

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ НЕФТИ И
ГАЗА (НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)
ИМЕНИ И.М. ГУБКИНА»

ФАКУЛЬТЕТ КОМПЛЕКСНОЙ БЕЗОПАСНОСТИ ТЭК
КАФЕДРА БЕЗОПАСНОСТИ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ

Лабораторная работа №1

по дисциплине «Специализированные языки и технологии
программирования»

на тему «Основы работы с виджетами в Qt»

Выполнил студент:

группы КС-22-03

Абдуллин Тагир Ренатович

Преподаватель:

Греков Владимир Сергеевич

Москва, 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
ХОД РАБОТЫ	4
КОНТРОЛЬНЫЕ ВОПРОСЫ	20
ЗАКЛЮЧЕНИЕ	22

ВВЕДЕНИЕ

Цель лабораторной работы: изучить основные виджеты, доступные в Qt, научиться разрабатывать графический интерфейс пользователя (GUI) с использованием виджетов и разработать простое приложение с GUI, используя виджеты Qt.

Задание лабораторной работы:

Создать приложение "Персональный органайзер", в котором пользователь сможет добавлять, просматривать и удалять записи о своих задачах. Приложение должно содержать следующие элементы управления:

- Список задач (QListWidget или QTreeView).
- Кнопки для добавления, удаления и просмотра деталей задач.
- Форма для добавления новой задачи с полями:
- Название задачи (QLineEdit)
- Описание (QTextEdit)
- Дата выполнения (QDateEdit)

ХОД РАБОТЫ

Запустите Qt Creator и создайте новый проект "Приложение Qt Widgets":

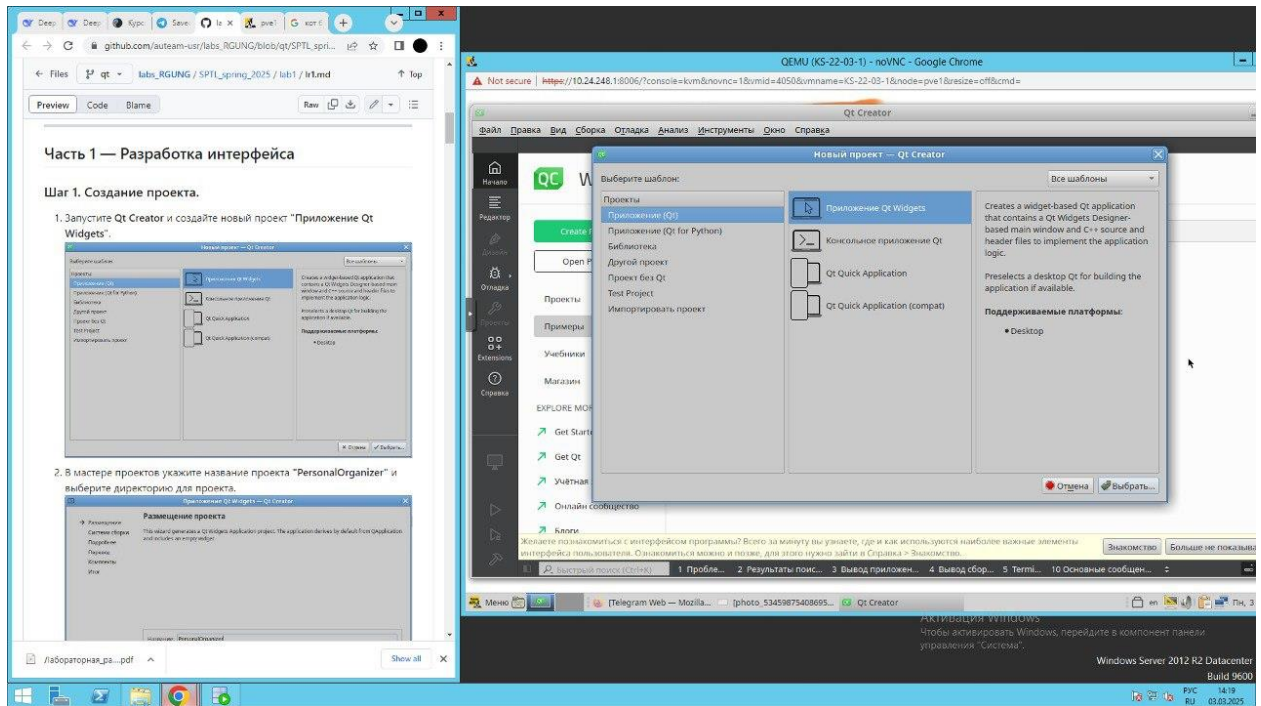


Рисунок 1. Создание нового проекта

В мастере проектов укажите название проекта "PersonalOrganizer" и выберите директорию для проекта

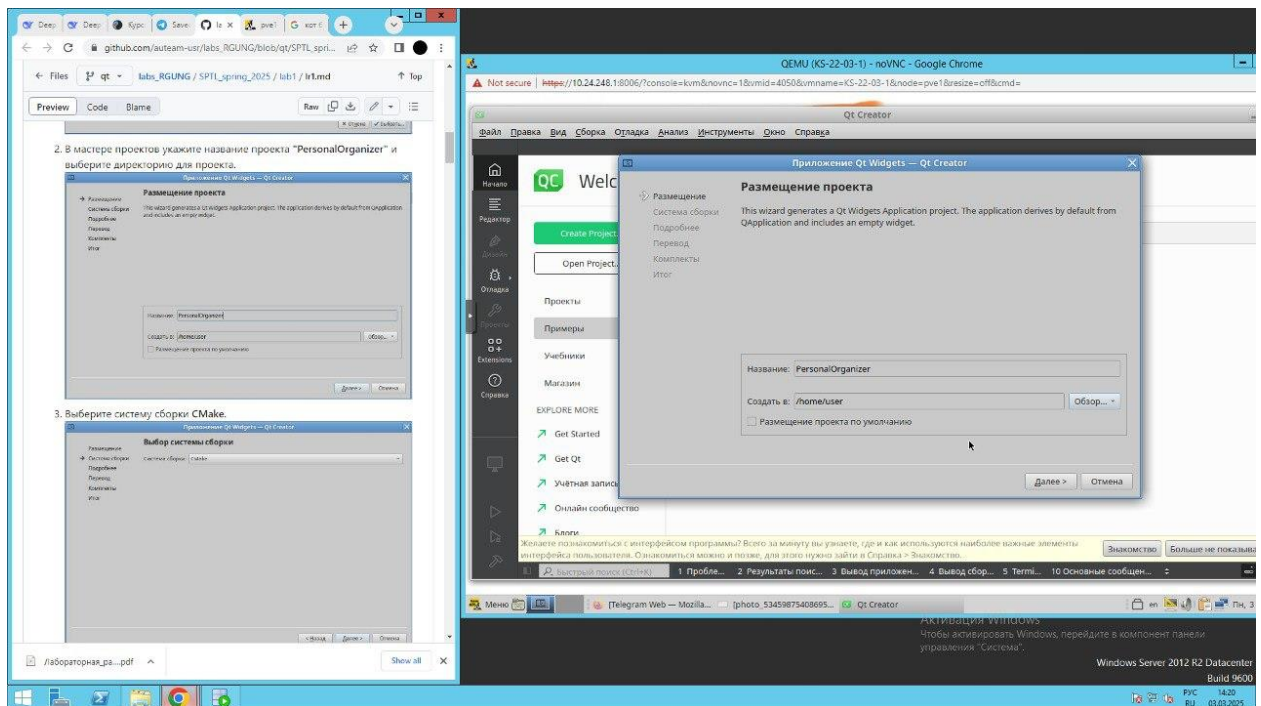


Рисунок 2. Название проекта и директория

Выберите систему сборки CMake:

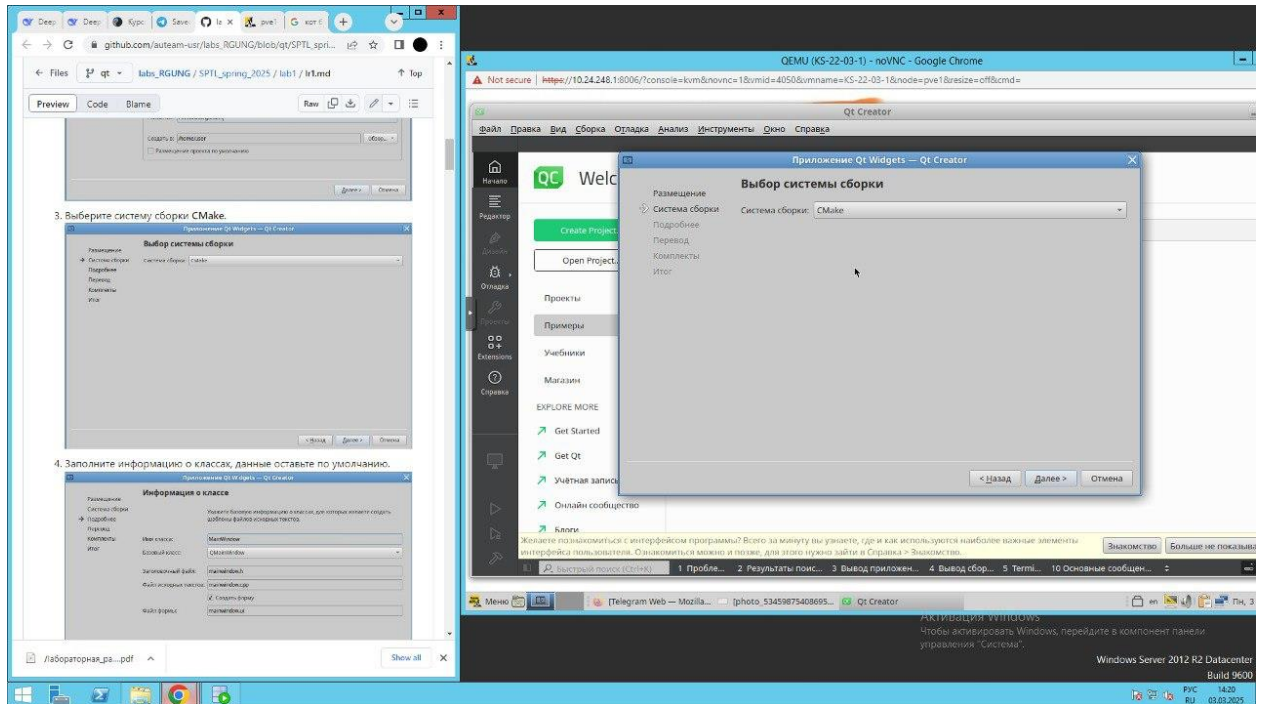


Рисунок 3. Система сборки

Заполните информацию о классах, данные оставьте по умолчанию:

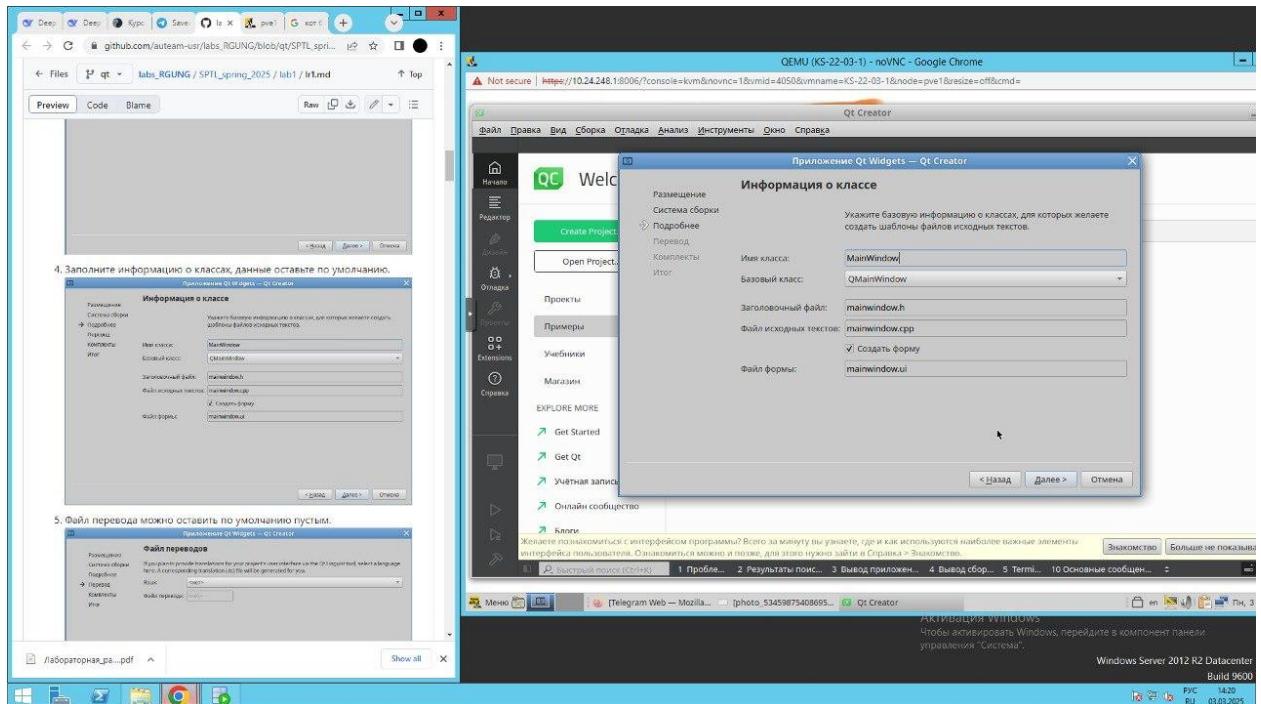


Рисунок 4. Информация о классах

Файл перевода можно оставить по умолчанию пустым:

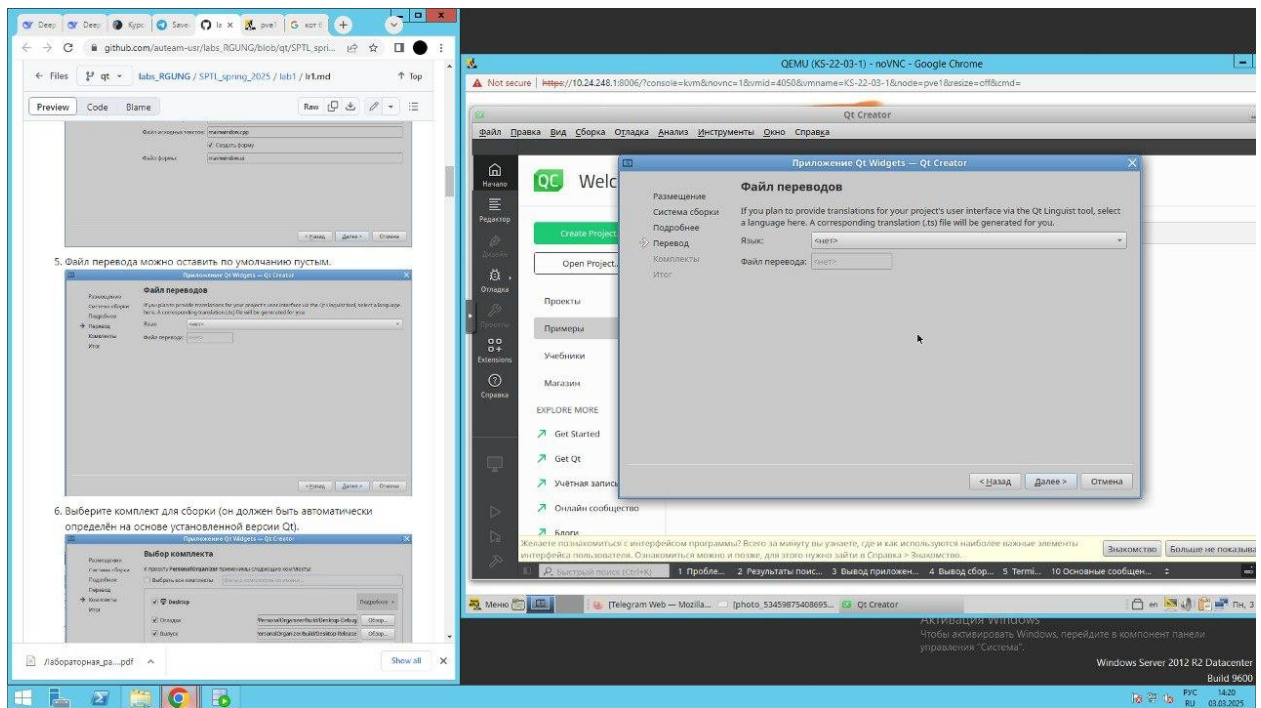


Рисунок 5. Файл переводов

Выберите комплект для сборки (он должен быть автоматически определён на основе установленной версии Qt)

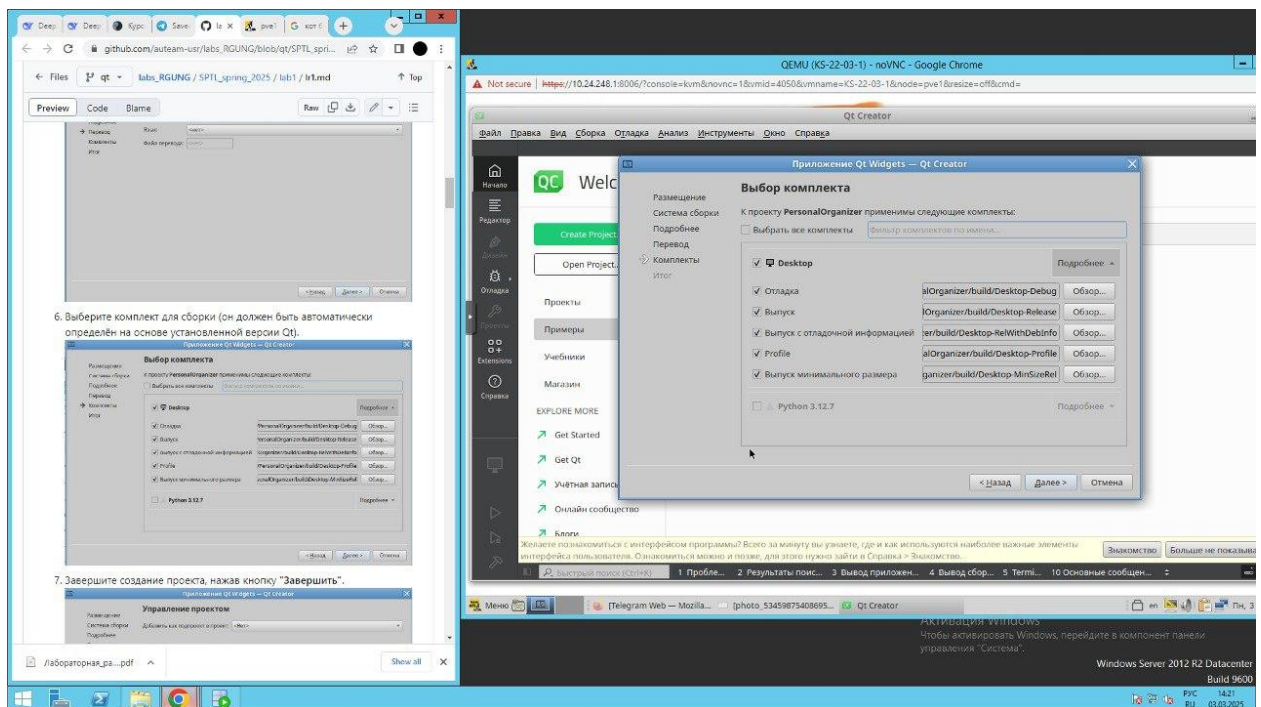


Рисунок 6. Комплект для сборки

Завершите создание проекта, нажав кнопку "Завершить":

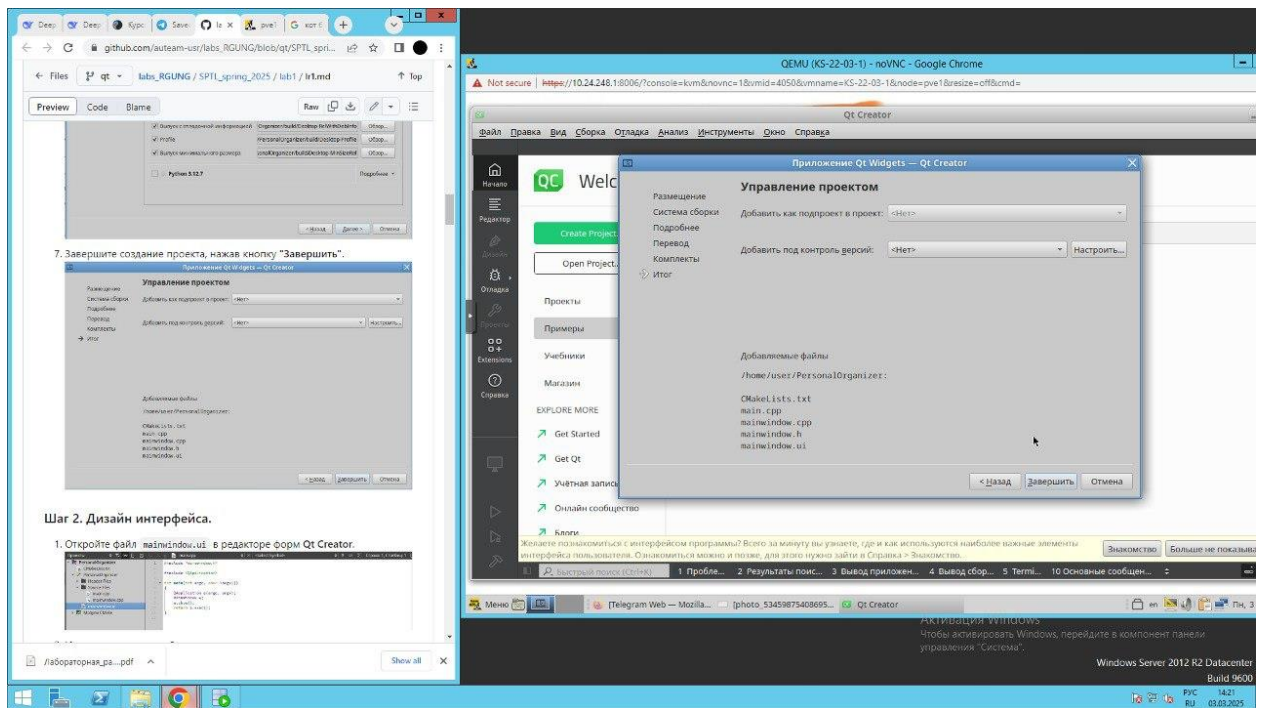


Рисунок 7. Управление проектом

Откройте файл `mainwindow.ui` в редакторе форм Qt Creator:

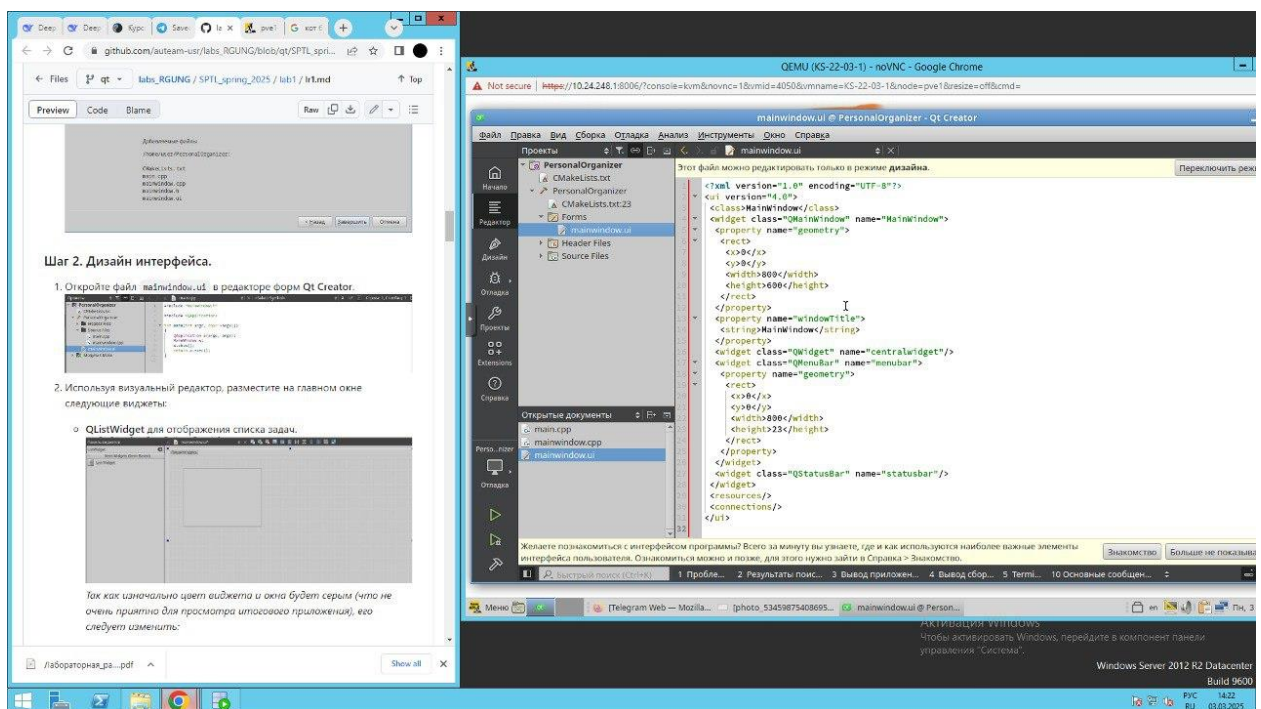


Рисунок 8. `mainwindow.ui`

QListWidget для отображения списка задач:

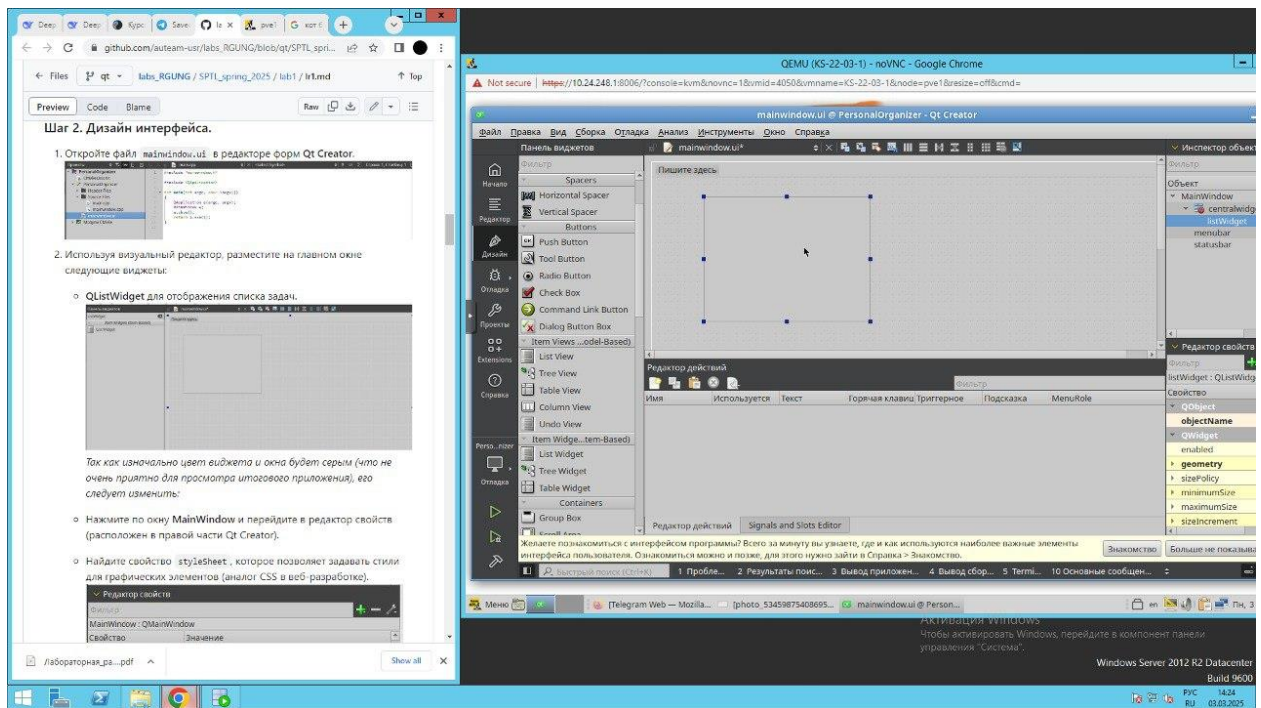


Рисунок 9. Добавление QListWidget

Так как изначально цвет виджета и окна будет серым (что не очень приятно для просмотра итогового приложения), его следует изменить:

- Нажмите по окну `MainWindow` и перейдите в редактор свойств (расположен в правой части Qt Creator).
- Найдите свойство `stylesheet`, которое позволяет задавать стили для графических элементов (аналог CSS в веб-разработке)

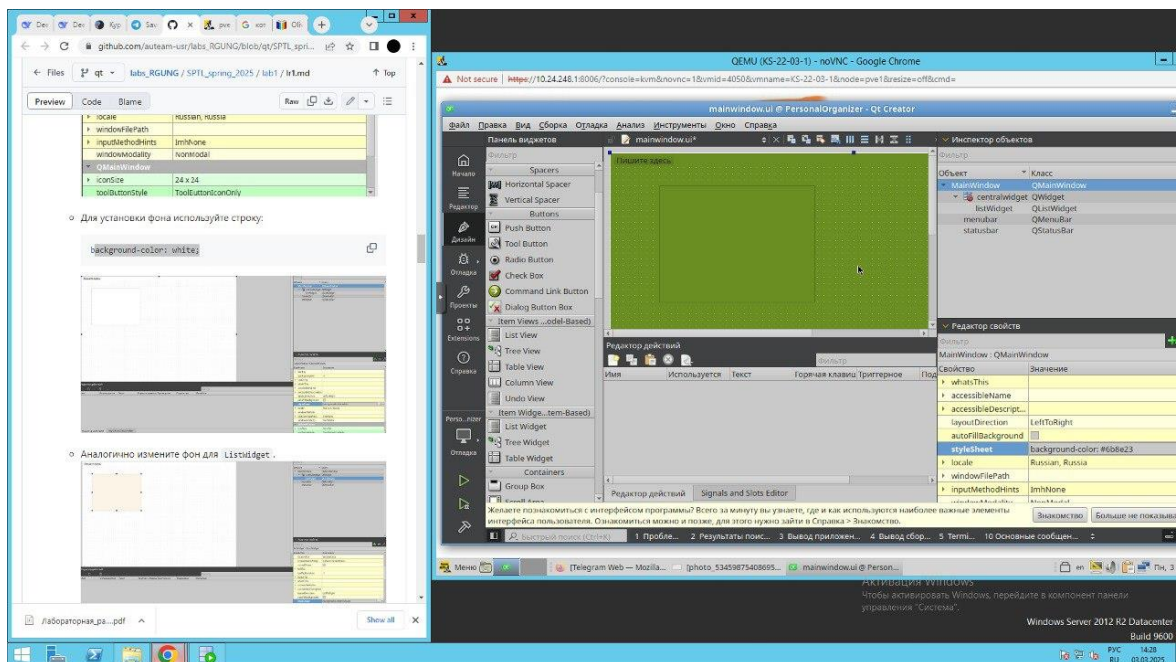


Рисунок 10. Изменение цвета

Для установки фона используйте строку:
background-color: white;

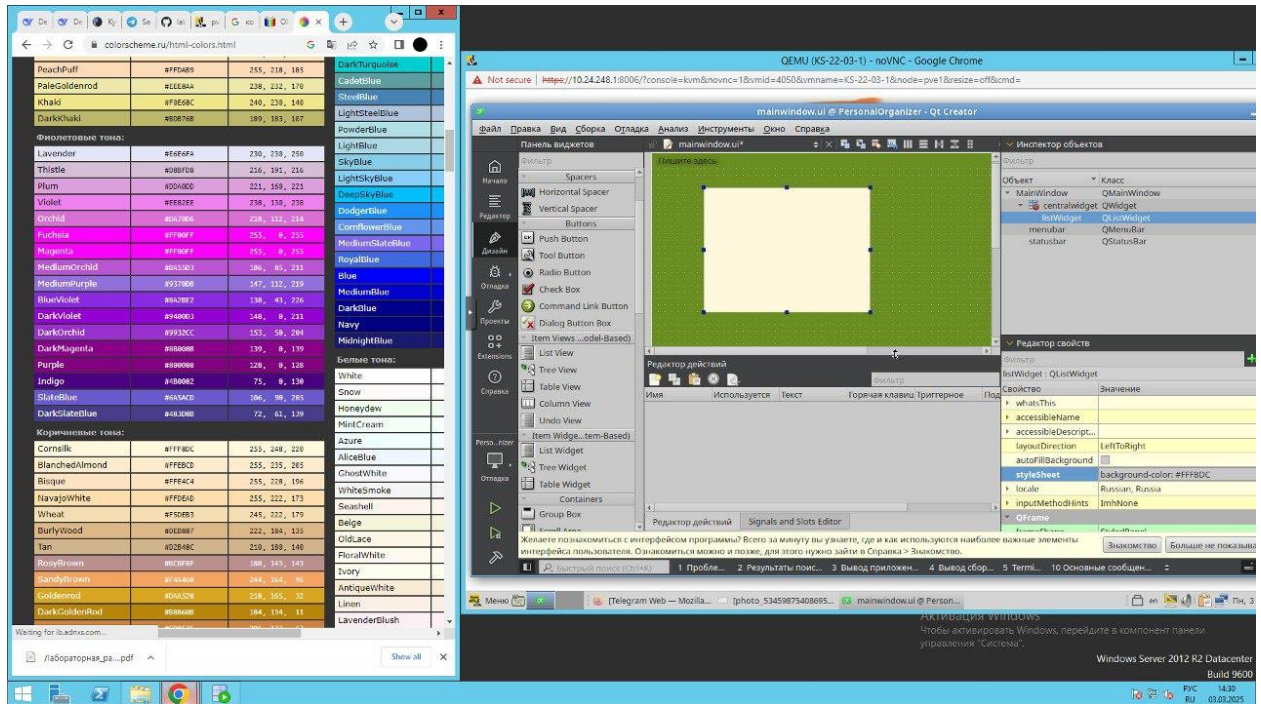


Рисунок 11. Изменение цвета

Аналогично измените фон для ListWidget:

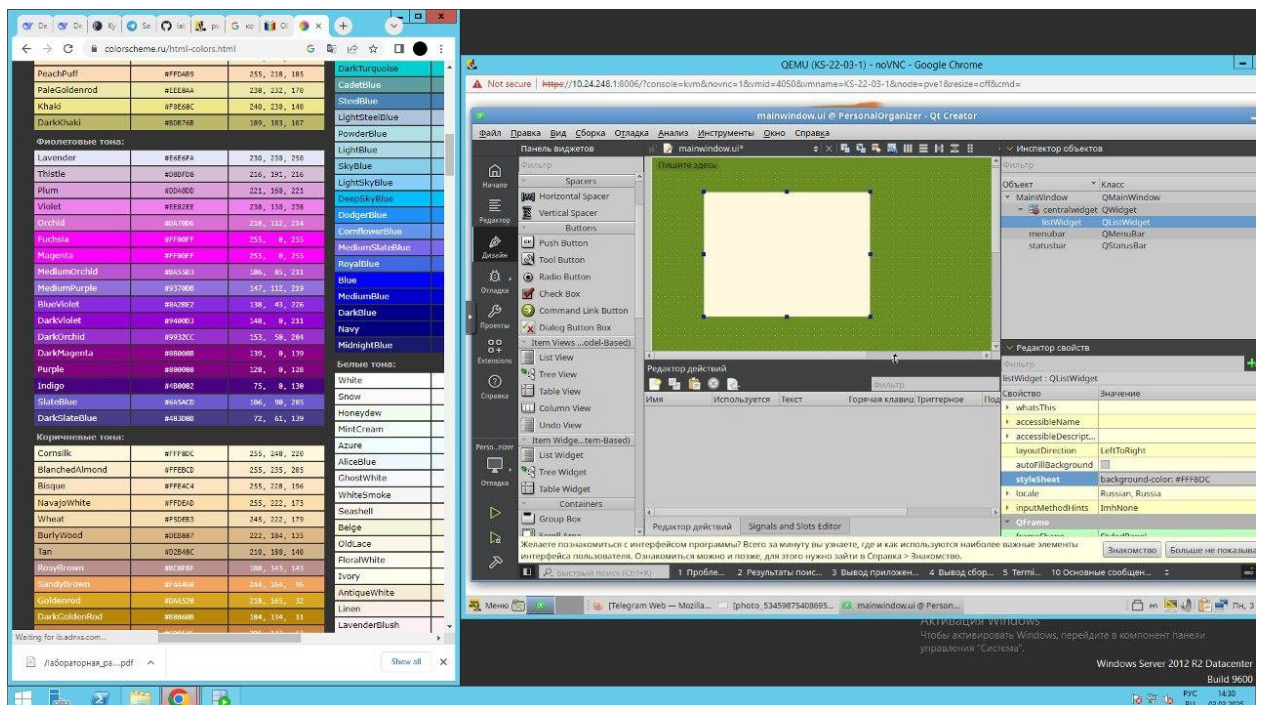


Рисунок 12. Изменение цвет

QPushButton для кнопок:

- "Добавить задачу"

- "Удалить задачу"
- "Подробности задачи"

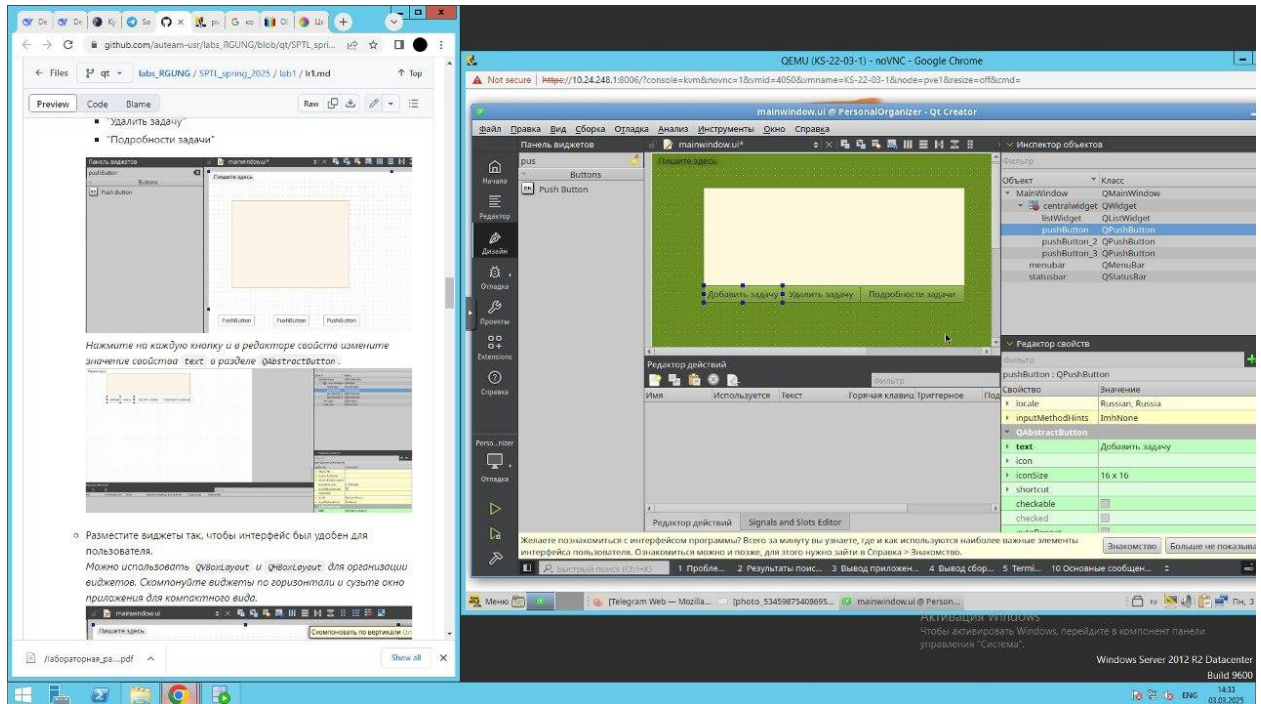


Рисунок 13. Добавление QPushButton

Нажмите на каждую кнопку и в редакторе свойств измените значение свойства text в разделе QAbstractButton:

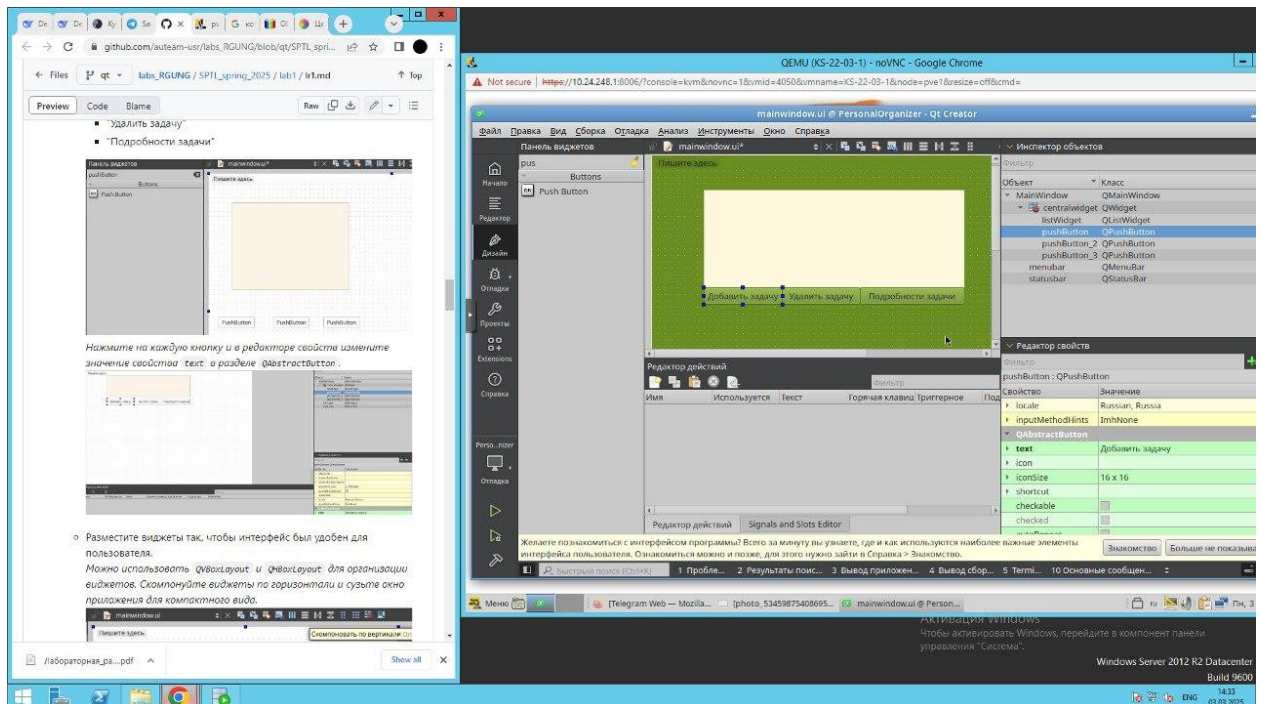


Рисунок 14. Изменение text в QAbstractButton

Разместите виджеты так, чтобы интерфейс был удобен для пользователя.

Можно использовать QVBoxLayout и QHBoxLayout для организации виджетов. Скомпонуйте виджеты по горизонтали и сузьте окно приложения для компактного вида:

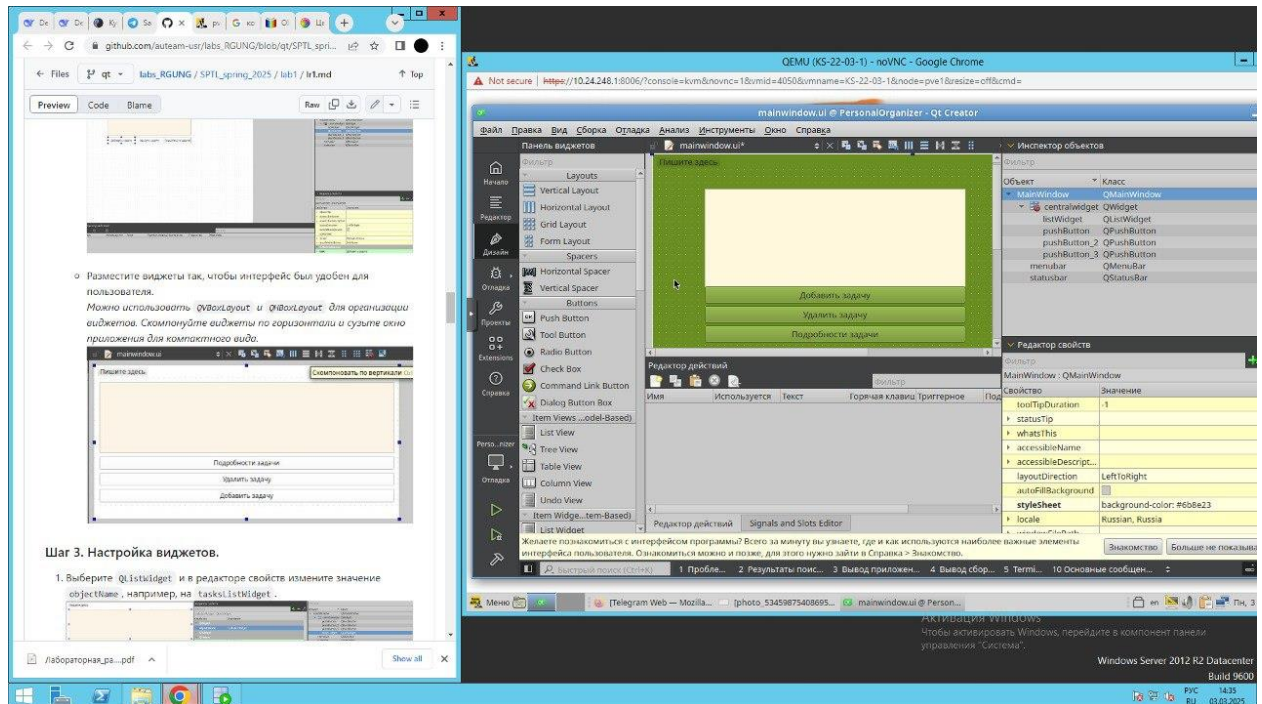


Рисунок 15. Компоновка виджетов

Выберите `QListWidget` и в редакторе свойств измените значение `objectName`, например, на `tasksListWidget`:

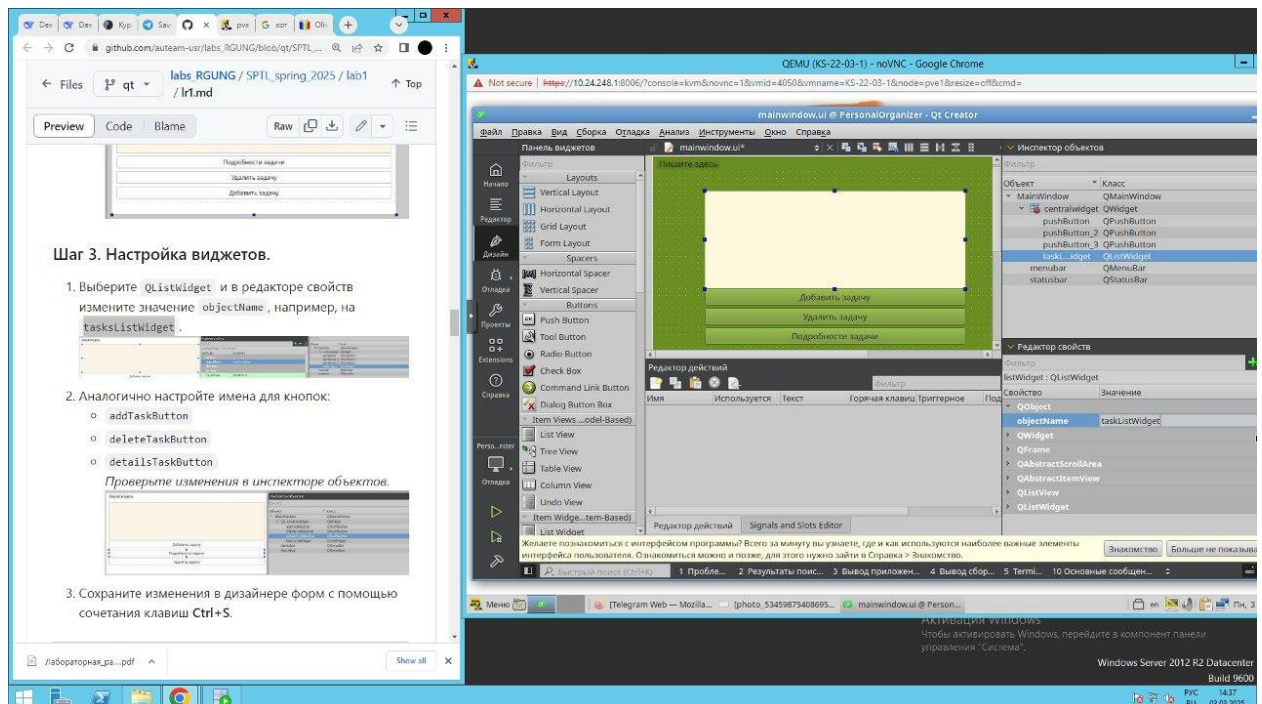


Рисунок 16. Изменение objectName

Аналогично настройте имена для кнопок:

- addTaskButton
- deleteTaskButton
- detailsTaskButton

Проверьте изменения в инспекторе объектов:

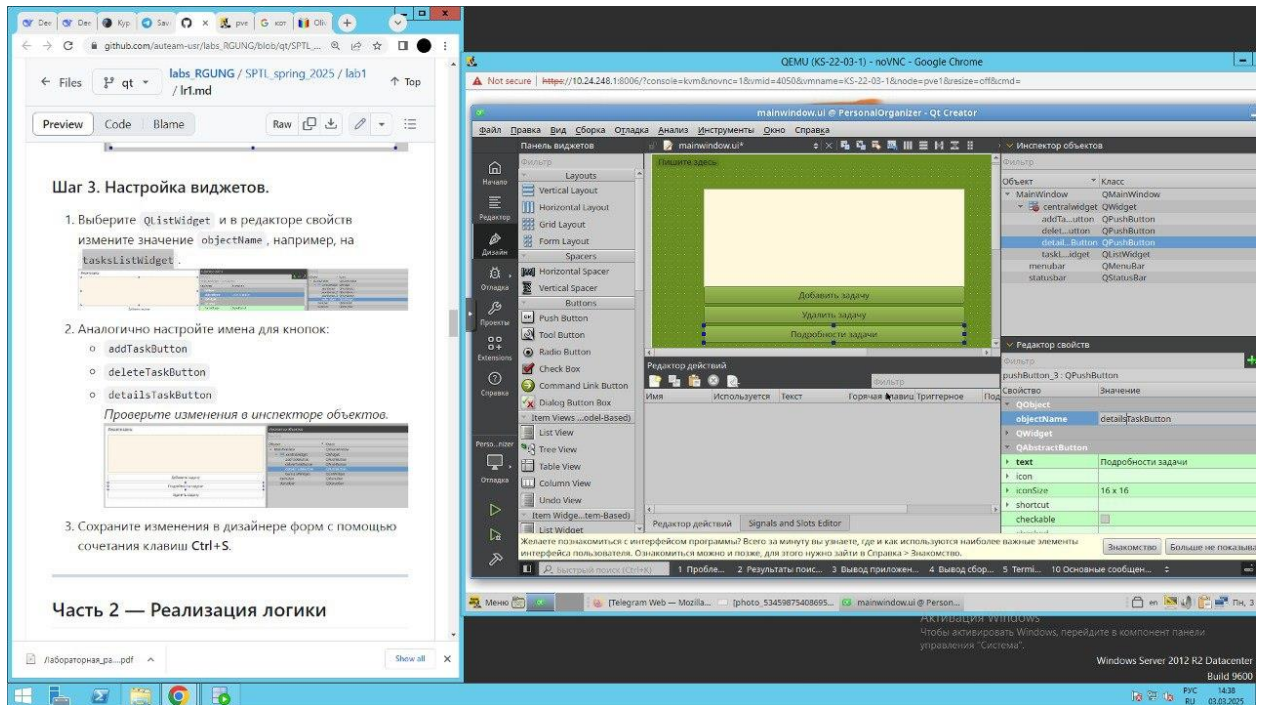


Рисунок 17. Проверка изменений

Перейдите в режим редактирования кода для mainwindow.cpp:

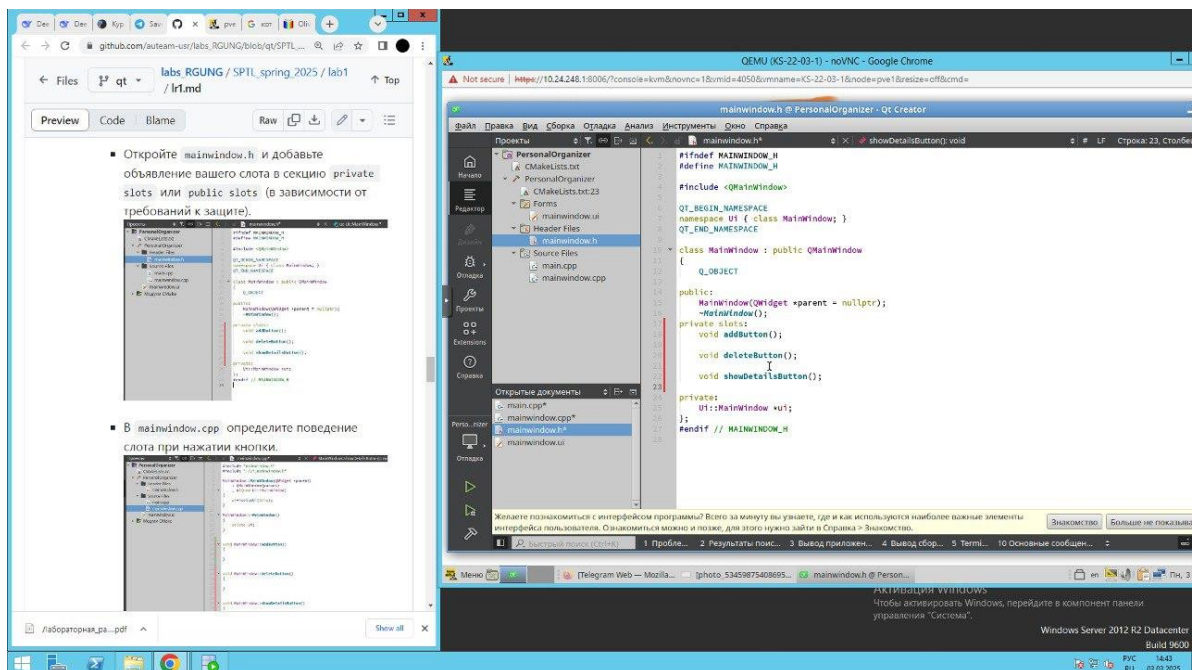
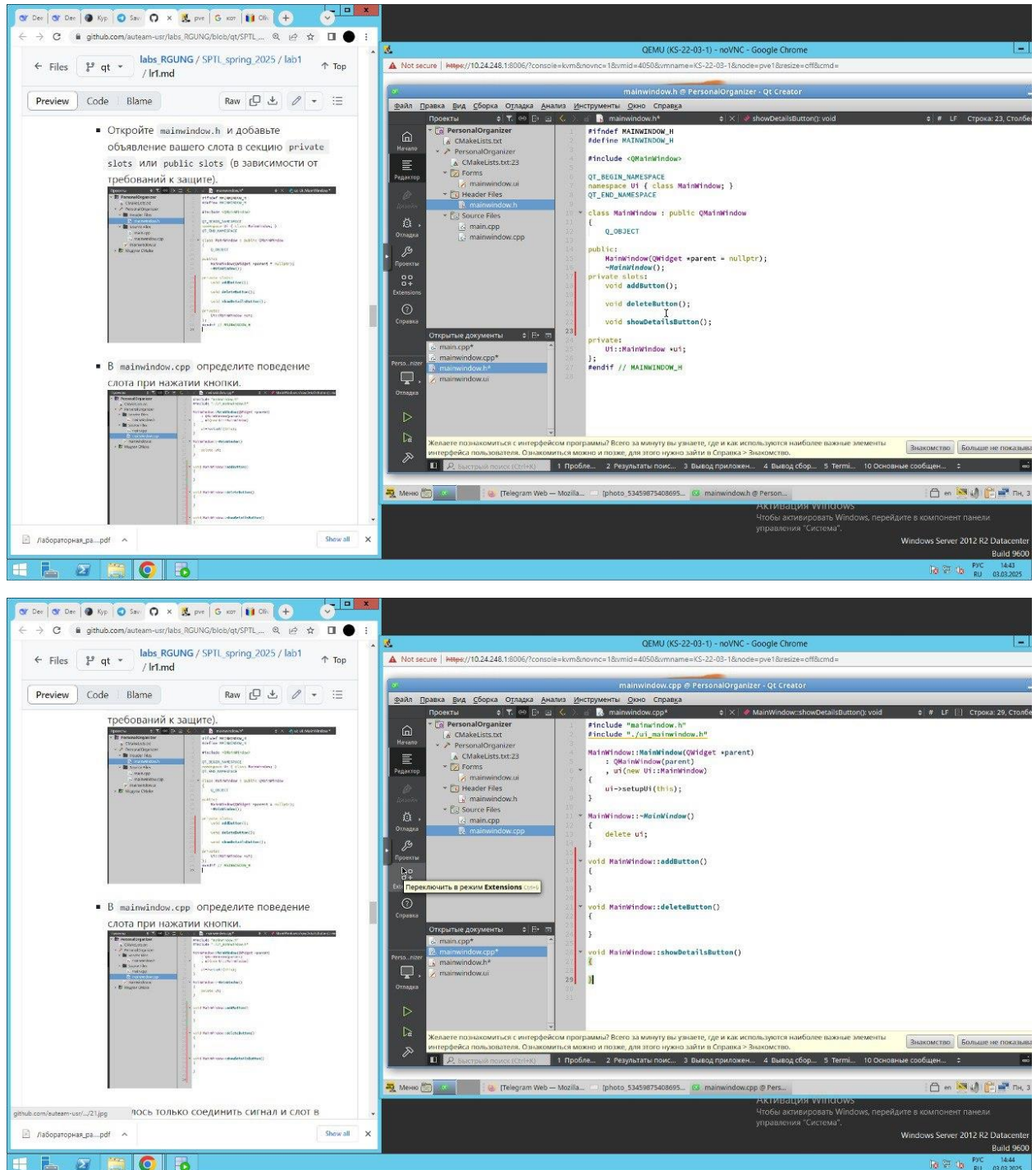


Рисунок 18. Редактирование mainwindow.cpp

Создайте слоты для обработки нажатий на кнопки. Это можно сделать двумя способами:



Рисунки 19-20. Создание слотов для обработки кнопок

Осталось только соединить сигнал и слот в конструкторе вашего класса. Перейдите в `mainwindow.cpp` и перейдите в конструктор класса:

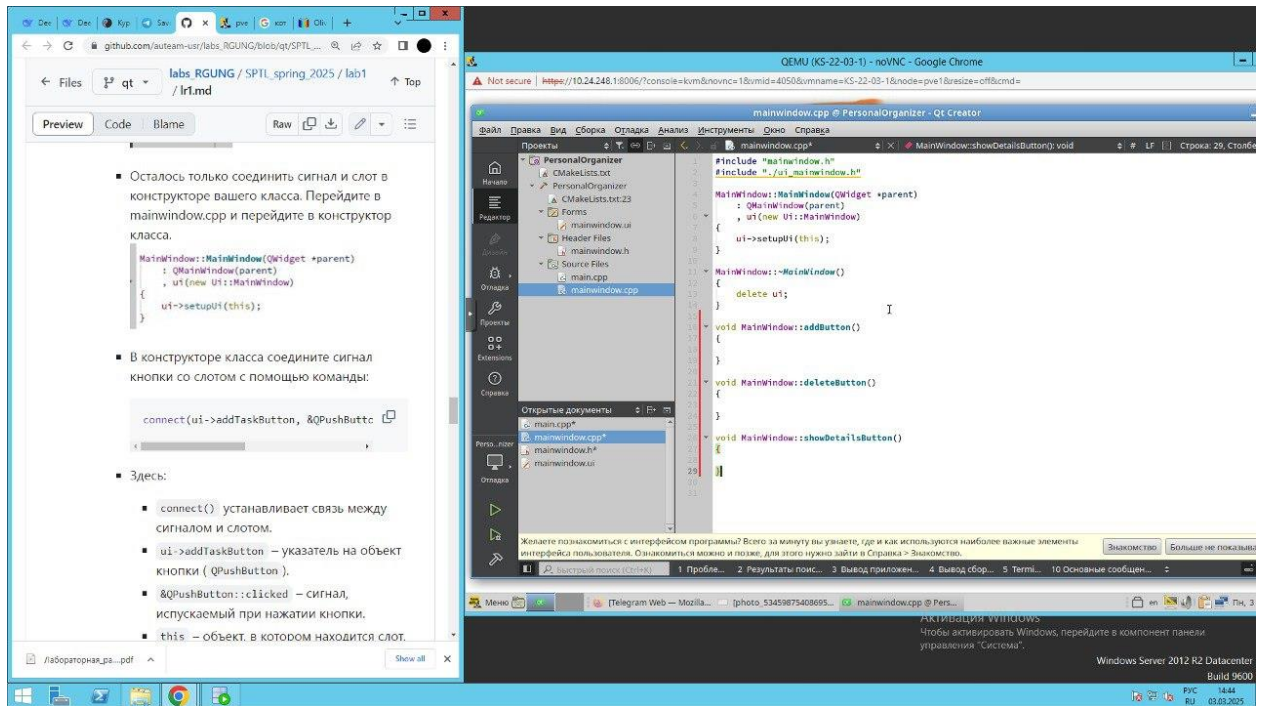


Рисунок 21. Соединение сигнала и слота

В конструкторе класса соедините сигнал кнопки со слотом с помощью команды:

connect(ui->addTaskButton, &QPushButton::clicked, this, &MainWindow::addButton);

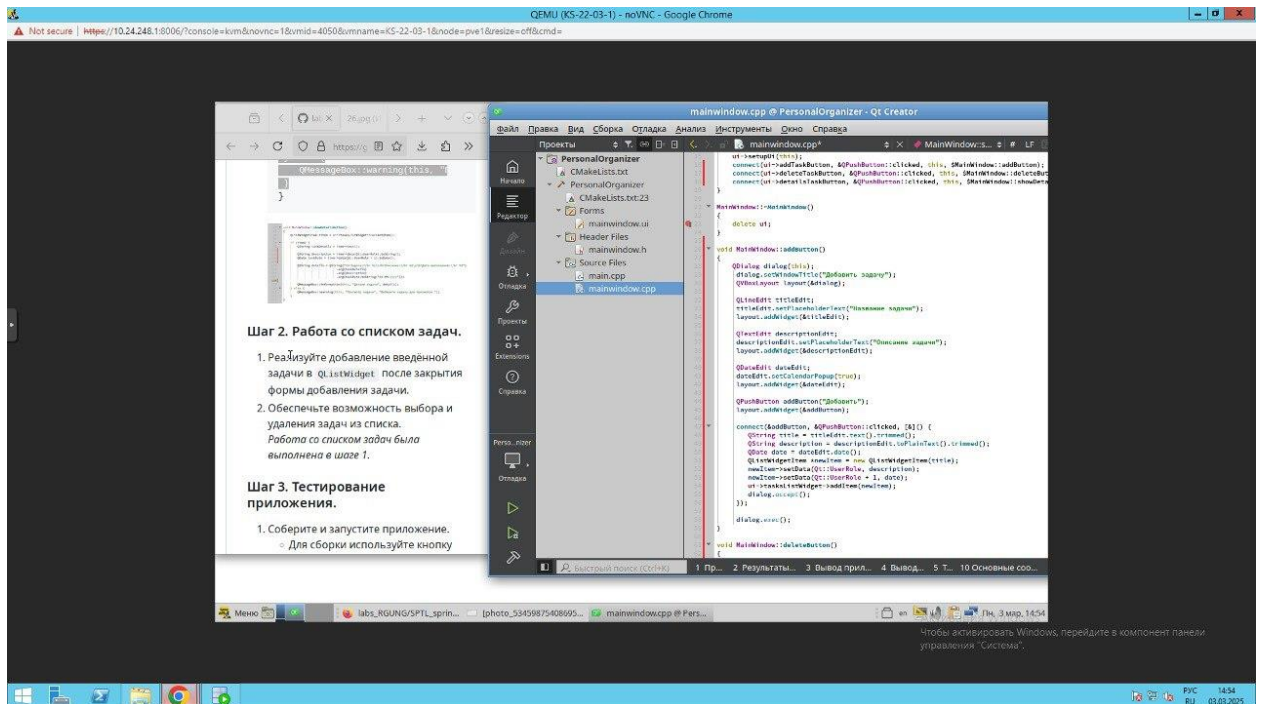


Рисунок 22. Логика добавления задачи

Реализуйте открытие диалогового окна с формой для ввода данных о новой задаче. При добавлении задачи должно открываться диалоговое окно с полями:

- Название задачи (QLineEdit)
- Описание (QTextEdit)
- Дата выполнения (QDateEdit)

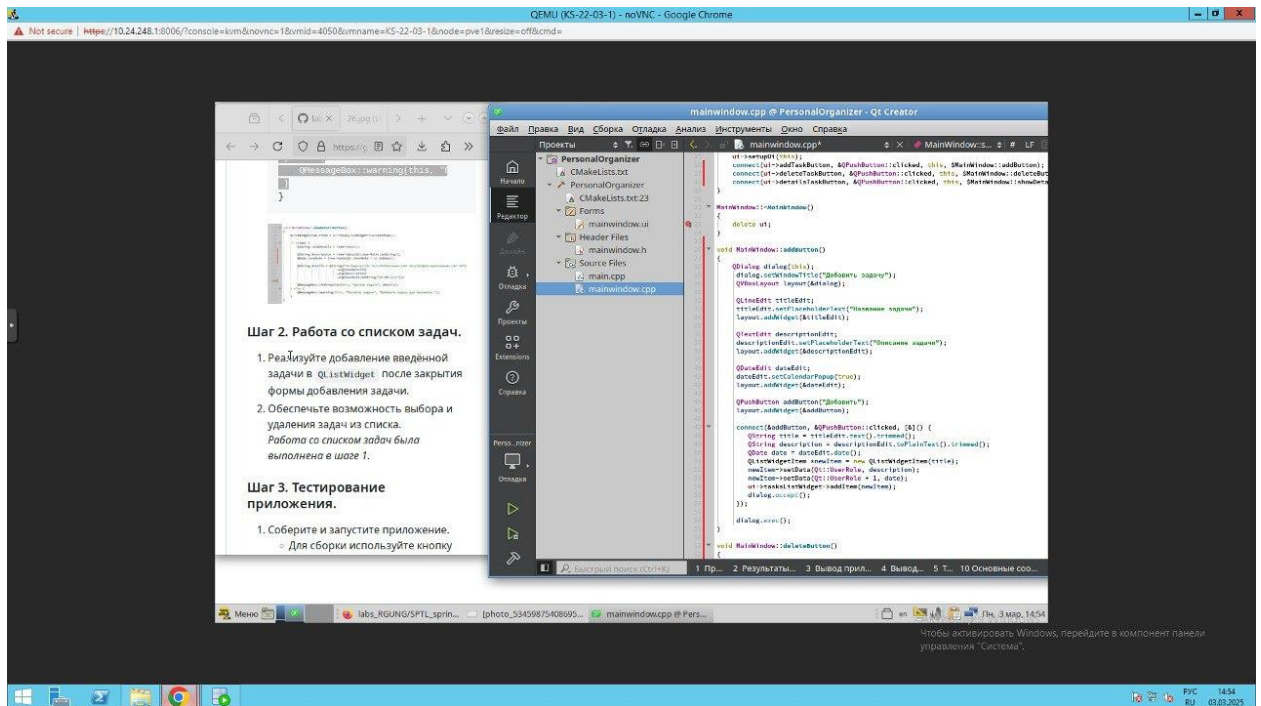


Рисунок 23. Реализация открытия диалогового окна

Реализуйте удаление выбранной задачи из списка:

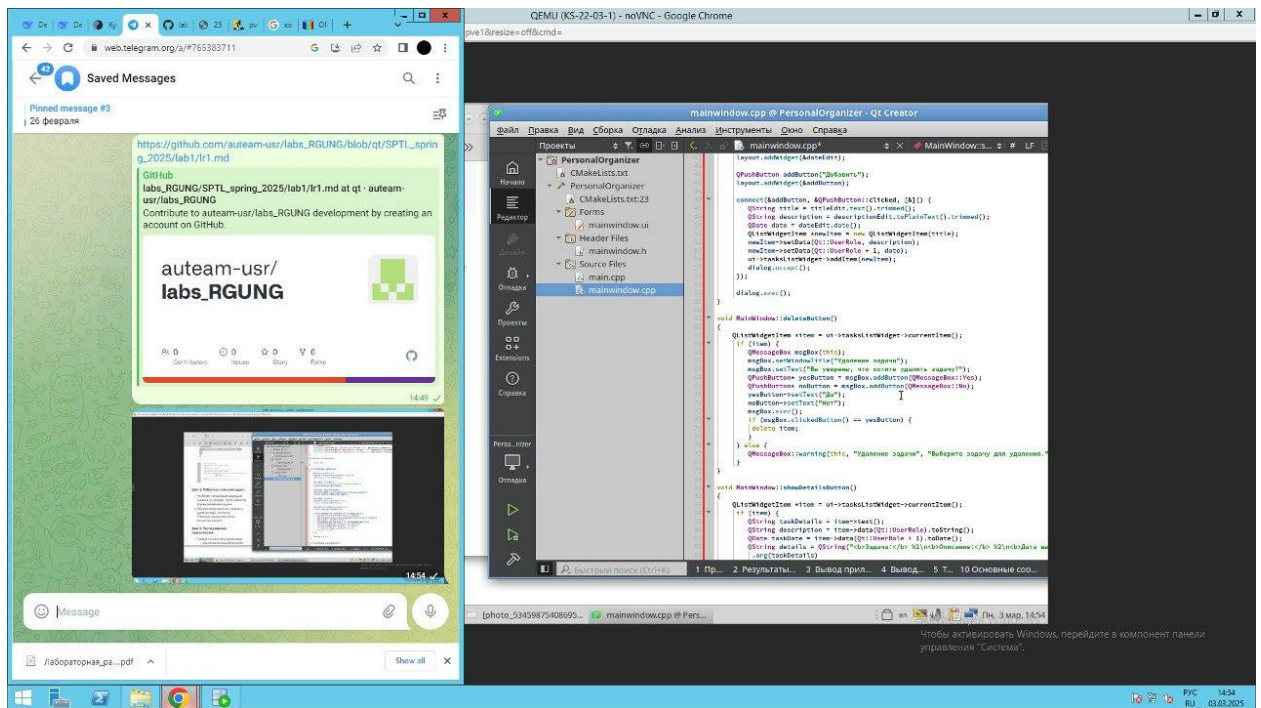


Рисунок 24. Удаление задачи

Реализуйте отображение информации о выбранной задаче, например, с помощью QMessageBox:

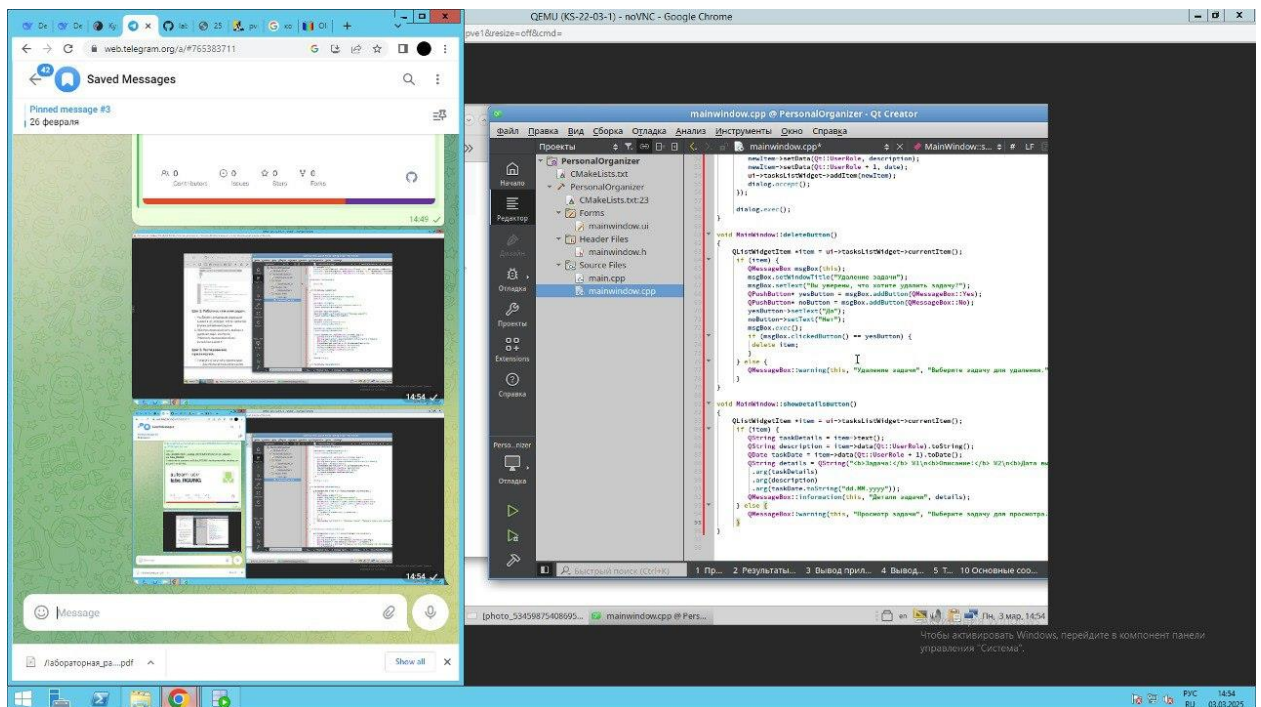


Рисунок 25. Отображение информации по задаче

Проверьте функциональность:

При нажатии кнопки «Добавить задачу» появляется диалоговое окно с необходимыми полями:

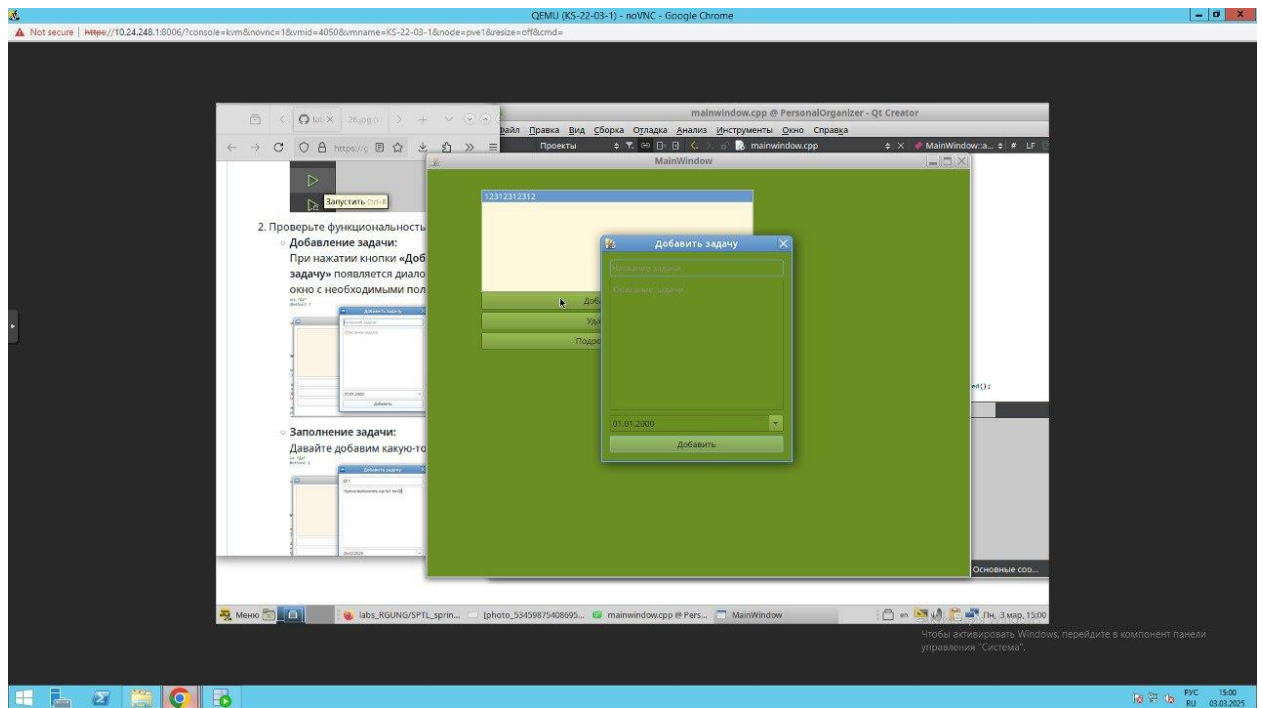


Рисунок 26. Проверка добавления задачи

Давайте добавим какую-то задачу:

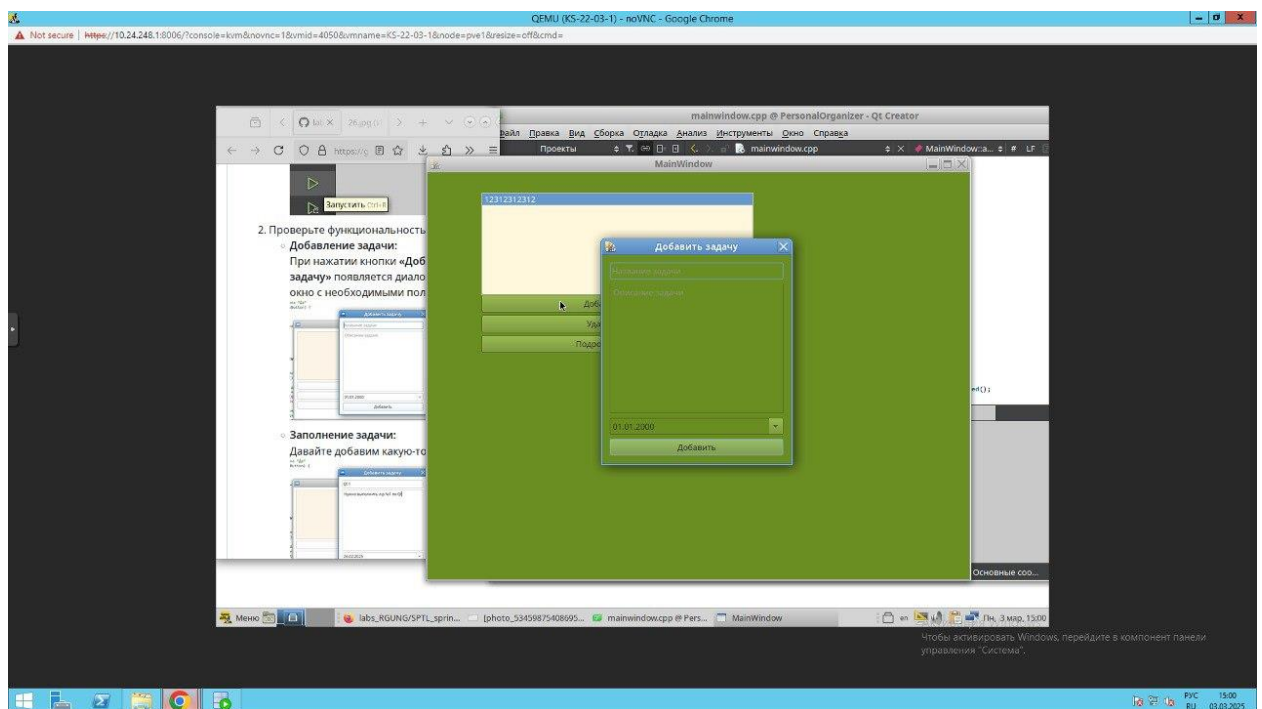


Рисунок 27. Проверка добавления задачи

Просмотр деталей задачи: выберите задачу и нажмите кнопку «Подробности задачи» для отображения деталей:

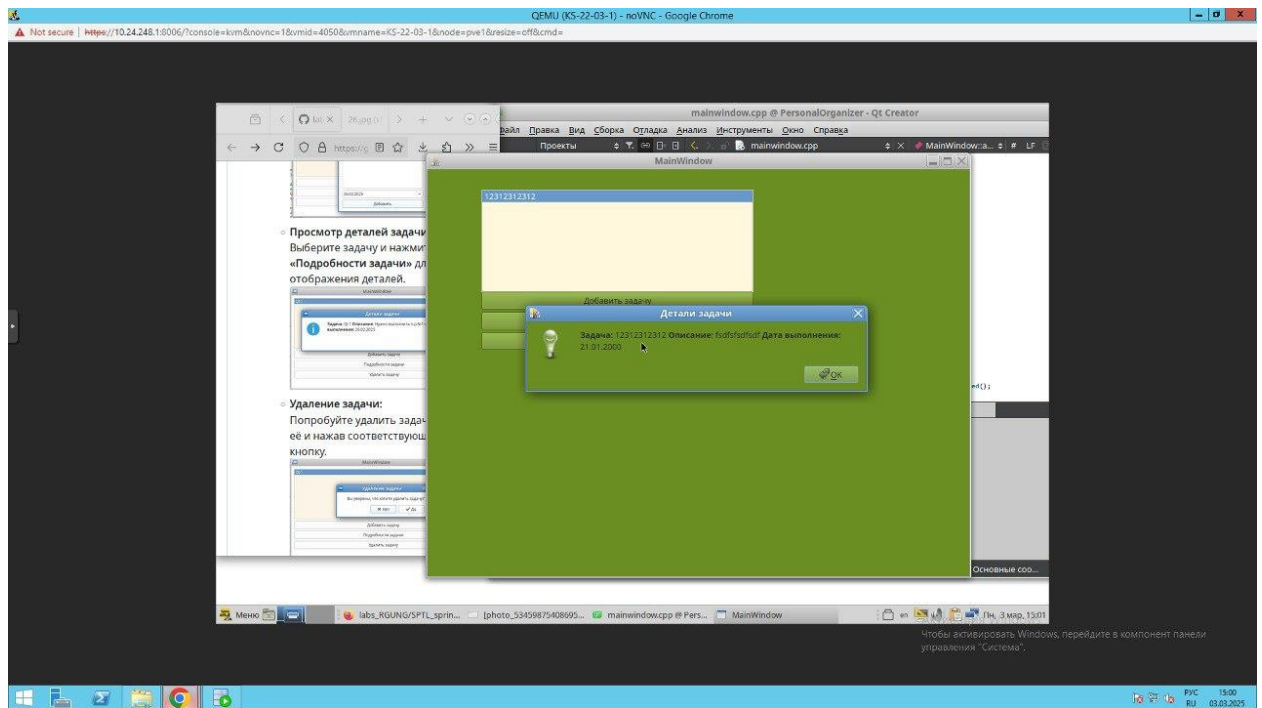


Рисунок 28. Подробности задачи

Попробуйте удалить задачу, выбрав её и нажав соответствующую кнопку:

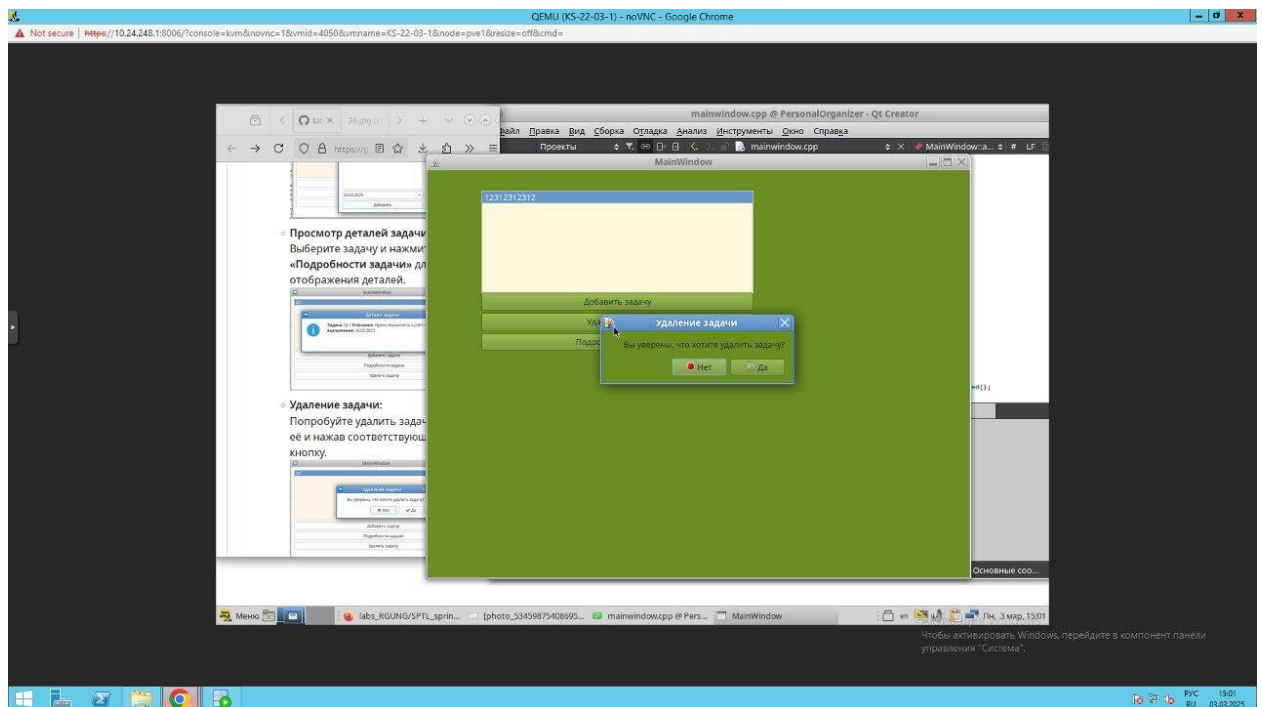


Рисунок 29. Удаление задачи

Добавьте валидацию данных формы добавления задачи (например, проверку на пустое название):

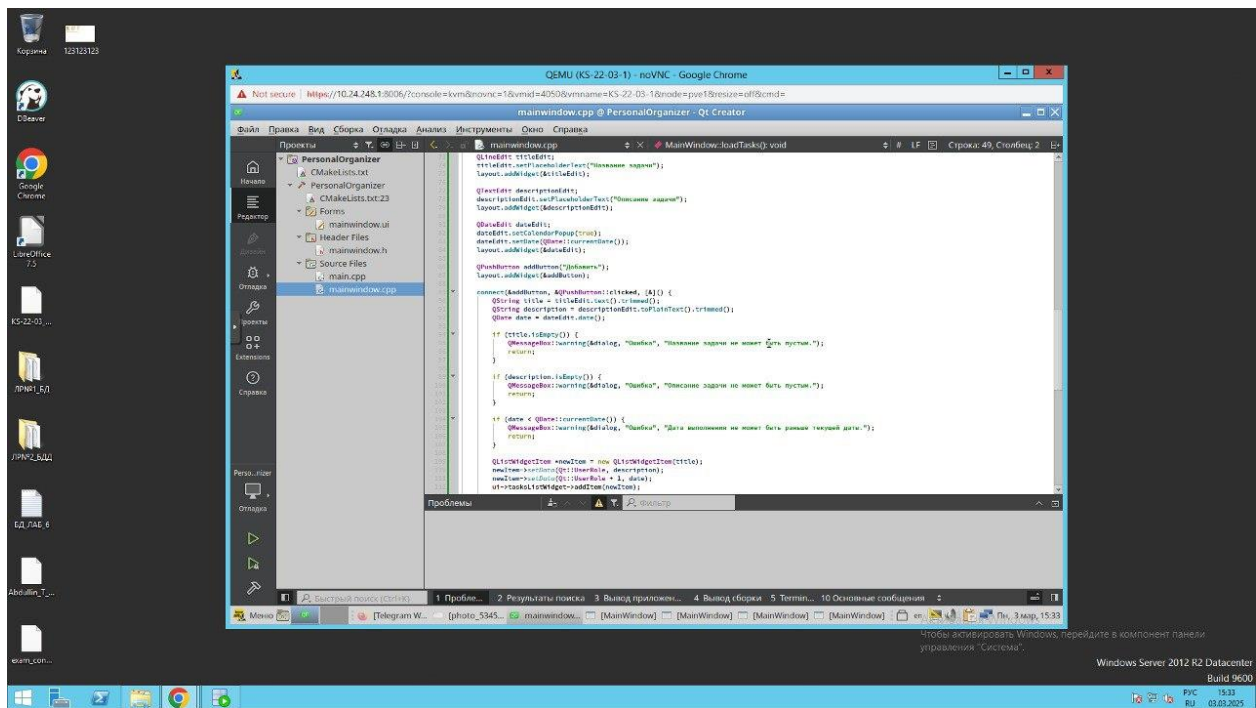


Рисунок 30. Валидация на добавление задачи

Реализуйте сохранение списка задач между сессиями работы приложения (можно использовать QSettings или работу с файлом):

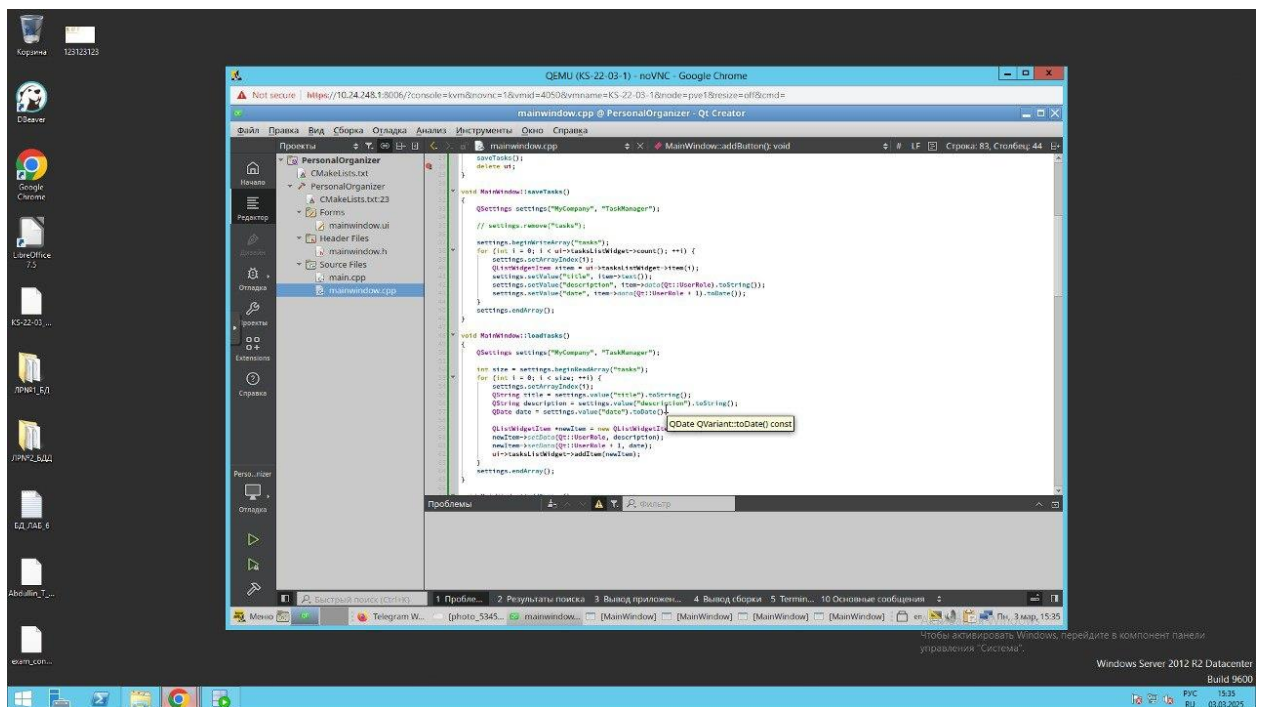


Рисунок 31. Сохранение списка задач

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как добавить виджеты на форму в Qt Creator?

- Откройте Qt Creator и создайте новый проект.
- В режиме Design (дизайнер форм) вы увидите пустую форму (окно).
- В левой панели (называется Widget Box) находятся все доступные виджеты (кнопки, метки, поля ввода и т.д.).
- Чтобы добавить виджет на форму:
 - Перетащите нужный виджет (например, QPushButton, QLabel, QLineEdit) из Widget Box на форму.
 - Вы можете изменять размер и положение виджета с помощью мыши.
- Чтобы изменить свойства виджета (например, текст кнопки или размер шрифта), используйте правую панель (Property Editor).

2. Как создать слот для обработки нажатия на кнопку?

- Добавьте кнопку на форму (например, QPushButton).
- Щелкните правой кнопкой мыши на кнопке и выберите Go to slot....
- В открывшемся окне выберите сигнал clicked() и нажмите ОК.
- Qt Creator автоматически создаст слот в вашем классе и откроет его в редакторе кода.

3. Как можно реализовать сохранение данных между сессиями приложения в Qt?

- QSettings — это класс, который позволяет сохранять настройки и данные в реестр (на Windows) или в файлы конфигурации (на Linux/macOS)

- Вы можете сохранять данные в текстовый файл, JSON, XML или бинарный файл.

ЗАКЛЮЧЕНИЕ

Я изучил основные виджеты, доступные в Qt, научился разрабатывать графический интерфейс пользователя (GUI) с использованием виджетов и разработал простое приложение с GUI, используя виджеты Qt.