Министерство образования и науки Российской Федерации

Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский**

**политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по творческой работе часть 1**

Дисциплина: «Основы алгоритмизации и программирования»

Тема: Калькулятор польской записи

Вариант 18

Выполнил:

Студент группы РИС-20-1б

Тараканов Д. М.

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

Пермь

2021 год

**Постановка задачи**

1. Разработать калькулятор с графическим интерфейсом, реализованным средствами Qt, который вычисляет выражения в польской инверсной записи.
2. Калькулятор на вход получает два действительных числа – операнда, а затем вводится знак операции и выдается ответ.

Допустимые операции:

1. Сложение
2. Вычитание
3. Умножение
4. Деление
5. Возведение в целую степень
6. На панели калькулятора отображаются исходные данные и результаты, операции и служебные клавиши, клавиши изменения знака числа. Калькулятор должен позволять корректировать последнее введенное число и выдавать сообщения об ошибках

**Анализ задачи**

1. Определить какие действия необходимо выполнить для решения задачи:

* Разработать интерфейс программы средствами qt
* Реализовать добавление цифр, изменения знака числа, удаление последнего введённого элемента при помощи нажатии соответствующих кнопок
* Реализовать вывод результата или сообщения об ошибке при нажатии на кнопки операций.

1. С какими типами данных действие надо сделать, в каком виде эти данные будут представлены:

* Экземпляры всех элементов интерфейса хранятся в переменной ui.

Ui::PolishNotation\* ui;

* Переменные счетчик кол-ва точек в числе, кол-ва цифр в числе и максимальное кол-во цифр.

int Dot\_count = 0, number\_count = 0;

*const* int max\_number\_count = 6;

* Переменная отвечающая за само число.

QString value;

1. Для решения задачи данные были представлены в виде результата арифметических операций типа double
2. Реализация операций ввода и вывода данных:

* Ввод элементов реализован при помощи добавления символов в строку результата и переменную value. На примере нажатия на кнопку “0” рассматривается как добавляются цифры в число.

void PolishNotation::**on\_Btn\_0\_clicked**()

{

*if*(flag==*true*)

{

ui->L\_Result\_out->setText("");

flag = 0;

}

*if*(number\_count < max\_number\_count)

{

ui->L\_Result\_out->setText(ui->L\_Result\_out->text() + '0');

value += '0';

number\_count++;

}

*else*

{

ui->L\_Error->setText("Error: Превышение максимального кол-ва разрядов в числе!");

}

}

* Вывод элементов реализован при помощи добавления строк в поля вывода результатов и привычной записи последней арифметической операции. На примере нажатия на кнопку “+” рассматривается как происходит вывод результатов вычислений

void PolishNotation::**on\_Btn\_Plus\_clicked**()

{

*if*(st.size() > 1)

{

re\_change('+');

ui->L\_PN->setText(ui->L\_PN->text() + "+ ");

double arg1, arg2 = st.top(), result;

st.pop();

arg1 = st.top();

st.pop();

result = arg1 + arg2;

st.push(result);

QString output = QString::number(result);

ui->L\_Result\_out->setText(output);

flag = 1;

}

*else*

{

ui->L\_Error->setText("Error: Недостаточно операндов!");

}

}

void PolishNotation::**re\_change**(char op)

{

stack<double> temp\_stack;

temp\_stack = st;

QString elem = QString::number(temp\_stack.top());

temp\_stack.pop();

*switch* (op)

{

*case* '+': ui->L\_Stack->setText(QString::number(temp\_stack.top())+ " + " + elem + " = "); *break*;

*case* '-': ui->L\_Stack->setText(QString::number(temp\_stack.top())+ " - " + elem + " = "); *break*;

*case* '/': ui->L\_Stack->setText(QString::number(temp\_stack.top())+ " / " + elem + " = "); *break*;

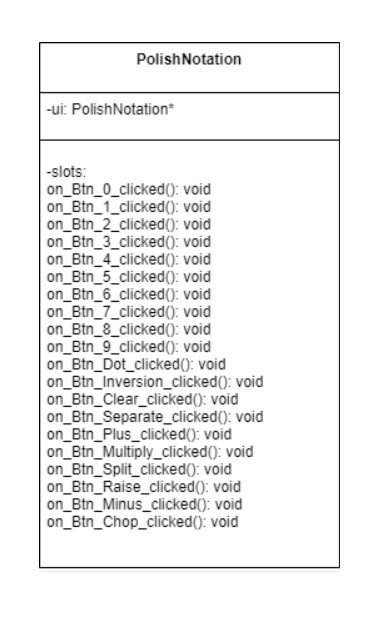
*case* '\*': ui->L\_Stack->setText(QString::number(temp\_stack.top())+ " \* " + elem + " = "); *break*;

*case* '^': ui->L\_Stack->setText(QString::number(temp\_stack.top())+ " ^ " + elem + " = "); *break*;

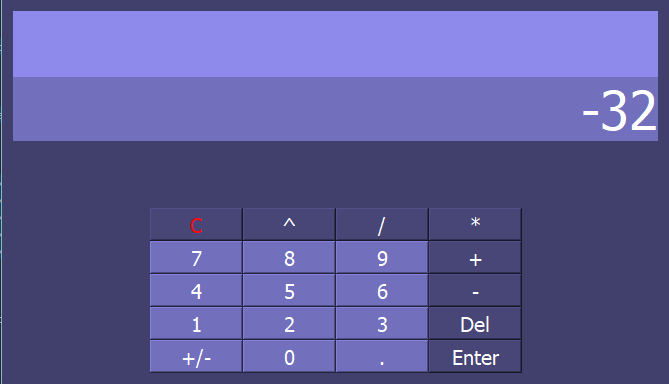
}

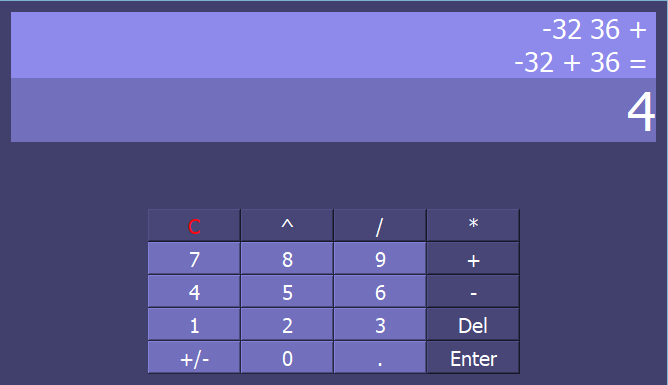
}

**UML диаграмма классов**



**Скриншоты тестов**

****

****

****