

Создание проекта с графическим интерфейсом с использованием библиотеки Qt

6.1. Постановка задачи.

В настоящей лабораторной работе необходимо разработать приложение, обладающее графическим интерфейсом с использованием средств библиотеки Qt. В приложении должно быть предусмотрено меню и панель инструментов. Ниже сформулирован ряд задач. Предлагаемые задачи достаточно просты, что позволяет сосредоточить внимание на вопросах организации интерфейса.

В настоящей лабораторной работе предлагаются две группы задач. в первой группе задач требуется разработать одну из разновидностей калькулятора. Ко второй группе задач относятся задачи, связанные с манипуляцией с содержимым текстового однострочного редактора.

6.2. Варианты заданий.

Первая группа заданий (задач). Калькуляторы.

Задача 1

На форме находятся три компонента типа LineEdit. Два из них служат для хранения операндов, третий – фиксации результата вычислений. Обеспечить выполнение четырех обычных арифметических операций. Организовать меню и панель инструментов.

Задача 2

На форме находятся два компонента типа LineEdit. Один из них служит для хранения операндов, второй – фиксации результата вычислений. Оба операнда заносятся в одно окно редактирования. Обеспечить выполнение четырех обычных арифметических операций. Организовать меню и панель инструментов.

Задача 3

На форме находятся два компонента типа LineEdit и один TextEdit. Компоненты LineEdit служат для хранения операндов, а TextEdit –вывода результатов вычислений. Результаты вычислений выводятся в виде таблицы. Назначения колонок: номер эксперимента, значение 1-го операнда, значение 2-го операнда, знак операции и, наконец – результат вычислений. Обеспечить выполнение четырех обычных арифметических операций. Организовать меню и панель инструментов.

Задача 4

На форме находятся два компонента: LineEdit и TextEdit. Компонент LineEdit служит для хранения операндов, а TextEdit –вывода результатов вычислений. В LineEdit вводятся оба операнда, разделенные одним или

несколькими пробелами. Результаты вычислений выводятся в виде таблицы. Назначения колонок: номер эксперимента, значение 1-го операнда, значение 2-го операнда, знак операции и, наконец – результат вычислений. Обеспечить выполнение четырех обычных арифметических операций. Организовать меню и панель инструментов

Задача 5

На форме находится один компонент LineEdit. В этот компонент вводятся исходные операнды, а после выполнения заданной операции размещается полученный результат вычислений. Исходные данные друг от друга отделяются пробелами. Обеспечить выполнение четырех обычных арифметических операций. Организовать меню и панель инструментов.

Вторая группа заданий (задач). Манипуляция содержимым компонентов LineEdit.

Рассмотрим возможные операции и введем некоторые обозначения. Будем под `strEdt1` понимать строку, которая является содержимым однострочного редактора с именем `Edt1`. Начнем с возможных операций с содержимым редактора.

1. `clear` – очистка окна редактора. Операция имеет один аргумент. Например, `clear(strEdtLeft)` – очистка окна редактора `EdtLeft`.
2. `clearAll` – очистка всех расположенных на форме редакторов.
3. `reverse` – обращение строки. Операция имеет один аргумент. Например, `reverse(strEdtRight)` – обращение строки в редакторе `EdtRight`.
4. `reverseAll` – обращение содержимого всех редакторов.
5. `copy` – копирование строк. Операция имеет два аргумента. Например, `copy(strEdtLeft, strEdtRight)` – содержимое редактора `EdtRight` копируется в окно редактора `EdtLeft`. Содержимое окна `EdtRight` остается неизменным.
6. `move` – перенос строки. Операция имеет два аргумента. Например, `move(strEdtLeft, strEdtRight)` – строка, являющаяся содержимым редактора `strEdtRight` переносится в окно редактора `EdtLeft`. Окно редактора `EdtRight` – очищается.
7. `concat` – объединение строк. Операция имеет два аргумента. Например, `concat(strEdtLeft, strEdtRight)` – строка, являющаяся содержимым редактора `strEdtRight` объединяется в окне редактора `EdtLeft` с его содержимым. Содержимое окна редактора `EdtRight` остается неизменным.

Задача 6

На форме имеются два компонента LineEdit, горизонтально скомпонованные. Обеспечить выполнение следующих операций:

- `clear(strEdtLeft)`

- clear(strEdtRight).
- clearAll().
- swap(strEdtLeft, strEdtRight).
- copy(strEdtLeft, strEdtRight).
- copy(strEdtRight, strEdtLeft).

Задача 7

На форме имеются два компонента LineEdit, горизонтально скомпонованные. Обеспечить выполнение следующих операций:

- reverse(strEdtLeft)
- reverse(strEdtRight).
- reverseAll().
- swap(strEdtLeft, strEdtRight).
- copy(strEdtLeft, strEdtRight).
- copy(strEdtRight, strEdtLeft).

Задача 8

На форме имеются два компонента LineEdit, горизонтально скомпонованные. Обеспечить выполнение следующих операций:

- clear(strEdtLeft)
- clear(strEdtRight).
- clearAll().
- swap(strEdtLeft, strEdtRight).
- move(strEdtLeft, strEdtRight).
- move(strEdtRight, strEdtLeft).

Задача 9

На форме имеются два компонента LineEdit, горизонтально скомпонованные. Обеспечить выполнение следующих операций:

- reverse(strEdtLeft)
- reverse(strEdtRight).
- reverseAll().
- swap(strEdtLeft, strEdtRight).
- concat(strEdtLeft, strEdtRight).
- concat(strEdtRight, strEdtLeft).

Задача 10

На форме имеются два компонента LineEdit, горизонтально скомпонованные. Обеспечить выполнение следующих операций:

- clear(strEdtLeft)

- `clear(strEdtRight)`.
- `clearAll()`.
- `copy(strEdtLeft, strEdtRight)`.
- `concat(strEdtLeft, strEdtRight)`.
- `move(strEdtRight, strEdtLeft)`.

Задача 11

На форме имеются два компонента `LineEdit`, вертикально скомпонованные. Обеспечить выполнение следующих операций:

- `clear(strEdtUpper)`
- `clear(strEdtLower)`.
- `clearAll()`.
- `move(strEdtUpper, strEdtLower)`.
- `copy(strEdtUpper, strEdtLower)`.
- `copy(strEdtLower, strEdtUpper)`.

Задача 12

На форме имеются два компонента `LineEdit`, вертикально скомпонованные. Обеспечить выполнение следующих операций:

- `reverse(strEdtUpper)`
- `reverse(strEdtLower)`.
- `reverseAll()`.
- `concat(strEdtUpper, strEdtLower)`.
- `concat(strEdtLower, strEdtUpper)`.
- `copy(strEdtUpper, strEdtLower)`.

Задача 13

На форме имеются два компонента `LineEdit`, вертикально скомпонованные. Обеспечить выполнение следующих операций:

- `clear(strEdtLeft)`
- `clear(strEdtRight)`.
- `clearAll()`.
- `move(strEdtUpper, strEdtLower)`.
- `move(strEdtLower, strEdtRight)`.
- `swap(strEdtRight, strEdtLeft)`.

Задача 14

На форме имеются два компонента `LineEdit`, вертикально скомпонованные. Обеспечить выполнение следующих операций:

- reverse(strEdtLower)
- reverse(strEdtUpper).
- reverseAll().
- swap(strEdtLower, strEdtUpper).
- concat(strEdtLower, strEdtUpper).
- concat(strEdtUpper, strEdtLower).

Задача 15

На форме имеются два компонента QLineEdit, вертикально скомпонованные. Обеспечить выполнение следующих операций:

- clear(strEdtLower)
- clear(strEdtUpper).
- clearAll().
- copy(strEdtLower, strEdtUpper).
- concat(strEdtLower, strEdtUpper).
- move(strEdtLower, strEdtUpper)

6.3. Порядок выполнения лабораторной работы

Рассмотрим возможный порядок выполнения настоящей лабораторной работы.

1. Вначале следует, воспользовавшись мастером системы Qt Creator, создать заготовку проекта GUI приложения. Основное окно приложения целесообразно породить от класса QMainWindow библиотеки Qt. Последнее позволит упростить построение меню приложения и его панели инструментов.
2. Выведя на экран форму приложения, разместите на ней необходимые визуальные компоненты.
3. Установить свойства размещенных на форме компонентов.
4. Определить с помощью Редактора действий среды Qt Creator требуемые для работы Action.
5. Используя созданную Редактором действий систему Action, постройте элементы меню и панели инструментов.
6. Воспользовавшись контекстным меню для Action, создайте заготовки обработчиков.
7. Напишите реализацию для обработчиков.
8. Выполните отладку созданного приложения.