|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |
| Институт искусственного интеллекта | | |
| Кафедра программного обеспечения систем радиоэлектронной аппаратуры | | |

|  |  |
| --- | --- |
| **КУРСОВАЯ РАБОТА** | |
| по дисциплине  «Технология создания программного продукта» | |
|  | |
| **Тема: «Редактор тестов с удалённой компиляцией»** | |
| Студент группы           КМБО-02-20 | *А. В. Волков* |
| Руководитель практики | *А. В. Завьялов* |

|  |  |  |
| --- | --- | --- |
| «Отчет представлен к рассмотрению» | «\_\_\_» \_\_\_\_\_\_\_\_\_\_2022 г. | *(подпись студента)* |
|  |  |  |
| «Отчет утвержден.  Допущен к защите» | «\_\_\_» \_\_\_\_\_\_\_\_\_\_2022 г. | *(подпись руководителя)* |

Москва 2022

# ОГЛАВЛЕНИЕ

[Введение 3](#_Toc519522571)

[1 Постановка задачи 3](#_Toc519522572)

[1.1 Задание 3](#_Toc519522573)

[2 Описание работ и результатов 3](#_Toc519522574)

[2.1 Используемые средства разработки 3](#_Toc519522575)

[2.2 Содержание и результат выполненных работ 3](#_Toc519522576)

3 [Заключение 8](#_Toc519522577)

[Список использованных источников 8](#_Toc519522578)

# Введение

На базовой кафедре разрабатывается система автоматизации проверки программ студентов, разрабатываемых в ходе изучения программирования. В связи с этим возникла необходимость создания и проверки автоматических тестов для таких программ. В рамках практики поставлена задача создания приложения для быстрой проверки правильности теста при наличии образца проверяемой программы.

# 1 Постановка задачи

1.1 Задание

Необходимо создать приложение на основе фреймворка Qt для проверки тестов, которое будет собирать отправлять на сервер тест и образец проверяемой программы, после чего принимать и отображать результат проверки.

1.2 Исходные данные

Ссылка на API удаленной компиляции теста и образца программы

* <http://vega.fcyb.mirea.ru/testcpp/api/v1/swagger/#/SpecRunner/post_specRun>

Архив с примерами тестов и образцов программ.

1.3 Планируемые результаты

- Приложение для проверки автоматических тестов.

- Образцы проверяемых программ и автоматических тестов к ним.

- Отчет с описанием возможностей и инструкций по использованию приложения.

, которое при наличии образца правильно работающей программы, будет отсылать требующие проверки тесты на сайт Веги для удаленной компиляции, после чего обрабатывать результаты работы компилятора.

# 2 Описание работ и результатов

2.1 Используемые средства разработки

Для написания исходного кода была использована IDE Qt Creator 4.11.2, Qt Widgets, а так же стандартные библиотеки Qt.

2.2 Содержание и результат выполненных работ

В соответствии с планом выполнения индивидуального задания, были выполнены следующие пункты:

* Была изучена работа с фреймворком Qt, в частности его встроенные библиотеки для отправки сетевых запросов.
* Был более подробно изучен процесс компиляции программ.
* Создан прототип приложения, отправляющего образец программы и тест для нее в формате JSON, для удаленной компиляции, (рисунок 2.1).

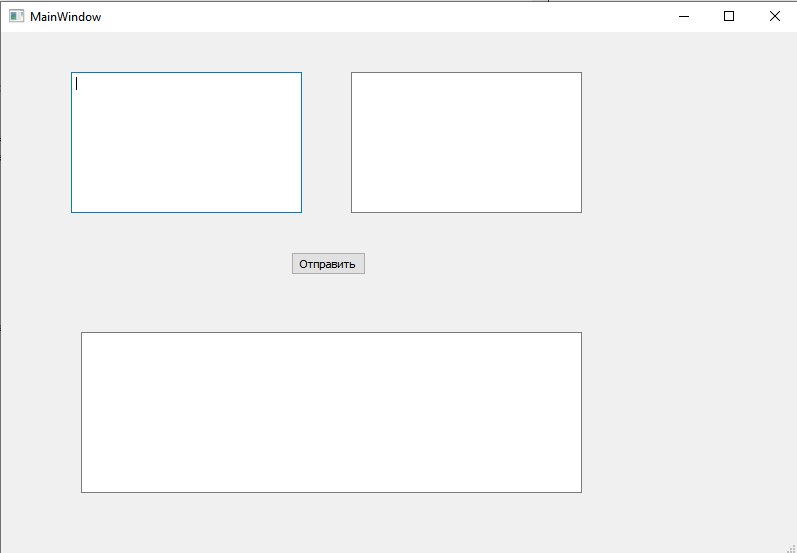


Рисунок 2.1 – прототип основной формы программы

* Добавлен текстовый редактор QCodeEditor для более удобной работы с текстом программы и теста, так как в стандартном текстовом окне отсутствует даже простейшая табуляция. QCodeEditor был добавлен из сторонней библиотеки, скомпилированной из исходного кода (<https://github.com/Megaxela/QCodeEditor>). Созданы отдельные вкладки для вывода результатов разных этапов компиляции (рисунок 2.2).

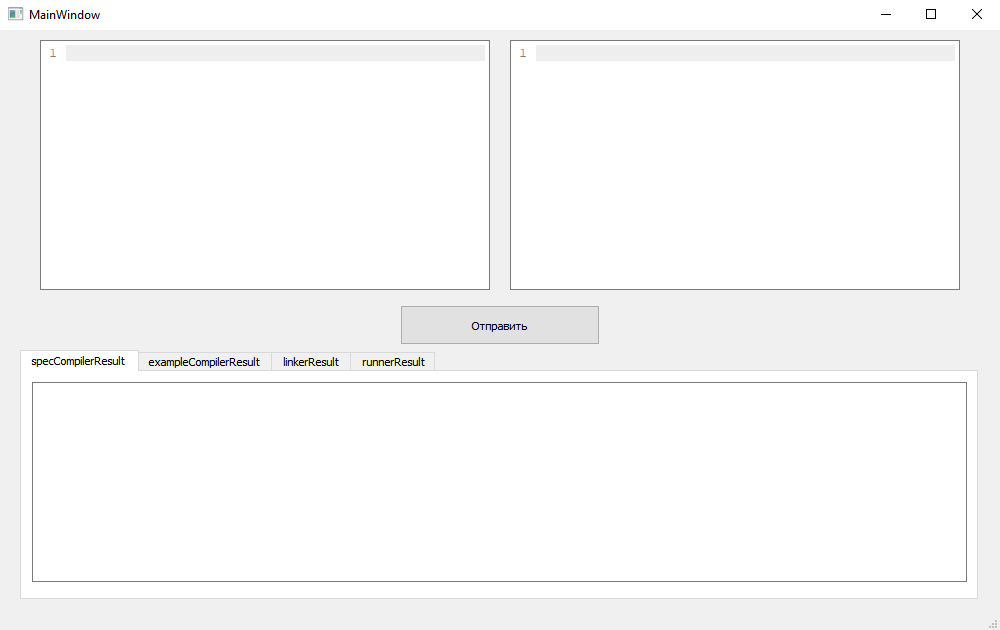


Рисунок 2.2 – основная форма итогового приложения

* Разработана функция обработки результата компиляции, полученного с сервера, в том числе обработки ANSI escape codes для корректного форматирования полученных данных. Результаты компиляции используют ANSII для форматирования текста, которые не поддерживаются в виджетах Qt. Для этого используется сторонняя функция (<https://stackoverflow.com/questions/26500429/qtextedit-and-colored-bash-like-output-emulation>). Используемая функция была модифицирована для поддержки разных форматов ANSI escape codes. Результат обработки ANSI-форматирования представлен на рисунке 2.3.

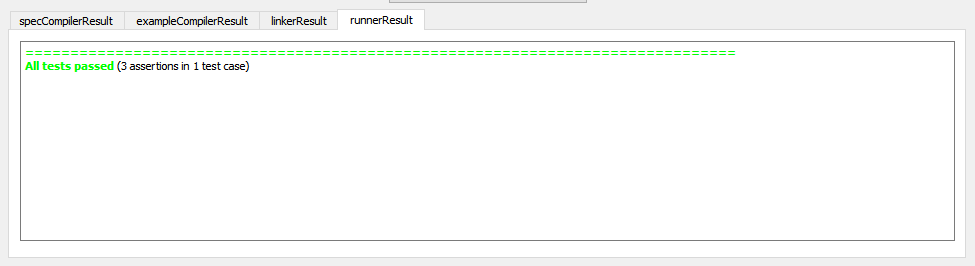


Рисунок 2.3 – пример корректной обработки результатов компиляции

* Разработана форма авторизации (рисунок 2.4), благодаря которой доступ к основной форме приложения возможен только после проверки логина и пароля пользователя на соответствие таковым на сайте кафедры 536 и получения с него данных cookie. Данные cookie после используется для запросов к API удаленной компиляции.

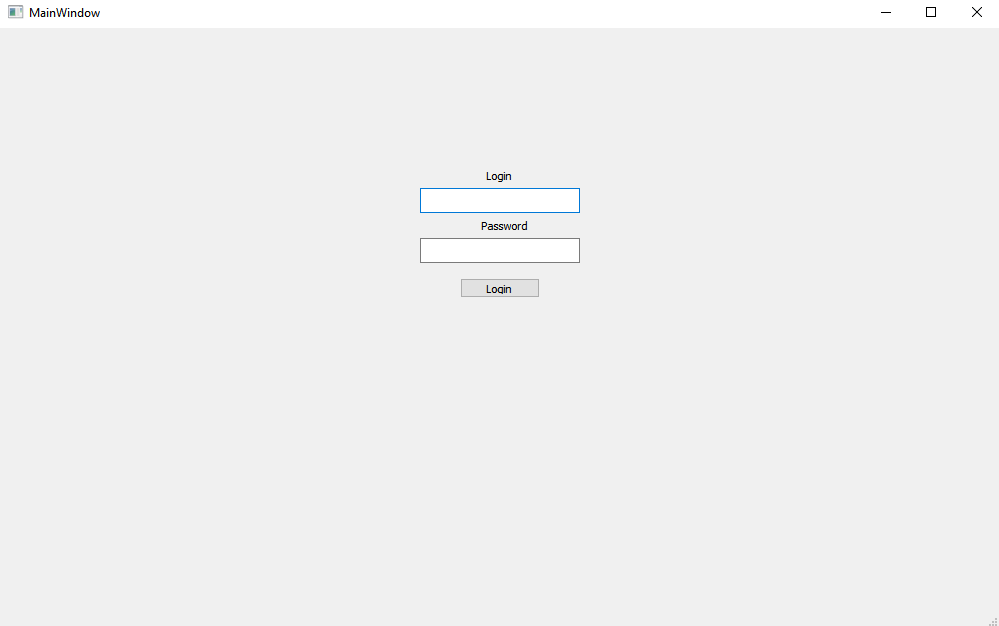


Рисунок 2.4 – форма авторизации

* Выполнена отладка обработки полученных результатов компиляции, ошибка на одном из этапов компиляции останавливает вывод последующих.
* Добавлено всплывающее сообщение в случае некорректных данных авторизации. Так как сайт кафедры не возвращает код ошибки в случае неправильной авторизации, используется адрес перенаправления для проверки статуса авторизации.

Для создания конечной версии исполняемого файла использовался qtwindeploy, позволяющий добавлять к исполняемому файлу все необходимые библиотеки. Возникли проблемы с тем, что в результатах qtwindeploy не хватало необходимых библиотек (libwinpthread-1.dll, libgcc\_s\_seh-1.dll, libstdc++6.dll), поэтому их пришлось вручную копировать из файлов Qt.

2.3 Доступ к результатам

Отчет и все полученные результаты практики расположены папке //MSERVER/praktika\_2021/Волков, а также в репозитории GitHub и отправлены руководителю в виде ссылки на репозиторий: <https://github.com/avolkov00/TestEditor/releases/tag/v1.1>

# 

# 3 Заключение

В ходе выполнения данного индивидуального проекта была создана программа для редактирования и проверки правильности тестов для системы автоматической проверки программ. Приложение полностью работает и интегрировано с системой авторизации сайта кафедры 536.

В ходе работы был изучен фреймворк Qt, получены базовые навыки работы с HTTP запросами и HTTP REST API, использования и подключения сторонних библиотек и виджетов, изучен формат данных JSON, процесс компиляции программ, ANSI коды и система авторизации сайта кафедры.

# Список использованных источников

1. <https://doc.qt.io/>
2. <https://stackoverflow.com/questions/26500429/qtextedit-and-colored-bash-like-output-emulation>
3. <https://github.com/Megaxela/QCodeEditor>
4. <https://lecturesnet.readthedocs.io/net/requests/qt.html>