Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**По творческому заданию**

Тема: Разработка калькулятора

Выполнил работу

студент группы РИС-20-1б

Бурмантов Е.П.

Проверила

Доцент кафедры ИТАС

к.т.н ПоляковаО.А.

Пермь, 2021

**Постановка задачи**

1. Разработать алгоритм калькулятора в соответствии с заданием.
2. Реализовать алгоритм в виде программы на алгоритмическом языке С++.
3. Разработать интерфейс средствами Qt.

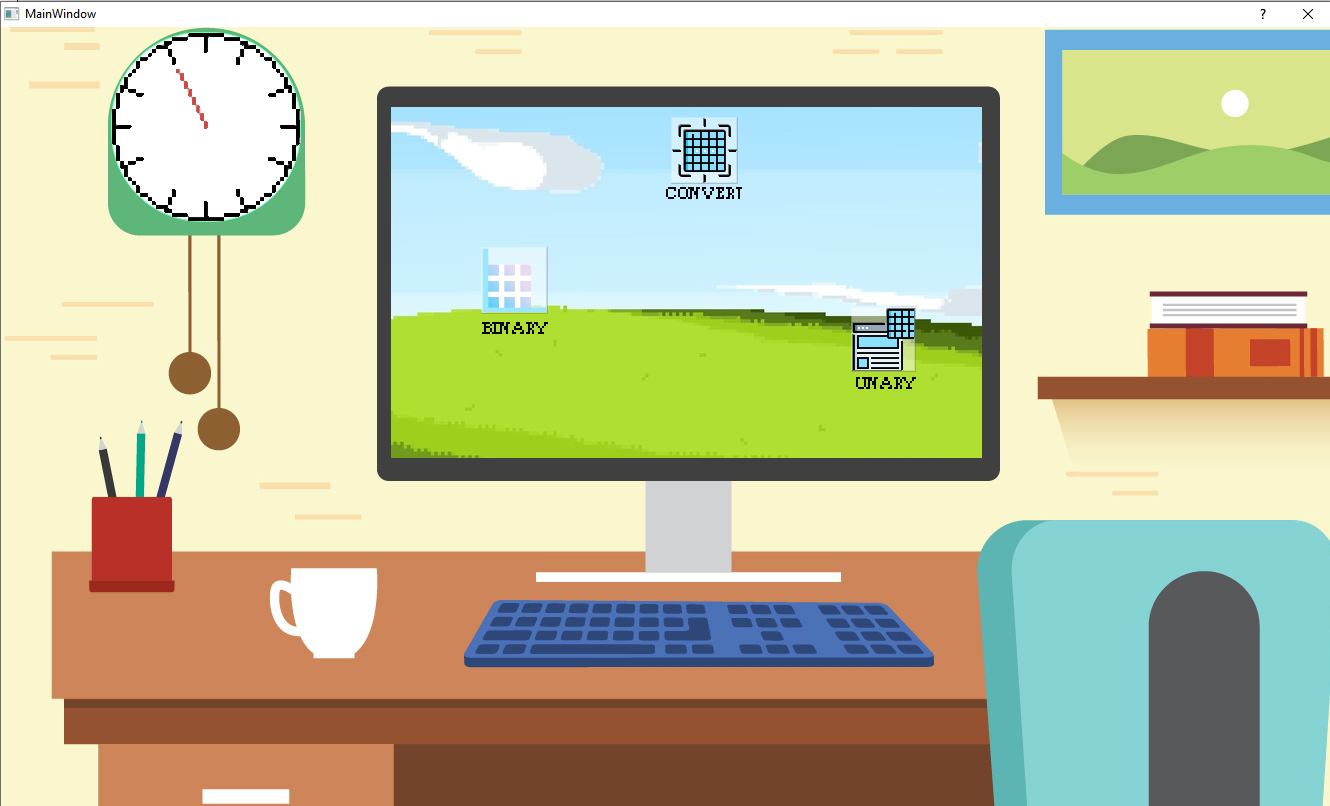
|  |  |  |
| --- | --- | --- |
| 3 | Калькулятор комплексных чисел | Комплексные числа в алгебраической, тригонометрической и экспоненциальной формах.  Реализовать сложение, вычитание, умножение, деление, возведение в целую степень, извлечение квадратного корня (по формуле Муавра), преобразование из одной формы в другую, сброс результата. |

**Анализ задачи**

1. Для решения задачи необходимо:

1.1. Создать четыре класса Qt Designer, для 4 окон работы калькулятора: начальный экран, окно бинарных операций, перевод чисел из одной формы в другую, окно унарных операций.

1.2. Для каждой формы разработать дизайн, прописать для всех модулей Qt Designer сигналы, функции и анимации.

1) MainWindow

В данном окне имеются 3 активных кнопки. Binary, unary и convert открывают окна для бинарных операций, унарных операций и перевода чисел соответственно.

void MainWindow::**on\_exp\_clicked**()

{

expWindow->show();

}

void MainWindow::**on\_trig\_clicked**()

{

trigWindow->show();

}

void MainWindow::**on\_alg\_clicked**()

{

algWindow->show();

}

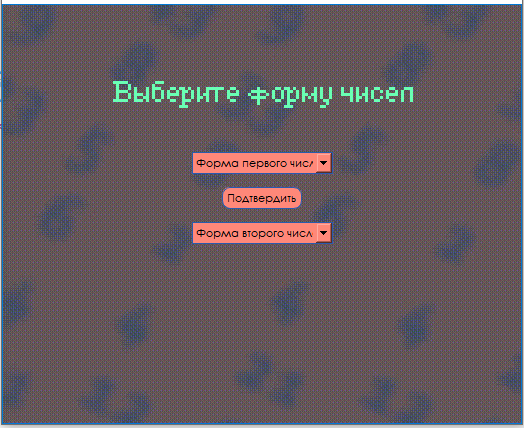
Так же в данном окне, средствами QT, подключена гиф-анимация(движение стрелок часов)

QMovie \*mov = new QMovie(":/clock.gif");

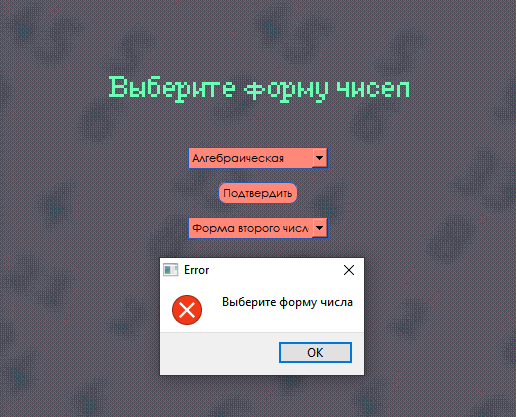
ui->clock->setMovie(*mov*);

mov->start();

2) alg1 (Бинарные действия)



В данном окне присутствуют такие активные модули как:

ComboBox  
Пользователь должен сделать выбор форм комплексных чисел для продолжения работы в программе. Если пользователь не заполняет поля и заполняет только одно, появляется сообщение об ошибке.

if((ui->first->currentIndex()==0)||(ui->second->currentIndex()==0))

{

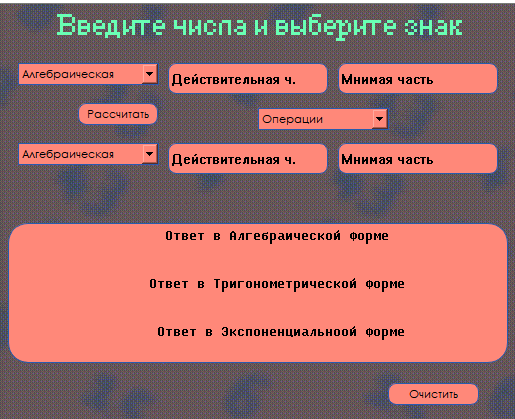
QMessageBox messageBox;

messageBox.critical(0,"Error","Выберите форму чисел");

messageBox.setFixedSize(500,200);

}

После верного ввода данных, с помощью простейшей анимации на экране появляются поля для ввода чисел(LineEdit), ComboBox для выбора знака.



QPropertyAnimation \*oper = new QPropertyAnimation(*ui->oper*, "geometry");

oper->setDuration(500);

oper->setStartValue(QRect(600, 100, 131, 22));

oper->setEndValue(QRect(260, 105, 131, 22));

oper->start();

QPropertyAnimation\* oper1 = new QPropertyAnimation(*ui->oper*);//Код для QPushButton

QGraphicsOpacityEffect\* grEffect2 = new QGraphicsOpacityEffect(*ui->oper*);

ui->oper->setGraphicsEffect(*grEffect2*);

oper1->setTargetObject(*grEffect2*);

oper1->setPropertyName("opacity");

oper1->setStartValue(0.0);

oper1->setEndValue(1.0);

oper1->setDuration(1000);

oper1->start();

Проверка на правильность ввода данных, осуществляется с помощью валидаторов, пользователь просто не может ввести в поля знаки, кроме положительных или отрицательных цифр.

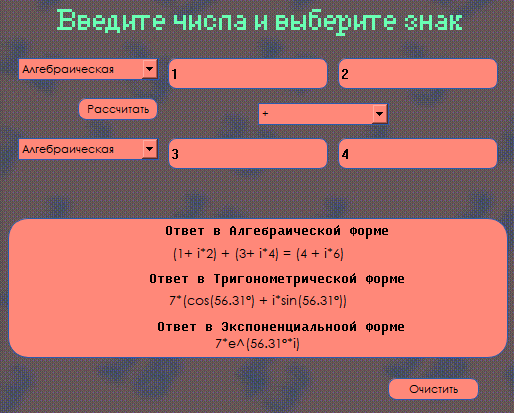
ui->real1->setValidator(new QRegExpValidator(QRegExp("[-1-9][0-9]{1,11}\\.[0-9]{1,5}|[0-9]\\.[0-9]{1,5}")));

ui->imag1->setValidator(new QRegExpValidator(QRegExp("[-1-9][0-9]{1,11}\\.[0-9]{1,5}|[0-9]\\.[0-9]{1,5}")));

ui->real2->setValidator(new QRegExpValidator(QRegExp("[-1-9][0-9]{1,11}\\.[0-9]{1,5}|[0-9]\\.[0-9]{1,5}")));

ui->imag2->setValidator(new QRegExpValidator(QRegExp("[-1-9][0-9]{1,11}\\.[0-9]{1,5}|[0-9]\\.[0-9]{1,5}")));

После ввода данных и выбора знака, пользователь получает ответ сразу во всех трех формах комплексного числа



if(ui->first->currentIndex()==1 && ui->second->currentIndex()==1){

if(ui->oper->currentIndex()==1)

{

ui->alg\_text->setText("(" + Qreal1 + "+ i\*"+ Qimag1+ ") + (" + Qreal2 + "+ i\*"+ Qimag2+ ") = (" + newText1 + " + i\*" + newText3 + ")");

}

if(ui->oper->currentIndex()==2)

{

ui->alg\_text->setText("(" + Qreal1 + "+ i\*"+ Qimag1+ ") - (" + Qreal2 + "+ i\*"+ Qimag2+ ") = (" + newText1 + " + i\*" + newText3 + ")");

}

if(ui->oper->currentIndex()==3)

{

ui->alg\_text->setText("(" + Qreal1 + "+ i\*"+ Qimag1+ ") \* (" + Qreal2 + "+ i\*"+ Qimag2+ ") = (" + newText1 + " + i\*" + newText3 + ")");

}

if(ui->oper->currentIndex()==4)

{

ui->alg\_text->setText("(" + Qreal1 + "+ i\*"+ Qimag1+ ") / (" + Qreal2 + "+ i\*"+ Qimag2+ ") = (" + newText1 + " + i\*" + newText3 + ")");

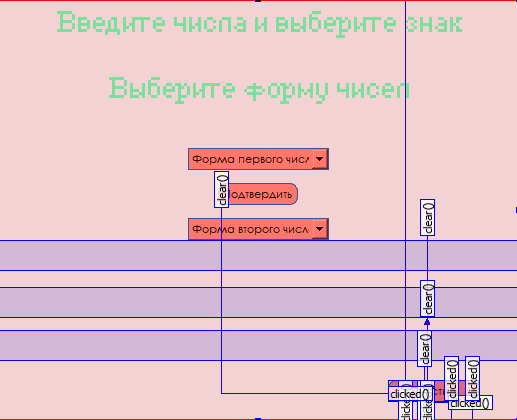
}

ui->AAA1->setText(newTextAA +"\*e^" + "("+Qatan+"°\*i)");

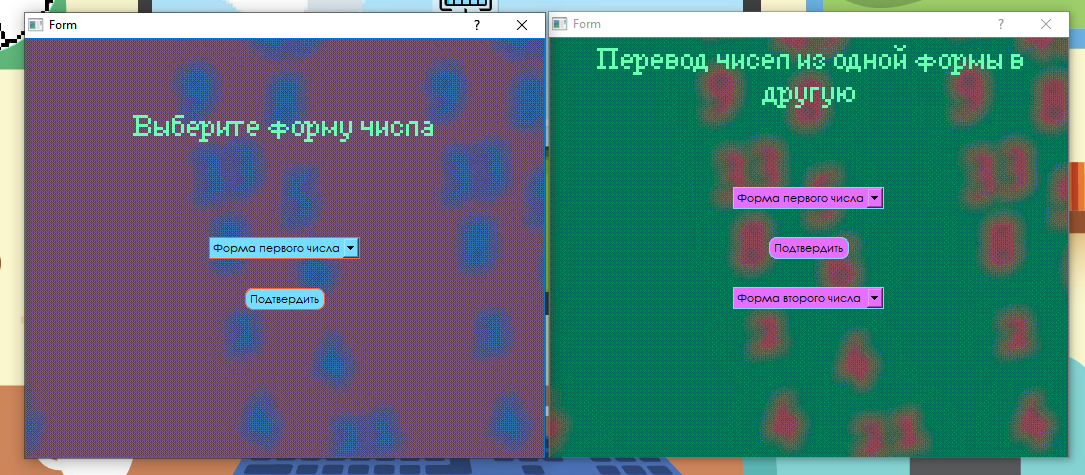
ui->atan->setText(newTextAA +"\*(" +"cos"+ "("+Qatan+"°) +" + " i\*sin(" + Qatan + "°))");

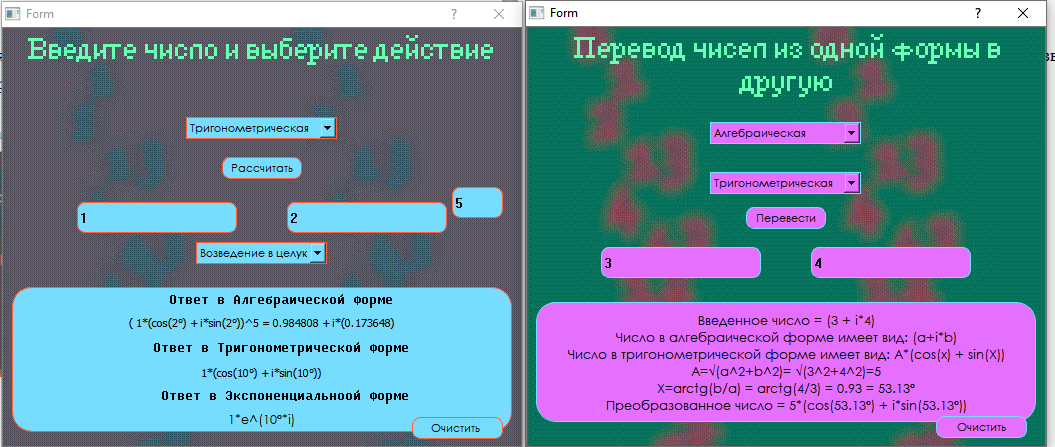
}

При нажатии кнопки очистить, все поля, заполненные ранее, очищаются, данное действие реализовано через привязку слотов в UI форме



**3)** Техническая реализация окон Unary и Convert ничем не отличается от реализации первого окна, отличия имеются только в операциях, проводимых над числами



****

Все поля ввода ограничены с помощью валидаторов, а ComboBox так же не могут остаться незаполненными

**4)** В качестве фоновых изображений для каждого окна были выбраны созданные самостоятельно гиф-анимации.

QMovie \*mov = new QMovie(":red.gif");

ui->fon->setMovie(*mov*);

mov->start();

QMovie \*mov = new QMovie(":/orange/orange.gif");

ui->back->setMovie(*mov*);

mov->start();

**2.** В ходе работы были использованы следующие типы данных:

**2.1.** Переменные типа double, для создания комплексных чисел, и переменные типа complex, для реализации требуемых операций.

double a,b,realCH,imagCH,sincos;

complex <double> c(a,b);

**3.** Для решения задачи, данные были представлены в следующем виде:

**3.1.** Ввод данных производится с помощью полей QLineEdit:

**3.2.** Вывод данных производится с помощью функции:

ui->lineEdit->setText(Qstring);

**3.3.** Преобразование типа QString к типу double производится с помощью функции:

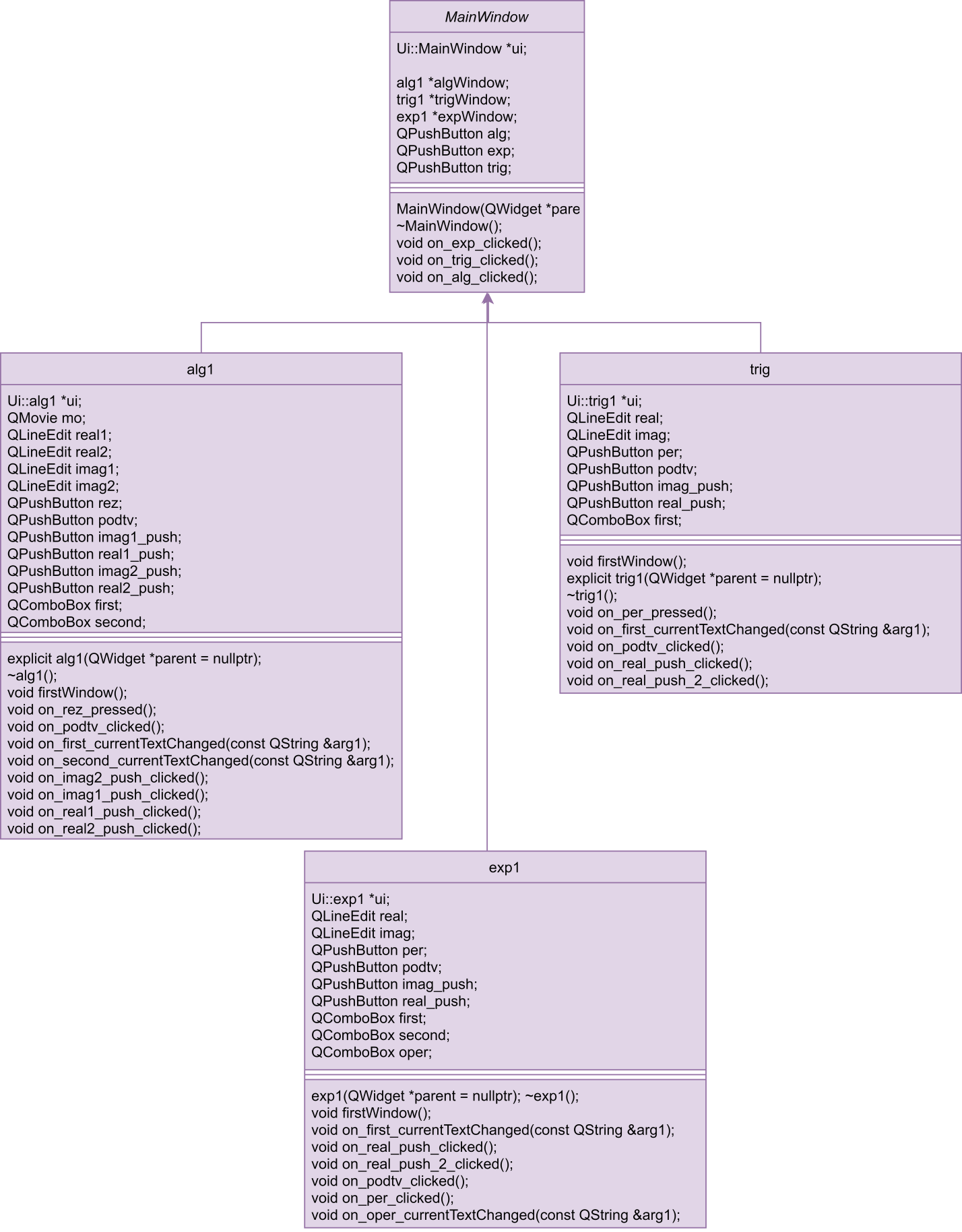
QString.toDouble;

**3.4** Выбор формы числа и знаков реализован с помощью текстового меню ComboBox

1. **Достоинства программы**

К достоинствам проекта можно отнести простоту и удобство интерфейса, многофункциональность программы. Реализация программы в нескольких окнах позволяет одновременно совершать несколько действий над одними числами, что заметно ускоряет работу

**UML-диаграмма**

****