**Курсовая работа  
по информационным технологиям  
по теме «Графическое приложение: вычисление дискретного преобразования Фурье для синусоиды с линейно изменяющейся частотой»**

|  |  |
| --- | --- |
| Выполнил студент ФРТ гр. 1183 | Чаминов Д. А. |
| Преподаватель | Ситников И. Ю. |

# Содержание

[Содержание 2](#_Toc98872395)

[Спецификация задания 3](#_Toc98872396)

[Постановка задачи 5](#_Toc98872397)

[Формализованное описание алгоритма решения задачи 7](#_Toc98872398)

# Спецификация задания

* 1. Требования к расчёту и программе:
     1. Программа должна выполнять дискретное преобразование Фурье (ДПФ) для сигнала вида в соответствии с введенными пользователем параметрами и визуализировать результат расчёта в виде графика модуля ДПФ;
     2. Программа должна строить график сигнальной функции с возможностью его масштабирования, смещения, использования логарифмического масштаба по оси ординат, изменения этих и параметров сигнальной функции пользователем, а также иметь возможность сохранения графика в файл в формате BMP (bitmap image);
     3. Поддержка универсальной символьной кодировки Unicode;
     4. Наличие графического интерфейса;
     5. Использование пользовательского типа данных и оперирование динамическими массивами;
     6. Программа должна разрабатываться в среде MS Visual Studio на языке программирования C++ с применением библиотек Microsoft Foundation Classes (MFC) как оконное приложение на базе диалогов для операционных систем MS Windows 7 (x32) и выше.
     7. Приложение должно иметь сведения о программе, авторе и авторских правах, название и иконку;
     8. Интерфейс должен использовать кнопки для управления приложением, поля ввода данных для задания сигнальной функции;
     9. В приложении должны содержаться списки выбора;
     10. Интерфейс должен быть русскоязычным, интуитивно понятным, соответствовать понятию «дружественный интерфейс»;
  2. Требования к отчёту:
     1. Отчёт должен соответствовать ГОСТу 19.701–90 единой системы программной документации;
     2. Отчет должен содержать описание программного интерфейса, диаграмму классов и диаграмму потоков данных, выбор и обоснование переменных, пользовательских типов и классов, код программы с комментариями, пример работы программы и контрольный пример;
     3. Контрольный пример должен быть представлен в виде графиков и расчётов в программе Mathcad, подтверждающих правильность результата.

Срок сдачи отчета:

Срок сдачи курсовой работы:

|  |  |  |
| --- | --- | --- |
| Преподаватель |  | Ситников И. Ю. |
| Студент |  | Чаминов Д. А. |

# Постановка задачи

Задача: для сигнала вида , где – параметры, задаваемые пользователем рассчитать дискретное преобразование Фурье, построить график сигнальной функции и график модуля ее ДПФ.

Дискретное преобразование Фурье переводит вектор в вектор частотных отсчетов . Такое преобразование можно рассматривать умножение матрицы на вектор , где – матрица преобразования (унитарная матрица для некоторого базиса пространства ).

Для компонента преобразования справедлива следующая формула:  
где – число отсчетов, – отсчет сигнальной функции с номером , – число Эйлера, – число пи и – мнимая единица. Применяя к этому выражению формулу Эйлера, получим следующее:

Учитывая четность и нечетность , а также то, что в данной задаче сигнал вещественный, получим:

Модуль

Для расчёта положения точки на экране необходимо перейти к другому масштабу и учесть смещение. Если отрезок длиной 1 должен на экране по оси совпадать с длиной пикселей, а по оси – , левая верхняя точка графика имеет координаты , а высота графика равна пикселей, то для перехода к координатам окна можно использовать следующие преобразования:

Разность координат показывает смещение точки относительно координат окна, но в координатах графика; умножение на коэффициент масштаба показывает длину в пикселях проекции радиус-вектора, проведенного из нового начала координат в эту точку; знак минус перед показывает отражение системы координат, а компенсирует вызванное таким поворотом смещение.

# Формализованное описание алгоритма решения задачи

Для выполнения расчётов в программе определено несколько пользовательских классов. Все классы диалогов являются наследниками CDialogEx.

Первый из них – класс начального окна courseworkDlg. Он содержит один CStatic элемент, в котором дано краткое описание задачи, кнопку открытия диалога с более подробными сведениями о задаче, кнопку открытия диалога с расчётом и кнопку закрытия программы.

Диалог с описанием – MyTaskAboutDlg содержит Cstatic с более подробным описанием задания.

Диалог с расчётом Calculator содержит в себе элементы управления для настройки параметров сигнальной функции и графиков: 7 CEdit (3 для параметров функции и 4 для настроек масштаба) , 1 CSliderCtrl для удобной настройки шага, CButton – check box (флажок) для установки логарифмического масштаба, 4 CMFCColorButton – выпадающие списки для выбора цвета, 4 кнопки (обновление графика, его сохранение, сброс к начальным параметрам и закрытие диалога), 15 CStatic (1 для отображения сигнальной функции и 14 для пояснения приглашений ко вводу), а также объект пользовательского класса CMyGraph для отрисовки графиков и объекты пользовательских классов SignalFunction и DFTFunction – объекты, описывающие сигнальную функцию и функцию дискретного преобразования Фурье, необходимые для построения графиков.

При нажатии кнопки сохранения должен вызываться метод для вызова диалога сохранения void Calculator::OnBnClickedButtonSaveGr(), в котором из объекта CMyGraph CGraph извлекается картинка и сохраняется в выбранное место.

При нажатии кнопки сброса вызывается метод void Calculator::OnBnClickedButtonreset(), в котором сбрасываются значения всех элементов управления.

При нажатии кнопки обновления вызывается метод void Calculator::OnBnClickedButtonreset(), в которой вызывается метод получения параметров из всех элементов управления void Calculator::UpdateCalculatorParams(), в котором происходит проверка заполнения элементов (с созданием диалогового окна об ошибке или присвоением соответствующих параметров через методы CMyGraph или напрямую объектам сигнальной или полученной функций). Помимо обновления параметров, из этого метода через CWnd::RedrawWindow() система отправляет элементу CMyGraph CGraph сообщение о необходимости перерисовать окно.

CMyGraph является наследником CStatic. Он содержит в себе автоматический динамический массив объектов типа MathFunction\* – ссылки на функции, графики которых необходимо построить, а также 2 структуры из 2-х чисел типа double – границы области построения.

При получении сообщений о перерисовке вызывается метод void CMyGraph::OnPaint(), который получает от каждой из функций массив точек (при помощи метода MathFunction::get\_points() и строит график, последовательно соединяя точки при помощи CPaintDC::LineTo(Point), предварительно залив фон области построения необходимым цветом.

Класс CMyGraph обладает множеством set’еров, которые вызывают соответствующие методы для каждой из функций.

MathFunction – абстрактный класс (интерфейс), объекты наследников (реализаций) которого содержат сведения об области определения, области построения, размерах расчётного промежутка, типе масштаба, цвете линий функции на графике, а также о том, являются ли рассчитанные точки актуальными; методы установки параметров, расчёта математических точек (чисто виртуальная функция), расчёта точек (в координатах окна), пересчёта точек в координаты окна, а также автоматический динамический массив объектов POINT – точек в координатах окна и метод его передачи для отрисовки.

Метод передачи точек MathFunction::get\_points() проверяет их актуальность и вызывает (в случае если данные устарели) метод пересчёта точек MathFunction::calulate(), который с заданным шагом на промежутке пересечения области определения и расчётной области вызывает метод MathFunction::f() расчёта функции в этой точке и метод перевода координат, полученной таким образом точки в координаты окна POINT MathFunction::to\_the\_new\_coords\_system(double x, double y) const, которая основываясь на формулах пересчета приведенных выше вычисляет координаты точки в окне.

Классы SignalFunction и DFTFunction содержат в себе некоторые дополнительные сведения (коэффициенты и значения функции в отсчетах для сигнальной функции и ссылка на сигнальную функцию для класса функции ДПФ) и методы установки этих параметров.

Расчёт в классе SignalFunction отличается тем, что значения функции в точках на расчетном промежутке сохраняются в vector<double> data для дальнейшего применения ДПФ.

В классе DFTFunction расчёт отличается тем, что в качестве параметра функции f передается номер отсчета, для которого производятся вычисления. В самой функции f происходит расчёт мнимой и действительной частей при помощи суммирования данных сигнальной функции с коэффициентами, соответствующими номерам отсчета. Затем находится модуль ДПФ путем извлечения корня из суммы квадратов мнимой и реальной частей.

# Варианты взаимодействия оператора и программы (Use Case)

# Диаграмма потоков данных (Data flow diagram)

# Выбор и обоснование типов переменных. Разработка структур данных

В программе для данных о координатах окна или о координатах объекта в окне используется типы POINT и Cpoint, которые хранят два целых числа int, диапазона которых достаточно, чтобы однозначно указать одну из координат точки экрана.

Для хранения данных о границах элементов используются типы CRect и RECT, которые хранят координаты двух противолежащих точек прямоугольника в виде целых чисел int.

Для представления цвета используется тип COLORREF, который является синонимом беззнакового 32-разрядного целого (unsigned long int) хранит данные о трех компонентах цвета (красной, зеленой и голубой) в виде битовых полей.

В качестве блока расчёта выступают несколько объектов типа MathFunction. Этот тип содержит базовые данные о функции (здесь под функцией MathFunction подразумевается некоторый объект, который численными методами генерирует последовательность точек в данной области с заданным шагом для отрисовки графика, причем точки определяются в координатах области построения): область определения некоторой функции (в виде 2х чисел с плавающей точкой двойной точности double), данные об области вычислений (в виде анонимной структуры из 4х чисел double, содержащей начальные и конечные значения по каждой из осей), данные об области отрисовки (CRect), шаг расчёта (double), данные о цвете графика функции (COLORREF), индикатор того, произведен ли уже расчёт (bool) и индикатор логарифмического масштаба (bool). Все вышеперечисленные параметры кроме цвета являются защищенными (недоступны вне дочерних классов), поскольку их изменение напрямую может привести к недостоверности сведений об актуальности рассчитанных координат (несоответствие индикатора is\_calculated действительности, которое в свою очередь приводит к сохранению неактуальных данных). Для этих параметров определены соответствующие set’теры и get’теры (только для массива точек и индикатора готовности).