*Федеральное государственное бюджетное образовательное учреждение* *высшего образования*

|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | ***«Московский государственный технический университет  имени Н.Э. Баумана»***  ***(национальный исследовательский университет)*** |

ФАКУЛЬТЕТ \_\_\_\_ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_\_Компьютерные Системы и сети (ИУ6)\_\_\_\_\_\_\_\_\_\_\_

**Отчет**

**по лабораторной работе № 9**

**Название лабораторной работы:** Программирование с использованием Qt.

**Вариант № 23**

**Дисциплина:** Объектно-ориентированное программирование

Студент гр. ИУ6-26(22) **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О. Фамилия)

Москва, 2018

**Часть 1**

**Задание:** Создать простое приложение, в котором предлагается ввести возраст с использованием одного из трех вариантов ввода:

1. непосредственного ввода числа,
2. посредством стрелок (элемент типа QSpinBox), последовательно увеличивающих или уменьшающих значение,
3. с помощью специального ползунка (слайдера – элемент типа QSlider).

Кроме того, задано верхнее ограничение вводимого возраста, что должно быть корректно отработано слайдером в крайних положениях, а изменение значения любым способом должно синхронизировать его положение.

**Код программы:**



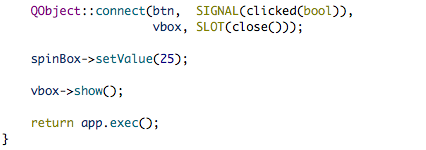
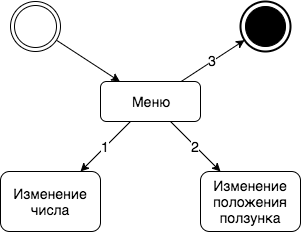


Рисунок 9.1*.1* — Код основной программы для ЛР9

2

**Диаграмма состояний интерфейса:**

1. Изменение положения ползунка
2. Изменение числа
3. Нажатие на кнопку “Выход”

Рисунок 9.1*.2* — Диаграмма состояний интерфейса для ЛР9

# Тестирование:

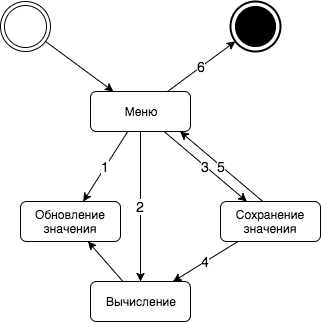
Рисунок 9.1*.3* — Тестирование для ЛР9

# Вывод: Согласно заданию была создана диаграмма состояний интерфейса (Рисунок 9.1.2), и по ней была разработана программа, изменяющее число при любом из трех способов его ввода. Проведено тестирование (Рисунок 9.1.3), показавшее, что программа работает корректно.

3

**Часть 2**

**Задание:** Создание программы­калькулятора.

**Диаграмма состояний интерфейса:**

1. Ввод новых цифр
2. Нажатие на унарную операцию
3. Нажатие на бинарную операцию
4. Нажатие на “=” или нажатие на другую операцию
5. Нажатие на сброс
6. Нажатие на “X”

Рисунок 9.2*.1* — Диаграмма состояний интерфейса для ЛР9

**Код программы:**

*Файл main.cpp:*

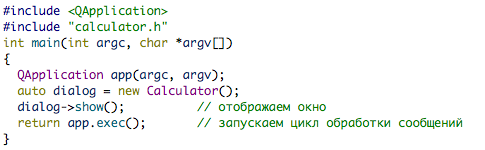
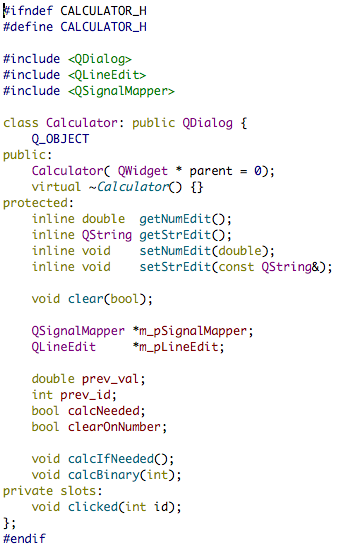


Рисунок 9.2*.2* — Код основной программы для ЛР9

4

*Файл calculator.h:*

**

*Файл calculator.cpp:*

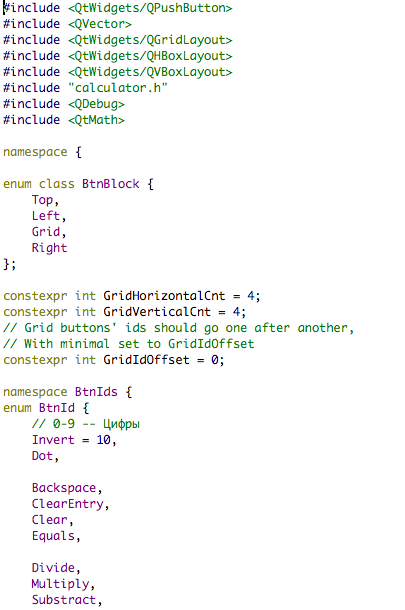


Рисунок 9.2*.2* — Продолжение

5

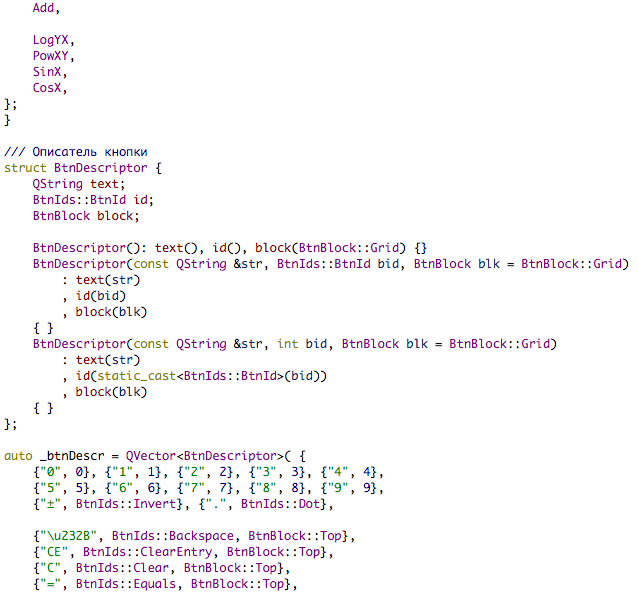


Рисунок 9.2*.2* — Продолжение

6



Рисунок 9.2*.2* — Продолжение

7

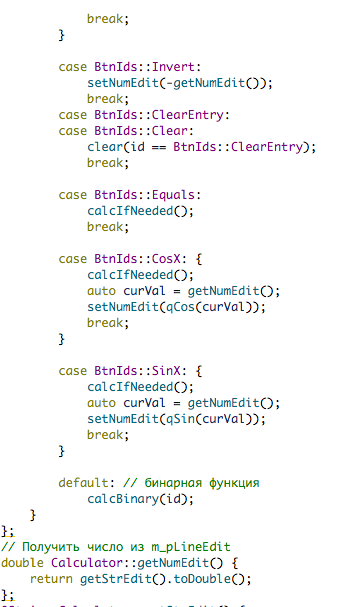




Рисунок 9.2*.2* — Продолжение

8



Рисунок 9.2*.2* — Продолжение

# Тестирование:

Рисунок 9.2*.3* — Тестирование программы для ЛР9

# Вывод: Была создана диаграмма состояний интерфейса калькулятора (Рисунок 9.2.1), по которой была разработана программа с использованием библиотеки Qt. Произведено тестирование, показавшее, что программа работает корректно (Рисунок 9.2.3).

9

**Часть 3**

**Задание:** Разработать приложение, имеющее строку ввода данных, кнопку запуска преобразования и текстовое поле, предназначенное только для отображения информации. Любой текст строки ввода должен отображаться в текстовом поле сразу после завершения ввода. В начале строки должна быть вставлена пометка «input:». При нажатии кнопки преобразования строка ввода должна быть преоб­ разована либо в верхний регистр, либо в нижний противоположно тому, что производи­ лось при предыдущем нажатии кнопки.

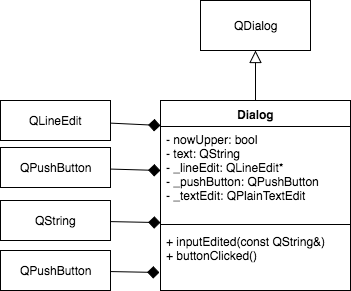
**Диаграмма интерфейсных классов:**

Рисунок 9.3.1 — Диаграмма интерфейсных классов для ЛР9

# Диаграмма состояний интерфейса:

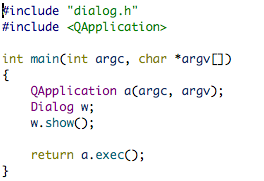
1. Изменение текста
2. Нажатие на “Convert”
3. Нажатие на «X»

Рисунок 9.3.2 — Диаграмма состояний интерфейса для ЛР9

10

**Код программы:**

*Файл main.cpp:*



*Файла dialog.h:*

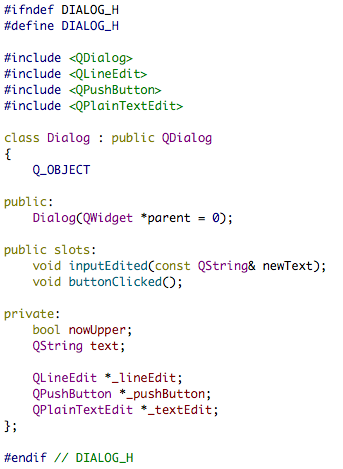


Рисунок 9.3.3 — Код основной программы для ЛР9

11

*Файла dialog.cpp:*

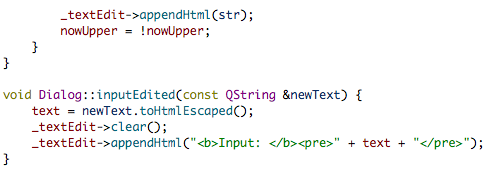


Рисунок 9.3.3 — Продолжение

12

# Тестирование:

Рисунок 9.3.4 — Тестирование программы для ЛР9

# Вывод:Согласно заданию были созданы диаграмма интерфейсных классов (Рисунок 3.1) и диаграмма состояний интерфейса (Рисунок 3.2). По ним была разработана программа с использованием библиотеки Qt. Во время тестирования программа показала ожидаемые результаты (Рисунок 3.3).

**Часть 4**

**Задание:** Использование QSplitter. Связывание сигналов и слотов в Qt Designer.

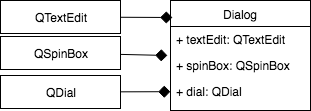
**Диаграмма интерфейсных классов:**

Рисунок 9.4.1 — Диаграмма интерфейсных классов для ЛР9

13

# Код программы:

*Файл main.cpp:*

# /Users/tatiana/Desktop/Снимок экрана 2018-06-29 в 11.16.01.png

*Файла dialog.cpp:*

# /Users/tatiana/Desktop/Снимок экрана 2018-06-29 в 11.16.17.png

Рисунок 9.4.2 — Код основной программы для ЛР9

14

*Файла dialog.h:*

**

Рисунок 9.4.2 — Продолжение

# Тестирование:

Рисунок 9.4.3 — Тестирование программы для ЛР9

15

# Вывод: По заданию была создана диаграмма классов (Рисунок 9.4.1), и по нему была разработана программа на языке C++ с использованием библиотеки Qt в среде Qt Creator с использованием Qt Designer. Проведенное тестирование (Рисунок 9.4.3) показало корректную работу программы.

16