Міністерство освіти і науки України

Львівський національний університет імені Івана Франка

Факультет електроніки та комп’ютерних технологій

Кафедра радіоелектронних

і комп’ютерних систем

КУРСОВА РОБОТА

З курсу «КРОСПЛАТФОРМИННЕ ПРОГРАМУВАННЯ» на тему

«Аудіо плеєр на мові програмування С++ з використанням бібліотеки QT»

Виконав:

ст 2 курсу гр. ФуІ-23

\_\_\_\_\_\_\_\_ Мінькач А.В.

Науковий керівник:

\_\_\_\_\_\_\_\_\_ Футей О.С.

Львів

2019

**Зміст**

Вступ

Вступ

В умовах бурхливого росту інформаційних технологій все більшу популярність і потрібність набувають додатки, що розробляються для використання їх на різних платформах - кросплатформені програми. Це обумовлено появою та розвитком всіляких пристроїв, на яких вони можуть функціонувати.

Одним з швидко розвинених напрямків в галузі багатоплатформеності в даний час є розробка додатків, що працюють на різних операційних системах, таких як десктопні - Windows, MacOS, Linux, мобільні - iOS, Android. Однак, існуючі методики їх розробки орієнтовані на створення додатків під конкретну операційну систему, що обмежує універсальність їх застосування. Наприклад, додатки, розроблені під систему Windows не завжди будуть працювати на мобільних платформах. А з урахуванням зростаючого попиту на мобільні додатки, актуальність вирішення даного протиріччя зростає. Крім того, досягнення високої якості розробки кросплатформних додатків може бути забезпечено ефективного організацією процесу їх автоматизованого тестування.

Анотація

В курсовій роботі присвячена створенню аудіо програвача, кросплатформній розробці з використанням мови програмування C++ та бібліотеки QT. Результатом роботи став створений додаток під операційні системи Windows та Linux, зроблений огляд основних можливостей бібліотеки QT, що використовуються при розробці додатків.

Практично апробовано створений додаток під операційну систему Linux та Windows.

**РОЗДІЛ 1**

**1.1 ПОНЯТТЯ КРОСПЛАТФОРМНОСТІ**

***1.1.1. Загальна характеристика***

**Багатоплатформність** (*кросплатформність, мультиплатформність*) –

властивість програмного забезпечення працювати більш ніж на одній програмній (

тому числі операційній системі) або апаратній платформі та технології, що дозволяють

досягти такої властивості.

Платформа (комп'ютерна) - апаратний або програмний комплекс, який служить основою для різних обчислювальних систем.

Кросплатформенними можна назвати більшість сучасних високоріннених мов програмування. Наприклад, C, C++ i Free Pascal — кросплатформні мови на рівні компіляції, тобто для цих мов є компілятори під різні платформи. Java i C# –кросплатформні мови на рівні виконання, тобто їх виконанні файли можна запускати нa різних платформах 6eз попередньої перекомпіляції. Це забезпечує двохетапна компіляція через проміжний код, для виконання якого в Java використовується віртуальна машина, а в С# — загальномовне середовища програмування CLR, Python, Ruby кросплатформні інтерпретовані мови, їх інтерпретатори існують для багатьох платформ.

Кростлатформність на рівні редакторів зв’язків досягається за допомогою кросплатформних бібліотек, які реалізують незалежний від платформи інтерфейс. Існус велика кількість нестандартних кросплатформних бібліотек: Qt, GTK+, FLTK, STL, OpenGL, SDL, OpenAL, OpenCL.

***1.1.2. Багатоплатформний користувацький інтерфейс***

На різних ОС — незалежно від того, як технічно досягнута робота в них — стандартні елементи інтерфейсу мають різні розміри. Тому просте жорстке позиціонування елементів інтерфейсу неможливо — під іншою ОС вони можуть налазити один на одного. Існує кілька підходів:

* Єдиний стиль, загальний для всіх ОС. Програми виглядають однаково під усіма ОС. Так працюють інтерфейсні бібліотеки Java на зразок Swing.
  + Плюс: можна жорстко розставляти елементи управління на манер Delphi; оригінальний стиль.
  + Мінус: системі доводиться мати свої екранні шрифти; стиль відрізняється від стилю ОС.
* Самоадаптований інтерфейс, який налаштовує сітку під реальні розміри елементів управління. Типові приклади — [wxWidgets](https://uk.wikipedia.org/wiki/WxWidgets), [XUL](https://uk.wikipedia.org/wiki/XUL).
  + Плюс: стандартний стиль ОС, дуже швидкий і [«скіни»](https://uk.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D1%96%D1%87%D0%BD%D0%B8%D0%B9_%D1%96%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BA%D0%BE%D1%80%D0%B8%D1%81%D1%82%D1%83%D0%B2%D0%B0%D1%87%D0%B0) під Windows XP, Vista і 7; деяка автоматизація локалізації.
  + Мінус: щоб зібрати самоадаптовану сітку, потрібен кваліфікований програміст; утруднене щільне компонування.
* Гібридний підхід реалізовано у GTK +.
  + Плюс: шрифти можна брати з системи, а не «тягнути» свої; деяка автоматизація локалізації.
  + Мінус: бере всі недоліки від перших двох підходів. Стиль відрізняється від стилю ОС; утруднене щільне компонування.

У будь-якому випадку, під іншими ОС потрібне хоча б мінімальне тестування, оскільки можливі помилки компонування.

**1.2 Графічний інтерфейс користувача**

***1.2.1. Загальна характеристика***

Графі́чний інтерфе́йс кори́стувача́ (англ. *GUI, Graphical user interface*) — тип інтерфейсу, який дозволяє користувачам взаємодіяти з електронними пристроями через графічні зображення та візуальні вказівки, на відміну від текстови інтерфейсів, заснованих на використанні тексту, текстовому наборі команд та текстовій навігації.

Виконання дій в ГІК — це безпосередня маніпуляція з графічними елементами. Окрім комп'ютерів, GUI використовується в мобільних пристроях, таких, як мобільні телефони, планшети, електронні книги, портативні медіаплеєри тощо. Термін ГІК зазвичай не вживають стосовно інтерфейсів з низькою [роздільною здатністю](https://uk.wikipedia.org/wiki/%D0%A0%D0%BE%D0%B7%D0%B4%D1%96%D0%BB%D1%8C%D0%BD%D0%B0_%D0%B7%D0%B4%D0%B0%D1%82%D0%BD%D1%96%D1%81%D1%82%D1%8C_(%D0%BA%D0%BE%D0%BC%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0_%D0%B3%D1%80%D0%B0%D1%84%D1%96%D0%BA%D0%B0)). Наприклад, в відеоіграх використовують інтерфейс [HUD](https://uk.wikipedia.org/wiki/HUD_(%D0%B2%D1%96%D0%B4%D0%B5%D0%BE%D1%96%D0%B3%D1%80%D0%B8)).

Вперше концепція ГІК була запропонована вченими з дослідницької лабораторії Xerox PARC в 1970-х, але отримала комерційне втілення лише в продуктах корпорації Apple Computer. У операційній системі AMIGAOS ГІК з багатозадачністю був використаний 1985 року. Зараз цей інтерфейс — стандартна складова частина більшості доступних на ринку операційних систем і застосунків.

***1.2.2. Історія виникнення***

Графічний інтерфейс користувача винайшли завдяки дослідженням, проведеним у 60-ті роки в науково дослідному інституті Стенфорда Дугласом Енгельбартом.

Згодом концепцію GUI перейняли дослідники з лабораторії Xerox PARC у 1970-х. 1973 року в лабораторії Xerox PARC зібрали молодих учених і дали їм свободу досліджень. У підсумку, крім усього іншого, на світ з'явилася концепція графічного інтерфейсу WIMP (Windows, Icons, Menus, Point-n-Click) і в рамках цієї концепції було створено комп'ютер Alto. Його не виробляли як комерційний продукт, але він широко застосовувався на фірмі Xerox як корпоративний інструмент.

1979 року Three Rivers Computer Corporation побудувала робочу станцію PERQ, схожу за принципами роботи на Alto. 1981 року Xerox розробила наступника Alto — Star.

Комерційне втілення концепція GUI отримала 1984 року в продуктах корпорації [Apple](https://uk.wikipedia.org/wiki/Apple) Computer. В операційній системі [AmigaOS](https://uk.wikipedia.org/wiki/AmigaOS) GUI з [багатозадачністю](https://uk.wikipedia.org/wiki/%D0%91%D0%B0%D0%B3%D0%B0%D1%82%D0%BE%D0%B7%D0%B0%D0%B4%D0%B0%D1%87%D0%BD%D1%96%D1%81%D1%82%D1%8C) був застосований 1985 року.

Зараз GUI є стандартном для більшості доступних на ринку операційних систем і додатків. Приклади систем, що використовують GUI: [Mac OS](https://uk.wikipedia.org/wiki/Mac_OS), [GEM](https://uk.wikipedia.org/wiki/GEM), [Atari TOS](https://uk.wikipedia.org/w/index.php?title=Atari_TOS&action=edit&redlink=1), [Microsoft Windows](https://uk.wikipedia.org/wiki/Microsoft_Windows), [Solaris](https://uk.wikipedia.org/wiki/Solaris), [GNU/Linux](https://uk.wikipedia.org/wiki/GNU/Linux), [NeXTSTEP](https://uk.wikipedia.org/wiki/NeXTSTEP), [OS/2](https://uk.wikipedia.org/wiki/OS/2), [BeOS](https://uk.wikipedia.org/wiki/BeOS), [Android](https://uk.wikipedia.org/wiki/Android), [iOS](https://uk.wikipedia.org/wiki/IOS), [Bada](https://uk.wikipedia.org/wiki/Bada), [MeeGo](https://uk.wikipedia.org/wiki/MeeGo).

Хоча в переважній більшості систем GUI є надбудовою для операційної системи, існують і незалежні його реалізації. Відомий варіант графічної програми [BIOS](https://uk.wikipedia.org/wiki/BIOS) Setup, коли, ще до завантаження ОС, управління настройками [IBM PC](https://uk.wikipedia.org/wiki/IBM_PC)-сумісної ЕОМ здійснюється мишею, аналогічно повноцінному GUI. Втім, такий варіант BIOS не пройшов перевірку часом. Також є GUI для [Мікроконтролерів](https://uk.wikipedia.org/wiki/%D0%9C%D1%96%D0%BA%D1%80%D0%BE%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D0%B5%D1%80), які не потребують ОС.

***1.2.3. Компоненти графічного інтерфейсу***

До основних елементів інтерфейсу Windows належать:

* робочий стіл;
* панель завдань;
* головне меню;
* вікна програм і папок;
* діалогові вікна;
* значки програм, документів, папок і ярликів.

### Робочий стіл Windows

Після завантаження Windows на екрані монітора з’являється Робочий стіл, на якому можуть відображатися значки об’єктів операційної системи. **Робочий стіл Windows** — основне вікно в інтерфейсі операційної системи, яке займає весь екран та розташовується під усіма іншими вікнами. Він подібний до звичайного письмового стола, на якому замість книжок, журналів та різного канцелярського приладдя розташовано значки — стилізовані зображення системних засобів, папок, документів, програм або їхніх ярликів. **Ярлики** – це невеликі файли, що зв’язані з іншими об’єктами і створюються для швидкого доступу до них.

На робочому столі користувача, який уперше ввійшов у систему, відображується лише значок службової програми Кошик, призначеної для проміжного зберігання видалених документів. Згодом на робочому столі можна розташувати значки інших програм, файлів, папок, документів тощо, проте не слід надмірно захаращувати його, щоб не уподібнити до письмового стола недбалого та неохайного учня.

### Панель завдань Windows

У нижній частині екрана розміщена Панель завдань, яку можна приховати або перемістити в інше місце екрана. Призначення **Панелі завдань** — спростити перехід від однієї запущеної програми до іншої, відображати важливу системну інформацію (наприклад, поточний час. мову введення тексту), а також полегшити виконання найуживаніших операцій.

#### Меню Пуск

На Панелі завдань знаходиться **кнопка Пуск**, вибір якої відкриває Головне меню операційної системи. Меню Пуск можна відкрити також натисненням клавіші Windows [Клавіша Windows](http://informat.in.ua/wp-content/uploads/2011/11/3.klavisha_windows.jpg)або сполученням клавіш Ctrl + Esc. Щоб закрити меню, потрібно натиснути клавішу Esc або вибрати будь-яку точку екрана поза межами меню.

**Меню Пуск** умовно поділене на кілька частин.

**Ліва частина** містить команди для запуску програм, які встановлені в операційній системі. Ця область розділена горизонтальними лініями на три частини:

* У верхній розміщуються так звані **закріплені елементи** – команди запуску програм, які часто використовуються. Користувач може змінити список закріплених елементів.
* У середній частині розміщено список команд запуску **програм, що використовувались останнім часом**. Він дає змогу швидше знову запустити ці програми.
* У нижній частині знаходиться **команда Усі програми**, яка відкриває список програм, установлених у цій операційній системі.

**Права частина меню Пуск** теж розділена на три частини, в яких розміщено команди:

* доступу до місць зберігання даних (наприклад, Мій комп’ютер);
* запуску програм, які встановлюють режими роботи операційної системи і пристроїв комп’ютера (наприклад, Панель керування);
* отримання довідки по роботі ОС, пошуку даних і запуску на виконання програм (наприклад, Довідка та підтримка).

**Нижній рядок меню Пуск** містить команди Завершення сеансу та Вимкнути комп’ютер. Їх вибір дає змогу завершити сеанс роботи користувача, перезавантажити операційну систему або коректно завершити роботу комп’ютера.

На панелі завдань можуть також відображуватися одна чи кілька **панелей інструментів**, де зібрано кнопки запуску найуживаніших програм. Щоб побачити назви цих панелей, потрібно клацнути панель завдань правою кнопкою миші та **вибрати в контекстному меню команду Панелі інструментів**. Установивши або скинувши позначку поруч із назвою панелі. цю панель можна відобразити або сховати.

#### Панель Швидкий запуск

**Панель Швидкий запуск** містить кнопки зі значками програм. Запуск програм здійснюється вибором відповідної кнопки.

#### Область сповіщень

В **Області сповіщень** відображаються повідомлення про поточний час, стан виконання окремих програм тощо.

### Контекстне меню об’єкта

Клацнувши правою кнопкою миші майже будь-який об’єкт ОС Windows, ви відкриєте його **контекстне меню.**

**Контекстне меню** містить команди, які можна застосувати до вибраного об’єкта.

Набір команд контекстного меню змінюється залежно від того, на об’єкті якого типу користувач клацнув правою кнопкою миші. Контекстне меню також відкривається після клацання на об’єктах, які містяться в документах, створених у прикладних програмах, та після клацання на елементах інтерфейсу. Зазвичай контекстне меню містить команду Властивості, за допомогою якої можна змінювати параметри вибраного об’єкта.

**1.3 Основні характеристики бібліотеки Qt**

***1.3.1. Загальна характеристика***

**Qt** – кросплатформовий інструментарій розробки програмного забезпечення (ПЗ) мовою програмування C++. Дозволяє запускати написане за його допомогою ПЗ на більшості сучасних операційних систем (ОС). 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 нитками, 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, Node.js, Pascal, Perl, PHP (PHP-Qt), Ruby (QtRuby), та Python (PyQt, PySide).

***1.3.2. Історія***

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](https://uk.wikipedia.org/wiki/Trolltech).

Назва Qt з'явилася, бо літера «Q» добре виглядала в Гаавардівському [Emacs](https://uk.wikipedia.org/wiki/Emacs), а «t» позначало [toolkit](https://uk.wikipedia.org/w/index.php?title=Toolkit&action=edit&redlink=1).

Nokia, придбавши Trolltech ASA 17 червня 2008 р., змінила його назву спершу на [Qt Software](https://uk.wikipedia.org/wiki/Qt_Software), потім — на [Qt Development Frameworks](https://uk.wikipedia.org/wiki/Qt_Development_Frameworks). Відтоді Nokia сфокусувалася на розвитку Qt, щоб зробити його основною платформою для кожного свого мобільного пристрою, включаючи портування платформи [Symbian](https://uk.wikipedia.org/wiki/Symbian) S60. Версія Nokia Qt SDK 1.0 вийшла 23 червня 2010[[7]](https://uk.wikipedia.org/wiki/Qt#cite_note-7).

Але після укладання стратегічної угоди з [Microsoft](https://uk.wikipedia.org/wiki/Microsoft) із відмовою Nokia від будь-яких платформ для [смартфонів](https://uk.wikipedia.org/wiki/%D0%A1%D0%BC%D0%B0%D1%80%D1%82%D1%84%D0%BE%D0%BD), крім [Windows Phone](https://uk.wikipedia.org/wiki/Windows_Phone), Nokia втратила інтерес до технологій Qt, бо Windows Phone підтримує єдину мову програмування — [C#](https://uk.wikipedia.org/wiki/C-Sharp).

У березні 2011 фінська компанія [Digia](https://uk.wikipedia.org/w/index.php?title=Digia&action=edit&redlink=1), постачальник [ERP](https://uk.wikipedia.org/wiki/ERP)-систем та послуг і рішень у галузі мобільних систем і користувацьких інтерфейсів, оголосила про укладення угоди з [Nokia](https://uk.wikipedia.org/wiki/Nokia) про викуп у тої прав на комерційне ліцензування та надання послуг із підтримки розробки з використанням бібліотеки Qt[[8]](https://uk.wikipedia.org/wiki/Qt#cite_note-8). Відповідно до запропонованої угоди, близько 3500 компаній, що спеціалізуються на створенні [застосунків](https://uk.wikipedia.org/wiki/%D0%97%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA) 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[[9]](https://uk.wikipedia.org/wiki/Qt#cite_note-nokia-digia-9).

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](https://uk.wikipedia.org/w/index.php?title=Qt_Project&action=edit&redlink=1), у якому представники спільноти можуть безпосередньо брати участь у розвитку Qt відповідно до принципів [меритократії](https://uk.wikipedia.org/wiki/%D0%9C%D0%B5%D1%80%D0%B8%D1%82%D0%BE%D0%BA%D1%80%D0%B0%D1%82%D1%96%D1%8F). 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](https://uk.wikipedia.org/wiki/KDE) отримує права переліцензувати код Qt під [ліцензією BSD](https://uk.wikipedia.org/wiki/%D0%9B%D1%96%D1%86%D0%B5%D0%BD%D0%B7%D1%96%D1%8F_BSD) або будь-якою іншою відкритою ліцензією та продовжити розробку своїми силами. Ця угода продовжує діяти й при зміні власника Qt.

***1.3.2. Властивості бібліотеки Qt***

Qt Creator – інтегроване середовище розробки, призначене для створеннякрос-платформових застосунків з використанням бібліотеки Qt. Вінпідтримує розробку як класичних програм мовою C++, так і використаннямови QML для визначення сценаріїв, в якій використовують JavaScript, аструктура і параметри елементів інтерфейсу задаються CSS-подібнимиблоками. Qt Creator може використовувати GCC або Microsoft VC++ яккомпілятор і GDB як зневаджувач.

Для версій Windows бібліотекаукомплектована компілятором, заголовними і об'єктними файлами MinGW. Qtвикористовує C++ з кількома нестандартними розширеннями, які реалізованіза допомогою додаткового препроцесора, який генерує стандартний код на C++ перед компіляцією. Qt також можна використовувати і з іншими мовамипрограмування через спеціальні розроблені прив'язки. Переважно Qt використовують для розробки програм з графічнимінтерфейсом користувача, а також для консольних і серверних додатків.Дозволяє запускати написане за його допомогою програмне забезпечення набільшості сучасних операційних систем, просто компілюючи текст програмидля кожної операційної системи без зміни вихідного коду. Qtрозповсюджується на умовах ліцензії GNU Lesser General Public License і євільним програмним забезпеченням. Найвідомішими проектами, яківикористовують Qt є стільничне середовище KDE, Google Earth, Skype, QtExtended, Adobe Photoshop Album, VirtualBox та OPIE.

Найважливішою помічницею під час розробки з використанням Qt є інтегрована довідка. Документація Qt напрочуд зручна у використанні та вдало створена для швидкого пошуку серед багатого інструментарію Qt. Вона містить не тільки описи класів, які входять до складу модулів, але й короткі приклади використання методів та класів, повні тексти демонстраційних програм, які висвітлюють можливості Qt. Також тут можна знайти кілька покрокових інструкцій для початківців та статті присвячені опису і поясненню механізмів роботи та різних аспектів використання інструментарію. Для перегляду інтегрованої довідки можна скористатися як середовищем Qt Creator, так і спеціальною окремою програмою, яка зветься Qt Assistantта є частиною інструментарію Qt..

**1.4 Основні характеристики класу QMediaPlayer**

***1.4.1. Загальна характеристика***

Клас QMediaPlayer - клас високого рівня відтворення мультимедіа. Його можна використовувати для відтворення такого вмісту, як пісні, фільми та інтернет-радіо. Вміст для відтворення визначається як об'єкт QMediaContent, який можна розглядати як основну або канонічну URL з додатковою інформацією. Якщо передбачено відтворення QMediaContent, його можна відтворити.

Оскільки QMediaPlayer є об'єктом QMediaObject, ви можете використовувати декілька функцій QMediaObject для таких речей, як:

* Доступ до медіаданних, що відтворюються в даний час (QMediaObject :: metaData () і попередньо визначені ключі медіаданних)
* Перевірка чи поточний мультимедіа сервіс доступний (QMediaObject :: availability ())

***1.4.2. Публічні функції для взаємодії з медіа файлами***

|  |  |
| --- | --- |
| * void | **pause**() ставить відтворюваний файл на паузу. |
| * void | **play**() відтворює файд. |
| * void | **setMedia**(const QMediaContent &*media*, QIODevice \**stream* = nullptr) |
| * void | **setMuted**(bool *muted*) |
| * void | **setPlaybackRate**(qreal *rate*) |
| * void | **setPlaylist**(QMediaPlaylist \**playlist*) створює плейліст. |
| * void | **setPosition**(qint64 *position*) |
| * void | **setVolume**(int *volume*) задає гучність файлу. |
| * void | **stop**() припиняє відтворення файлу. |

.

**РОЗДІЛ 2**

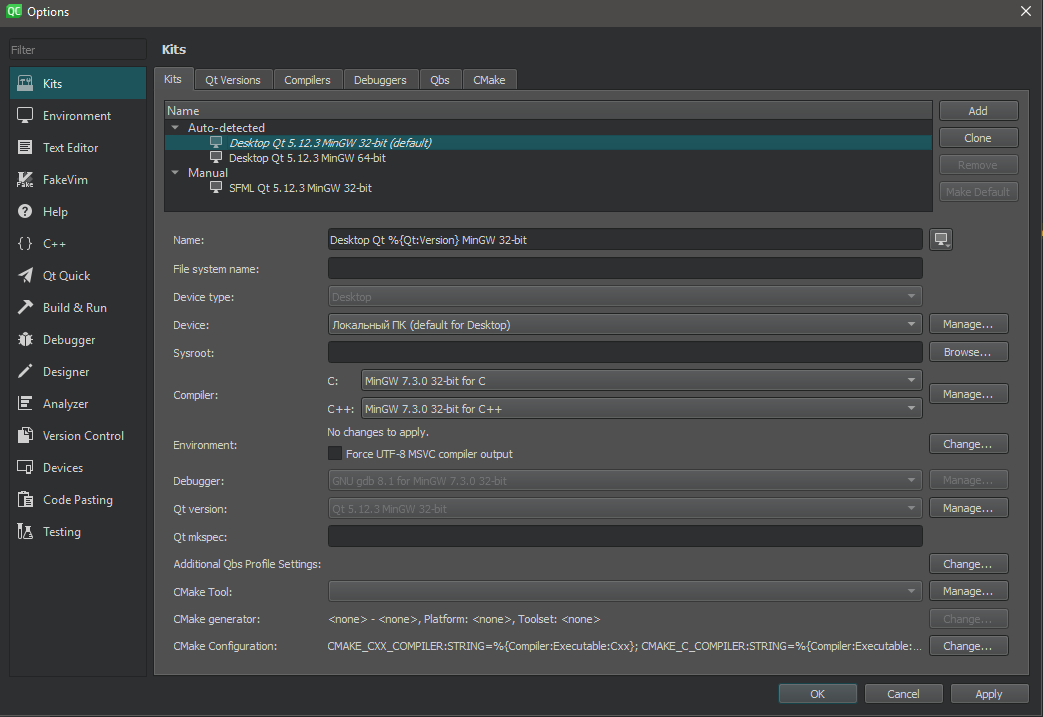
**2.1 ОПИС АЛГОРИТМУ ПРОГРАМИ**

***2.1.1. Налаштування Qt Creator.***

### Для початку на комп’ютері має бути встановлене середовище розробки – Qt Creator. Так як Qt є Open Source, слід лише ознайомитись з правами перед встановленням.

### Спершу торкнемося налаштувань компіляції. Для керування налаштуваннями, які стосуються побудови проекту, QtCreator використовує поняття інструментарію.

**Інструментарій (*Kit*) — це конфігурація, яку складають версія Qt, компілятор та ще деякі додаткові налаштування.** Таким чином, QtCreator дає змогу працювати з кількома різними версіями Qt, кількома компіляторами в системі, вибирати та налаштовувати їх комбінацію для побудови проекту.

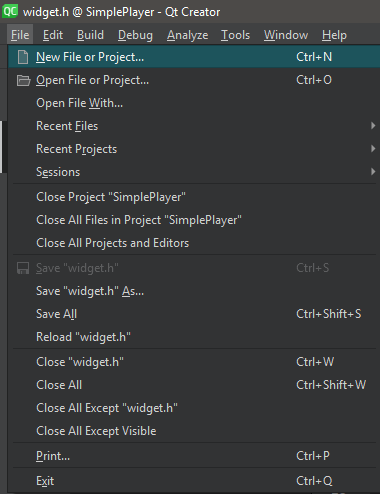
Щоб дізнатись який **інструментарій (*Kit*) встановлений в нашій версію Qt перейдемо** «***Tools***» **–>** «***Options***» **(Мал.1.1)**

Мал 1.1

Як бачимо я використовую компілятор версії MinGW 7.3.0 32-bit.

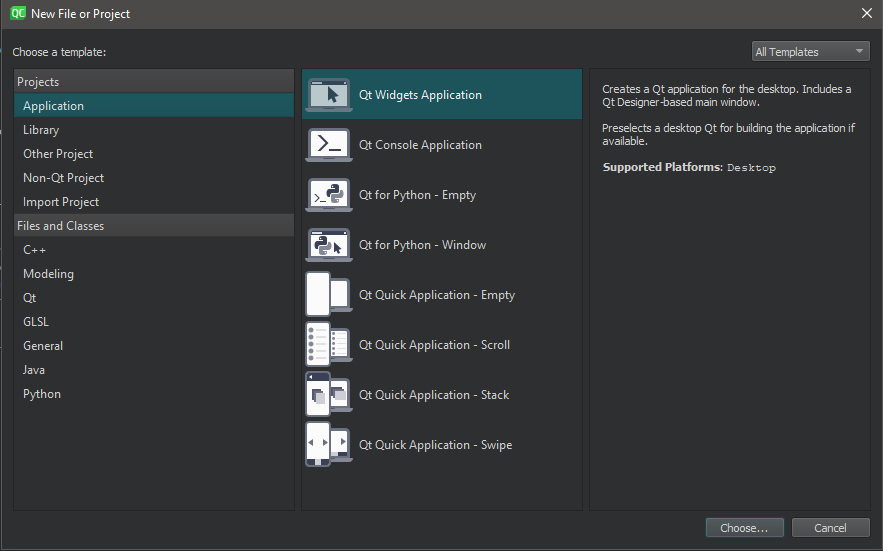
***2.1.1. Створення проекту.***

Після перевірки компілятора ми можемо розпочати роботу. Спершу потрібно створити проект **“*File*” –> “*New File of Project*” або нажавши клавіші Ctrl+N(Мал.1.1)**

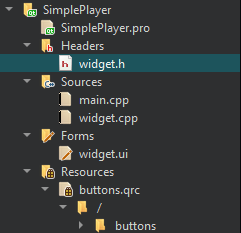


Мал 1.2

Після цього ми обираємо який тип проекту ми бажаємо створити. Обираємо **QT Widgets Application** (Мал. 1.3). Заповнивши потрібні поля в нас створиться проект (Мал. 1.4).



Мал 1.3



Мал 1.4

***2.1.1. Опис коду програми.***

# У main.cpp підключмо хедер файл, щоб наша програма мала гарний вигляд та якщо ми захочемо внести певні зміни то не було нагромадження коду. Також я сворив об’єкт Widget та використав метод show.

# У файлі widget.ui ми створимо форму нашого аудіо плеєра Мал(1.5) заповнивши її запропонованими нам компонентами графічного інтерфейсу.Я використав 5 QToolButton для кнопок(додати до playlist, минулий трек, запустити трек, наступний трек, пауза, принити ), 1 QLabel (для відображення вибраної назви пісні), 1 Spacer(для відображення назви пісні), 1 QTableView(головне вікно відображення всього playlist).

Мал 1.5

# Підключаємо клас QMediaPlayer до нашого проету, а також інші потрібні для подільшої роботи модулі. У файлі «widget.h» описуємо клас Widget.

У файлі widget.cpp будуть описані всі головні дії програми. Модель QStandardItemModel представляє універсальний інтерфейс, тому з його допомогою можна швидко створити будь-яку структуру даних. Встановимо заголовок нашої таблиці «Audio Track» setHorizontalHeaderLabels(QStringList() << tr("Audio Track")<<tr("File Path")) та одразу заберемо колонку в якій зберігається шлях до файлів ui->playlistView->hideColumn(1).

Включаємо виділення пісні в плейлісті setSelectionBehavior (QAbstractItemView::*SelectRows*) щоб було зручно та одразу зрозуміло яка пісня зараз грає.

Дозволяємо виділяти лише одну пісню setSelectionMode (QAbstractItemView::*SingleSelection*) адже виділяти декілька пісень в плейлісті немає сенсу.

Підганяємо розмір вікна з плейліста під довжину назви пісні щоб було зрозуміло яка пісня виконується horizontalHeader()->setStretchLastSection(*true*).

Ініціалізуємо плеєр m\_player = *new* QMediaPlayer(*this*) та ініціалізуємо плейліст m\_playlist = *new* QMediaPlaylist(m\_player).

Підключаю функцію плейліста в функцію плеєра setPlaylist(m\_playlist), та одразу встановлюю гучність програвання пісень m\_player->setVolume(70), встановлюю циклічність для зручності setPlaybackMode(QMediaPlaylist::*Loop*).

Підключаю кнопки управління до слотів управління connect(ui->playlistView, &QTableView::doubleClicked, [*this*](*const* QModelIndex &index){m\_playlist->setCurrentIndex(index.row()).

При додаванні треку в таблицю встановлюю трек в плейліст, робимо багатовибіркову кількість mp3 файлів QStringList files = QFileDialog::getOpenFileNames(*this*,

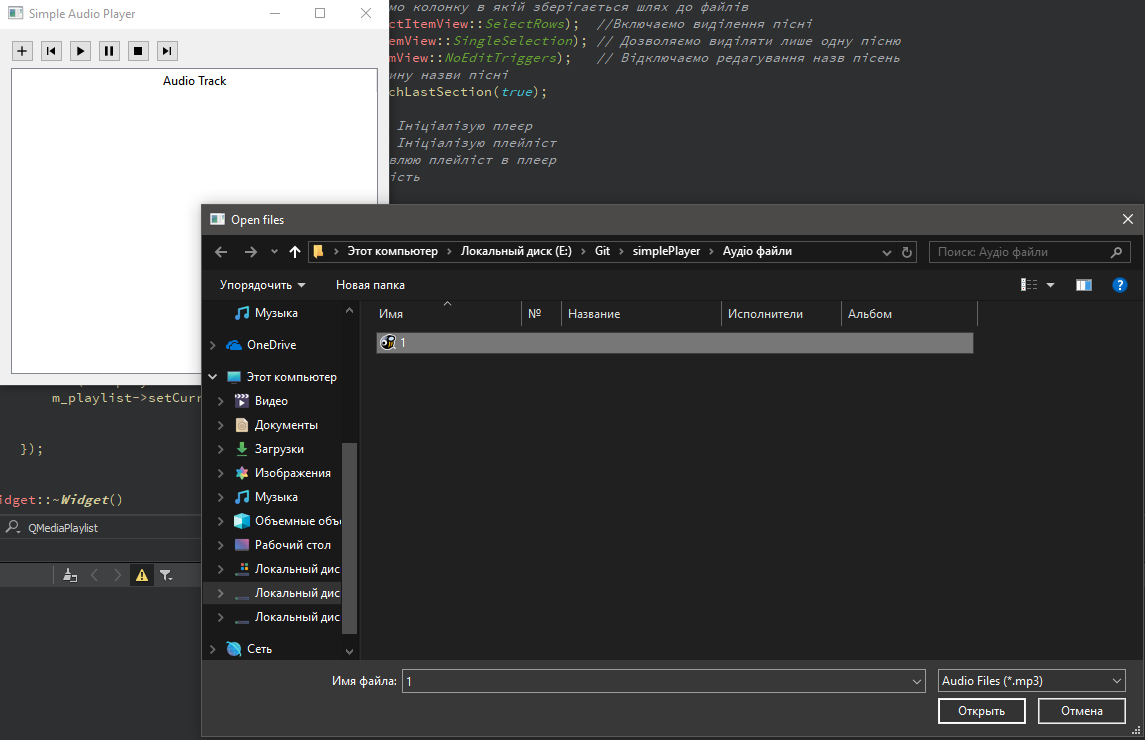
tr("Open files"),QString(),tr("Audio Files (\*.mp3)")).

**РОЗДІЛ 3**

**3.1 ДЕМОНСТАЦІЯ ПРОГРАМИ**

У цій курсовій роботі був реалізований аудіо програвач на мові C++ використовуючи засоби бібліотеки Qt. Ця програма дозволяє нам наступні маніпуляції з аудіо файлами:

1. ***Відкриття та відтворення пісень.***



1. ***Відтворення пісень.***