

Лабораторная работа №11 (3-й семестр)

Сначала учите науку программирования и всю теорию. Далее выработаете свой программистский стиль. Затем забудьте все и просто программируйте.
— George Carrette

Построение графиков с помощью wxMathPlot

WxMathPlot представляет собой надстройку на библиотеку для WxWidgets, которая обеспечивает основу для рисования графиков математических функций и данных по точкам (2D).

Ядром библиотеки wxMathPlot является класс mpWindow, который представляет собой 2D холст. mpWindow может встраиваться как в окна (wxPane, wxFrame), так и любой другой компонент wxWindow. mpWindow обеспечивает масштабирование и сдвиг слоев. Текущим видом можно управлять с помощью: мыши, полос прокрутки и контекстного меню. Основные классы приведены библиотеки wxMathPlot приведены в табл. 1.

Таблица 1 — Основные классы wxMathPlot

Класс	Описание
mpFX	Класс для построения функций вида $y=f(x)$
mpFXY	Класс для построения функций вида $[x=f(t); y=f(t)]$
mpFXYVector	Класс для построения функций по точкам
mpFY	Класс для построения функций вида $x=f(y)$
mpInfoCoords	Базовый класс, который реализует окно для вывода координат точки под указателем мыши в текущей системе координат
mpInfoLayer	Базовый класс для вывода информации на график
mpInfoLegend	Класс легенды
mpLayer	Базовый класс отображения на графике. Родитель для mpFX, mpFY...
mpText	Класс для вывода текста на экран
mpWindow	Класс окна графика

Для начала работы с компонентом разместите его на форме (MathPlot→mpWindow). Для отображения осей координат необходимо добавить два компонента mpAxis, в настройках которых axis type (тип оси) выбрать X-axis (ось X) или Y-axis (ось Y) соответственно. Также необходимо снять галочку Show Tick Marks для отображения сетки. Более детально принцип работы с компонентом рассмотрим на примере.

Пример. Разработать программу, которая выводит на экран графики функций $y=\sin(x)$, $y=\cos(2x)$ и функции, заданной по точкам. Результат построения сохраняется в файл plot.png. Результат работы программы показан на рис. 1.

```
class mySin : public mpFX // Наследуем класс от mpFX
{
public:
    mySin() : mpFX( wxT("sin(x)") ) {}
    virtual double GetY( double x )
    {
        return sin(x);
    }
};
class myCos2x : public mpFX // Наследуем класс от mpFX
{
public:
    myCos2x() : mpFX( wxT("cos(2x)") ) {}
};
```

```

virtual double GetY( double x )
{
    return cos (2*x);
}
};

void plotgrFrame::OnButton1Click(wxCommandEvent& event)
{
    mpLayer *sx = new mySin(); // Создаем экземпляр класса mySin
    mpLayer *c2x = new myCos2x();
    // Создаем экземпляр класса myCos2x
    MathPlot1->AddLayer(sx); // выводим их на экран
    MathPlot1->AddLayer(c2x);
    MathPlot1->Fit(M_PI, 4*M_PI, -1, 1);
    // масштабируем окно  $\pi < x < 4\pi$ ;  $-1 < y < 1$ 
    sx->SetPen(wxPen(*wxRED, 4, wxPENSTYLE_SOLID));
    c2x->SetPen(wxPen(*wxBLUE, 4, wxPENSTYLE_SOLID));
    // задаем цвет и стиль линий, которыми выводятся графики
    // Построение графика по точкам
    mpFXYVector* vectorLayer = new mpFXYVector(_("Vector"));
    // создаем два вектора для x, y и заполняем их данными
    std::vector<double> vectorx, vectory;
    double xcoord;
    for (unsigned int i = 0; i < 100; i++) {
        xcoord = (double)i/100.0*4*M_PI+M_PI;
        vectorx.push_back(xcoord);
        vectory.push_back((rand()%200)/100.0-1.0);
    }
    vectorLayer->SetData(vectorx, vectory);
    // Привязываем вектора к слою отображения
    vectorLayer->SetContinuity(true);
    // Выводим график непрерывной линией
    wxPen vectorpen(*wxGREEN, 2, wxSOLID);
    vectorLayer->SetPen(vectorpen);
    vectorLayer->SetDrawOutsideMargins(false);
    MathPlot1->AddLayer(vectorLayer);
    // Настраиваем цвет и стиль линии графика, выводим на экран
    MathPlot1->SaveScreenshot(_("shot.png"), wxBITMAP_TYPE_PNG,
    wxSize(1024, 1024), false); // Сохраняем полученный график в файл
}

```

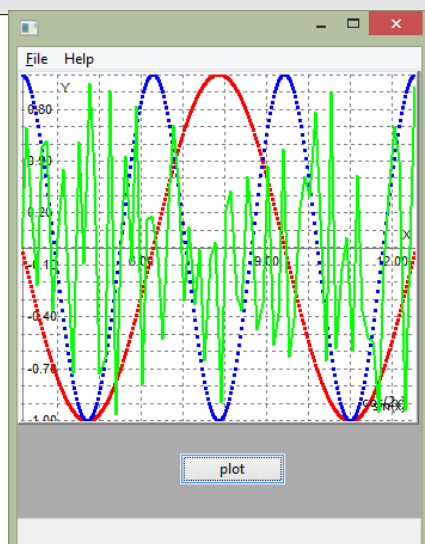


Рис.1 — Пример работы программы