МОСКОВСКАЯ ПРЕДПРОВЕССИОНАЛЬНАЯ ОЛИМПИАДА 2020/21

ИНФОРМАЦИОННО-КОНСТРУКТОРСКОЕ НАПРАВЛЕНИЕ

ПРОФИЛЬ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

**Информационная система**

**автоматизированного тестирования учащихся.**

|  |  |
| --- | --- |
|  | 11 класс  ГБОУ школа 641 им. С. Есенина  Трифонов Дмитрий Алексеевич |
|  |  |

Москва — 2021

**ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ**

**OS:** [Linux/X11](https://doc.qt.io/qt-5/linux.html), [macOS](https://doc.qt.io/qt-5/macos.html), [Windows](https://doc.qt.io/qt-5/windows.html)

**Оперативная память**: от 128 мб.

При продолжении разработки возможно использование на [IOS](https://doc.qt.io/qt-5/ios.html) и [Android](https://doc.qt.io/qt-5/android.html).

Все зависимости проекта исходят от зависимостей Qt (для серверного приложения добавляются зависимости PostgreSQL).

**ОБОСНОВАНИЕ ВЫБОРЯ ЯЗЫКА ПРОГРАММИРОВАНИЯ И**

**ИСПОЛЬЗУЕМЫХ ПРОГРАММНЫХ СРЕДСТВ**

**Используемые языки программирования:**

* **SQL**
* **C++**

Выбор C++ в качестве основного языка программирования обоснован по нескольким причинам:

* Глубокие исторические корни языка
* Надёжность и безопасность
* Высокая скорость
* Наличие ООП

**Используемые программные средства:**

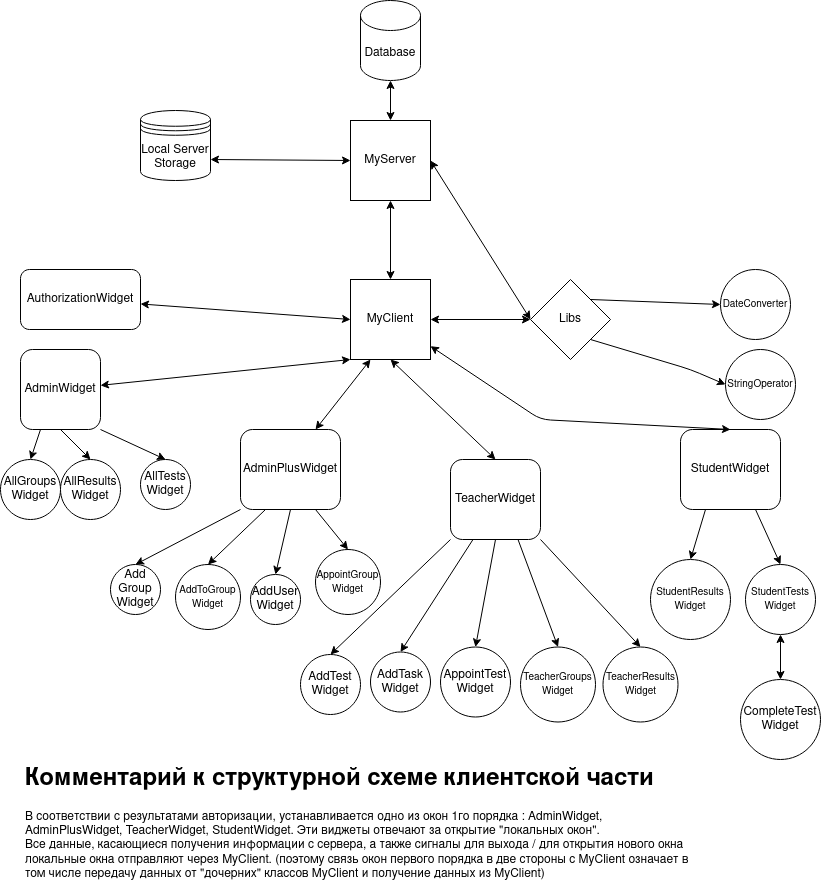
* **Среда разработки Qt 5.15 ( + необходимые компоненты (qmake, gcc, gdb, …) )**

Причины выбора данного фреймворка:

* Надёжность и безопасность (разрабатывается более 20 лет)
* Наличие документации
* Большое разнообразие предлагаемых средств
* Кроссплатформенность
* Возможность бесплатного использования для некоммерческих проектов(GPL)
* **Система управления базами данных PostgreSQL**

Причины выбора данной системы управления базами данных:

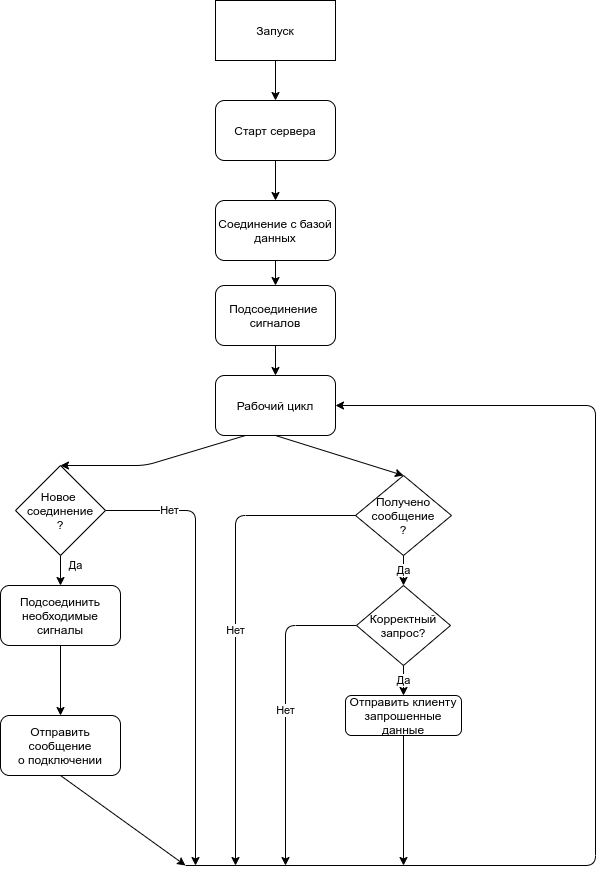
* Распространяется под лицензией postgreSQL
* Бережное отношение к данным
* Поддерживается на большом количестве ОС
* Следствие из первого пункта — наличие в Qt встроенного драйвера QPSQL
* **Средства для разработки - git, pgAdmin4**

**СТРУКТУРНАЯ СХЕМА ПРОГРАММНОГО ПРОДУКТА**

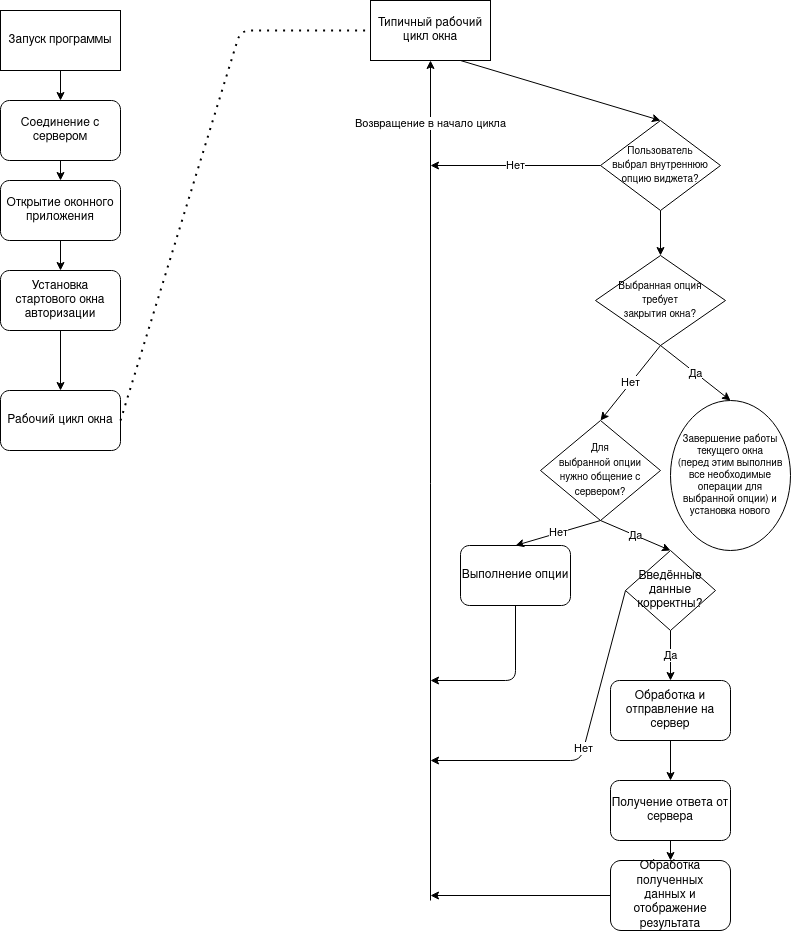
**Комментарий к структурной схеме клиентской части**

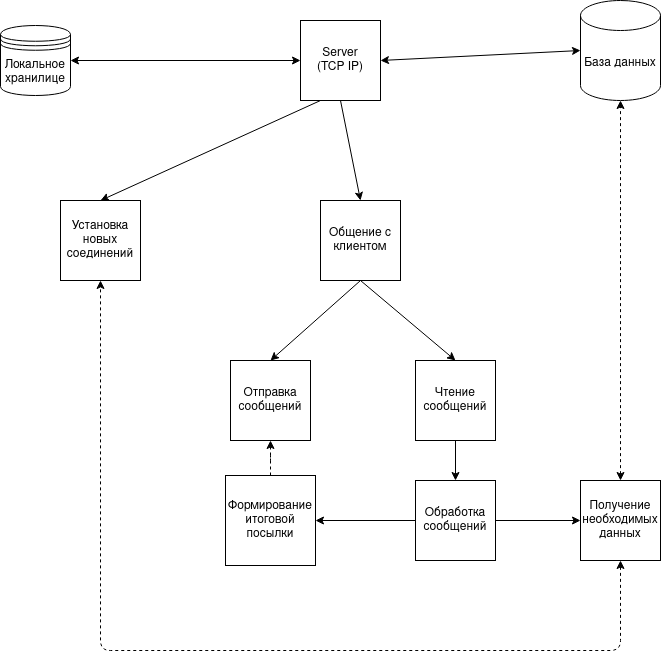
В соответствии с результатами авторизации, если авторизация прошла успешно, устанавливается одно из окон 1-го порядка: AdminWidget(окно администратора учебного процесса), AdminPlusWidget(окно системного администратора), TeacherWidget(окно учителя), StudentWidget(окно ученика). Эти виджеты отвечают за открытие «локальных окон». Все данные, касающтеся получения информации с сервера, а также сигналы для выхода / для открытия нового окна локальные окна отправляют с помощью класса MyClient (поэтому связь окон первого порядка двойной стрелкой с MyClient означает в том числе передачу данных от устанавливаемых окон а также получение ими данных из MyClient).

**WORKFLOW ДИАГРАММА СЕРВЕРНОЙ ЧАСТИ**

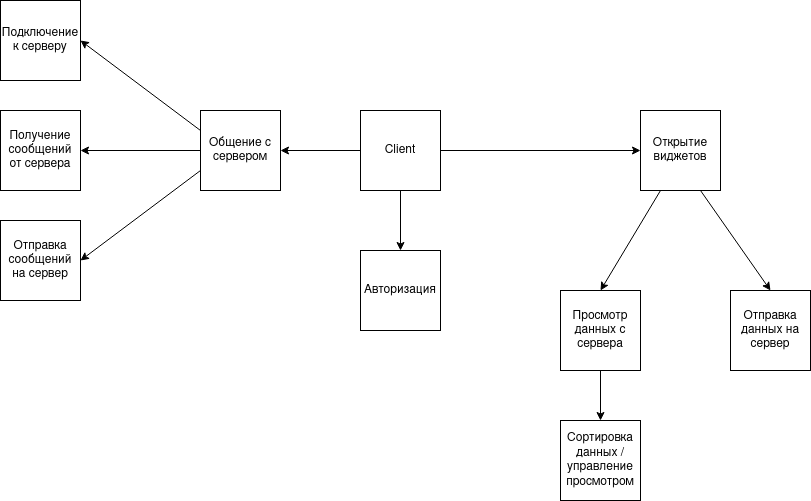
**Важное замечание:** в приведённой схеме приведён общий принцип работы, без излишней детализации проверок на ошибки (диагностики), но стоит учитывать, что диагностика есть на всех этапах работы.

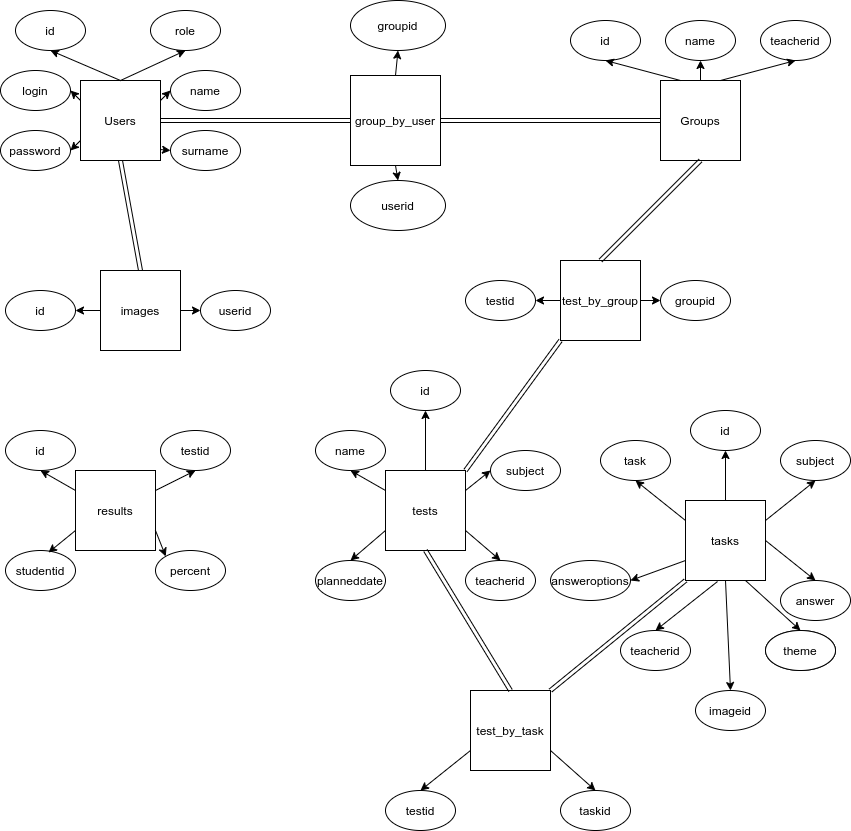
**WORKFLOW ДИАГРАММА КЛИЕНТСКОЙ ЧАСТИ**

**Важное замечание:** в приведённой схеме приведён общий принцип работы, без излишней детализации проверок на ошибки (диагностики), но стоит учитывать, что диагностика есть на всех этапах работы.

**ФУНКЦИОНАЛЬНАЯ СХЕМА СЕРВЕРНОЙ ЧАСТИ**

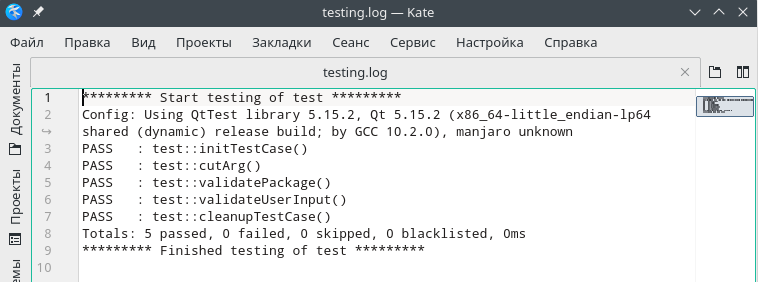
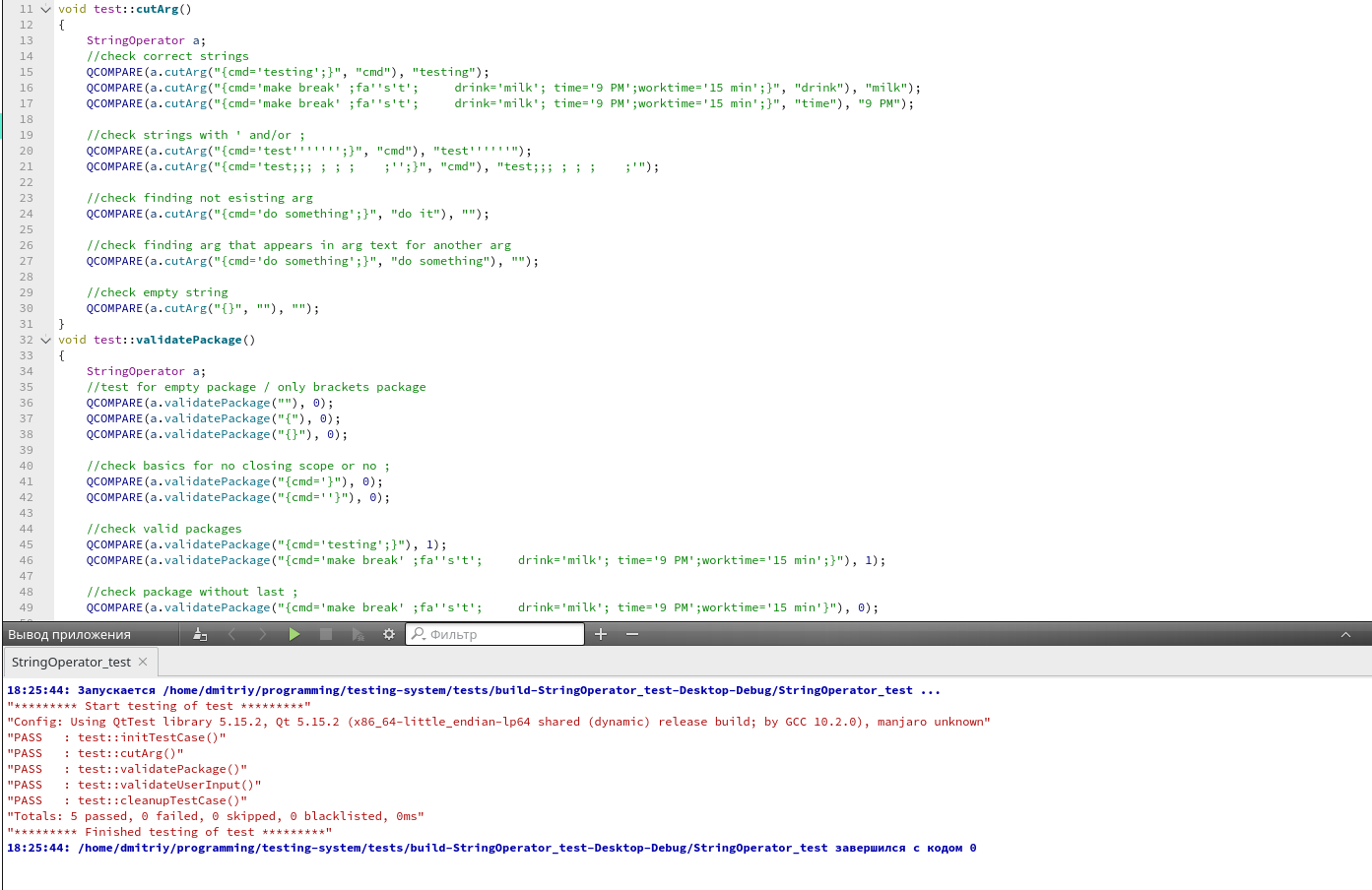
**ФУНКЦИОНАЛЬНЯ СХЕМА КЛИЕНТСКОЙ ЧАСТИ**

****

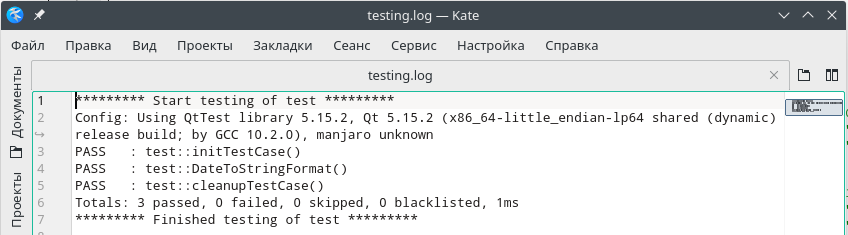
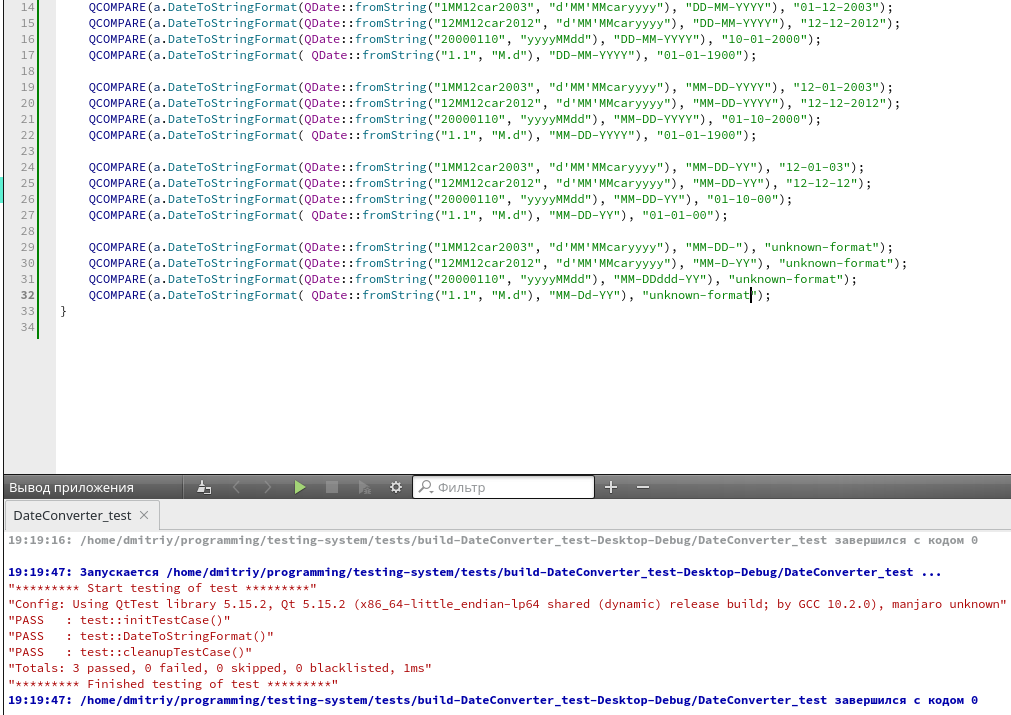
**ER МОДЕЛЬ**

**ПРОВЕДЁННЫЕ ИСПЫТАНИЯ**

**Обработчик строк, используемых в составлении посылок для общения с сервером:**

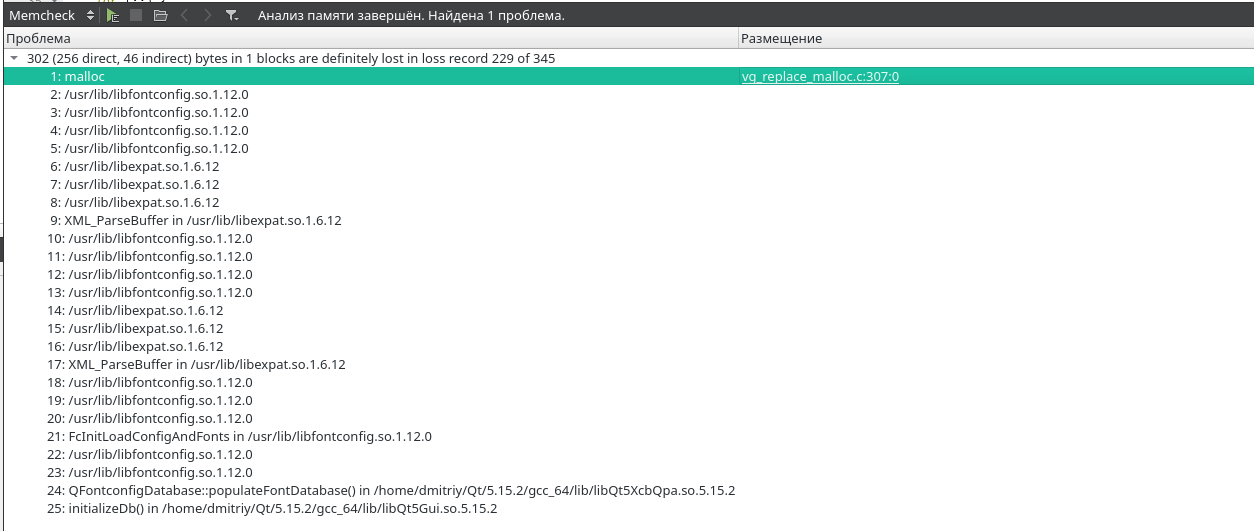


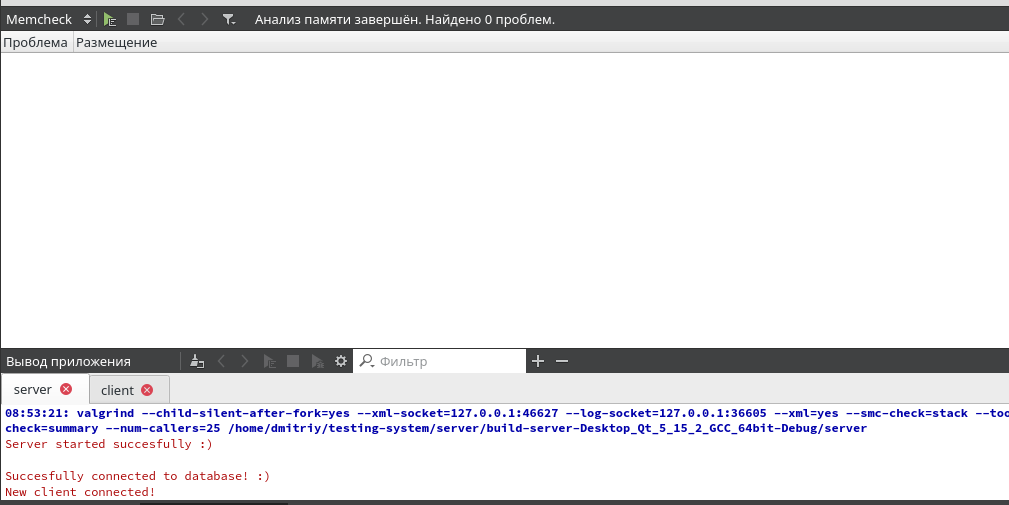
**Обработчик типа данных QDate, конвертирующий тип в строку по заданному параметру форматирования:**

****

**АНАЛИЗ VALGRIND**

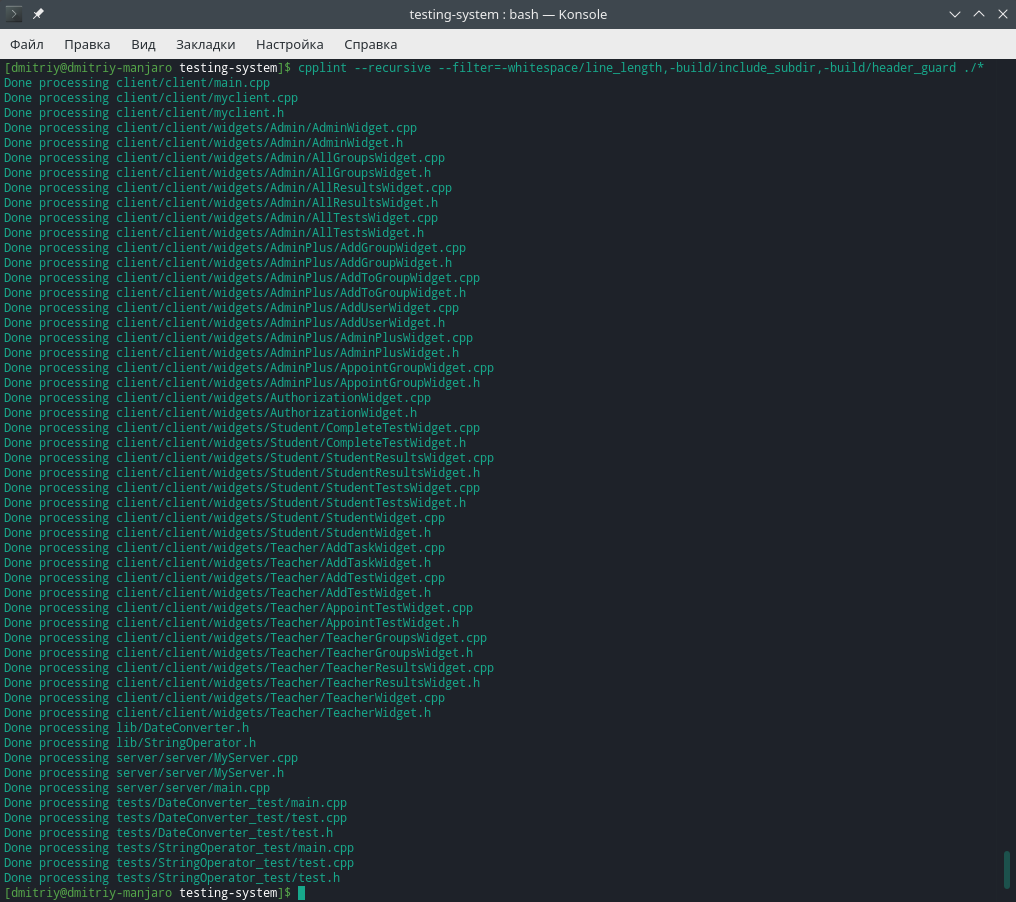
**Важное замечание:** все ошибки, полученные при анализе клиента — это внутренние ошибки Qt, появляющиеся из-за использования GUI (и не связанные с какими-то ошибками в коде).

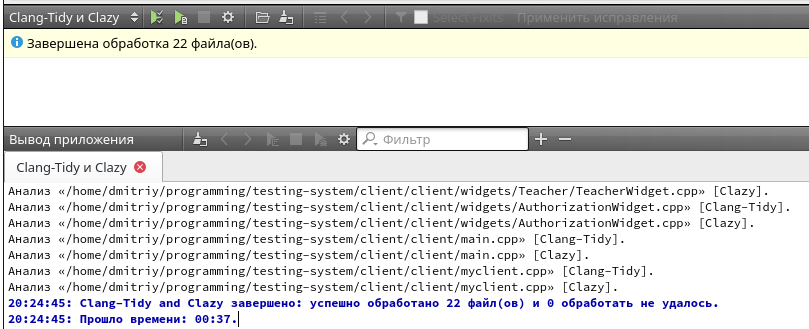
Client

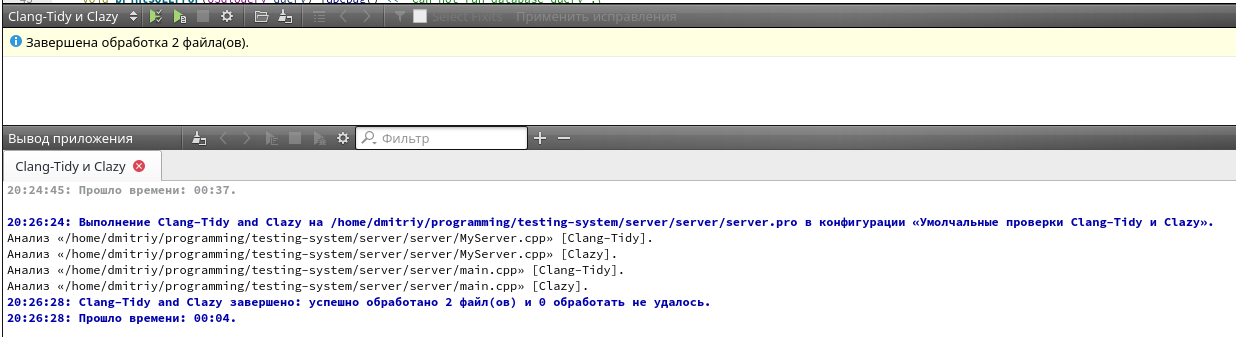
Server

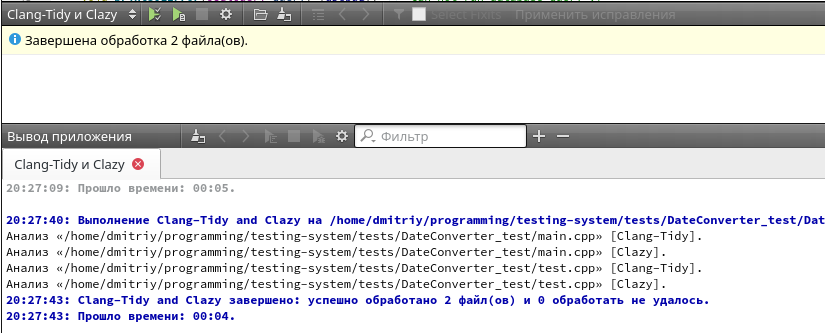
**АНАЛИЗ КОДА**

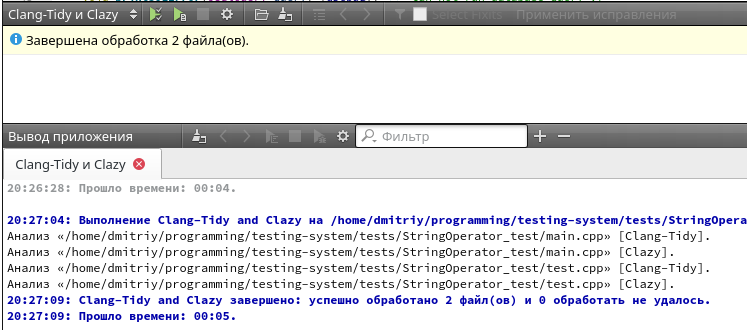
[**Cpplint**](https://github.com/google/styleguide/blob/gh-pages/cpplint/cpplint.py) **(google c++ styleguide):**

**Clang-Tidy, Clazy:**

Client

Server

Date Converter Test

String Operator Test

**ИСХОДНЫЙ КОД**

<https://github.com/Djivs/testing-system>