# МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Національний університет “Львівська політехніка”**



**Інститут післядипломної освіти**

**РЕФЕРАТ**

На тему:

**«Платформа для розробки ПЗ Qt. Інтеграція з Android Studio»**

з дисципліни

**«Інженерія кросплатформного програмного забезпечення»**

Виконав:

слухач групи ПЗС-21

Гринчук Тарас

Прийняв:

ст. викл.Сенів М.М.

« »\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2015 р.

∑ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

ЛЬВІВ – 2015

## Зміст

## 1. QT – загальні відомості……………………………………………………………….3

## 2. Платформи та ПЗ, що використовується з Qt............................................................5

## 3. Середовище для розробки Qt Creator..........................................................................7

## 4. Розробка з допомогою Qt в Android Studio………………………………………....8

## Список використаних джерел.......................................................................................13

## 1. QT – загальні відомості

Qt (варіант вимови від розробників cute — к'ют) — крос-платформовий інструментарій розробки програмного забезпечення (ПЗ) мовою програмування C++. Дозволяє запускати написане за його допомогою ПЗ на більшості сучасних операційних систем (ОС), просто компілюючи текст програми для кожної операційної системи без зміни сирцевого коду. Містить всі основні класи, які можуть бути потрібні для розробки прикладного програмного забезпечення, починаючи з елементів графічного інтерфейсу й закінчуючи класами для роботи з мережею, базами даних, OpenGL, SVG і XML. Бібліотека дозволяє керувати нитями, працювати з мережею та забезпечує крос-платформовий доступ до файлів.

Qt також може бути використаним в багатьох інших мовах програмування: Ada (QtAda), C# (Qyoto/Kimono), Java (Qt Jambi), Pascal, Perl, PHP (PHP-Qt), Ruby (QtRuby), та Python (PyQt,PySide).

Гаавард Норд та Айрік Чемб-Інг почали розробку Qt у 1991 році, 3 роками пізніше було створено компанію Quasar Technologies, яка згодом була перейменована в Troll Tech, а потім — в Trolltech.

Назва Qt з'явилася, бо літера "Q" добре виглядала в Гаавардівському Emacs, а "t" позначало toolkit.

Nokia, придбавши Trolltech ASA 17 червня 2008 р., змінила його назву спершу на Qt Software, потім — на Qt Development Frameworks. Відтоді Nokia сфокусувалася на розвитку Qt, щоб зробити його основною платформою для кожного свого мобільного пристрою, включаючи портування платформи Symbian S60. Версія Nokia Qt SDK 1.0 вийшла 23 червня 2010[6].

Але після укладання стратегічної угоди з Microsoft із відмовою Nokia від будь-яких платформ для смартфонів, крім Windows Phone, Nokia втратила інтерес до технологій Qt, бо Windows Phone підтримує єдину мову програмування — C#.

У березні 2011 фінська компанія Digia, постачальник ERP-систем та послуг і рішень у галузі мобільних систем і користувацьких інтерфейсів, оголосила про укладення угоди з Nokia про викуп у тої прав на комерційне ліцензування та надання послуг із підтримки розробки з використанням бібліотеки Qt. Відповідно до запропонованої угоди, близько 3500 компаній, що спеціалізуються на створенні застосунків для десктопів і мобільних пристроїв, перейдуть під заступництво Digia. У вересні 2012 Nokia повністю відмовилася від Qt й Digia купила в неї весь бізнес і програмні технології, пов'язані з Qt.

Компанія Nokia до продажу пов'язаного з Qt бізнесу передала управління над проектом Qt та пов'язані з прийняттям рішень повноваження окремому некомерційному проекту Qt Project, у якому представники спільноти можуть безпосередньо брати участь у розвитку Qt відповідно до принципів меритократії. В руки Qt Project передано розвиток як експериментальної гілки Qt 5, так і стабільної гілки Qt 4. Крім того, між компанією Nokia і організацією KDE Free Qt Foundation укладено угоду, яка передбачає, що в разі посилення ліцензійної політики, банкрутства Nokia або припинення розвитку Qt проект KDE отримує права переліцензувати код Qt під ліцензією BSD або будь-якою іншою відкритою ліцензією та продовжити розробку своїми силами. Ця угода продовжує діяти й при зміні власника Qt.

Першим значним кроком у напрямі збільшення відкритості Qt стала зміна на початку 2009 року ліцензії з GPL на LGPL 2.1, що дозволило безперешкодно використовувати Qt в закритих проектах без необхідності купівлі комерційної ліцензії або відкриття сирцевих текстів свого продукту під ліцензією GPL. Через кілька місяців був уведений в дію публічний Git-репозиторій, що дозволило стороннім розробникам безперервно відстежувати розвиток Qt (до цього час від часу публікувалися снапшоти коду) і пропонувати свої поліпшення або виправлення. Отже, починаючи з версії 4.5 Qt поширюється за трьома ліцензіями (незалежно від ліцензії, сирцевий код Qt один і той же):

Qt Commercial — для розробки ПЗ з власницькою ліцензією, що допускає модифікацію самої Qt без розкриття змін;

GNU GPL — для розробки відкритого програмного забезпечення, що поширюється на умовах GNU GPL;

GNU LGPL — для розробки ПЗ з власницькою ліцензією, але без внесення змін до Qt.

До версії 4.0 під вільною ліцензією розповсюджувалися лише Qt/Mac, Qt/X11, Qt/Embedded, але, починаючи з 4.0.0 (випущеної в кінці червня 2005), Qt Software «звільнила» й Qt/Windows. Слід зазначити, що існували сторонні вільні версії Qt/Windows <4.0.0, зроблені на основі Qt/X11.

## 2. Платформи та ПЗ, що використовується з Qt

Існують різні версії бібліотеки для таких платформ:

* Qt/X11 — для X Window System (Unix / Linux)
* Qt/Mac — для Mac OS X
* Qt/Windows — для Microsoft Windows
* Qt/Embedded — для вбудованих платформ (PDA, смартфонів…)
* Qt/WinCE — для Windows CE
* Qt/Maemo — для Maemo
* Qt/Symbian — для Symbian

Версія Qt/Embedded розповсюджувалася до недавнього часу. Тепер ця платформа перейменована в Qtopia Core і поширюється як окремий продукт, який забезпечує базову функціональність розробки застосунків усієї лінійки платформ, призначених для вбудовуваних і мобільних пристроїв (КПК, смартфонів тощо).

Qt 4 поширюєтсья в чотирьох редакціях:

* Qt Console — для розробки застосунків без графічного інтерфейсу (мережні демони, консольні застосунки тощо).
* Qt Desktop Light — полегшена версія для розробки застосунків із графічним інтерфейсом, але без підтримки баз даних, мережі та XML.
* Qt Desktop — повна версія, містить усі компоненти.
* Qt Open Source Edition — повна версія для розробки відкритого програмного забезпечення, що поширюєтсья під ліцензією GNU GPL. Не підтримує комерційні компілятори.

З часу своєї появи в 1996 році комерційна версія бібліотеки Qt лягла в основу тисяч успішних проектів у всьому світі. Крім того, Qt є фундаментом популярного робочого середовища KDE, що входить до складу багатьох дистрибутивів GNU/Linux.

Серед відомих проектів особливо треба відзначити:

* програма для IP-телефонії Skype;
* програма для обробки зображень Adobe Photoshop Album;
* мережева карта світу Google Earth.

Відмінна особливість Qt від інших бібліотек — використання Meta Object Compiller (MOC) — попередньої системи обробки початкового коду (загалом, Qt — це бібліотека не для чистого C++, а для його особливого діалекту, з якого й «перекладає» MOC для подальшої компіляції будь-яким стандартним C++ компілятором). MOC дозволяє в багато разів збільшити потужність бібліотек, вводячи такі поняття, як слоти (slots) і сигнали (signals). Qt комплектується графічним середовищем розробки графічного інтерфейсу QTDesigner, що дозволяює створювати діалоги і форми «мишею». Ідеологія створення форм у Qt базується на використанні менеджерів розташування, котрі надають «гумовий» дизайн, при якому розмір і розташування елементів форм визначаються автоматично, що значно прискорює розробку графічного інтерфейсу. В поставці Qt є «Qt Linguist» — могутня графічна утиліта, що дозволяє спростити локалізацію й переклад вашої програми багатьма мовами, та «Qt Assistant» — довідкова система Qt, що спрощує роботу з документацією для бібліотек і дозволяє створювати крос-платформову довідку для ПЗ, розробленого на основі Qt.

Бібліотека розділена на декілька модулів, для четвертої версії бібліотеки — це:

* QtCore — класи ядра бібліотеки для використання іншими модулями;
* QtGui — компоненти графічного інтерфейсу;
* QtNetwork — набір класів для мережевого програмування. Підтримка різних високорівневих протоколів може мінятися від версії до версії. У версії 4.2.x присутні класи для роботи з протоколами FTP і HTTP. Для роботи з протоколами TCP/IP і UDP призначені такі класи як: QTcpServer, QTcpSocket для TPC/IP і QUdpSocket для UDP;
* QtOpenGL — набір класів для роботи з OpenGL;
* QtSql — набір класів для роботи з базами даних мовою структурованих запитів SQL. Основні класи цього модуля у версії 4.2.х: QSqlDatabase — клас для надання з'єднання з базою, для роботи з якоюсь конкретною базою даних, вимагає об'єкт, успадкований від класу QSqlDriver — абстрактного класу, який реалізується для конкретної бази даних і може вимагати для компіляції SDK бази даних. Наприклад, для збірки драйвера під базу даних FireBird/InterBase вимагає .h файли й бібліотеки статичної лінковки, що входять у комплект постачання цієї БД;
* QtScript — класи для роботи з Qt Scripts;
* QtSvg — класи для відображення й роботи зі Scalable Vector Graphics (SVG) даними;
* QtXml — модуль для роботи з XML, підтримується SAX і DOM моделі роботи;
* QtDesigner — класи створення розширень QtDesigner'а для своїх власних віджетів;
* QtUiTools — класи для обробки в застосунку форм Qt Designer;
* QtAssistant — довідкова система;
* Qt3Support — модуль з класами, необхідними для сумісності з бібліотекою Qt версії 3.х.х;
* QtTest — модуль для роботи з UNIT тестами;
* QtWebKit — модуль WebKit, інтегрований в Qt і доступний через її класи;
* QtXmlPatterns — модуль для підтримки XQuery 1.0 і XPath 2.0;
* Phonon — модуль для підтримки відтворення та запису відео й аудіо, як локально, так і з пристроїв із мережі;
* QtCLucene — модуль для підтримки повнотекстового пошуку, застосовується в новій версії Assistant в Qt 4.4;
* ActiveQt — модуль для роботи з ActiveX і COM технологіями для Qt-розробників під Windows. Модуль доступний тільки в комерційній редакції Qt.

Також реалізована технологія WOC — widgets on canvas, за допомогою якої буде реалізована Plasma в KDE 4.1, Буде можливим використовувати віджети бібліотеки Qt прямо в аплетах. Забезпечує розташування віджетів на QGraphicsView із можливістю масштабування й різних графічних ефектів.

Бібліотека використовує власний формат проекту, що іменується .pro файлом, у якому зібрана інформація про те, які файли будуть скомпільовані, за якими шляхами шукати заголовні файли та багато іншої інформації. Згодом за допомогою утиліти qmake з них виходять makefile для make-утиліти компілятора. Також є можливість роботи за допомогою інтеграторів з такими середовищами програмування, як Microsoft Visual Studio 2003/2005, а зовсім недавно стала доступна інтеграція в Eclipse, для версії бібліотеки 4.х.х.

Розробники на Java можуть використовувати Qt за допомогою фреймворка Qt Jambi від того ж виробника, проте його розробка була припинена 2009 року.

## 3. Середовище для розробки Qt Creator

Qt Creator — інтегроване середовище розробки, призначене для створення крос-платформових застосунків з використанням бібліотеки Qt. Підтримується розробка як класичних програм мовою C++, так і використання мови QML, для визначення сценаріїв, в якій використовується JavaScript, а структура і параметри елементів інтерфейсу задаються CSS-подібними блоками. Qt Creator може використовувати GCC або Microsoft VC++ в якості компілятора і GDB як зневаджувач. Для Windows версій бібліотека комплектується компілятором, заголовними і об'єктними файлами MinGW.

Сирцеві тексти Qt Creator поширюються в рамках ліцензії LGPL.

Qt Creator розроблена норвезькою компанією Trolltech, яку у 2008 році поглинула Nokia. Анонс проекту відбувся на Qt Developer Days в жовтні 2008 року. Публічна бета-версія проекту була опублікована 30 жовтня 2008. Фінальний реліз відбувся 3 березня 2009 року (разом з виходом Qt 4.5).

Після укладення стратегічного союзу з Microsoft Nokia втратила інтерес до розвитку технологій Qt. У березні 2011 фінська компанія Digia, постачальник ERP-систем, послуг і рішень в області мобільних систем і користувацьких інтерфейсів, оголосила про укладення угоди з Nokia про викуп у тої прав на комерційне ліцензування та надання послуг з підтримки розробки з використанням бібліотеки Qt. У вересні 2012 Nokia повністю відмовилася від Qt і Digia купує у Nokia весь бізнес і програмні технології, пов'язані з Qt.

Особливості:

* Зроблений спеціально для розробки на Qt
* Вбудований редактор форм (Qt Designer) і довідкова система (Qt Assistant)
* Контекстно-залежна система допомоги
* Розширюваність плагінами
* Є графічний фронтенд для GDB
* Підтримка зневадження за допомогою CDB
* Для створення проектів використовується qmake (планується підтримка Makefile та тестується підтримка CMake)
* Узагальнене підсвічування синтаксису, підтримується велика кількість мов програмування і розмітки. Є можливість створення своїх стилів підсвічування
* Можливість редагувати етапи складання проекту
* Підтримка розробки на мовах C/C++, JavaScript, QML
* QML-дизайнер
* Можливість розробки під Symbian і Maemo зі зневадженням в симуляторі або на пристрої

## 4. Розробка з допомогою Qt в Android Studio

На даний момент екосистема Qt дозволяє писати додаток повністю засобами цього фреймворку, навіть більше - компілювати його для цілого набору операційних систем, серед яких знаходиться і Android. Але рухаючись цим шляхом дуже складно розробити інтерфейс, який буде виглядати і вести себе відповідно Гайд платформ. Та й стає неможливим використовувати бібліотеки, написані на Java для оформлення зовнішнього вигляду і поведінки програми. Саме тому хотілося б випробувати модель розробки, при якій вся бізнес логіка писалася на С ++, а користувальницький інтерфейс - засобами цільової платформи. Надалі це забезпечить просте перенесення програми на інші платформи із зовнішнім виглядом, характерним для цільової системи і функціоналом, який вже перевірений і відкладений.

Для роботи нам знадобиться бібліотека Qt і ide Qt Creator, налаштовані для роботи з Android-проектами (це зробити досить просто, докладно про це можна прочитати за посиланням), Android Studio, а так само Android ndk.

Для початку потрібно створити андроїд проект (назвемо його QTtests) і в якості імені пакета задамо «penguin.in.flight.qttests». На наступних етапах майстра створення проектів можна вибирати будь-які, зручні для роботи, настройки. Головною умовою для того, щоб можна було слідувати подальшим інструкціям є дотримання імені пакета.

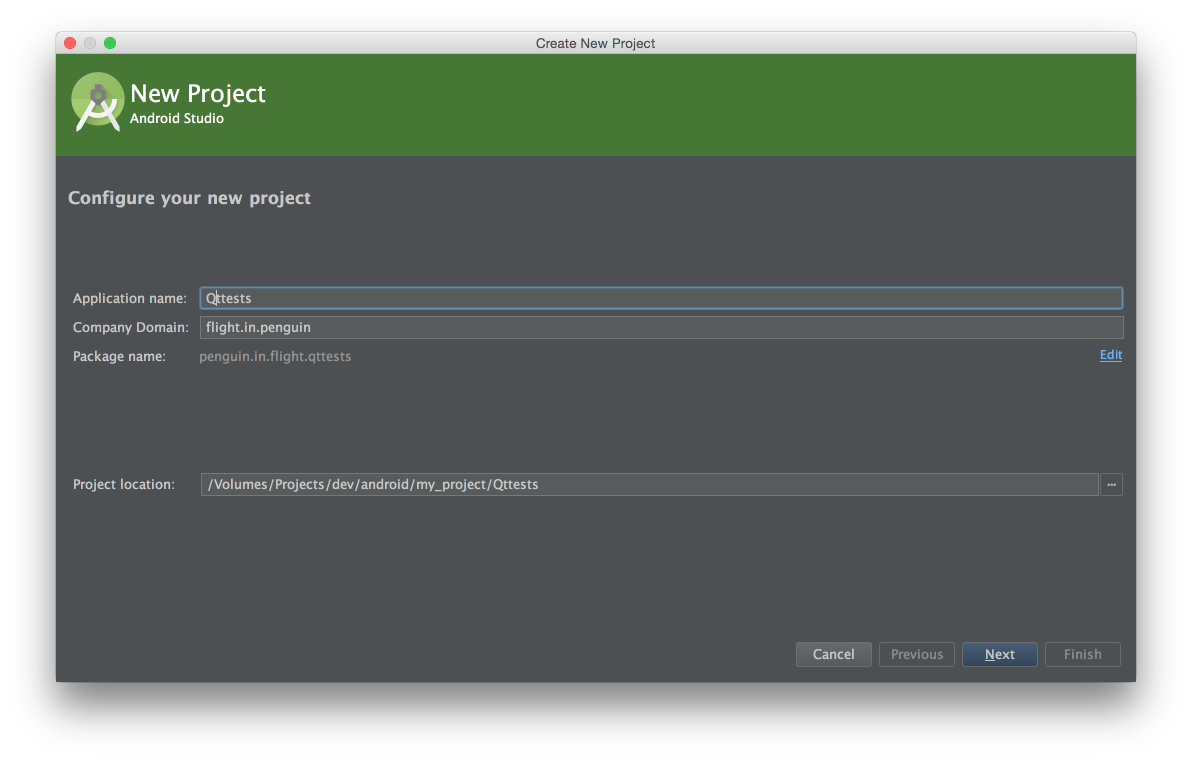
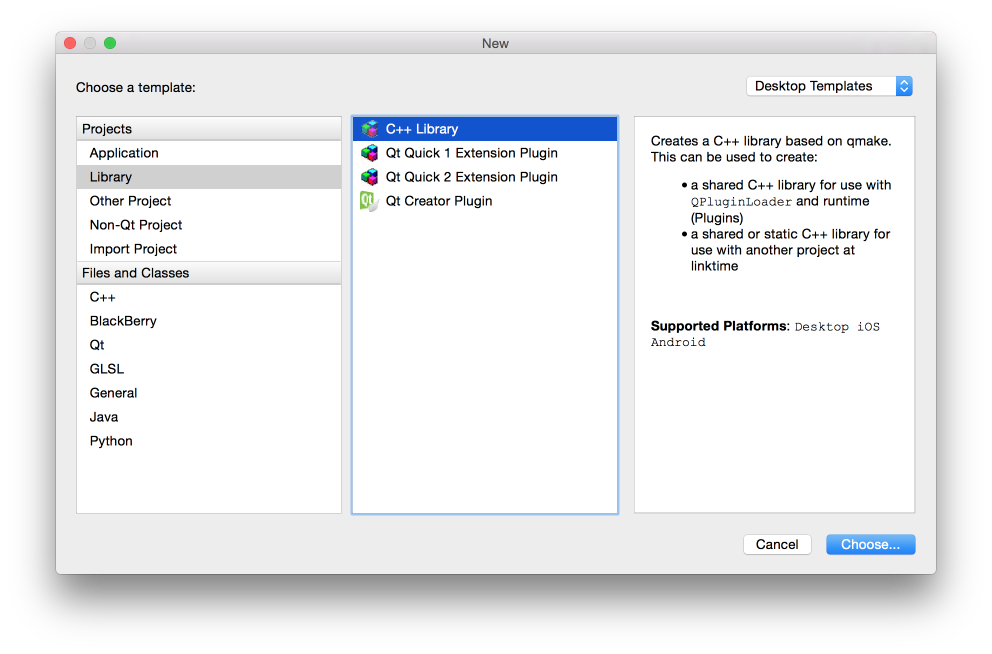
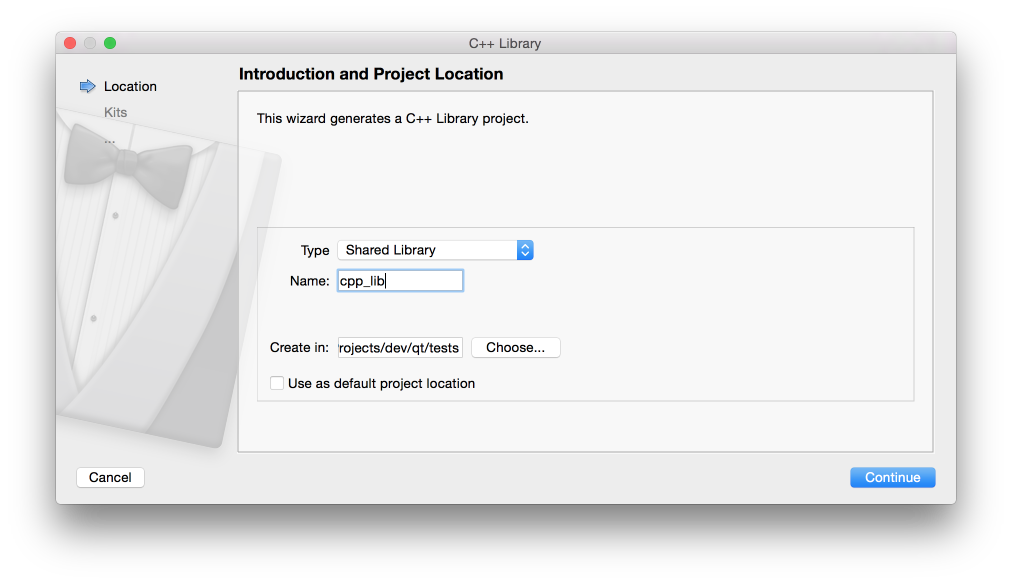


Рис. 1. Створення проекту в Android Studio

Коли Android проект вже є, можна на якийсь час забути про нього і перейти до створення Qt частини. Першим кроком буде створення (бажано в корені основного проекту) стандартного проекту типу бібліотека. Дамо йому ім'я cpp\_lib. При створенні проекту в якості цільової платформи потрібно вибрати armeabi-v7 (рис. 2).





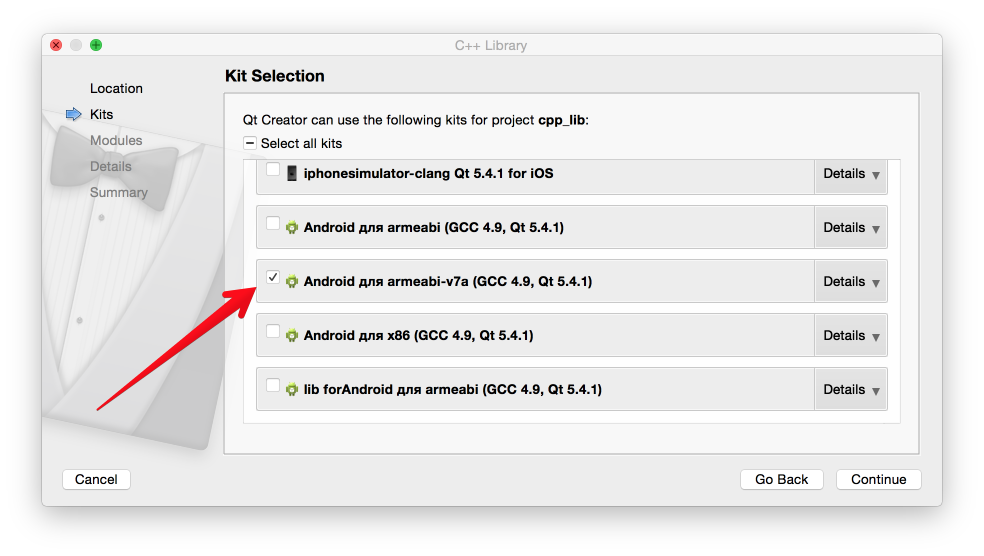


Рис. 2. Створення бібліотеки

Шаблон проекту у нас є, тепер саме час написати наш Hello Word на Qt.

cpp\_lib.h:





Як видно по коду, додаток у нас буде дуже просте. Воно містить дві функції - say і Java\_penguin\_in\_flight\_qttests\_utils\_JavaNatives\_sayHello. Перша - проста і зрозуміла. Вони приймає на вхід об'єкт, який у фреймворку є текстовим рядком, і віддає новий рядок. Набагато цікавіше наступна функція. У ній JNIEXPORT і JNICALL позначаю те, що вона може бути викликана з коду, написаного на джава. В якості повертається типу вона має jstring (який доступний нам завдяки цьому інклуд #include <jni.h>). Цей тип є типом String в джава.

На завершення розберемо аргументи методу. \* Env - це посилання на структуру, яка містить методи, необхідні для роботи з сутностями Джави, а obj є об'єктом, який викликав метод.

Після того, як код написаний, потрібно зробити так, щоб наша бібліотека коректно збиралася і ми могли використовувати її в своїх проектах. Для цього нам потрібно прибрати ось цей рядок TEMPLATE = lib у файлі налаштувань проекту (у нашому випадку це cpp\_lib.pro). Якщо цього не зробити то при складанні ми отримаємо помилку, зразок цієї:

Internal Error: Could not find .pro file.

Error while building / deploying project cpp\_lib (kit: Android для armeabi-v7a (GCC 4.9, Qt 5.4.1))

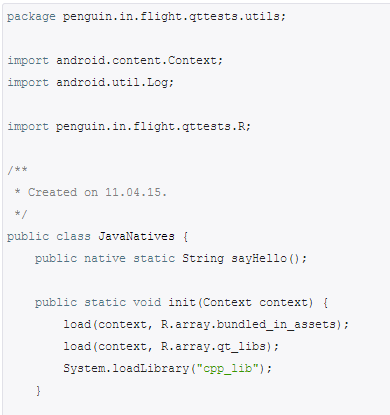
When executing step «Build Android APK»

Далі потрібно зробити так, щоб Qt Creator збирав бібліотеку по такому адресою: {путь\_к\_проекту} / QTtests / app / src / main / qt\_output:

Там же, де ми вказували місце складання, потрібно перевірити налаштування збірки додатку. А саме - цільову платформу збірки. Потрібно вибрати останню (у мене це android 22). Далі поставити галочку навпроти Build Qt libraries in APK. Це потрібно для того, щоб система збирання згенерувала всі необхідні бібліотеки. Так само потрібно поставити галочку навпроти Use Gradle - для того, щоб система збирання згенерувала проект потрібного нам типу.

Тепер ми можемо зібрати бібліотеку і отримати всі необхідні файли.

Тепер у нас є збори бібліотеки і вони навіть підключені до проекту. Далі потрібно підключити їх під час старту самої програми. Для цього будемо використовувати наступний клас:





Як видно з коду, клас знаходиться в пакеті penguin.in.flight.qttests.utils і називається JavaNatives. Це важливо для того, щоб забезпечити зв'язок між тим кодом, який ми написали в нативной частини. Щоб зрозуміти як, давайте згадаємо метод з ім'ям Java\_penguin\_in\_flight\_qttests\_utils\_JavaNatives\_sayHello. Його ім'я складається з префікса Java\_, потім йде ім'я пакету, де імена саб пакетів розділені символом "\_" (penguin\_in\_flight\_qttests\_utils). Далі йде ім'я класу і, нарешті, ім'я методу

Щоб використовувати написаний нами клас в onCreate у стартовій активують, потрібно вставити цей рядок:

JavaNatives.init(this);

Тепер залишилося тільки використовувати те, що ми написали. Для цього додамо кнопку і в обробці натискання на неї вставимо наступний код:

AlertDialog.Builder dialog = new AlertDialog.Builder(this); dialog.setMessage(JavaNatives.sayHello()); dialog.setPositiveButton("Ok", null); dialog.show();

## Список використаних джерел

1. S. Logan, Cross-Platform Development in C++: Building Mac OS X, Linux, and Windows Applications. –Addison-Wesley Professional; 2 edition . – 2007 . – 576 pp.

2. J. Blokh, Effective Java (2nd edition) . –Addison-Wesley Professional; 2 edition . – 2008 . – 346 pp.

3. M. Summerfield, Advanced Qt Programming: Creating Great Software with C++ and Qt 4 (Prentice Hall Open Source Software Development Series) . –Prentice Hall; 1 edition . – 2010 . – 553 pp.