

ОБУЧЕНИЕ РАЗРАБОТКЕ НА QT ДЛЯ МОБИЛЬНЫХ УСТРОЙСТВ

СПО: ОТ ОБУЧЕНИЯ ДО РАЗРАБОТКИ, 16 ИЮНЯ 2021



О ДОКЛАДЧИКЕ

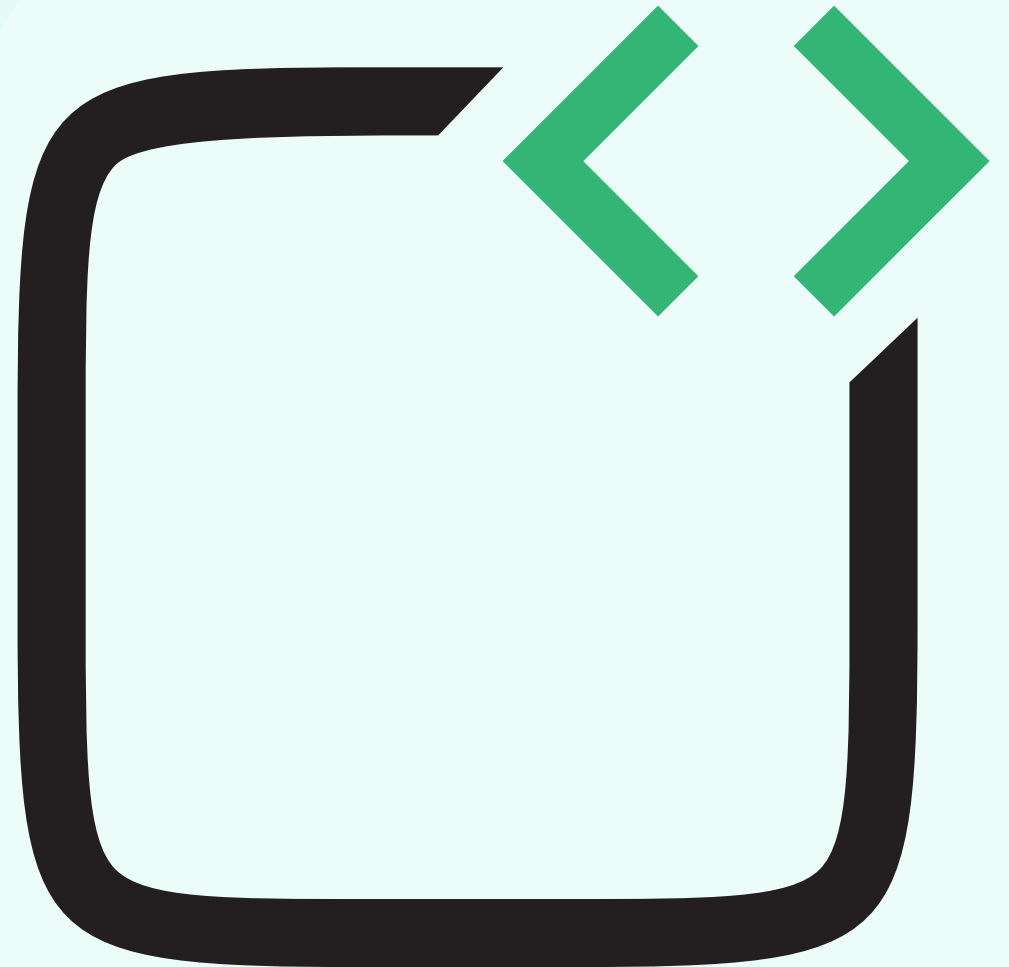
Кирилл Чувилин

- › Начальник отдела развития и поддержки разработчиков в компании «Открытая мобильная платформа»
- › к.т.н., доцент МФТИ
- › Более 10 лет опыта разработки на Qt
- › k.chuvilin@omp.ru



ПЛАН ПРЕЗЕНТАЦИИ

1. Мотивация
2. Фреймворк Qt
3. Важные технические аспекты
4. Кого чему и как обучаем



МОТИВАЦИЯ

1. ОС АВРОРА
2. ИСТОРИЯ РАЗВИТИЯ
3. АРХИТЕКТУРА
4. РАЗВИТИЕ И ПОДДЕРЖКА РАЗРАБОТЧИКОВ



ОС АВРОРА

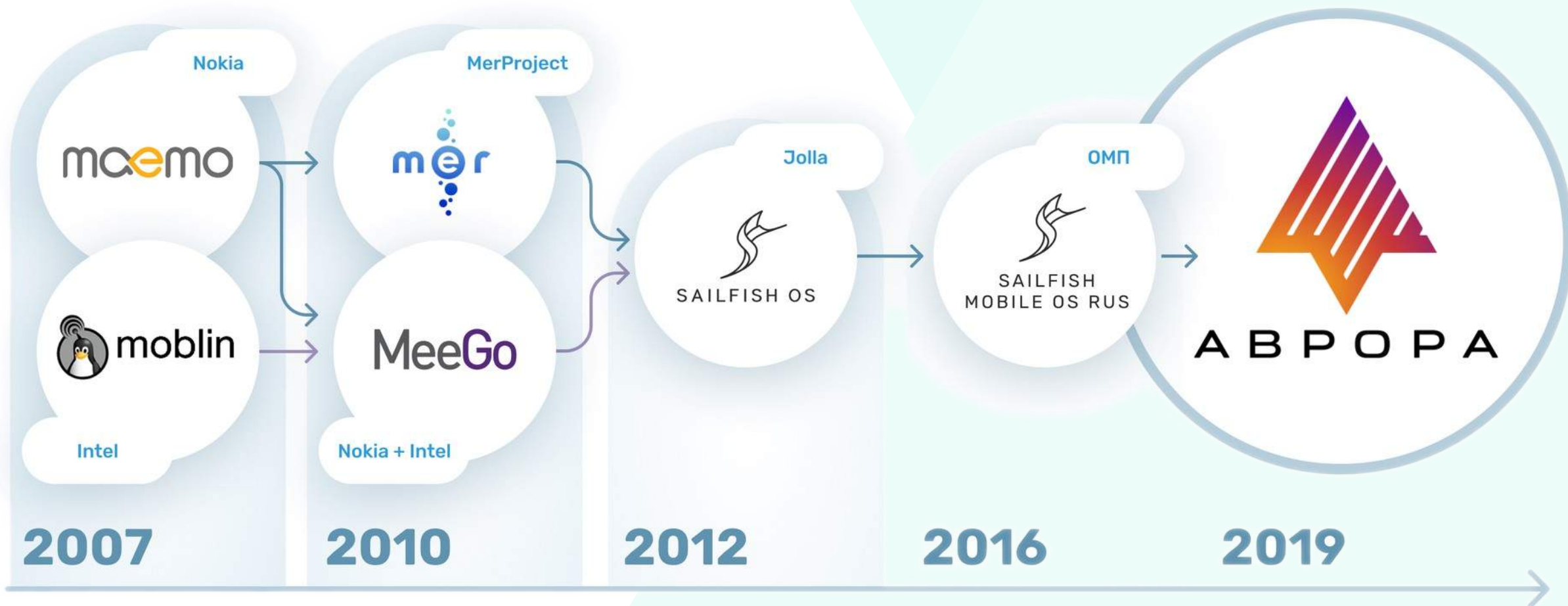
- › Операционная система для смартфонов и планшетов
- › POSIX-совместимый дистрибутив Linux
- › Архитектуры: ARM, x86
- › Встроенные СКЗИ
- › Сертификаты
 - » Сертификат ФСБ АК2/КС2
 - » Сертификат ФСТЭК А4/УД4
- › Разработчик



- › Партнёры-разработчики прикладного ПО на корпоративном рынке РФ



ИСТОРИЯ РАЗВИТИЯ ОС АВРОРА



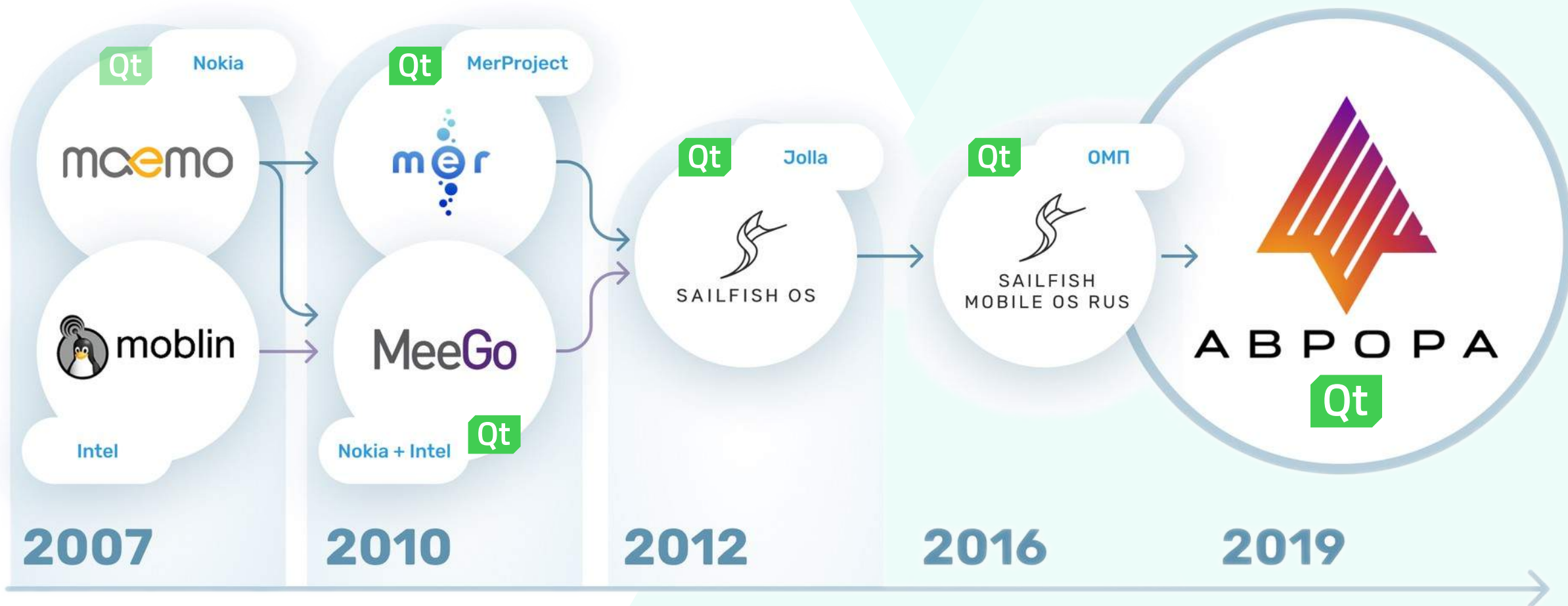
NOKIA



jolla



ИСТОРИЯ РАЗВИТИЯ ОС АВРОРА



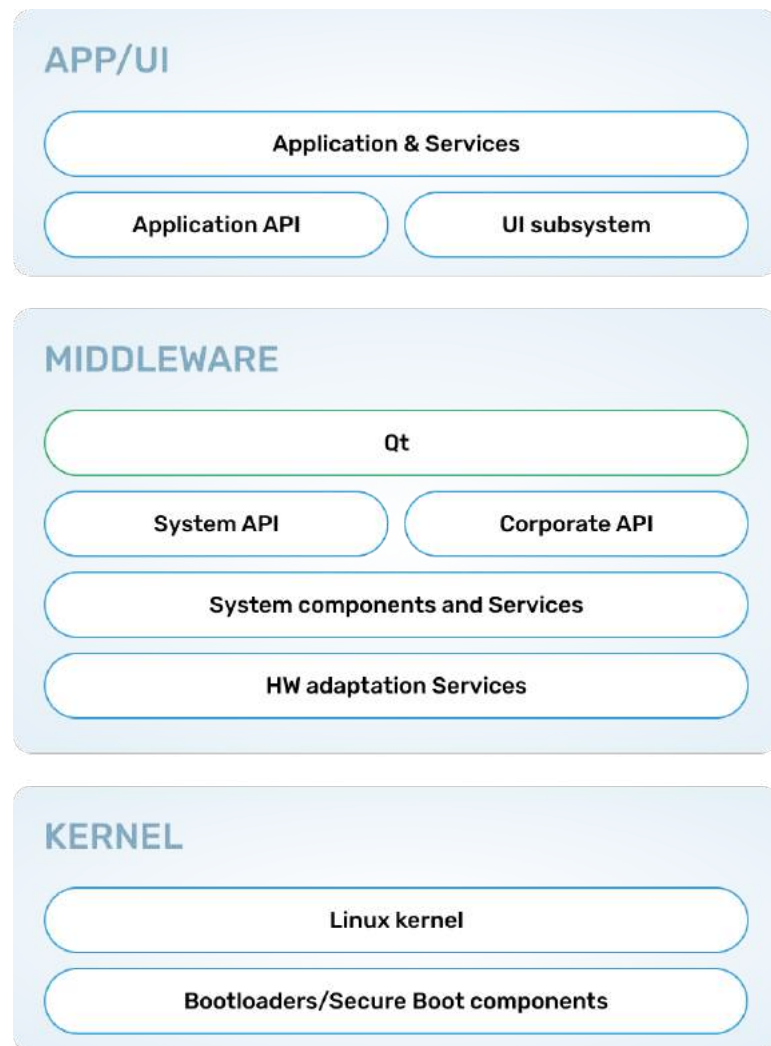
NOKIA



jolla



АРХИТЕКТУРА ОС АВРОРА



- > Ядро Linux
- > Wayland
- > systemd, d-bus
- > Qt
- > RPM, 800+ пакетов

РАЗВИТИЕ И ПОДДЕРЖКА РАЗРАБОТЧИКОВ

- › Сопровождение разработки на всех этапах dev-support@omp.ru
 - › Предоставление информации по разработке
 - › Ответы на технические вопросы
 - › Предоставление доступа к ресурсам
 - › Сбор и учёт требований к продуктам ОМП
- › Консультации по техническим вопросам
- › Проведение R&D по технологиям разработки
- › Помощь в подборе исполнителей
- › Разработка и предоставление документации и обучающих материалов
- › Взаимодействие с образовательными организациями

Квалифицированные (обученные) разработчики нужны для...

- › Уменьшения проблем партнёров
- › Повышения качества прикладного ПО
- › Снижения расходов на техническую поддержку
- › Упрощения поиска и найма разработчиков

1. ИСТОРИЯ
2. ЧТО СЕЙЧАС
3. МОДУЛИ
4. API

ФРЕЙМВОРК QT



2

ИСТОРИЯ QT

- › 1991 – Quasar Technologies (в будущем – Trolltech) начинает разработку Qt для Unix и Windows
- › 1995, 20 мая – выпущен Qt 0.90 под Qt Free Edition License
- › 1996, 24 сентября – выпущен Qt 1.0
- › 1998, 12 июля – выпущен KDE 1.0, лицензия Qt изменяется на Q Public License
- › 2000 – выпущен Qt/X11 2.2 под GPL v2
- › 2001 – выпущен Qt 3.0 с поддержкой Mac OS X
- › 2005, июнь – выпущен Qt 4.0 с полноценной поддержкой СПО
- › 2008, 17 июня – Nokia приобретает Trolltech
- › 2009, 14 января – выпущен Qt 4.5 с поддержкой LGPL
- › 2010, 23 июня – выпущен Nokia Qt SDK 1.0 с поддержкой S60 и Maemo
- › 2011, март – Nokia передаёт Qt лицензирование и обслуживание компании Digia для поддержки мобильных платформ
- › 2012, август – Digia приобретает Qt у Nokia
- › 2012, октябрь – выпущен Qt 5.0 с аппаратно ускоренной графикой, основная роль отдана развитию QML и JS
- › 2014, сентябрь – Digia создала Qt Company
- › 2016, май – Qt Company отделилась от Digia, Qt 5 переведён на открытое управление
- › 2020, 8 декабря – выпущен Qt 6.0
- › ТВС...

QT СЕЙЧАС

- › Среда разработки ПО для различных типов устройств
 - » ПК (Linux, Windows, macOS)
 - » Мобильные (Аврора, Sailfish и др. Linux, Android, iOS)
 - » Встраиваемые
- › Сообщество разработчиков более 1 500 000 человек
- › Qt Creator – SDK с открытым исходным кодом
- › Лицензии
 - » Коммерческая – платно, без ограничений
 - » LGPL – можно создавать ПО с закрытым исходным кодом, но с ограничением по линковке
 - » GPL – для открытого исходного кода
 - » Академическая лицензия



МОДУЛИ QT 5

Qt Essentials		Qt Add-Ons		
Qt Core	Qt Quick Controls 2	Active Qt	Qt Network Authorization	Qt Speech
Qt GUI	Qt Quick Dialogs	Qt 3D	Qt NFC	Qt SVG
Qt Multimedia	Qt Quick Layouts	Qt Android Extras	Qt Platform Headers	Qt UI Tools
Qt Multimedia Widgets	Qt Quick Test	Qt Bluetooth	Qt Positioning	Qt WebChannel
Qt Network	Qt SQL	Qt Canvas 3D	Qt Print Support	Qt WebEngine
Qt QML	Qt Test	Qt Concurrent	Qt Purchasing	Qt WebSockets
Qt Quick	Qt Widgets	Qt D-Bus	Qt Quick Controls	Qt WebView
		Qt Gamepad	Qt Quick Extras	Qt Windows Extras
		Qt Graphical Effects	Qt Quick Widgets	Qt X11 Extras
		Qt Help	Qt SCXML	Qt XML
		Qt Image Formats	Qt Sensors	Qt XML Patterns
		Qt Location	Qt Serial Bus	Qt Wayland
		Qt Mac Extras	Qt Serial Port	Compositor



QT API

Общее

- › Модели и представления
- › Файловая система
- › Базы данных
- › Проигрывание и запись мультимедиа
- › Сетевые соединения
- › Взаимодействие приложений и системных служб
- › Многопоточность
- › WebView
- › ...

Для мобильных устройств

- › Элементы интерфейса пользователя
- › Сенсорный экран
- › Геолокация и картография
- › Датчики
- › NFC

ВАЖНЫЕ ТЕХНИЧЕСКИЕ АСПЕКТЫ

1. КОНЦЕПЦИИ QT
2. МОБИЛЬНЫЕ ПРИЛОЖЕНИЯ С GUI
3. РАСШИРЕНИЕ C++ КЛАССОВ
4. СИГНАЛЫ И СЛОТЫ
5. КОНТЕЙНЕРЫ
6. QT QUICK
7. МОДЕЛИ И ПРЕДСТАВЛЕНИЯ
8. ЧТО ДАЁТ ОПЫТ ИСПОЛЬЗОВАНИЯ QT



3

КОНЦЕПЦИИ QT

- > Абстракция GUI
 - » UI/UX ориентирован на платформу
 - » Общая бизнес-логика
- > Сигналы и слоты
 - » Удобный способ отправки и обработки информации о событиях
 - » Возможность отслеживать значения свойств
- > Метаобъектный компилятор
 - » Обрабатывает макросы
 - » Создает добавленный исходный код с метаинформацией
 - » Расширяет синтаксис языка C++

Code less, create more, deploy everywhere



МОБИЛЬНЫЕ ПРИЛОЖЕНИЯ С GUI

Декларативное описание UI с помощью QML

- > Стандартный модуль QtQuick
- > Специфичный модуль Silica
- > Доступ к некоторым API
- > «Лёгкая» логика на «JavaScript»

Запуск с помощью исполняемого кода на C++

- > Основной файл QML встраивается в QQuickView
- > Доступ к библиотекам Qt и C++
- > Низкоуровневый доступ к API устройства
- > «Тяжёлые» вычисления

РАСШИРЕНИЕ C++ КЛАССОВ

```
class Counter
{

public:
    Counter() { m_value = 0; }
    int value() const { return m_value; }

    void setValue(int value);

private:
    int m_value;
};
```

```
#include <QObject>

class Counter : public QObject
{
    Q_OBJECT
    Q_PROPERTY (int value READ value
                WRITE setValue NOTIFY valueChanged)

public:
    Counter() { m_value = 0; }
    int value() const { return m_value; }

public slots:
    void setValue(int value);

signals:
    void valueChanged(int newValue);

private:
    int m_value;
};
```

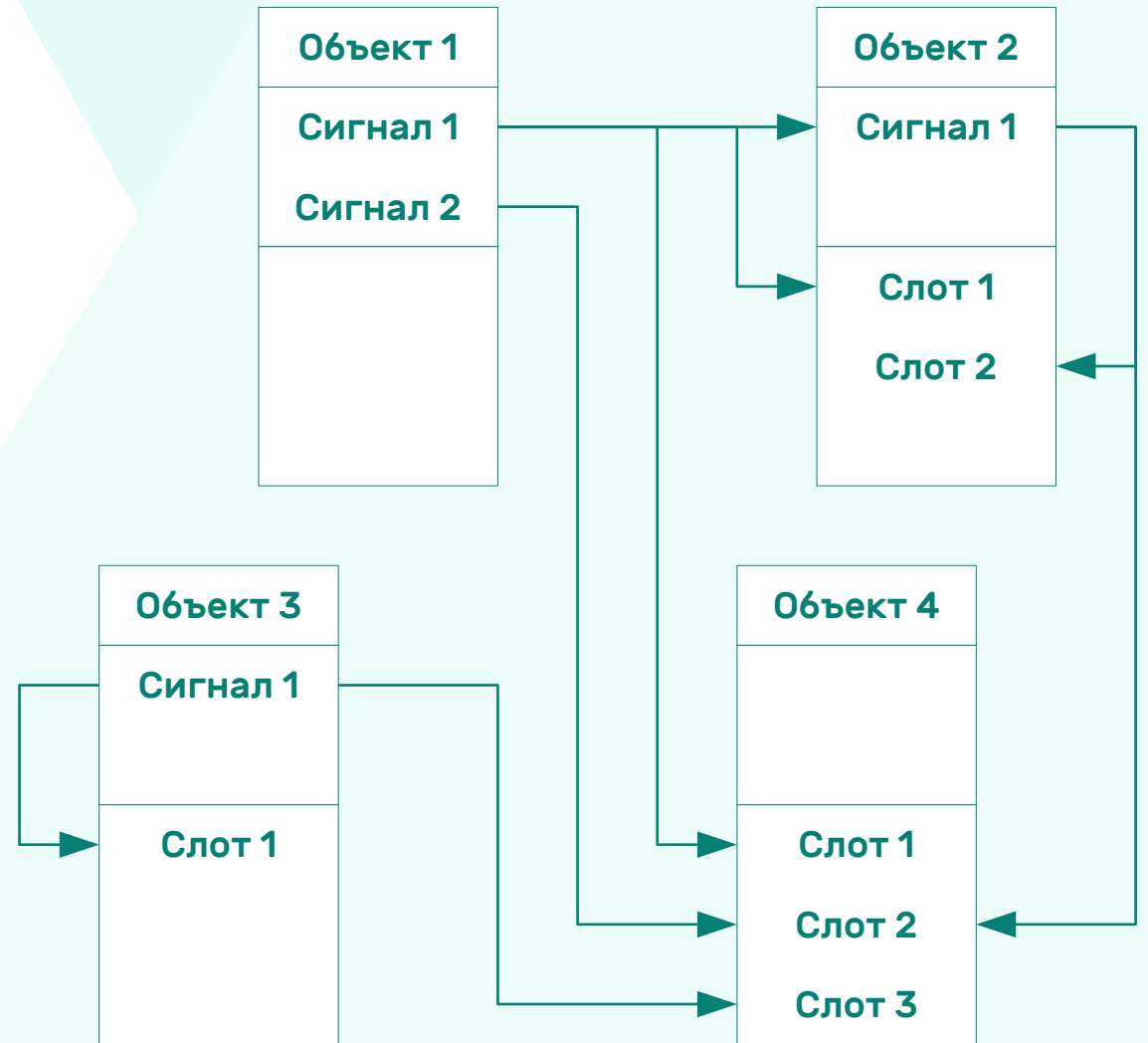
СИГНАЛЫ И СЛОТЫ

> Сигналы

- » Испускаются объектом при изменении его состояния
- » Публичные методы, доступны отовсюду (рекомендуется испускать только владельцем)
- » Не возвращают значений
- » Могут принимать аргументы
- » Реализация генерируется с помощью moc

> Слоты

- » Методы, к которым можно подключать сигналы
- » Не должны возвращать значение
- » Могут быть приватными и публичными
- » Могут быть вызваны как обычные методы
- » Вызываются при испускании присоединённых сигналов (даже приватные)



КОНТЕЙНЕРЫ

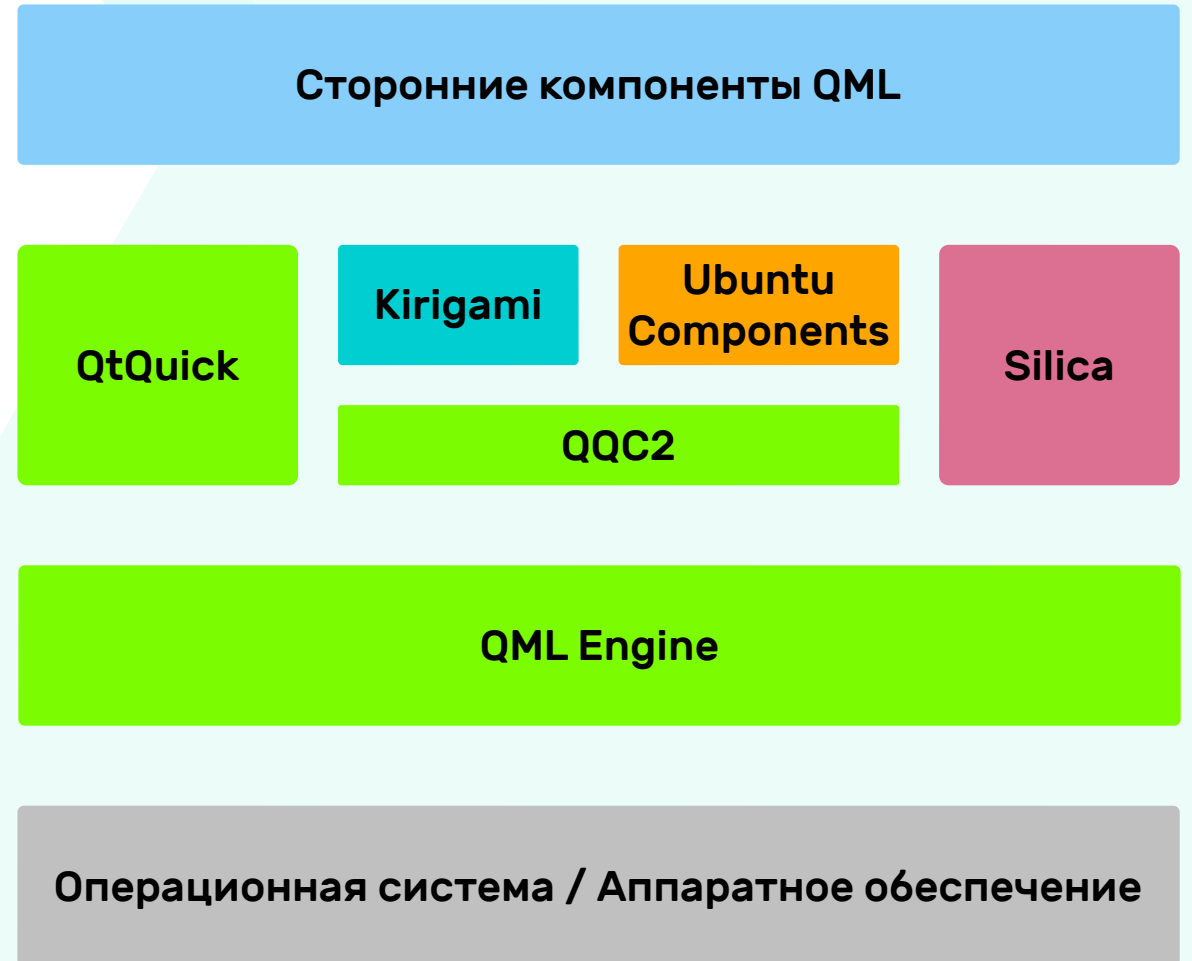
- › Готовые к использованию структуры данных
 - › Range-based for
 - › Итераторы
 - » Java-Style
 - » STL-Style
 - › Совместимы с STL-алгоритмами
 - › Работа с потоками
 - › Implicit Sharing
- › `QList<T>`
 - › `QLinkedList<T>`
 - › `QVector<T>`
 - › `QVarLengthArray<T, Prealloc>`
 - › `QStack<T>`
 - › `QQueue<T>`
 - › `QSet<T>`
 - › `QMap<Key, T>`
 - › `QMultiMap<Key, T>`
 - › `QHash<Key, T>`
 - › `QMultiHash<Key, T>`

QT QUICK

Технология быстрой разработки

- › QML – язык разметки
 - » Декларативный
 - » Поддерживает вставки на «JavaScript»
 - » Доступны свойства объектов C++
- › QML Engine – интерпретатор QML
- › QtQuick – общие компоненты QML, поставляемые Qt

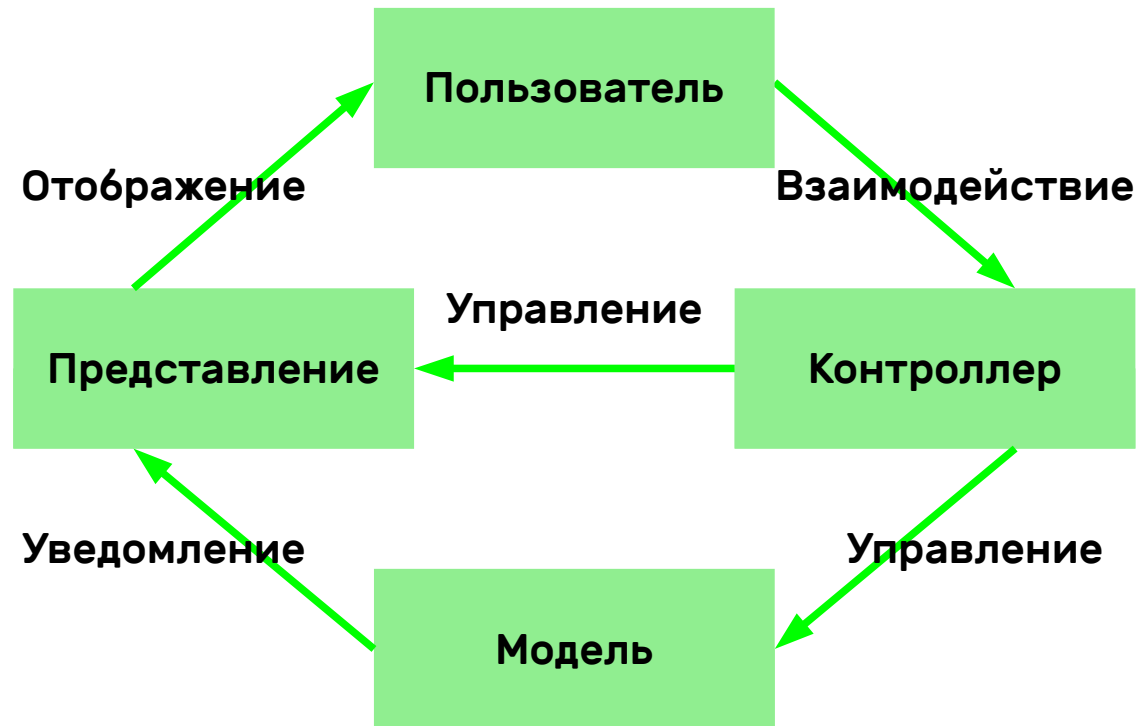
```
import QtQuick 2.0
```



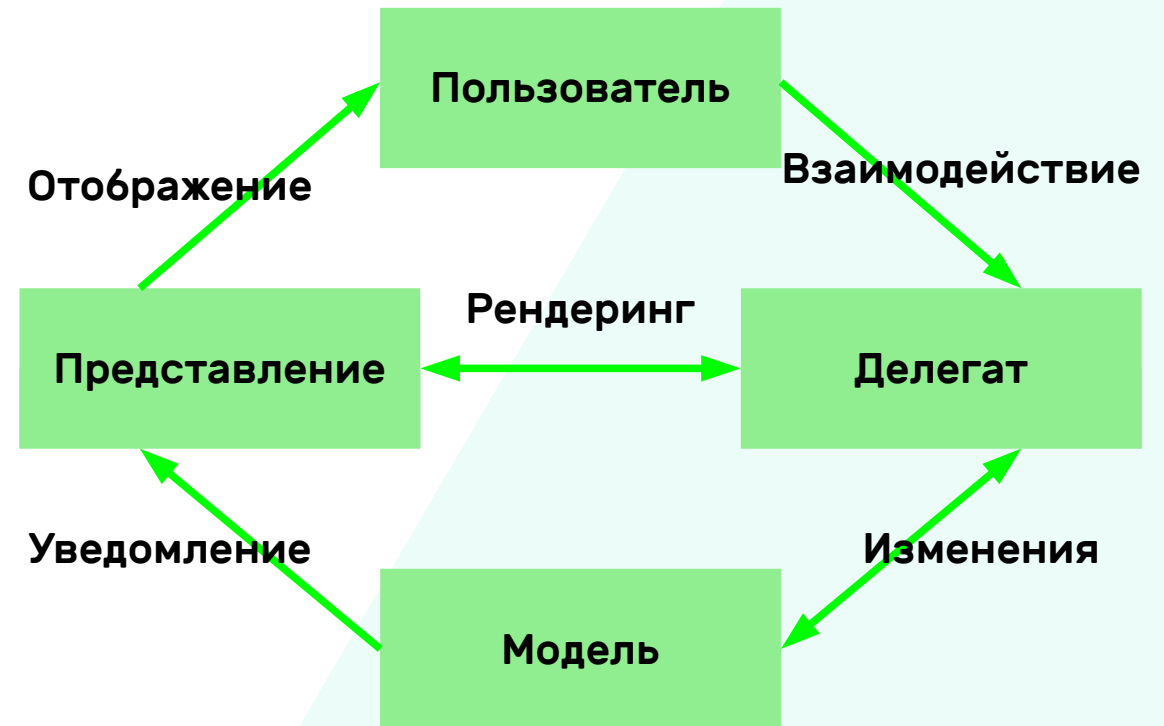
МОДЕЛИ И ПРЕДСТАВЛЕНИЯ

- › Активно используются для отображения и взаимодействия с однородным контентом
- › Можно реализовывать свои

Model-View-Controller



Model-View-Delegate



ЧТО ДАЁТ ОПЫТ ИСПОЛЬЗОВАНИЯ QT

Знания

- > Императивный и декларативный подходы
- > Мета-объектная парадигма
- > Понимание контейнеров и действий с ними
- > Поток
- > Паттерны
 - » Implicit Sharing
 - » MVC, MVD
 - » PIMPL
- > События
 - » Очередь
 - » Обработчики
 - » Сигналы и слоты как альтернатива обработчикам

Практические навыки

- > Написание кода на QML, C++ и JS
- > Проектирование ПО
 - » Гибридная связка (декларатив + императив)
 - » Модели-представления
- > Работа с API
 - » Мультимедиа
 - » Сети
 - » БД
 - » Файлы
 - » Датчики
 - » Картография
 - » ...

КОГО ЧЕМУ И КАК ОБУЧАЕМ

1. ПЕРВЫЕ АКТИВНОСТИ
2. ЧЕГО ХОТЯТ СТУДЕНТЫ И РАЗРАБОТЧИКИ?
3. ЧТО НУЖНО СТУДЕНТАМ И РАЗРАБОТЧИКАМ?
4. СТАТЬИ НА ХАБРЕ
5. ПРИМЕРЫ ПО
6. РАБОТА С ВУЗАМИ
7. СЕМЕСТРОВЫЙ КУРС ДЛЯ ВУЗОВ
8. КУРС ДЛЯ ПАРТНЁРОВ

4

ПЕРВЫЕ АКТИВНОСТИ

- › Серия тренингов в 2016 г
 - » 5 городов: Санкт-Петербург, Москва, Ярославль, Петрозаводск, Нижний Новгород
 - » 300+ участников
 - » 15 уникальных презентаций и видео по разработке
- › Летняя школа в Иннополисе, 27–30 июля 2016 г.
 - » 67 участников:
 - » 17 презентаций, 29 видеозаписей:
 - » 5 вводных, 7 учебных, 4 от партнеров, 13 с проектами
- › Как результат
 - » Сформированы материалы по разработке приложений
 - » 8 обучающих видеозаписей на YouTube
 - » Собраны первые «шишки»
 - » Начат процесс по структурированию и оформлению учебных материалов



ОБУЧЕНИЕ РАЗРАБОТКЕ НА QT ДЛЯ МОБИЛЬНЫХ УСТРОЙСТВ

ЧТО ХОТЯТ СТУДЕНТЫ И РАЗРАБОТЧИКИ?

ЧТО ХОТЯТ СТУДЕНТЫ И РАЗРАБОТЧИКИ?

Студенты

- › «Закодировать» свои идеи
- › Получить зачёт и уйти на каникулы

ЧТО ХОТЯТ СТУДЕНТЫ И РАЗРАБОТЧИКИ?

Студенты

- > «Закодировать» свои идеи
- > Получить зачёт и уйти на каникулы

Разработчики

- > Полнейшую документацию на все случаи жизни
- > Перенести код с Android/iOS
- > Чтобы за них решили все сложные проблемы

ОБУЧЕНИЕ РАЗРАБОТКЕ НА QT ДЛЯ МОБИЛЬНЫХ УСТРОЙСТВ

ЧТО НУЖНО СТУДЕНТАМ И РАЗРАБОТЧИКАМ?

ЧТО НУЖНО СТУДЕНТАМ И РАЗРАБОТЧИКАМ?

Студенты

- > Не привязанные к технологии концепты
 - » Декларативный и императивный подходы
 - » Паттерны
 - » Структуры данных, работа с ними
- > Практика реализации ПО
- > Возможность применить знания C/C++ для прикладных задач

Разработчики

- > Получить знания, как решать конкретные задачи

СТАТЬИ НА ХАБРЕ

1. Начало разработки
2. Основы
3. Стилизация приложения
4. Меню
5. Архитектура FLUX в QML на примере приложения для запоминания литературных терминов
6. Управляем стандартным плеером с помощью голосовых команд
7. Таймеры и реализация экспорта в файл на примере приложения для ведения списка дел
8. Работа с D-Bus
9. Работа с LocalStorage
10. Работа с уведомлениями на примере приложения для ведения заметок
11. Тестирование QML-компонентов
12. Создание собственных QML-компонент на C++
13. Модульное тестирование Qt/C++
14. Особенности работы с датами и временными зонами
15. Тестирование QML-кода, зависящего от C++
16. Работа с календарём и списком контактов
17. Использование системного API
18. Использование системных функций D-Bus
19. Работа с Bluetooth
20. Работа со звуком на примере приложения DayTimer
21. Использование датчиков
22. Разработка сканера Wi-Fi сетей

ПРИМЕРЫ ПО

› Лицензия: BSD 3-Clause

Примеры

gitlab.com/AuroraOS/examples

1. Silica Use Cases
2. Silica Browser
3. Map Viewer
4. Maintainer
5. QML Camera
6. QML Registration Form
7. QML Compass
8. QML File Browser
9. QML Media Picker
10. Hybrid Dictaphone
11. QML Page Stack Example

Руководства (пошаговые примеры)

gitlab.com/AuroraOS/tutorials

- › Chat
- › Notes
- › Media Player
- › Sensor Inspector

РАБОТА С ВУЗАМИ

1. ПетрГУ, 2016–2017

» Составлен семестровый курс разработки для Sailfish Mobile OS RUS

2. Университет Иннополис, весна 2017

» Проведены недельных элективных модуля разработки для Sailfish Mobile OS RUS

3. Запись первой онлайн-версии

Введение в разработку приложений на Sailfish OS

4. ОмГУ, СибАДИ, весна 2017

» Введены семестровые курсы разработки для Sailfish Mobile OS RUS

5. Университет Иннополис и МФТИ, 2018–2019

» Семестровый курс по мобильной разработке на Qt (ОС Аврора + Android)

6. Тиражирование курса

» ННГУ, ЯрГУ, СКФУ, КФУ...

» Мы открыты для сотрудничества!



СЕМЕСТРОВЫЙ КУРС ДЛЯ ВУЗОВ

1. Инструменты и технологии
 2. Основы QML
 3. Продвинутый QML
 4. Нативная разработка, экспорт C++ классов
 5. Компоненты UI
 6. Файлы и базы данных
 7. Мультимедиа
 8. Карты
 9. Сеть
 10. Датчики
 11. Многопоточность
 12. D-Bus
 13. Создание библиотек
 14. Сборка пакета, подготовка к публикации
- › Всего 60 ак. часов очных занятий
 - » 30 часов лекций
 - » 30 часов практики, включая задания и проекты
 - › Входные требования
 - » Навыки C++
 - » Понимание алгоритмов и структур данных

КУРС ДЛЯ ПАРТНЁРОВ

1. Знакомство с ОС Аврора
2. Инструменты разработки
3. Основы QML
4. Организация UI
5. Нативная разработка, экспорт C++ классов
6. Работа с базами данных и файлами
7. Мультимедиа
8. Работа с сетью
9. Работа с датчиками
10. Координаты и карты
11. QML-плагины
12. Публикация и распространение

- > Очный курс
 - » 3 дня лекций и практики
 - » Сертификационное тестирование
- > Онлайн-курс
 - » На stepik.org
 - » Бесплатно для партнёров
 - » ~4 часа видео
 - » Практические задания

По сравнению с курсом для студентов

- » Больше конкретных заданий
- » Но меньше общей практики
- » Ориентирован на ОС Аврора

- > **Мы открыты для сотрудничества по размещению и улучшению курса!**

РЕСУРСЫ ДЛЯ РАЗРАБОТЧИКОВ В ИНТЕРНЕТЕ

- › Портал сообщества разработчиков
community.omprussia.ru
- › Документация по разработке ПО
community.omprussia.ru/documentation/software_development.html
- › Примеры ПО
gitlab.com/AuroraOS/examples
- › Документация Qt
doc.qt.io
- › Qt QML
doc.qt.io/qt-5/qtqml-index.html



aurora-os.ru/vk



aurora-os.ru/telegram



aurora-os.ru/stepik



aurora-os.ru/youtube



aurora-os.ru/habr

ВОПРОСЫ?

EDU@OMPRUSSIA.RU

