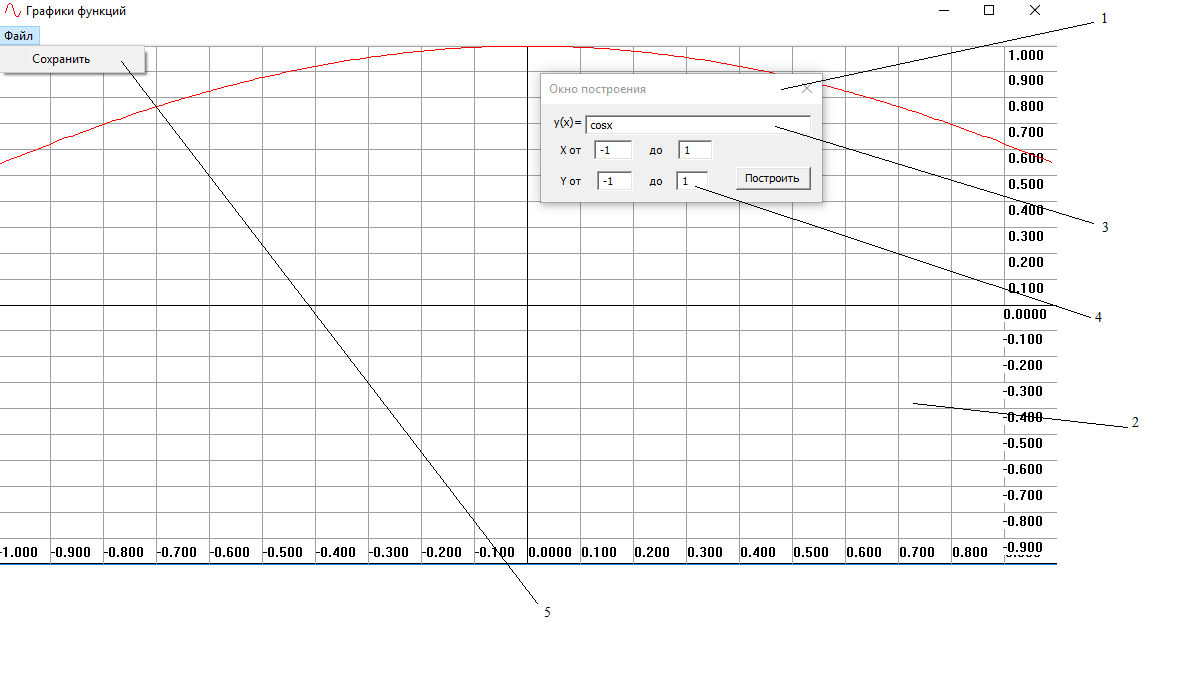


Рис 4: Интерфейс приложения , (1) "Окно построения",(2) главное окно,(3)форма ввода функции, (4) формы ввода пределов построения, (5) пункт меню для сохранения графика

Для построения графика введите в форму ввода функции (3) выражение представляющее собой функцию заданную в виде формулы от x. При вводе разрешается использовать:

1. Знаки операций(“+” -сложение,”-“-вычитание, “\*” -умножение,”/“-деление, “^” –возведение в степень );
2. Скобки “)” и “(“;
3. Латинская буква “x”;
4. Числа целые и десятичные дроби(“2” или “10.5”).
5. Функции(“cos”-тригонометрические функции принимают аргумент в радианах,”tan”,”sin”,”log”-натуральный логарифм,”exp”,”-”-используется как функция после других операций, после открывающей скобки, в начале выражения или после других функций)

Выражение должно содержать правильную скобочную последовательность. Допустимо использовать функции без скобок в случае если операнд передаваемый в функцию представляет собой число или “x”, например “cosx” эквивалентно “cos(x)” и “sin10.5” эквивалентно “sin(10.5)”. Числа используемые в выражении должны лежать между 0.001 и 100, исключением является использование в выражении нуля.

Примеры корректных выражений:

1. (1-x)\*sinx
2. cos-x, эквивалентно cos(-x)
3. cossinexpx, эквивалентно cos(sin(exp(x)))
4. x^-2
5. logcosx
6. (sin(2\*x) )

Далее необходимо ввести пределы построения графика в формы (4), при этом разница между крайними значениями по каждой оси не должен превышать 200, и не должен быть меньше 0.002.

Также в программе предусмотрена возможность сохранения графика в файле формата BMP.Для этого необходимо выбрать пункт системного меню Файл->Сохранить и далее выбрать в стандартном диалоговом окне сохранения windows место для сохранения файла.

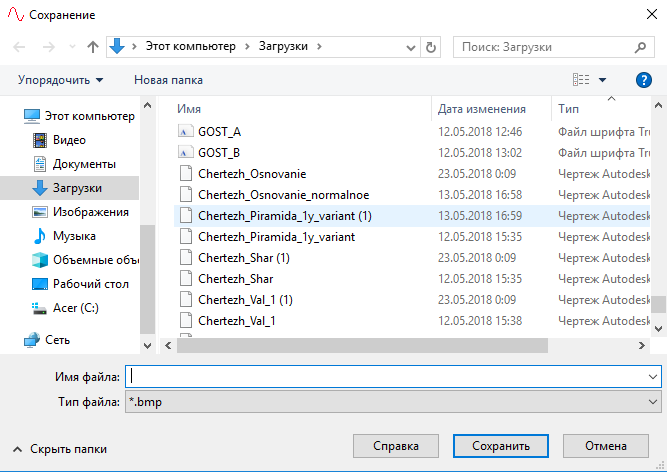


Рис 5: стандартное диалоговое окно сохранения

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Статья о парсинге арифметических выражений[Электронный ресурс]. – Режим доступа: URL: https://habr.com/post/263775/. (дата обращения: 10.10.2018)

2. Начальные сведения о программировании с помощью WinAPI [Электронный ресурс]. – Режим доступа: URL: http://dims.karelia.ru/win32/. (дата обращения: 15.10.2018)

3. Введение в программирование WinAPI[Электронный ресурс]. – Режим доступа: URL: http://www.firststeps.ru/mfc/winapi/. (дата обращения: 15.10.2018)

4. Официальная документация по WinAPI[Электронный ресурс]. – Режим доступа: URL: https://docs.microsoft.com/en-us/windows/desktop/api/. (дата обращения: 16.10.2018)

5. Перевод английской документации по WinAPI[Электронный ресурс]. – Режим доступа: URL: http://vsokovikov.narod.ru/. (дата обращения: 16.10.2018)

6. Требование к курсовому проекту [Электронный ресурс]. – Режим доступа:URL:https://gist.github.com/drewxa/61053007c05d4e03811b919d396632ce. (дата обращения: 22.10.2018)

7. Известный сайт для помощи в программировании [Электронный ресурс]. – Режим доступа: URL: https://stackoverflow.com/. (дата обращения: 26.10.2018)

8. Форум программистов [Электронный ресурс]. – Режим доступа: URL: http://www.cyberforum.ru/ (дата обращения: 26.11.2018)