

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский Авиационный Институт»  
(Национальный Исследовательский Университет)

Факультет №8 «Информационные технологии и прикладная математика»  
Кафедра 805 «Прикладная математика»

**Реферат**  
**по курсу «Вычислительные системы»**  
**1 семестр**

**Тема:**  
**Создание кроссплатформенного приложения на базе**  
**фреймворка QT**

**Автор:**  
Студент 1 курса, гр. М80-103Б-21  
Белоносов Кирилл Алексеевич  
**Руководитель:**  
Севастьянов Виктор Сергеевич

Москва, 2021

## Содержание

1. Введение.....	2
2. Знакомство с фреймворком QT.....	3
3. Создание приложения.....	5
Заключение .....	13
Список литературы .....	14
Приложение.....	15

## **Введение**

Основной целью проекта является создание кроссплатформенного приложения на базе фреймворка QT. В качестве примера я реализую простейший калькулятор для устройств на базе ОС Windows и Android.

Основные задачи:

1. Знакомство с фреймворком QT
2. Создание интерфейса приложения
3. Создание логики приложения
4. Сборка проекта под различные ОС
5. Создание инсталлятора для ОС Windows

## Знакомство с фреймворком QT



Qt (произносится ['kju:t] (кьют) как «cute» или неофициально Q-T (кью-ти, кутэ, кью-тэ)) — фреймворк для разработки кроссплатформенного программного обеспечения на языке программирования C++. Для многих языков программирования существуют наборы библиотеки, позволяющие использовать преимущества Qt: Python — PyQt, PySide; Ruby — QtRuby; Java — Qt Jambi; PHP — PHP-Qt и другие.

Qt позволяет запускать написанное с его помощью программное обеспечение в большинстве современных операционных систем путём простой компиляции программы для каждой системы без изменения исходного кода. Включает в себя все основные классы, которые могут потребоваться при разработке прикладного программного обеспечения, начиная от элементов графического интерфейса и заканчивая классами для работы с сетью, базами данных и XML. Является полностью объектно-ориентированным, расширяемым и поддерживающим технику компонентного программирования.

Отличительная особенность — использование метаобъектного компилятора — предварительной системы обработки исходного кода. Расширение возможностей обеспечивается системой плагинов, которые возможно

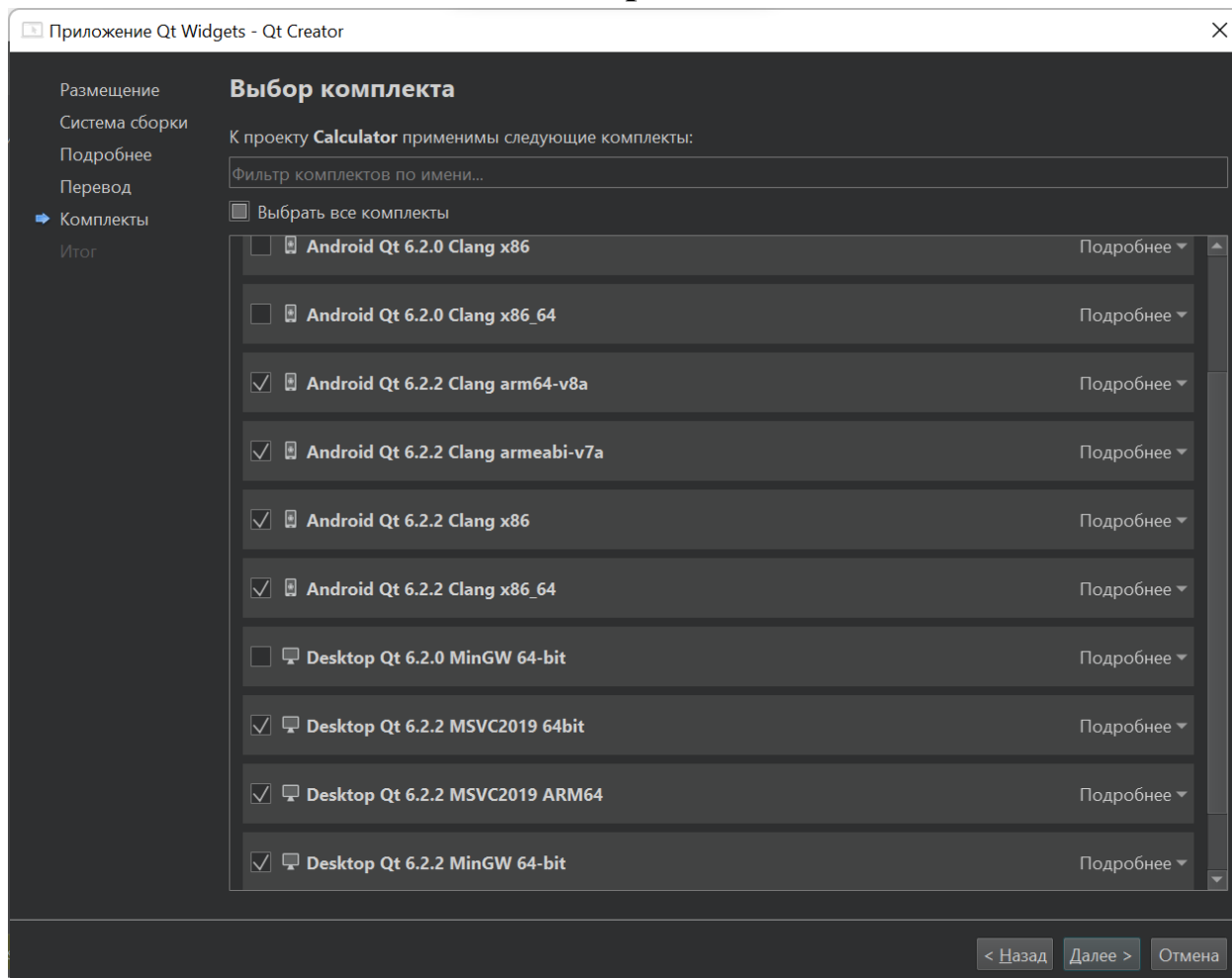
размещать непосредственно в панели визуального редактора. Также существует возможность расширения привычной функциональности виджетов, связанной с размещением их на экране, отображением, перерисовкой при изменении размеров окна.

QT поддерживается на операционных системах Windows, macOS, Linux и устройствах под управлением IOS, Android

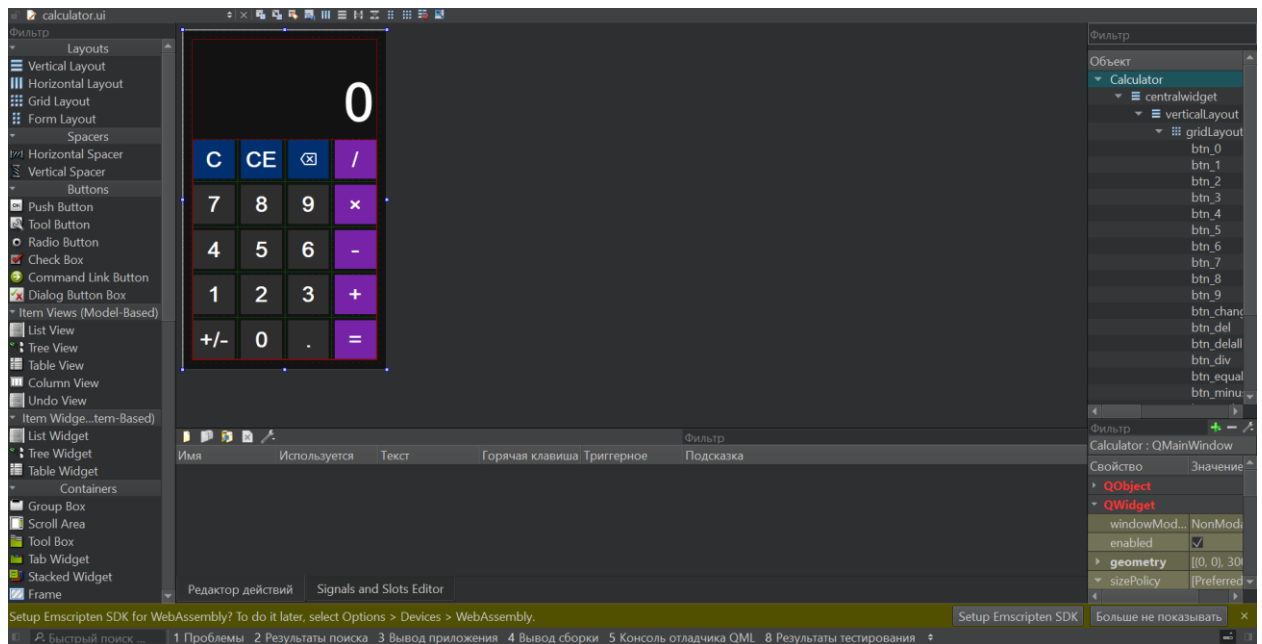
### **Кроссплатформенность**

**Кроссплатформенность** — способность программного обеспечения работать с несколькими аппаратными платформами или операционными системами. Обеспечивается благодаря использованию высокоуровневых языков программирования, сред разработки и выполнения, поддерживающих условную компиляцию, компоновку и выполнение кода для различных платформ. Типичным примером является программное обеспечение, предназначенное для работы в операционных системах Linux и Windows одновременно.

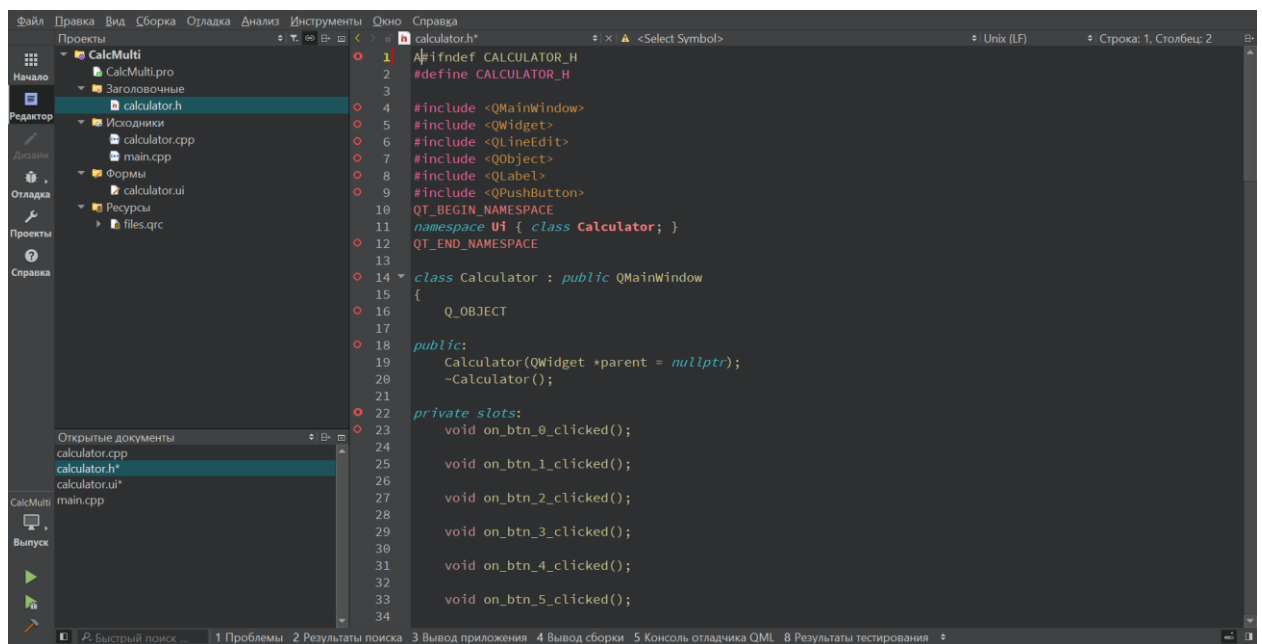
## Создание приложения



Перед работой над проектом я загрузил компоненты для сборки под разные устройства. Также я загрузил Android Studio и Android SDK для отладки проекта под Android.

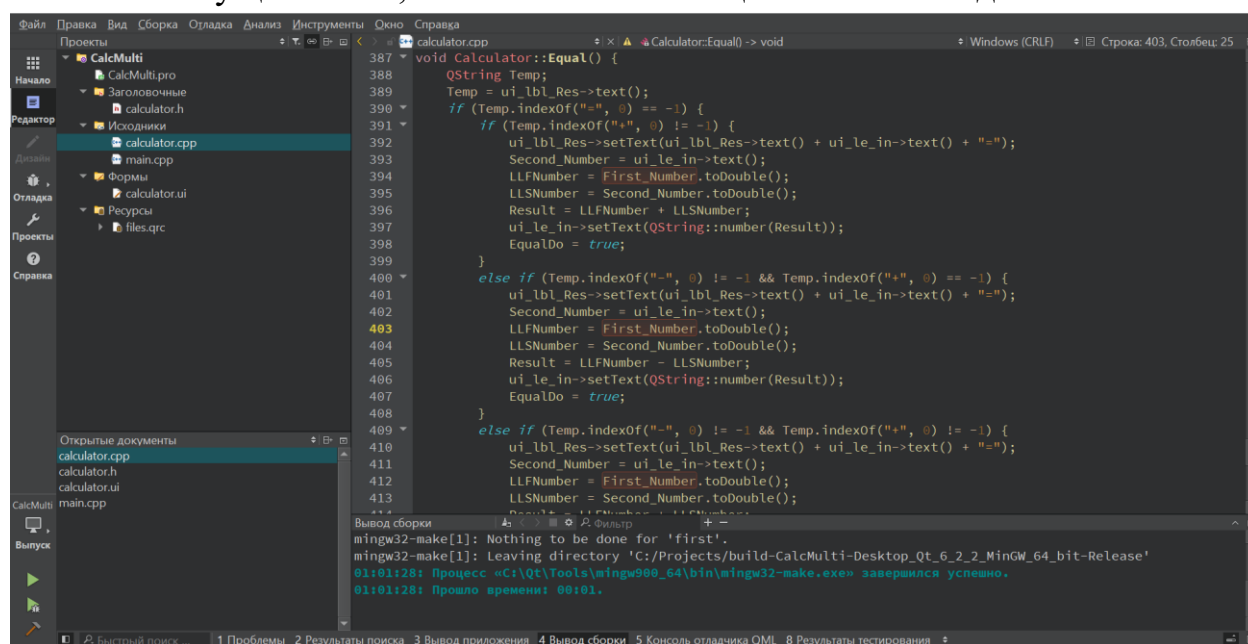


Создам дизайн в QT Designer. Для каждой кнопки пропишу горячие клавиши. Для кнопки удаления последнего символа добавлю икону (все использованные иконки и файлы находятся в файле ресурсов files.qrc). Настрою стиль приложения и кнопок, используя язык CSS.



Для каждой реализую систему слотов и сигналов каждой кнопки. Сигнал вырабатывается когда происходит определенное событие. Слот это функция, которая вызывается в ответ на определенный сигнал. Виджеты Qt имеют много предопределенных сигналов и слотов.

Пропишу логику для слотов. Кнопки цифр будут добавлять в QLineEdit, кнопка смены знака будет добавлять знак - в QLineEdit, если – уже есть в QLineEdit есть, то он будет удаляться. Кнопки арифметических операций будут добавлять число записанное в QLineEdit в QLabel и знак самой операции. «.» будет записывать в QLineEdit точку. С помощью CE можно очистить текущее число, а C полностью очищает панель ввода.



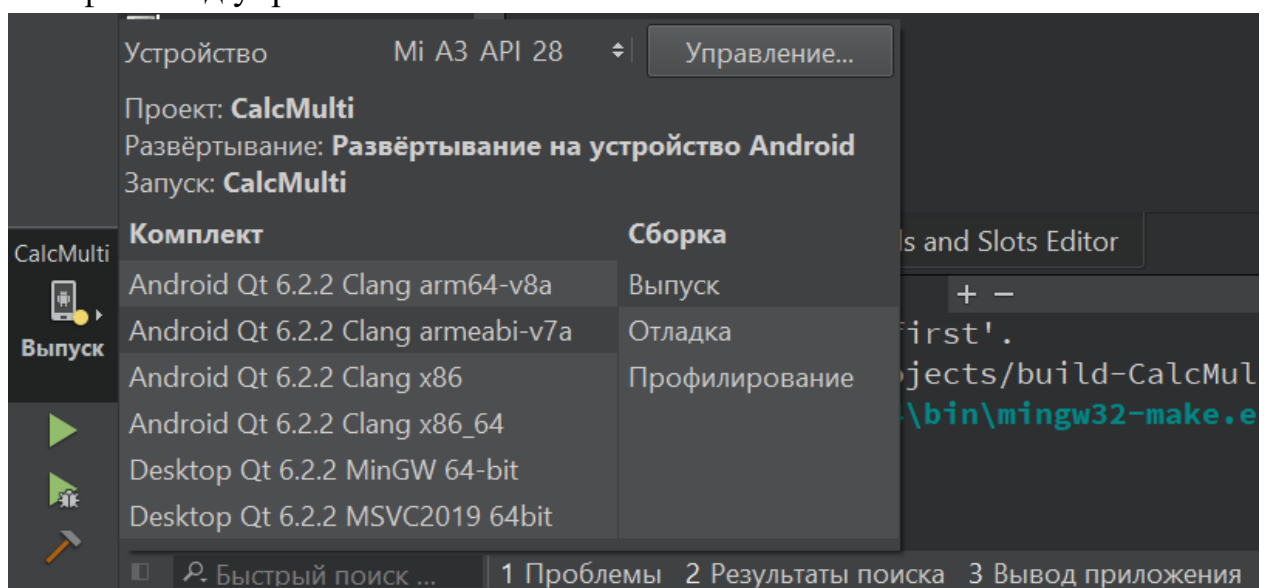
Пропишу логику для кнопки =, она будет считывать значения из QLabel и преобразовывать в double и исходя из символа в QLabel будет производить арифметическую операцию и записывать результат в QLineEdit.

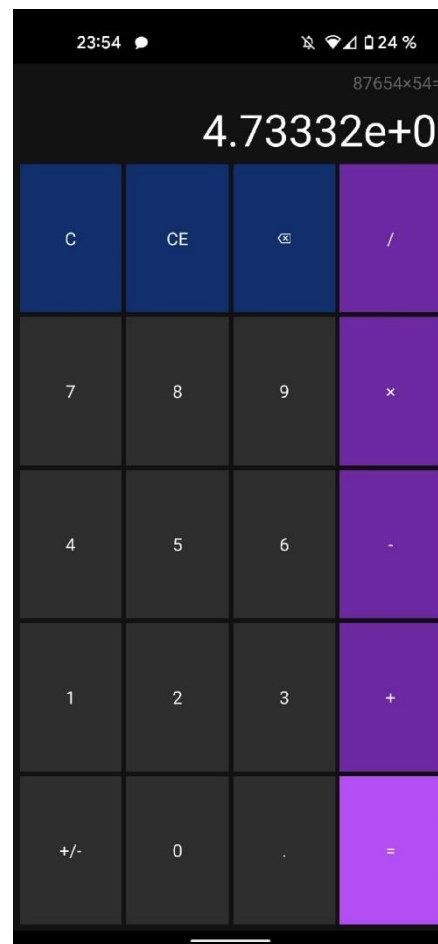
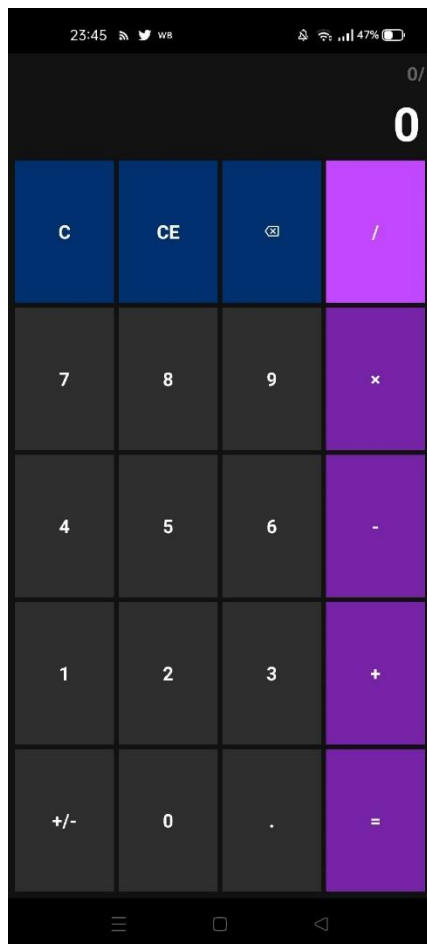




Также реализовал вывод ошибки при делении на 0.

После отладки и выявления ошибок собираю файл .apk проверю работу на телефоне под управлением Android





Проверив работу приложения на разных устройствах перейду к собору файл .exe и подключу необходимые библиотеки

```
Qt 6.2.0 (MinGW 8.1.0 64-bit)
C:\Projects\CalcMulti>windeployqt.exe
Please specify the binary or folder.

Usage: windeployqt.exe [options] [files]
Qt Deploy Tool 6.2.0

The simplest way to use windeployqt is to add the bin directory of your Qt
installation (e.g. <QT_DIR\bin>) to the PATH variable and then run:
    windeployqt <path-to-app-binary>
If ICU, etc. are not in the bin directory, they need to be in the PATH
variable. If your application uses Qt Quick, run:
    windeployqt --qmldir <path-to-app-qml-files> <path-to-app-binary>

Options:
  -?, -h, --help           Displays help on commandline options.
  --help-all              Displays help including Qt specific options.
  -v, --version            Displays version information.
  --dir <directory>       Use directory instead of binary directory.
  --qmake <path>          Use specified qmake instead of qmake from PATH.
  --libdir <path>         Copy libraries to path.
  --plugindir <path>      Copy plugins to path.
  --debug                 Assume debug binaries.
  --release                Assume release binaries.
  --pdb                   Deploy .pdb files (MSVC).
  --force                 Force updating files.
  --dry-run               Simulation mode. Behave normally, but do not
                          copy/update any files.
  --no-patchqt            Do not patch the Qt6Core library.
  --ignore-library-errors Ignore errors when libraries cannot be found.
  --no-plugins            Skip plugin deployment.
```

Перейду к созданию инсталлятора на базе Qt Installer Framework

Пропишу файлы config.xml, package.xml и installscript.qs

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <Installer>
3     <Name>Calculator</Name>
4     <Version>1.0.0</Version>
5     <Title>Calculator installation</Title>
6     <Publisher>KiRiLLBE1</Publisher>
7     <StartMenuDir>Calculator</StartMenuDir>
8     <TargetDir>@RootDir@Program Files (x86)\Calculator</TargetDir>
9     <AllowSpaceInPath>>true</AllowSpaceInPath>
10 </Installer>
```

config.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <Package>
3     <DisplayName>Calculator</DisplayName>
4     <Description>Very cool calculator</Description>
5     <Version>1.0.0</Version>
6     <ReleaseDate>2021-12-17</ReleaseDate>
7     <Name>ru.kirillbel.calculator</Name>
8     <Default>true</Default>
9     <Script>installscript.qs</Script>
10    <UserInterfaces>
11        <UserInterface>readmecheckboxform.ui</UserInterface>
12    </UserInterfaces>
13 </Package>
```

package.xml

```
1 function Component()
2 {
3     // default constructor
4 }
5
6 Component.prototype.createOperations = function()
7 {
8     // This actually installs the files
9     component.createOperations();
10
11     if (systemInfo.productType == "windows") {
12         // Start menu shortcut
13         component.addOperation("CreateShortcut",
14                                "@TargetDir@/CalcMulti.exe",
15                                "@StartMenuDir@/Calculator.lnk",
16                                "workingDirectory=@TargetDir@",
17                                "iconPath=@TargetDir@/Icon/IconCalc.ico");
18
19         // Desktop Shortcut
20         component.addOperation("CreateShortcut",
21                                "@TargetDir@/CalcMulti.exe",
22                                "@DesktopDir@/Calculator @Version@.lnk",
23                                "workingDirectory=@TargetDir@",
24                                "iconPath=@TargetDir@/Icon/IconCalc.ico");
25     }
26 }
```

Installscript.xml

С помощью скрипта на рабочем столе будет появляться ярлык

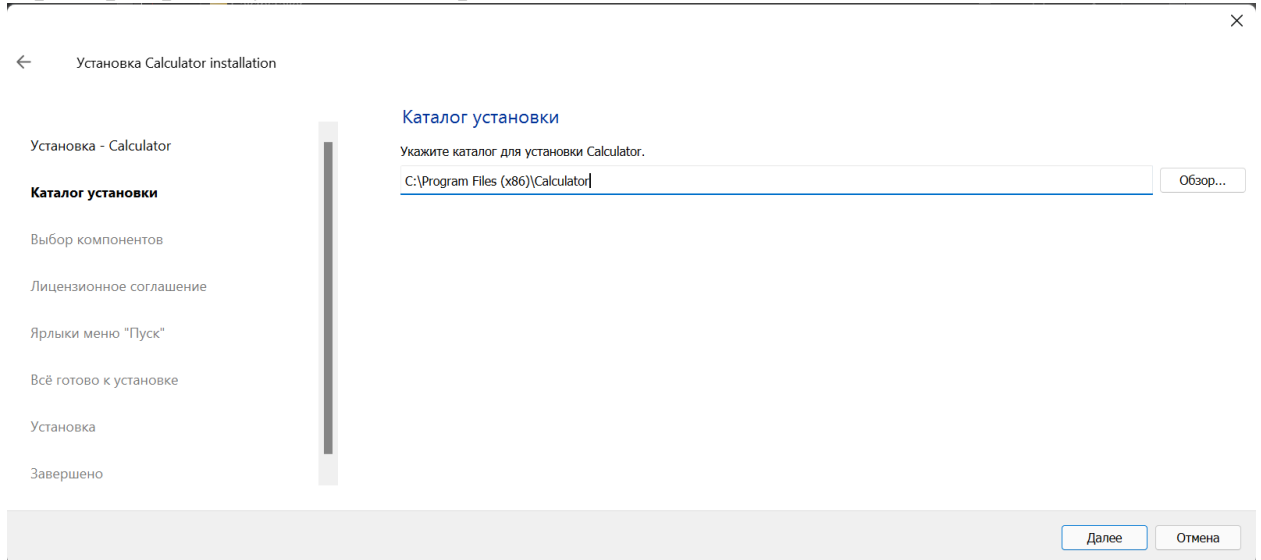
Соберу инсталлятор с помощью инструмента binarycreator.exe

```
Qt 6.2.0 (MinGW 8.1.0 64-bit)
Setting up environment for Qt usage...

C:\Qt\6.2.0\mingw81_64>cd /d "C:\Projects\CalcInstaller"

C:\Projects\CalcInstaller>"C:\Qt\Tools\QtInstallerFramework\4.2\bin\binarycreator.exe" --offline-only -c "config/config.xml" -p "packages" "CalcInstaller.exe"
```

Проверю работу инсталлятора





## **Заключение**

В результате работы было получено кроссплатформенное приложение работающее на ОС Windows и Android. Я научился работать с фреймворком QT, создавать приложения для нескольких платформ и упаковывать его в инсталлятор. Этот проект интересен тем, что он имеет пути расширения, например добавление более сложных математических функций и построение графиков. Были выполнены все поставленные задачи и полученный продукт полностью удовлетворяет намеченным планам.

## Список литературы

1. <https://doc.qt.io/qt-5>
2. <https://habr.com/ru/post/262211/>
3. <https://evileg.com/ru/knowledge/qt/>
4. <https://progtips.ru/qt/razrabotka-android-prilozheniya-chast-1-ustanovka-qt.html>
5. <https://ravesli.com/uroki-po-qt5/>

## Приложение

Исходники: [https://drive.google.com/drive/folders/1sUWq1B-yWw2\\_7LRnoZF8IG3qZEnbxtQd2?usp=sharing](https://drive.google.com/drive/folders/1sUWq1B-yWw2_7LRnoZF8IG3qZEnbxtQd2?usp=sharing)

Инсталлятор: <https://drive.google.com/drive/folders/1upU39GPhyE-MP9vYgQWR7u02JfDNIwsE?usp=sharing>

Приложение: [https://drive.google.com/drive/folders/1YRHkD\\_hmV-qgCyZDy7FYHHPGEH\\_GWI5c?usp=sharing](https://drive.google.com/drive/folders/1YRHkD_hmV-qgCyZDy7FYHHPGEH_GWI5c?usp=sharing)