

СЕМИНАР 9

Работа с графиками в Qt Charts

Добавление модуля QtCharts

- 1. Проверить наличие Maintenance Tool на вашей Ubuntu:
- `$locate -i maintenance | grep qt`

или

`$ls /opt/Qt | grep -i maintenance`

- 2. Если по результатам команды, вы нашли Maintenance Tool, запустите его:

`$sudo "путь к Maintenance Tool"`

```
amakashov@ROS-VirtualBox:~/Downloads$ /opt/Qt/MaintenanceTool
QApplication: invalid style override passed, ignoring it.
[634] Warning: Could not find Qt Account settings from the disk
```

- 3. Если Maintenance Tool отсутствует, то установить модули qt5-default и qtdeclarative5-dev:

`$sudo apt-get install qt5-default qtdeclarative5-dev`

Добавление модуля QtCharts

- 3. Запустить online-установщик Qt (который лежит в папке с материалами к этому семинару):
 - 3.1. Сделаем файл исполняемым:
\$ chmod +x qt-unified-linux-x64-3.0.1-online.run
 - 3.2 Запускаем установщик:
\$ sudo ./qt-unified-linux-x64-3.0.1-online.run

Добавление модуля QtCharts

Maintain Qt

Qt Account - Your unified login to everything Qt

Please log in to Qt Account

Qt

Login Email Password [Forgot password?](#)

Need a Qt Account?

Sign-up Valid email address Password Confirm Password

☐ I accept the [service terms](#).

☐ Uninstall only

Settings Skip Cancel

Maintain Qt

Setup - Qt

Qt

Welcome to open source Qt setup.

☒ Add or remove components
☐ Update components
☐ Remove all components

Settings < Back Next > Quit

Maintain Qt

Select Components

Select the components to install. Deselect installed components to uninstall them. Any components already installed will not be updated.

Component Name Inst. Latest Qt and related package preview snapshots

Preview

- ☐ Qt 5.10.0 RC
- ☐ Qt Creator 4.5.0-rc1
- ☒ Qt 5.9.3 1.0. 5.9. 5.9.
- ☒ Desktop gcc 64-bit 5.9.
- ☐ Android x86
- ☐ Android ARMv7
- ☐ Sources
- ☒ Qt Charts 5.9.
- ☐ Qt Data Visualization 5.9.
- ☐ Qt Purchasing
- ☐ Qt Virtual Keyboard 5.9.
- ☐ Qt WebEngine
- ☐ Qt Network Auth (TP)
- ☐ Qt Remote Objects (TP)
- ☐ Qt Speech (TP)
- ☐ Qt Script (Deprecated)
- ☐ Qt 5.9.2
- ☐ Qt 5.9.1
- ☐ Qt 5.9.0
- ☐ Qt 5.8
- ☐ Qt 5.7
- ☐ Qt 5.6.3

Reset Select All Deselect All

< Back Next > Cancel

Графики в Qt

Для отрисовки графиков в Qt можно использовать следующие модули:

- 1). Модуль QtCharts (ранее была доступна только для коммерческой лицензии);
- 2). Qwt;
- 3). etc

QtCharts

Основан на механизме модель-представление.

Основные классы:

- QtCharts – это элемент сцены (QGraphicWidget), который можно добавить на сцену (QGraphicsScene). Управляет графическим представлением различных типов «серий» данных и прочих объектов представления графиков, таких как легенда, оси и т.п.
- QtChartView – виджет для отображения графиков.
- Класс представления серий данных [QAbstractSeries](#). (Данные для графиков, диаграмм, точечных графиков и прочего)

QtCharts. Типы графиков

QtCharts может создавать различные типы графиков и диаграмм.

Тип представления задается в классах, наследниках от QAbstractSeries.

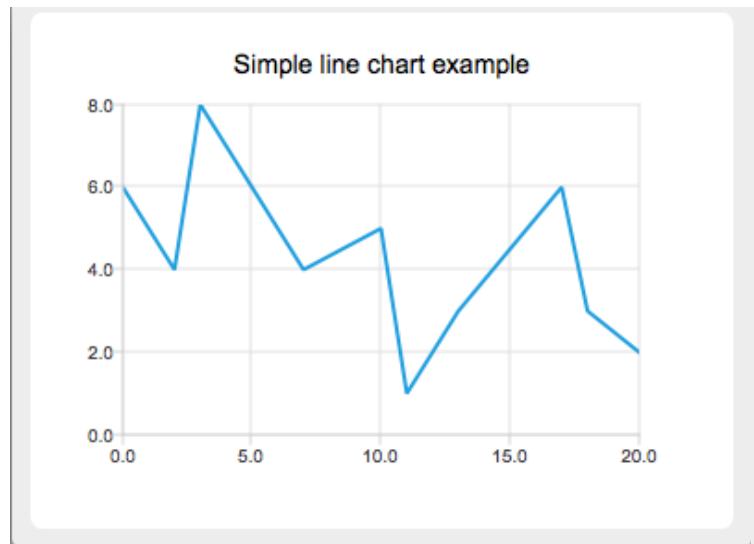
Классы наследники QAbstractSeries :

- QLineSeries;
- QSplineSeries;
- QAreaSeries;
- QScatterSeries;
- QPieSeries;
- QBarSeries;
- и т.п.

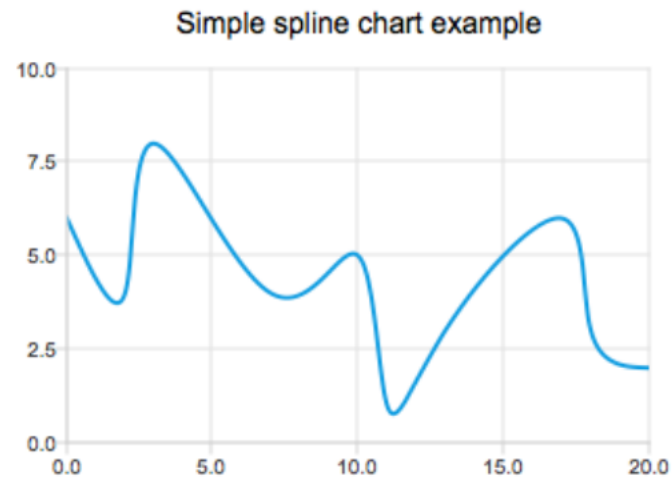
QtCharts. Типы графиков

QtCharts может создавать различные типы графиков и диаграмм.
Тип представления задается в классах, наследниках от `QAbstractSeries`.

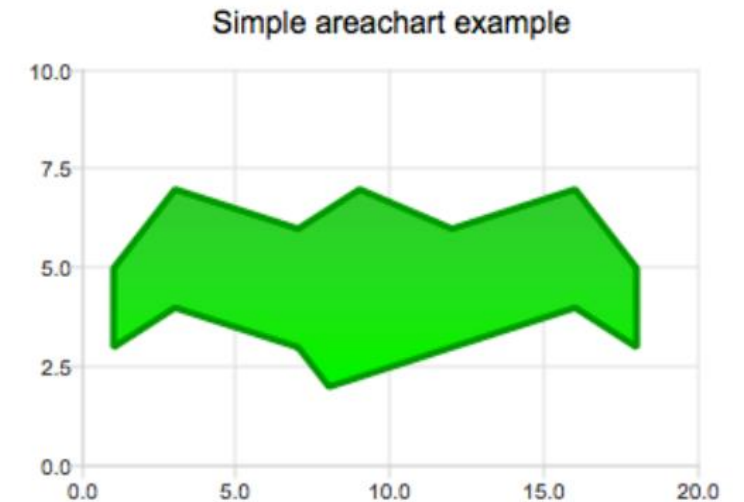
QLineCharts



QSplineCharts



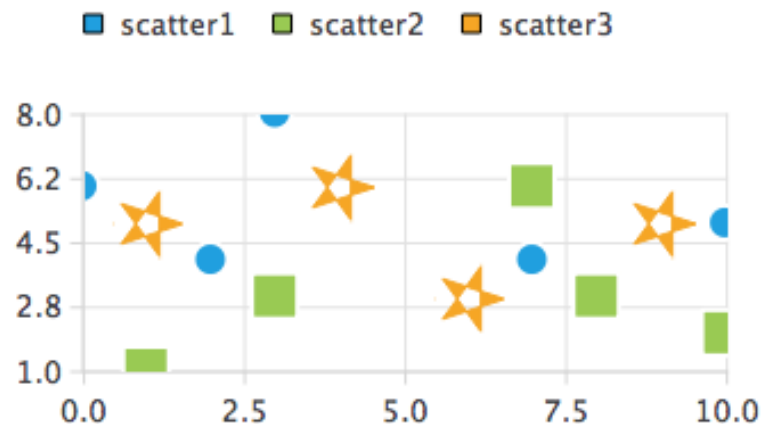
QAreaCharts



QtCharts. Типы графиков

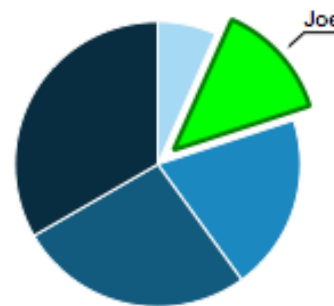
QScatterCharts

Simple scatterchart example



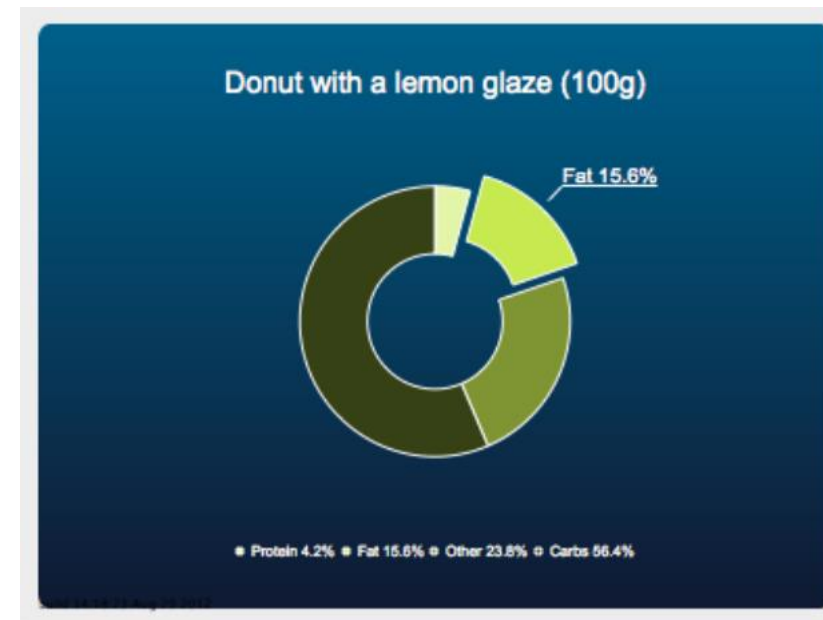
QScatterSeries

Simple piechart example



QPieSeries

QPierCharts

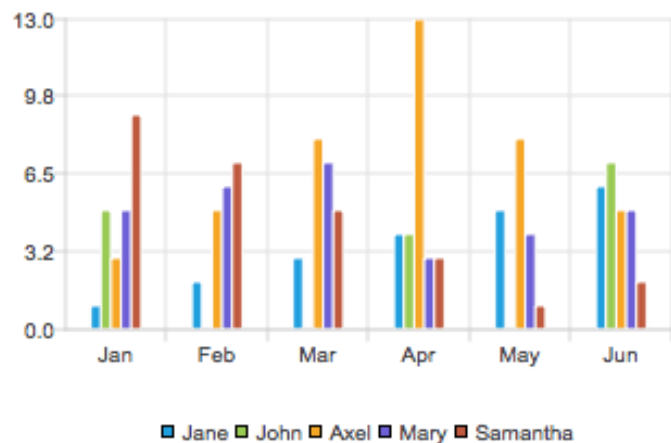


QPieSeries

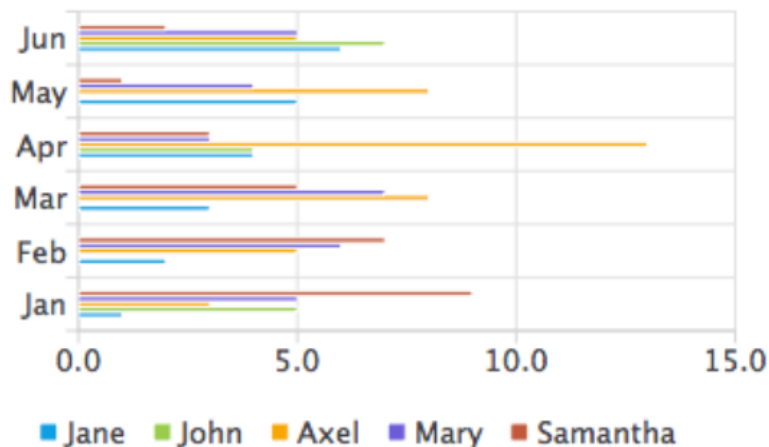
QtCharts. Типы графиков

QBarCharts

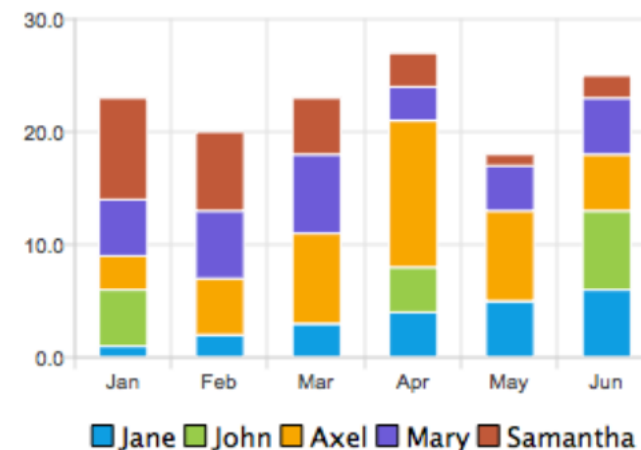
Simple barchart example



Simple horizontal barchart example



Simple stackedbarchart example



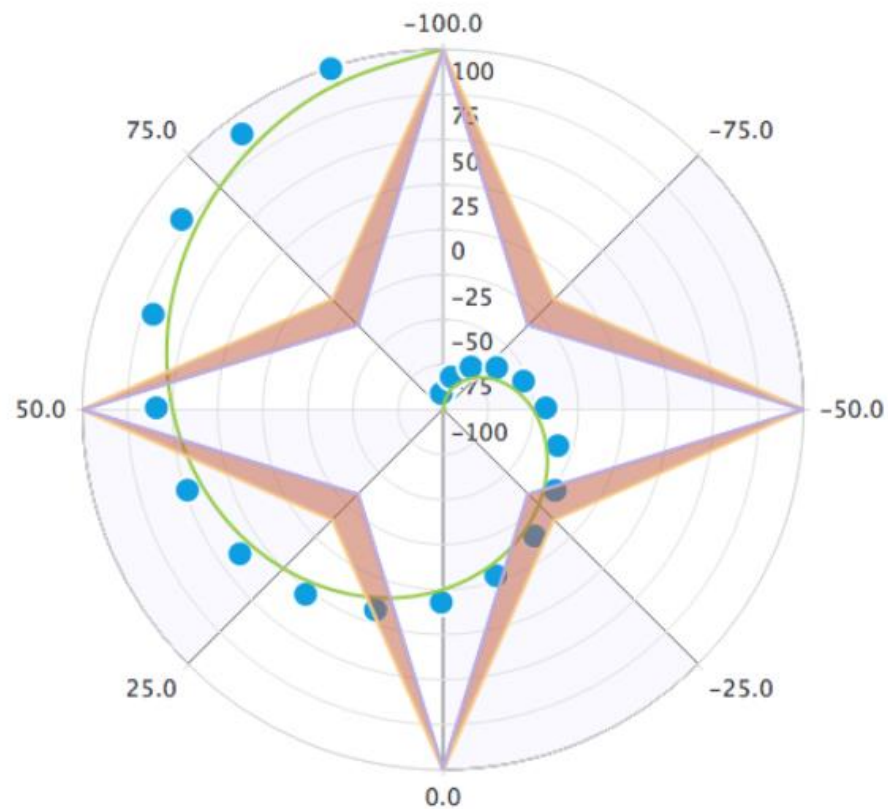
QBarSeries

QtCharts. Типы графиков

QPolarChart

Use arrow keys to scroll, +/- to zoom, and space to switch chart type.

■ scatter ■ spline ■ star outer ■ star inner ■ star area



При создании графиков в полярных координатах вместо класса Qchart создается класс QPolarChart

QtCharts. Типы осей.

Типы шкал графиков:

- Числовые значения (QValueAxis);
- Категории (QCategoryAxis);
- Временные (время/дата) (QDateTimeAxis);
- Логарифмические (QLogValueAxis).

QtCharts. Графики – это просто!

За вас в классе QtCharts уже реализовано множество полезных функций для работы с графиками:

Например:

Метод	Описание метода
QChart::zoomIn()	Масштабирование графика (по умолчанию увеличивает в 2 раза)
QChart::zoomOut()	Масштабирование графика (по умолчанию уменьшает в 2 раза)
QChart::zoomReset	Возвращает изображение к масштабу, который у него был до применения какого либо масштабирования.
QChart::scroll(dx,dy)	Перемещает отображаемую часть графика на переданные функции параметры dx и dy.

QtCharts. Начало работы.

1). Подключить модуль charts в файле описания вашего проекта *.pro:

```
Qt+=core gui widgets charts
```

2). Подключить библиотеку в заголовочный файл .h:

```
#include <QtCharts>
```

```
using namespace QtCharts; //открыть доступ к пространству имен  
QtCharts
```

QtCharts

Для того, чтобы отобразить наши данные на графике необходимо:

- 1) Создать нужные вам объекты представления данных `QLineSeries`, `QSplineSeries` и т.п. в зависимости от типа диаграммы/графика, который вы хотите создать;
- 2) Создать оси и привязать к ним данные;
- 3). Добавить объект данных к объекту `QChart`;
- 4). Настроить отображение легенды, цвета, типы линий и прочее.
- 5) Добавить график `Qchart` на сцену для отображения.

1. Создание объектов представления данных

- Например, если мы хотим создать двумерный график, то создадим объект `QSplineSeries` :
- `QSplineSeries *splineSeries = new QSplineSeries ;`
- //если хотим добавить на карту ПА, то можем создать объект для графика разброса параметров `QScatterSeries`:
- `QScatterSeries *scatterSeries = new QScatterSeries();`
- //добавить данные (координаты точек) для отображения
- //можно используя метод `append()`
- `splineSeries >append(0, 6);`
- `splineSeries >append(2, 4);`
- //или оператор “<<”
- `splineSeries << QPointF(11, 1) << QPointF(13,3)<<QPointF(17,6);`
- Если хотим очистить значения в `scatterSeries`, то используем функцию `clear`:
- `scatterSeries->clear();`

2. Добавление объекта данных к объекту QChart

```
QChart *chart = new QChart();//создать объект QChart
```

```
    chart->addSeries(splineSeries); //добавить данные на график
```

```
    chart->addSeries(scatterSeries ) //можно добавлять другие данные,  
даже другого типа (если до этого создать объект scatterSeries класса  
QScatterSeries)
```

3. Создание осей графика и привязка к ним данных

- Для создания осей в Qt существует специальный класс `QValueAxis`:
- `QValueAxis *xAxis= new , QValueAxis() ;`
- `QValueAxis *yAxis= new QValueAxis ();`
- `//установка максимального и минимального значения по оси (например +/- 100)`
- `xAxis->setRange(-100,100);`
- `//установка количества отметок (штрихов) на оси (по умолчанию =5)`
- `xAxis->setTickCount(11);`
- `//так устанавливается надпись для оси графика:`
- `xAxis->setTitleText("X, м");`
- Привязка данных к горизонтальной оси:
- `chartView->chart()->setAxisX(xAxis, lineSeries);`
- `chartView->chart()->setAxisY(yAxis, lineSeries);`
- `chartView->chart()->setAxisX(xAxis,scatterSeries);`
- `chartView->chart()->setAxisY(yAxis,scatterSeries);`

5. Добавление графика на сцену для отображения

```
QChartView *chartView = new QChartView(chart);  
chartView->setRenderHint(QPainter::Antialiasing);
```

QtCharts. Практическая часть.

- В проекте Map добавить карту, отображающую координаты ПА с использованием QtCharts, а также добавить управление картой с кнопок.
- При этом:
- Widget.ui и класс Widget – форма и класс основного окна, на которую добавляется виджет класса MapForm. Класс MapForm и есть тот класс, в котором вам необходимо запрограммировать карту.
- На форме widget.ui вводятся значения скорости и курса ПА. По нажатию на кнопку «Старт», с определенной периодичностью, класс Widget высылает сигнал setXY(X,Y) с координатами ПА (эти координаты вам и необходимо отобразить на карте). У класса MapForm есть слот setXY(X,Y), который принимает сигналы с координатами X,Y с основной формы. Эти сигналы и слоты уже соединены, вам нужно добавить код, который будет выводить принятые координаты X,Y на карту.
- На основном виджете есть кнопки «+», «-» и т.п. по их нажатию график должен масштабироваться, сдвигаться вправо, вверх, и т.п. В классе MapForm есть функции, которые реализуют данные возможности. Ваша задача – соединить сигналы и слоты, а также дописать необходимый код в эти функции, чтобы реализовать основной функционал карты.