Кафедра электронно-вычислительной аппаратуры

**Математическая программа "Производная"**

Москва 2009

# Аннотация

Работа представляет собой программу (реализованную на языке Microsoft Visual Basic 6), основной функцией которой является нахождение формулы производной на основании введенной пользователем исходной формулы, представляющей собой суперпозицию (произвольное сочетание) элементарных функций (предусмотрены программой sin, cos, tg, ctg, arcsin, arccos, arctg, arcctg, ln, exp, +, -, \*, /, ^, константы).

Для удобства программа обладает дополнительными функциями: нахождение значения формулы при данных аргументах, приближенное нахождение интеграла, работа с переменными (изменение аргументов), вызов справки.

Программа использует стандартные библиотеки Microsoft Windows (98, XP, 2000), поэтому для корректной работы программы необходимо использовать эту операционную систему.

# Техническое задание

Основное окно программы в рабочем режиме показано на рис 1.

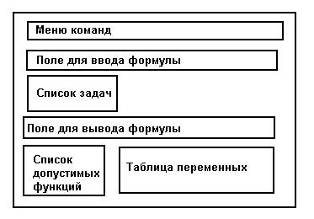


Рис. 1

Компоненты изображения:

* + меню команд в верхней части экрана;
  + поле для ввода формулы, включающее
  + список переменных,
  + поле для ввода строки;
  + список задач, включающий:
  + три кнопки «Производная», «Интеграл», «Значение»;
  + два текстовых поля для ввода границ интегрирования;
  + текстовое поле для вывода формулы;
  + список допустимых функций (объект класса ListBox);
  + таблица переменных, включающая:
  + таблицу класса MSHFlexGrid библиотеки «MSHFLXGD.OCX»,
  + три кнопки управления таблицей «Значение», «Новая переменная», «Сброс».

**Выбор команды**

* Для выбора команды используется одноуровневое горизонтальное меню.
* Для перемещения по меню используются клавиши навигации и / или мышь.
* Для выбора пункта используется клавиша Enter и / или левая клавиши мыши.

**Команды меню**

**Файл -> Выход**

При выходе программа заканчивает свою работу.

**Действия -> Производная**

Программа анализирует введенную формулу и в соответствующем поле выводит формулу производной. (При введении неверной формулы выводится ошибка).

**Действия -> Интеграл**

Программа анализирует введенную формулу и в соответствующем поле выводит значение частичных сумм под графиком функции. (При введении неверной формулы выводится ошибка).

**Действия -> Значение**

Программа анализирует введенную формулу и в соответствующем поле выводит значение функции при аргументах, указанных пользователем. (При введении неверной формулы выводится ошибка).

**Справка -> Мне нужна помощь**

На экране появляется окно, содержащее руководство пользователя.

**Справка -> Об авторе**

На экране появляется окно, содержащее информацию о создателе.

**Поле для ввода формулы**

Текст из этого поля используется в качестве исходной функции пользователя.

**Список задач**

Команды аналогичны командам меню.

**Поле для вывода функции**

Это поле служит для выведения конечного результата.

**Список допустимых функций**

Служит подсказкой пользователю как правильно написать функцию.

Двойной щелчок на элементе добавляет его в поле для ввода формулы.

**Таблица переменных**

Служит подсказкой пользователю, какие переменные можно использовать;

Двойной щелчок на элементе позволяет изменить значение переменной.

Таблица связана с тремя кнопками.

**Значение переменной**

Изменить значение выбранной переменной на указанной пользователем.

**Новая переменная**

Добавляет в таблицу новую переменную, имя которой указал пользователь.

**Сброс переменных**

Очищает таблицу переменных и создает переменные по умолчанию (x=0, y=0, z=0, t=0).

# Введение

Алгоритм программы основан на теории математического анализа. В частности, в основу положены два утверждения:

– Все элементарные функции дифференцируемы на области определения. (Sin, cos, tg, ctg, arcsin, arccos, arctg, arcctg, ln, exp, +, -, \*, /, ^, константы и т.д.)

– Суперпозиция дифференцируемых функций дифференцируема: (f (g(x)))’=g’ (x) f’ (g(x)).

Из этих утверждений следует, что любая суперпозиция элементарных функций дифференцируема.

В основу вычисления производной решено положить *таблицу элементарных функций*, т. к. любой суперпозиции элементарных функций соответствует список последовательных элементарных функций. Работа программы (т.е. выполнение необходимых процедур) сводится к составлению и анализу этой таблицы, а также проверки входных данных.

Например, при вводе функции y(x)=2^2+2sinx и нажатии кнопки «Производная» программа:

– вставляет символ «\*» между 2 и sin x;

– берет x после sin в скобки;

– проверяет формулу на наличие синтаксических ошибок;

– составляет таблицу функции, которая выглядит следующим образом:

&F1=sin(x);

&F2=2^2;

&F3=2\*&F1;

&F0=&F2+&F3.

(&F0 аналогична функции main в С, т.е. с нее начинается вычисление)

– упрощает таблицу функции:

&F1=sin(x);

&F3=2\*&F1;

&F0=4+&F3.

– по таблице рекурсивно (сводя к случаям x’=1 и const’=0) вычисляет производную по правилам дифференцирования:

y’ (x)=(4+2\*sin(x))’=(4)’+(2\*sin(x))’=0+(2’\*sin(x)+2\*(sin(x))’)= 0+(0\*sin(x)+2\*(x’\*cos(x)))= 0+(0\*sin(x)+2\*(1\*cos(x)))

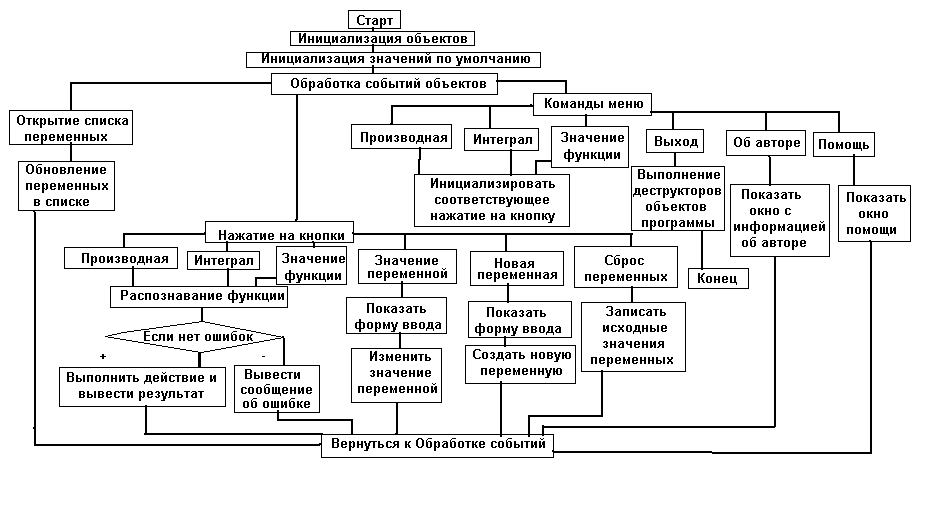
Данная функция является производной, но имеет трудно читаемый вид. Поэтому в алгоритм включен следующий пункт:

– Упрощение функции:

y’ (x)= 0+0\*sin(x)+2\*(1\*cos(x))= 0+0+2\*cos(x)= 2\*cos(x).

# 1. Функциональная структура программы

На основе технического задания составлена функциональная структура программы, которая отображает иерархию основных функций и отвечает на вопрос «Что должна делать программа?». Порядок выполнения функций, способы их оформления в виде подпрограмм и модулей, способы реализации функциональная структура не отображает.



### 2. Принципы построения программы

При построении программы решены следующие проблемы.

**2.1 Управление обработкой входных событий**

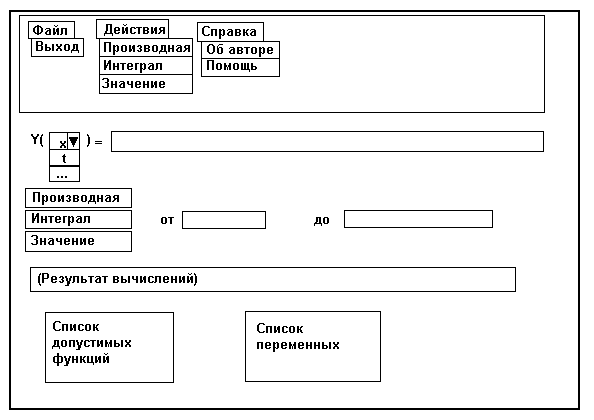
Входными событиями могут быть события нажатия клавиш и кнопок ручного манипулятора типа «мышь».

В объектах программы реализован цикл ввода и обработки событий.

**2.2 Вывод объектов на экран**

Предусмотрена работа программы в среде Windows на графическом адаптере VGA и в режиме монитора 800х600 точек (и выше). В программе явным образом указаны координаты видимых объектов относительно левого верхнего угла окна приложения.

На рисунке показаны координаты основных видимых объектов.



### 3. Основные структуры данных и алгоритмы

### 

### 3.1 Основной алгоритм

Начало

Инициализировать объекты

Цикл

Обработать события

КЦикл

Конец

**Инициализировать объекты**

Инициализировать Командное меню и добавить его элементы.

Инициализировать Кнопки «Производная», «Интеграл», «Значение функции», «Значение переменной», «Новая переменная», «Сброс переменных».

Инициализировать Список Допустимых Функций и добавить его элементы

Инициализировать Таблицу Переменных

**Установить Таблицу Переменных по умолчанию**

Инициализировать Таблицу Элементарных Функций

Инициализировать Поле Ввода Функции, Поле Вывода Результата

Инициализировать Список переменных

**Обновить Список переменных**

Конец

Алгоритм **Обработать события**

Если Развернут Список переменных

**Обновить Список переменных**

КЕсли

Если Нажата кнопка «Производная»

**Найти производную**

КЕсли

Если Нажата кнопка «Интеграл»

**Найти Интеграл**

КЕсли

Если Нажата кнопка «Значение функции»

**Найти Значение функции**

КЕсли

Если Нажата кнопка «Значение переменной»

**Изменить** **Значение текущей переменной**

КЕсли

Если Нажата кнопка «Новая переменная»

**Создать новую переменную пользователя**

КЕсли

Если Нажата кнопка «Сброс переменных»

**Сбросить переменные**

КЕсли

Если Команда меню «Производная»

**Найти производную**

КЕсли

Если Команда меню «Интеграл»

**Найти Интеграл**

КЕсли

Если Команда меню «Значение функции»

**Найти Значение функции**

КЕсли

Если Команда меню «Выход»

*Завершить программу*

КЕсли

Если Команда меню «Справка»

*Показать справку*

КЕсли

Если Команда меню «Об авторе»

*Показать Окно Об авторе*

КЕсли

Конец

Алгоритм **Установить Таблицу Переменных по умолчанию**

*Стереть таблицу переменных*

**Объявить новую переменную пользователя** («x»)

**Объявить новую переменную пользователя** («y»)

**Объявить новую переменную пользователя** («z»)

**Объявить новую переменную пользователя** («t»)

Конец

Алгоритм **Создать новую переменную пользователя**

*Строка = Ввести строку*

Если **Хорошее** **Имя переменной** (строка)

**Объявить новую переменную пользователя** (строка)

КЕсли

Конец

### 3.2 Реализация алгоритма распознавания функции

Реализация алгоритма распознавания функции нахождения производной, интеграла и значения функции описана ниже:

**Распознавание функции**

* Создание таблицы связанных элементарных (содержащих одно действие) функций, последовательное вычисление которых приводит к вычислению сложной функции.
  + Создание размеченной строки: выделение чисел в служебные символы []; переменных в {}; функций в <>. (Контекстно, т.е. в строке «exp» не будет выделена переменная «x»).
  + Создание на основании размеченной строки карты функции, т.е. выделение из неё только символов () [] {} <> и определение их позиций в размеченной строке.
  + Анализ карты функции и вставка в размеченную строку соответствующих замен (например, знака умножения между числом и переменной), а также поиск синтаксических ошибок (например, если строка начинается со знака «/»)
  + Обработанная таким образом строка, в которой не были найдены синтаксические ошибки, подлежит разложению на элементарные функции, которое заключается в следующем:
    - поиск самой внутренней скобки
    - нахождение первого оператора и его операндов в порядке важности (^,\*, /, +,–)
    - Если в скобке больше нет операторов, заменить всю скобку, иначе заменить только данный оператор с операндами на новую переменную (вида &F#) и записать это в таблицу.
    - Если во всей строке еще есть операторы, повторить иначе создать последнюю переменную &F0
  + Дополнить таблицу переменными пользователя.

**Нахождение значения функции в точке (x, y, z, t,…)**

* + Найти переменную (элементарную функцию) в таблице.
  + Считать оператор и операнды; если операнды – переменные, то
  + Если операнд переменная не дифференцируемая, то найти рекурсивно (этой же функцией).
  + Если операнд переменная дифференцируемая, то вернуть текущее значение дифференцируемой переменной.
  + Выполнить действие над операндами в зависимости от оператора и вернуть значение.

**Нахождение частичных сумм площадей под графиком функции (~ интеграла)**

* Установить текущую переменную интегрирования / дифференцирования (на основании сведений полученных от пользователя).
* Просуммировать произведения значений функции на изменения аргумента (на основании сведений полученных от пользователя и пользуясь определением частичных сумм).
* Вернуть результат

**Нахождение формулы производной**

* + Найти переменную (элементарную функцию) в таблице.
  + Считать оператор и операнды; если операнды – переменные, то
  + Если операнд переменная не дифференцируемая, то найти рекурсивно (этой же функцией).
  + Если операнд переменная дифференцируемая, то вернуть «1».
  + Найти производную по правилам дифференцирования и вернуть значение.

# 

# Заключение

В результате проектирования создана полностью работоспособная программа в соответствии с техническим заданием. Программа оформлена в виде структурных записей. Написание программы «Производная» на Visual Basic 6.0 является примером использования объектно-ориентированных средств языка Visual Basic.

Программа может иметь практическое применение при математических вычислениях.

# Список литературы

1. Microsoft Visual Basic для профессионалов. Шаг за шагом: Практ. пособ. / Пер. с англ. - М: Издательство ЭКОМ, 1999. – 720 с.