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 Творческой работе №1**

Дисциплина: основы алгоритмизации и программирования

Тема: Разработка калькулятора

Вариант 3

Выполнил работу

студент группы ИВТ-20-2б

Солдатов А.П.

Проверил

Доцент кафедры ИТАС

Полякова О.А.

Пермь, 2021

Постановка задачи

Разработать алгоритм калькулятора в соответствии с заданием.

Реализовать алгоритм в виде программы на алгоритмическом языке С++.

Разработать интерфейс средствами QT.

Комплексные числа в алгебраической, тригонометрической и экспоненциальной формах.

Сложение, вычитание, умножение, деление, возведение в целую степень, извлечение квадратного корня (по формуле Муавра), преобразование из одной формы в другую, отмена последней операции, сброс результата.

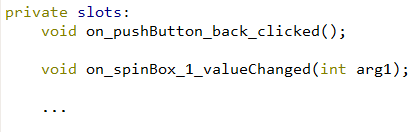
Анализ задачи

Калькулятор основан на функциях и классах.

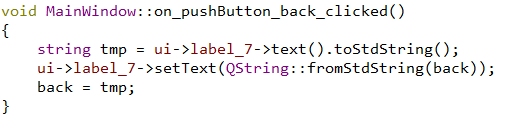
При вызове определенной функции, советующей выбранной пользователем операцией, начинается её выполнение. Связано это с отсутствием необходимости в добавлении пользовательских классов. Например, вызов операции сложения подразумевает ввод пары чисел, состоящих из действительной и мнимой части. В процессе выполнения функции используются глобальные переменные, хранящие в себе действительную и мнимую части обоих чисел, и передаются в локальные, где с ними происходят соответствующие операции.

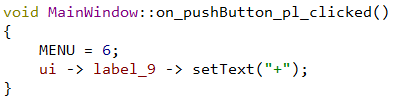
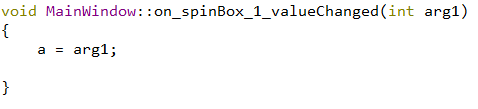
Определить какие операции должны быть выполнены по заданию:

Создание приватных слотов для каждой кнопки и каждого spinbox:

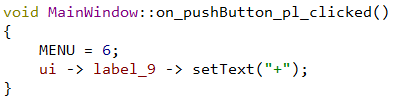


Создание функций для каждой кнопки и каждого spinbox:



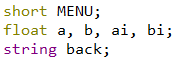


Создание функции для кнопки «=», в которой описывается выбор нужной операции:



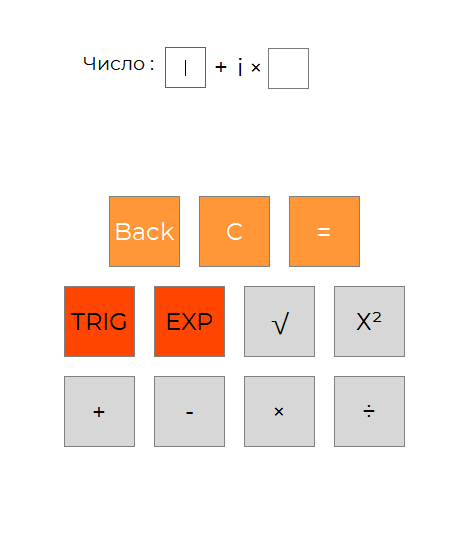
Для решения задачи используются глобальные переменные:

* 4 переменные, типа float, для хранения действительной и мнимой частей чисел.
* 1 переменная, типа short, для хранения номера операции.
* 1 строка для хранения предыдущего значения, необходимого для реализации кнопки back.



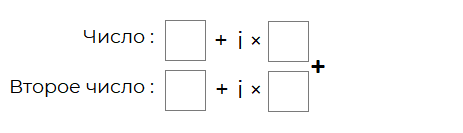
Анализ задачи со стороны дизайна:

Изначально пользователю доступны все операции на выбор:

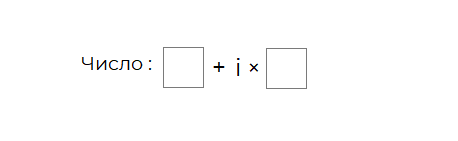


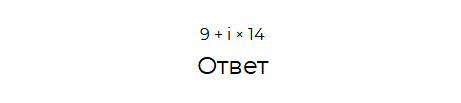
После выбора операции появляются окна для ввода значений. Тип окон зависит от выбранной операции:

Например, при вызове операции сложения появляются окна для ввода второго комплексного числа:

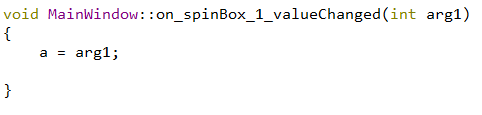


При вызове операции вычисление экспоненциальной формы числа пропадает окно для ввода второй градусной меры:

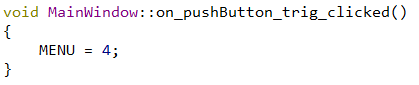


После ввода всех значений, пользователь нажимает на кнопку “=” и появляется окно для вывода ответа:  


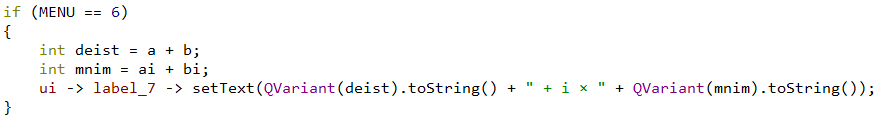
Ввод данных осуществляется посредством виджетов QSpinBox



Также ввод данных в переменную MENU, которая отвечает за выбранную операцию, происходит посредством фиксирования нажатия на кнопку



Вывод данных осуществляется посредством виджетов Label



В калькуляторе используются ограничители вводимых данных. Это связано с областью определения функций арксинуса и арккосинуса. Ограничители настроены свойствами соответствующих QDouble Spin Box-ов.

Код основных методов:

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include <iomanip>

#include <string>

#include <iostream>

using namespace std;

short MENU;

float a, b, ai, bi;

string back;

MainWindow::MainWindow(QWidget \*parent)

: QMainWindow(parent)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

ui -> label\_3 -> hide();

ui -> label\_4 -> hide();

ui -> label\_6 -> hide();

ui -> label\_7 -> hide();

ui -> label\_8 -> hide();

ui -> spinBox\_3 -> hide();

ui -> spinBox\_4 -> hide();

ui -> spinBox\_1 -> *clear*();

ui -> spinBox\_2 -> *clear*();

ui -> spinBox\_3 -> *clear*();

ui -> spinBox\_4 -> *clear*();

}

MainWindow::~*MainWindow*()

{

delete ui;

}

void MainWindow::on\_pushButton\_back\_clicked()

{

string tmp = ui->label\_7->text().toStdString();

ui->label\_7->setText(QString::fromStdString(back));

back = tmp;

}

void MainWindow::on\_pushButton\_C\_clicked()

{

ui -> label\_7 -> hide();

ui -> label\_8 -> hide();

ui -> label\_9 -> hide();

ui -> label\_6 -> hide();

ui -> label\_3 -> hide();

ui -> label\_4 -> hide();

ui -> spinBox\_3 -> hide();

ui -> spinBox\_4 -> hide();

}

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

{

MENU = 4;

ui -> label\_9 -> setText("");

}

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

{

MENU = 5;

ui -> label\_9 -> setText("");

}

void MainWindow::on\_pushButton\_pl\_clicked()

{

MENU = 6;

ui -> label\_9 -> setText("+");

}

void MainWindow::on\_pushButton\_mi\_clicked()

{

MENU = 7;

ui -> label\_9 -> setText("−");

}

void MainWindow::on\_pushButton\_um\_clicked()

{

MENU = 8;

ui -> label\_9 -> setText("×");

}

void MainWindow::on\_pushButton\_de\_clicked()

{

MENU = 9;

ui -> label\_9 -> setText("÷");

}

void MainWindow::on\_pushButton\_sq\_clicked()

{

MENU = 10;

ui -> label\_9 -> setText("");

}

void MainWindow::on\_pushButton\_st\_clicked()

{

MENU = 11;

ui -> label\_9 -> setText("");

}

void MainWindow::on\_pushButton\_an\_clicked()

{

back = ui->label\_7->text().toStdString();

//сложение

if (MENU == 6)

{

int deist = a + b;

int mnim = ai + bi;

ui -> label\_7 -> setText(QVariant(deist).toString() + " + i × " + QVariant(mnim).toString());

}

//вычитание

if (MENU == 7)

{

int deist = a - b;

int mnim = ai - bi;

ui -> label\_7 -> setText(QVariant(deist).toString() + " + i × " + QVariant(mnim).toString());

}

//умножение

if (MENU == 8)

{

int mnim = a \* bi + b \* ai;

int deist = a \* b - ai \* bi;

ui -> label\_7 -> setText(QVariant(deist).toString() + " + i × " + QVariant(mnim).toString());

}

//деление

if (MENU == 9)

{

float deist = (a \* b + ai \* bi)/(pow(b,2) + pow(bi,2));

float mnim = (b \* ai - a \* bi)/(pow(b,2) + pow(bi,2));

ui -> label\_7 -> setText(QVariant(deist).toString() + " + i × " + QVariant(mnim).toString());

}

//возведение в степень

if (MENU == 11)

{

int deist = pow(a,2);

int mnim = pow(ai,2);

ui -> label\_7 -> setText(QVariant(deist).toString() + " + i × " + QVariant(mnim).toString());

}

//перевод в тригонометрическую форму

if (MENU == 4)

{

double abs = sqrt(pow(a,2) + pow(ai,2));

ui -> label\_7 -> setText(QVariant(abs).toString() + " × cos(arctg(" + QVariant(ai).toString() + "/" + QVariant(a).toString() + ") + i × (arctg(" + QVariant(ai).toString() + "/" + QVariant(a).toString() + ")))");

}

//перевод в экспоненциальную форму

if (MENU == 5)

{

double abs = sqrt(pow(a,2) + pow(ai,2));

double arg = (atan(ai/a) \* 180)/acos(-1);

ui -> label\_7 -> setText(QVariant(abs).toString() + " + e^(i × " + QVariant(arg).toString() + ")");

}

//вычисление корня

if(MENU == 10)

{

}

}

void MainWindow::on\_spinBox\_1\_valueChanged(int arg1)

{

a = arg1;

}

void MainWindow::on\_spinBox\_2\_valueChanged(int arg1)

{

ai = arg1;

}

void MainWindow::on\_spinBox\_3\_valueChanged(int arg1)

{

b = arg1;

}

void MainWindow::on\_spinBox\_4\_valueChanged(int arg1)

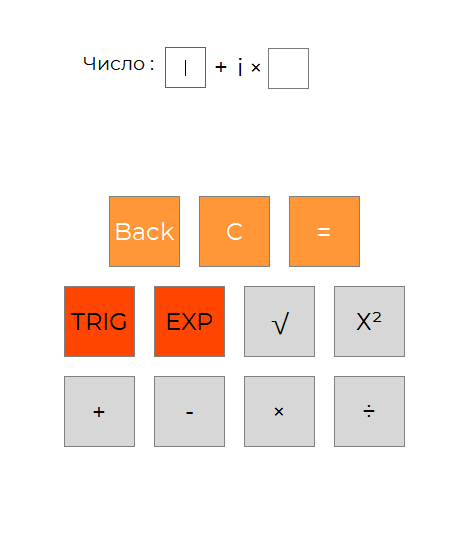
{

bi = arg1;

}

Работа калькулятора:

Калькулятор в первичном состоянии:



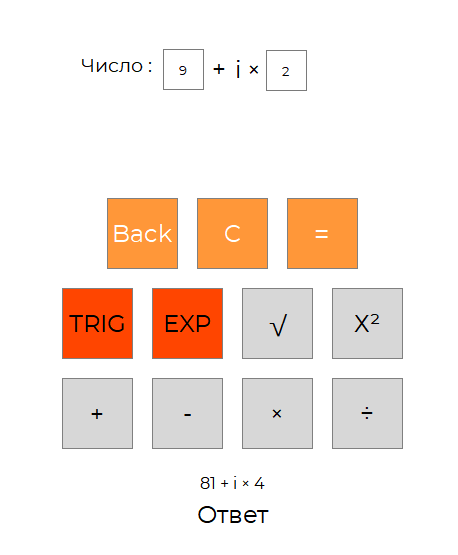
Калькулятор при выполнении операции сложения (ответом является число в алгебраической форме):



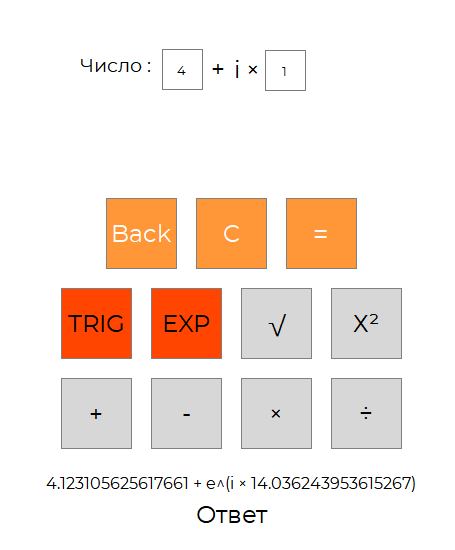
Калькулятор при выполнении операции умножения (ответом является число в алгебраической форме):



Калькулятор при выполнении операции возведения числа во ворую степень (ответом является число в алгебраической форме):



Калькулятор при выполнении операции перевода числа из алгебраической формы в экспоненциальную (ответом является число в экспоненциальной форме):



UML Диаграмма:

