

HarrixQtLibraryForQWebView v.1.15

А. Б. Сергиенко

11 июня 2014 г.

Аннотация

Библиотека HarrixQtLibraryForQWebView — библиотека для отображения различных данных в QWebView, включая графики.

Содержание

1	Введение	3
2	Список функций	4
3	Функции	6
3.1	Главные загрузочные функции	6
3.1.1	HQt_AddHtml	6
3.1.2	HQt_BeginHtml	6
3.2	Показ графиков	6
3.2.1	HQt_DrawLine	6
3.2.2	THQt_ShowChartOfLine	7
3.2.3	THQt_ShowChartsOfLineFromMatrix	9
3.2.4	THQt_ShowIndependentChartsOfLineFromMatrix	10
3.2.5	THQt_ShowTwoChartsOfLine	11
3.2.6	THQt_ShowTwoIndependentChartsOfLine	13
3.2.7	THQt_ShowTwoIndependentChartsOfPointsAndLine	14
3.3	Показ математических выражений	16
3.3.1	THQt_ShowMatrix	16
3.3.2	THQt_ShowMatrix2	16
3.3.3	THQt_ShowNumber	17

3.3.4	THQt_ShowVector	17
3.3.5	THQt_ShowVector2	17
3.3.6	THQt_ShowVectorT	18
3.4	Показ текста	18
3.4.1	HQt_ShowAlert	18
3.4.2	HQt_ShowH1	19
3.4.3	HQt_ShowHr	19
3.4.4	HQt_ShowSimpleText	19
3.4.5	HQt_ShowText	20
3.4.6	THQt_NumberToText	20

1 Введение

Библиотека `HarrixQtLibraryForQWebView` — это библиотека для отображения различных данных в `QWebView`, включая графики..

Последнюю версию документа можно найти по адресу:

<https://github.com/Harrix/HarrixQtLibraryForQWebView>

Об установке библиотеки можно прочитать тут:

<http://blog.harrix.org/?p=1196>

С автором можно связаться по адресу sergienkoanton@mail.ru или <http://vk.com/harrix>.

Сайт автора, где публикуются последние новости: <http://blog.harrix.org/>, а проекты располагаются по адресу <http://harrix.org/>.

2 Список функций

Главные загрузочные функции

1. **HQt_AddHtml** — Функция добавляет код html к существующему и сохраняет его в temp.html.
2. **HQt_BeginHtml** — Функция обнуляет переменную HTML. Требуется когда нужно перезапустить показ информации в QWebView.

Показ графиков

1. **HQt_DrawLine** — Функция возвращает строку с HTML кодом отрисовки линии по функции Function. Для добавление в html файл.
2. **THQt_ShowChartOfLine** — Функция возвращает строку с выводом некоторого графика по точкам с HTML кодами. Для добавление в html файл.
3. **THQt_ShowChartsOfLineFromMatrix** — Функция возвращает строку с выводом графиков из матрицы по точкам с HTML кодами. Для добавление в html файл.
4. **THQt_ShowIndependentChartsOfLineFromMatrix** — Функция возвращает строку с выводом графиков из матрицы по точкам с HTML кодами. Для добавление в html файл.
5. **THQt_ShowTwoChartsOfLine** — Функция возвращает строку с выводом некоторого двух графиков по точкам с HTML кодами. Для добавление в html файл. У обоих графиков одинаковый массив значений X.
6. **THQt_ShowTwoIndependentChartsOfLine** — Функция возвращает строку с выводом некоторого двух независимых графиков по точкам с HTML кодами. Для добавление в html файл.
7. **THQt_ShowTwoIndependentChartsOfPointsAndLine** — Функция возвращает строку с выводом некоторого двух независимых графиков по точкам с HTML кодами. Для добавление в html файл. Один график выводится в виде точек, а второй в виде линии. Удобно для отображения регрессий. У обоих графиков разные массивы значений X и Y.

Показ математических выражений

1. **THQt_ShowMatrix** — Функция возвращает строку с выводом некоторой матрицы VMHL_Matrix с HTML кодами. Для добавление в html файл.
2. **THQt_ShowMatrix2** — Функция возвращает строку с выводом некоторой матрицы VMHL_Matrix с HTML кодами. Для добавление в html файл. В качестве матрицы выступает массив QStringList, где количество QStringList - это количество строк. Каждый QStringList — это одна строка.
3. **THQt_ShowNumber** — Функция возвращает строку с выводом некоторого числа VMHL_X с HTML кодами. Для добавление в html файл.
4. **THQt_ShowVector** — Функция возвращает строку с выводом некоторого вектора VMHL_Vector с HTML кодами. Для добавление в html файл.

5. **THQt_ShowVector2** — Функция возвращает строку с выводом некоторого списка строк VMHL_Vector с HTML кодами. Для добавление в html файл.
6. **THQt_ShowVectorT** — Функция возвращает строку с выводом некоторого вектора VMHL_Vector в транспонированном виде с HTML кодами. Для добавление в html файл.

Показ текста

1. **HQt_ShowAlert** — Функция возвращает строку с выводом некоторого предупреждения. Для добавление в html файл.
2. **HQt_ShowH1** — Функция возвращает строку с выводом некоторой строки в виде заголовка. Для добавление в html файл.
3. **HQt_ShowHr** — Функция возвращает строку с выводом горизонтальной линии. Для добавление в html файл.
4. **HQt_ShowSimpleText** — Функция возвращает строку с выводом некоторой строки с HTML кодами без всякого излишества. Для добавление в html файл.
5. **HQt_ShowText** — Функция возвращает строку с выводом некоторой строки с HTML кодами. Для добавление в html файл.
6. **THQt_NumberToText** — Функция выводит число VMHL_X в строку.

3 Функции

3.1 Главные загрузочные функции

3.1.1 HQt_AddHtml

Функция добавляет код html к существующему и сохраняет его в temp.html.

Код 1. Синтаксис

```
void HQt_AddHtml(QString Html);
```

Входные параметры:

Html — добавляемый текст.

Возвращаемое значение:

Отсутствует.

3.1.2 HQt_BeginHtml

Функция обнуляет переменную HTML. Требуется когда нужно перезапустить показ информации в QWebView.

Код 2. Синтаксис

```
void HQt_BeginHtml(QString Path);
```

Входные параметры:

Path — путь к папке, в которой надо будет сохранять html код. В этой папке должен содержаться файл index.html.

Возвращаемое значение:

Отсутствует.

3.2 Показ графиков

3.2.1 HQt_DrawLine

Функция возвращает строку с HTML кодом отрисовки линии по функции Function. Для добавление в html файл.

Код 3. Синтаксис

```
QString HQt_DrawLine (double Left, double Right, double h, double (*Function)(double)  
    , QString TitleChart, QString NameVectorX, QString NameVectorY, QString NameLine,  
    bool ShowLine, bool ShowPoints, bool ShowArea, bool ShowSpecPoints, bool RedLine);  
QString HQt_DrawLine (double Left, double Right, double h, double (*Function)(double)  
    , QString TitleChart, QString NameVectorX, QString NameVectorY, bool ShowLine,  
    bool ShowPoints, bool ShowArea, bool ShowSpecPoints, bool RedLine);
```

```
QString HQt_DrawLine (double Left, double Right, double h, double (*Function)(double)
, QString TitleChart, QString NameVectorX, QString NameVectorY, QString NameLine);
QString HQt_DrawLine (double Left, double Right, double h, double (*Function)(double)
);
```

Входные параметры:

Left — левая граница области;

Right — правая граница области;

h — шаг, с которым надо рисовать график;

Function — указатель на вычисляемую функцию;

TitleChart — заголовок графика;

NameVectorX — название оси Ox;

NameVectorY — название оси Oy;

NameLine — название первого графика (для легенды);

ShowLine — показывать ли линию;

ShowPoints — показывать ли точки;

ShowArea — показывать ли закрашенную область под кривой;

ShowSpecPoints — показывать ли специальные точки;

RedLine — рисовать ли красную линию, или синюю.

Возвращаемое значение:

Строка с HTML кодом.

3.2.2 THQt_ShowChartOfLine

Функция возвращает строку с выводом некоторого графика по точкам с HTML кодами. Для добавление в html файл.

Код 4. Синтаксис

```
template <class T> QString THQt_ShowChartOfLine (T *VMHL_VectorX, T *VMHL_VectorY, int
VMHL_N, QString TitleChart, QString NameVectorX, QString NameVectorY, QString
NameLine, bool ShowLine, bool ShowPoints, bool ShowArea, bool ShowSpecPoints, bool
RedLine);
template <class T> QString THQt_ShowChartOfLine (T *VMHL_VectorX, T *VMHL_VectorY, int
VMHL_N, QString TitleChart, QString NameVectorX, QString NameVectorY, bool
ShowLine, bool ShowPoints, bool ShowArea, bool ShowSpecPoints, bool RedLine);
template <class T> QString THQt_ShowChartOfLine (T *VMHL_VectorX, T *VMHL_VectorY, int
VMHL_N, QString TitleChart, QString NameVectorX, QString NameVectorY, QString
NameLine);
template <class T> QString THQt_ShowChartOfLine (T *VMHL_VectorX, T *VMHL_VectorY, int
VMHL_N);
```

Входные параметры:

VMHL_VectorX — указатель на вектор координат X точек;

VMHL_VectorY — указатель на вектор координат Y точек;

VMHL_N — количество точек;

TitleChart — заголовок графика;

NameVectorX — название оси Ox;

NameVectorY — название оси Oy;

NameLine — название первого графика (для легенды);

ShowLine — показывать ли линию;

ShowPoints — показывать ли точки;

ShowArea — показывать ли закрашенную область под кривой;

ShowSpecPoints — показывать ли специальные точки;

RedLine — рисовать ли красную линию, или синюю.

Возвращаемое значение:

Строка с HTML кодами с выводимым графиком.

Примечание:

Используются случайные числа, так что рекомендуется вызвать в программе инициализатор случайных чисел `qrand`. Рекомендую так: `qrand(QDateTime::currentMSecsSinceEpoch ())`

Требуется наличие в папке с html файлом файлы `jsxgraph.css` и `jsxgraphcore.js` из библиотеки JSXGraph.

Код 5. Пример использования

```
QString DS=QDir::separator();
QString path=QGuiApplication::applicationDirPath()+DS;//путь к папке

QString Html;
Html=HQt_BeginHtml ();

int N=6;
double *dataX=new double [N];
double *dataY=new double [N];
dataX[0]=7;dataY[0]=6;
dataX[1]=8;dataY[1]=4;
dataX[2]=10;dataY[2]=7;
dataX[3]=5;dataY[3]=12;
dataX[4]=14;dataY[4]=4;
dataX[5]=13;dataY[5]=8;

Html += THQt_ShowChartOfLine (dataX,dataY,N,"Тестовый график","x","y","линия",true,
    true,true,true,false);

delete []dataX;
delete []dataY;

Html+=HQt_EndHtml();
HQt_SaveFile(Html, path+"temp.html");
```



```
ui->webView->setUrl(QUrl::fromLocalFile(path+"temp.html"));
```

3.2.3 THQt_ShowChartsOfLineFromMatrix

Функция возвращает строку с выводом графиков из матрицы по точкам с HTML кодами. Для добавление в html файл.

Код 6. Синтаксис

```
template <class T> QString THQt_ShowChartsOfLineFromMatrix (T **VMHL_MatrixXY, int
    VMHL_N, int VMHL_M, QString TitleChart, QString NameVectorX, QString NameVectorY,
    QString *NameLine, bool ShowLine, bool ShowPoints, bool ShowArea, bool ShowSpecPoints
);
template <class T> QString THQt_ShowChartsOfLineFromMatrix (T **VMHL_MatrixXY, int
    VMHL_N, int VMHL_M, QString TitleChart, QString NameVectorX, QString NameVectorY,
    bool ShowLine, bool ShowPoints, bool ShowArea, bool ShowSpecPoints);
template <class T> QString THQt_ShowChartsOfLineFromMatrix (T **VMHL_MatrixXY, int
    VMHL_N, int VMHL_M, QString TitleChart, QString NameVectorX, QString NameVectorY,
    QString *NameLine);
template <class T> QString THQt_ShowChartsOfLineFromMatrix (T **VMHL_MatrixXY, int
    VMHL_N, int VMHL_M);
```

Входные параметры:

VMHL_MatrixXY — указатель на матрицу значений X и Y графиков;

VMHL_N — количество точек;

VMHL_M — количество столбцов матрицы (1+количество графиков);

TitleChart — заголовок графика;

NameVectorX — название оси Ox;

NameVectorY — название оси Oy;

NameLine — указатель на вектор названий графиков (для легенды) количество элементов VMHL_M—1 (так как первый столбец — это X значения);

ShowLine — показывать ли линию;

ShowPoints — показывать ли точки;

ShowArea — показывать ли закрашенную область под кривой;

ShowSpecPoints — показывать ли специальные точки.

Возвращаемое значение:

Строка с HTML кодами с выводимым графиком.

Примечание:

Используются случайные числа, так что рекомендуется вызвать в программе инициализатор случайных чисел qsrnd. Рекомендую так: qsrnd(QDateTime::currentMSecsSinceEpoch ())

Требуется наличия в папке с html файлом файлы jsxgraph.css и jsxgraphcore.js из библиотеки JSXGraph.

Код 7. Пример использования

```
QString DS=QDir::separator();
QString path=QGuiApplication::applicationDirPath()+DS;//путь к папке

QString Html;
Html=HQt_BeginHtml ();

int N=6;
int M=2;
double **data;
data=new double*[N];
for (int i=0;i<N;i++) X[i]=new double[M];
data[0][0]=7;data[0][1]=6;
data[1][0]=8;data[1][1]=4;
data[2][0]=10;data[2][1]=7;
data[3][0]=5;data[3][1]=12;
data[4][0]=14;data[4][1]=4;
data[5][0]=13;data[5][1]=8;

QString *NameLine=new QString[M-1];
NameLine[0]="Первая линия";

Html += THQt_ShowChartsOfLineFromMatrix (data,N,M, "График", "x", "y",NameLine,true,
true,true,true);

for (int i=0;i<N;i++) delete [] data[i];
delete [] data;
delete [] NameLine;

Html+=HQt_EndHtml();
HQt_SaveFile(Html, path+"temp.html");
ui->webView->setUrl(QUrl::fromLocalFile(path+"temp.html"));
```

3.2.4 THQt_ShowIndependentChartsOfLineFromMatrix

Функция возвращает строку с выводом графиков из матрицы по точкам с HTML кодами. Для добавление в html файл.

Код 8. Синтаксис

```
template <class T> QString THQt_ShowIndependentChartsOfLineFromMatrix (T **
VMHL_MatrixXY,int *VMHL_N_EveryCol,int VMHL_M, QString TitleChart, QString
NameVectorX, QString NameVectorY,QString *NameLine, bool ShowLine,bool ShowPoints,
bool ShowArea,bool ShowSpecPoints);
template <class T> QString THQt_ShowIndependentChartsOfLineFromMatrix (T **
VMHL_MatrixXY,int *VMHL_N_EveryCol,int VMHL_M, QString TitleChart, QString
NameVectorX, QString NameVectorY,bool ShowLine,bool ShowPoints,bool ShowArea,bool
ShowSpecPoints);
template <class T> QString THQt_ShowIndependentChartsOfLineFromMatrix (T **
VMHL_MatrixXY,int *VMHL_N_EveryCol,int VMHL_M, QString TitleChart, QString
NameVectorX, QString NameVectorY,QString *NameLine);
template <class T> QString THQt_ShowIndependentChartsOfLineFromMatrix (T **
VMHL_MatrixXY,int *VMHL_N_EveryCol,int VMHL_M);
```

Входные параметры:

VMHL_MatrixXY — указатель на матрицу значений X и N графиков;

VMHL_N_EveryCol — количество элементов в каждом столбце (так как столбцы идут по парам, то число элементов в нечетном и

следующем за ним четном столбце должны совпадать, например 10,10,5,5,7,7);

VMHL_M — количество столбцов матрицы (должно быть четным числом конечно);

TitleChart — заголовок графика;

NameVectorX — название оси Oх;

NameVectorY — название оси Oу;

NameLine — указатель на вектор названий графиков (для легенды) количество элементов VMHL_M/2;

ShowLine — показывать ли линию;

ShowPoints — показывать ли точки;

ShowArea — показывать ли закрашенную область под кривой;

ShowSpecPoints — показывать ли специальные точки.

Возвращаемое значение:

Строка с HTML кодами с выводимым графиком.

Примечание:

Используются случайные числа, так что рекомендуется вызвать в программе инициализатор случайных чисел qrand. Рекомендую так: qrand(QDateTime::currentMSecsSinceEpoch ())

Требует наличия в папке с html файлом файлы jsxgraph.css и jsxgraphcore.js из библиотеки JSXGraph.

Нечетные столбцы - это значения координат X графиков. Следующие за ними четные столбцы - соответствующие значения Y. То есть графики друг от друга независимы.

3.2.5 THQt_ShowTwoChartsOfLine

Функция возвращает строку с выводом некоторого двух графиков по точкам с HTML кодами. Для добавление в html файл. У обоих графиков одинаковый массив значений X.

Код 9. Синтаксис

```
template <class T> QString THQt_ShowTwoChartsOfLine (T *VMHL_VectorX, T *VMHL_VectorY1, T *VMHL_VectorY2, int VMHL_N, QString TitleChart, QString NameVectorX, QString NameVectorY, QString NameLine1, QString NameLine2, bool ShowLine, bool ShowPoints, bool ShowArea, bool ShowSpecPoints);
template <class T> QString THQt_ShowTwoChartsOfLine (T *VMHL_VectorX, T *VMHL_VectorY1, T *VMHL_VectorY2, int VMHL_N, QString TitleChart, QString NameVectorX, QString NameVectorY, bool ShowLine, bool ShowPoints, bool ShowArea, bool ShowSpecPoints);
template <class T> QString THQt_ShowTwoChartsOfLine (T *VMHL_VectorX, T *VMHL_VectorY1, T *VMHL_VectorY2, int VMHL_N, QString TitleChart, QString NameVectorX, QString NameVectorY, QString NameLine1, QString NameLine2);
template <class T> QString THQt_ShowTwoChartsOfLine (T *VMHL_VectorX, T *VMHL_VectorY1, T *VMHL_VectorY2, int VMHL_N);
```

Входные параметры:

VMHL_VectorX — указатель на вектор координат X точек;

VMHL_VectorY1 — указатель на вектор координат Y точек первой линии;

VMHL_VectorY2 — указатель на вектор координат Y точек второй линии;

VMHL_N — количество точек;

TitleChart — заголовок графика;

NameVectorX — название оси Ox;

NameVectorY — название оси Oy;

NameLine1 — название первого графика (для легенды);

NameLine2 — название второго графика (для легенды);

ShowLine — показывать ли линию;

ShowPoints — показывать ли точки;

ShowArea — показывать ли закрашенную область под кривой;

ShowSpecPoints — показывать ли специальные точки.

Возвращаемое значение:

Строка с HTML кодами с выводимым графиком.

Примечание:

Используются случайные числа, так что рекомендуется вызвать в программе инициализатор случайных чисел `qrand`. Рекомендую так: `qrand(QDateTime::currentMSecsSinceEpoch ())`

Требуется наличие в папке с html файлом файлы `jsxgraph.css` и `jsxgraphcore.js` из библиотеки JSXGraph.

Код 10. Пример использования

```
QString DS=QDir::separator();
QString path=QGuiApplication::applicationDirPath()+DS;//путь к папке

QString Html;
Html=HQt_BeginHtml ();

int N=6;
double *dataX=new double [N];
double *dataY1=new double [N];
double *dataY2=new double [N];
dataX[0]=7;dataY1[0]=6;dataY2[0]=1;
dataX[1]=8;dataY1[1]=4;dataY2[0]=2;
dataX[2]=10;dataY1[2]=7;dataY2[0]=3;
dataX[3]=5;dataY1[3]=12;dataY2[0]=4;
dataX[4]=14;dataY1[4]=4;dataY2[0]=4;
dataX[5]=13;dataY1[5]=8;dataY2[0]=3;

Html += THQt_ShowTwoChartsOfLine (dataX,dataY1,dataY2,N,"Тестовый график","x","y","количество деревьев","количество домов",true,true,true,true);
```

```

delete []dataX;
delete []dataY1;
delete []dataY2;

Html+=HQt_EndHtml();
HQt_SaveFile(Html, path+"temp.html");
ui->webView->setUrl(QUrl::fromLocalFile(path+"temp.html"));

```

3.2.6 THQt_ShowTwoIndependentChartsOfLine

Функция возвращает строку с выводом некоторого двух независимых графиков по точкам с HTML кодами. Для добавление в html файл.

Код 11. Синтаксис

```

template <class T> QString THQt_ShowTwoIndependentChartsOfLine (T *VMHL_VectorX1,T *
VMHL_VectorY1,int VMHL_N1,T *VMHL_VectorX2,T *VMHL_VectorY2, int VMHL_N2, QString
TitleChart, QString NameVectorX, QString NameVectorY,QString NameLine1, QString
NameLine2,bool ShowLine,bool ShowPoints,bool ShowArea,bool ShowSpecPoints);
template <class T> QString THQt_ShowTwoIndependentChartsOfLine (T *VMHL_VectorX1,T *
VMHL_VectorY1,int VMHL_N1,T *VMHL_VectorX2,T *VMHL_VectorY2, int VMHL_N2, QString
TitleChart, QString NameVectorX, QString NameVectorY,bool ShowLine,bool ShowPoints
,bool ShowArea,bool ShowSpecPoints);
template <class T> QString THQt_ShowTwoIndependentChartsOfLine (T *VMHL_VectorX1,T *
VMHL_VectorY1,int VMHL_N1,T *VMHL_VectorX2,T *VMHL_VectorY2, int VMHL_N2, QString
TitleChart, QString NameVectorX, QString NameVectorY,QString NameLine1, QString
NameLine2);
template <class T> QString THQt_ShowTwoIndependentChartsOfLine (T *VMHL_VectorX1,T *
VMHL_VectorY1,int VMHL_N1,T *VMHL_VectorX2,T *VMHL_VectorY2, int VMHL_N2);

```

Входные параметры:

VMHL_VectorX1 — указатель на вектор координат X точек первой линии;

VMHL_VectorY1 — указатель на вектор координат Y точек первой линии;

VMHL_N1 — количество точек первой линии;

VMHL_VectorX2 — указатель на вектор координат X точек второй линии;

VMHL_VectorY2 — указатель на вектор координат Y точек второй линии;

VMHL_N2 — количество точек второй линии;

TitleChart — заголовок графика;

NameVectorX — название оси Ox;

NameVectorY — название оси Oy;

NameLine1 — название первого графика (для легенды);

NameLine2 — название второго графика (для легенды);

ShowLine — показывать ли линию;

ShowPoints — показывать ли точки;

ShowArea — показывать ли закрашенную область под кривой;

ShowSpecPoints — показывать ли специальные точки.

Возвращаемое значение:

Строка с HTML кодами с выводимым графиком.

Примечание:

Используются случайные числа, так что рекомендуется вызвать в программе инициализатор случайных чисел qrand. Рекомендую так: qrand(QDateTime::currentMSecsSinceEpoch ())

Требуется наличия в папке с html файлом файлы jsxgraph.css и jsxgraphcore.js из библиотеки JSXGraph.

Код 12. Пример использования

```
QString DS=QDir::separator();
QString path=QGuiApplication::applicationDirPath()+DS;//путь к папке

QString Html;
Html=HQt_BeginHtml ();

int N1=6;
double *dataX1=new double [N1];
double *dataY1=new double [N1];
dataX1[0]=7;dataY1[0]=6;
dataX1[1]=8;dataY1[1]=4;
dataX1[2]=10;dataY1[2]=7;
dataX1[3]=5;dataY1[3]=12;
dataX1[4]=14;dataY1[4]=4;
dataX1[5]=13;dataY1[5]=8;

int N2=3;
double *dataX2=new double [N1];
double *dataY2=new double [N2];
dataX2[0]=1;dataY2[0]=8;
dataX2[1]=2;dataY2[1]=4;
dataX2[2]=3;dataY2[2]=5;

Html += THQt_ShowTwoIndependentChartsOfLine (dataX1,dataY1,N1,dataX2,dataY2,N2,"Тестовый график", "x", "y", "количество деревьев", "количество домов", true,true,true,true);

delete []dataX1;
delete []dataY1;
delete []dataX2;
delete []dataY2;

Html+=HQt_EndHtml();
HQt_SaveFile(Html, path+"temp.html");
ui->webView->setUrl(QUrl::fromLocalFile(path+"temp.html"));
```

3.2.7 THQt_ShowTwoIndependentChartsOfPointsAndLine

Функция возвращает строку с выводом некоторого двух независимых графиков по точкам с HTML кодами. Для добавления в html файл. Один график выводится в виде точек, а второй в виде линии. Удобно для отображения регрессий. У обоих графиков разные массивы значений X и Y.

Код 13. Синтаксис

```
template <class T> QString THQt_ShowTwoIndependentChartsOfPointsAndLine (T *
    VMHL_VectorX1,T *VMHL_VectorY1,int VMHL_N1,T *VMHL_VectorX2,T *VMHL_VectorY2, int
    VMHL_N2, QString TitleChart, QString NameVectorX, QString NameVectorY,QString
    NameLine1, QString NameLine2,bool ShowLine,bool ShowPoints,bool ShowArea,bool
    ShowSpecPoints);
template <class T> QString THQt_ShowTwoIndependentChartsOfPointsAndLine (T *
    VMHL_VectorX1,T *VMHL_VectorY1,int VMHL_N1,T *VMHL_VectorX2,T *VMHL_VectorY2, int
    VMHL_N2, QString TitleChart, QString NameVectorX, QString NameVectorY,bool
    ShowLine,bool ShowPoints,bool ShowArea,bool ShowSpecPoints);
template <class T> QString THQt_ShowTwoIndependentChartsOfPointsAndLine (T *
    VMHL_VectorX1,T *VMHL_VectorY1,int VMHL_N1,T *VMHL_VectorX2,T *VMHL_VectorY2, int
    VMHL_N2, QString TitleChart, QString NameVectorX, QString NameVectorY,QString
    NameLine1, QString NameLine2);
template <class T> QString THQt_ShowTwoIndependentChartsOfPointsAndLine (T *
    VMHL_VectorX1,T *VMHL_VectorY1,int VMHL_N1,T *VMHL_VectorX2,T *VMHL_VectorY2, int
    VMHL_N2);
```

Входные параметры:

VMHL_VectorX1 — указатель на вектор координат X точек первой линии;

VMHL_VectorY1 — указатель на вектор координат Y точек первой линии;

VMHL_N1 — количество точек первой линии;

VMHL_VectorX2 — указатель на вектор координат X точек второй линии;

VMHL_VectorY2 — указатель на вектор координат Y точек второй линии;

VMHL_N2 — количество точек второй линии;

TitleChart — заголовок графика;

NameVectorX — название оси Ox;

NameVectorY — название оси Oy;

NameLine1 — название первого графика (для легенды);

NameLine2 — название второго графика (для легенды);

ShowLine — показывать ли линию;

ShowPoints — показывать ли точки;

ShowArea — показывать ли закрашенную область под кривой;

ShowSpecPoints — показывать ли специальные точки.

Возвращаемое значение:

Строка с HTML кодами с выводимым графиком.

Примечание:

Используются случайные числа, так что рекомендуется вызвать в программе инициализатор случайных чисел `qrand`. Рекомендую так: `qrand(QDateTime::currentMSecsSinceEpoch ())`

Требует наличия в папке с html файлом файлы `jsxgraph.css` и `jsxgraphcore.js` из библиотеки `JSXGraph`.

3.3 Показ математических выражений

3.3.1 THQt_ShowMatrix

Функция возвращает строку с выводом некоторой матрицы VMHL_Matrix с HTML кодами. Для добавление в html файл.

Код 14. Синтаксис

```
template <class T> QString THQt_ShowMatrix (T *VMHL_Matrix, int VMHL_N, int VMHL_M,
    QString TitleMatrix, QString NameMatrix);
template <class T> QString THQt_ShowMatrix (T *VMHL_Matrix, int VMHL_N, int VMHL_M,
    QString NameMatrix);
template <class T> QString THQt_ShowMatrix (T *VMHL_Matrix, int VMHL_N, int VMHL_M);
```

Входные параметры:

VMHL_Matrix — указатель на выводимую матрицу;

VMHL_N — количество строк в матрице;

VMHL_M — количество столбцов в матрице;

TitleMatrix — заголовок выводимой матрицы;

NameMatrix — обозначение матрицы.

Возвращаемое значение:

Строка с HTML кодами с выводимой матрицей.

3.3.2 THQt_ShowMatrix2

Функция возвращает строку с выводом некоторой матрицы VMHL_Matrix с HTML кодами. Для добавление в html файл. В качестве матрицы выступает массив QStringList, где количество QStringList - это количество строк. Каждый QStringList — это одна строка.

Код 15. Синтаксис

```
QString THQt_ShowMatrix (QStringList *VMHL_Matrix, int VMHL_N, QString TitleMatrix,
    QString NameMatrix);
QString THQt_ShowMatrix (QStringList *VMHL_Matrix, int VMHL_N, QString NameMatrix);
QString THQt_ShowMatrix (QStringList *VMHL_Matrix, int VMHL_N);
```

Входные параметры:

VMHL_Matrix — указатель на выводимую матрицу;

VMHL_N — количество строк в матрице;

TitleMatrix — заголовок выводимой матрицы;

NameMatrix — обозначение матрицы.

Возвращаемое значение:

Строка с HTML кодами с выводимой матрицей.

3.3.3 THQt_ShowNumber

Функция возвращает строку с выводом некоторого числа VMHL_X с HTML кодами. Для добавление в html файл.

Код 16. Синтаксис

```
template <class T> QString THQt_ShowNumber (T VMHL_X, QString TitleX, QString NameX);  
template <class T> QString THQt_ShowNumber (T VMHL_X, QString NameX);  
template <class T> QString THQt_ShowNumber (T VMHL_X);
```

Входные параметры:

VMHL_X — выводимое число;

TitleX — заголовок выводимого числа;

NameX — обозначение числа.

Возвращаемое значение:

Строка с HTML кодами с выводимым числом.

3.3.4 THQt_ShowVector

Функция возвращает строку с выводом некоторого вектора VMHL_Vector с HTML кодами. Для добавление в html файл.

Код 17. Синтаксис

```
template <class T> QString THQt_ShowVector (T *VMHL_Vector, int VMHL_N, QString  
    TitleVector, QString NameVector);  
template <class T> QString THQt_ShowVector (T *VMHL_Vector, int VMHL_N, QString  
    NameVector);  
template <class T> QString THQt_ShowVector (T *VMHL_Vector, int VMHL_N);
```

Входные параметры:

VMHL_Vector — указатель на выводимый вектор;

VMHL_N — количество элементов вектора;

TitleVector — заголовок выводимого вектора;

NameVector — обозначение вектора.

Возвращаемое значение:

Строка с HTML кодами с выводимым вектором.

3.3.5 THQt_ShowVector2

Функция возвращает строку с выводом некоторого списка строк VMHL_Vector с HTML кодами. Для добавление в html файл.

Код 18. Синтаксис

```
QString THQt_ShowVector (QStringList VMHL_Vector, QString TitleVector, QString
    NameVector);
QString THQt_ShowVector (QStringList VMHL_Vector, QString NameVector);
QString THQt_ShowVector (QStringList VMHL_Vector);
```

Входные параметры:

VMHL_Vector — указатель на список строк QStringList;

VMHL_N — количество элементов вектора;

TitleVector — заголовок выводимого вектора;

NameVector — обозначение вектора.

Возвращаемое значение:

Строка с HTML кодами с выводимым вектором.

3.3.6 THQt_ShowVectorT

Функция возвращает строку с выводом некоторого вектора VMHL_Vector в транспонированном виде с HTML кодами. Для добавление в html файл.

Код 19. Синтаксис

```
template <class T> QString THQt_ShowVectorT (T *VMHL_Vector, int VMHL_N, QString
    TitleVector, QString NameVector);
template <class T> QString THQt_ShowVectorT (T *VMHL_Vector, int VMHL_N, QString
    NameVector);
template <class T> QString THQt_ShowVectorT (T *VMHL_Vector, int VMHL_N);
```

Входные параметры:

VMHL_Vector — указатель на выводимый вектор;

VMHL_N — количество элементов вектора;

TitleVector — заголовок выводимого вектора;

NameVector — обозначение вектора.

Возвращаемое значение:

Строка с HTML кодами с выводимым вектором.

3.4 Показ текста

3.4.1 HQt_ShowAlert

Функция возвращает строку с выводом некоторого предупреждения. Для добавление в html файл.

Код 20. Синтаксис

```
QString HQt_ShowAlert (QString String);
```

Входные параметры:

String — непосредственно выводимая строка.

Возвращаемое значение:

Строка с HTML кодами с выводимым предупреждением.

3.4.2 HQt_ShowH1

Функция возвращает строку с выводом некоторой строки в виде заголовка. Для добавление в html файл.

Код 21. Синтаксис

```
QString HQt_ShowH1 (QString String);
```

Входные параметры:

String — непосредственно выводимая строка.

Возвращаемое значение:

Строка с HTML кодами с выводимой строкой.

3.4.3 HQt_ShowHr

Функция возвращает строку с выводом горизонтальной линии. Для добавление в html файл.

Код 22. Синтаксис

```
QString HQt_ShowHr ();
```

Входные параметры:

Отсутствуют.

Возвращаемое значение:

Строка с HTML кодами с выводимой строкой.

3.4.4 HQt_ShowSimpleText

Функция возвращает строку с выводом некоторой строки с HTML кодами без всякого излишества. Для добавление в html файл.

Код 23. Синтаксис

```
QString HQt_ShowSimpleText (QString String);
```

Входные параметры:

String — непосредственно выводимая строка.

Возвращаемое значение:

Строка с HTML кодами с выводимой строкой.

3.4.5 HQt_ShowText

Функция возвращает строку с выводом некоторой строки с HTML кодами. Для добавление в html файл.

Код 24. Синтаксис

```
QString HQt_ShowText (QString TitleX);
```

Входные параметры:

TitleX — непосредственно выводимая строка.

Возвращаемое значение:

Строка с HTML кодами с выводимой строкой.

3.4.6 THQt_NumberToText

Функция выводит число VMHL_X в строку.

Код 25. Синтаксис

```
template <class T> QString THQt_NumberToText (T VMHL_X);
```

Входные параметры:

VMHL_X — выводимое число.

Возвращаемое значение:

Строка, в которой записано число.