# CoChat

Создано системой Doxygen 1.8.13

# Оглавление

# Глава 1

# Иерархический список классов

## 1.1 Иерархия классов

Иерархия классов.

# Глава 2

# Алфавитный указатель классов

## 2.1 Классы

Классы с их кратким описанием.

# Глава 3

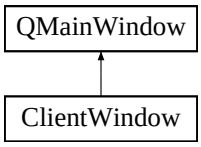
# Классы

## 3.1 Класс ClientWindow

Класс, предоставляющий интерфейс пользователя

#include <clientwindow.h>

Граф наследования:ClientWindow:




### Открытые слоты

- void on_SearchLine_returnPressed ()

  Слот, отвечающий за отправку поискового запроса на сервер по нажатию кнопки Return.
- void onRead ()

  Слот, отвечающий за получение входящих данных
- void on_NameInput_returnPressed ()

  Слот, отвечающий за сохранение и отправку на сервер имени пользователя
- void on_FriendList_itemDoubleClicked (QListWidgetItem *item)

  Слот, определяющий имя пира, который был выбран пользователем (используется для отправки сообщений выбранному пользователю)
- void on_MsgInput_returnPressed ()

  Слот, служащий для считывания текста из поля ввода по нажатию клавиши return, добавляющий сообщение в историю и вызывающий функцию отправки сообщения этому пиру
- void ConnDetector ()

  Слот для обнаружения входящих соединений
- void on_UpdateListButton_clicked ()

  Слот, отвечающий за запрос серверу на получение списка всех пиров в сети. Активируется по нажатию кнопки Update.
- void on_checkBox_toggled (bool checked)

Слот, информирующий о нажатии галочки "Private" при входе в сеть. Устанавливает соответствующий статус(Private, Public)текущему пользователю

- void on_pushButton_clicked ()

  Слот для открытия Issue Creator по нажатию соответствующей кнопки
- void on_SearchButton_clicked ()

  Слот для отправки запроса о поиске пира на сервер
- void on_LoginButton_clicked ()

  Слот, аналогичный on_NameInput_returnPressed()
- void closeEvent (QCloseEvent *e)

  Слот, перехватывающий закрытие окна приложения и отправляющий запрос на сервер о выходе из сети перед закрытием

## Открытые члены

- ClientWindow (int Port, QString address, QWidget *parent=0)

  Конструктор, принимающий на вход порт и адрес информационного сервера
- ~ClientWindow ()

  Деструктор

## Закрытые члены

- int Resolver (const QString &Data)

  Функция, предназначенная для распознавания типа данных, переданных в неё
- void SendMessageToPeer (const QString &PeerName, QString Message)

  Функция для отправки сообщения пиру
- void ConnectToPeer (const QString &IP, int Port, const QString &UserName)

  Функция для отправки запроса на соединение другому пиру
- void SendConnectRequest (const QString &PeerName)

  Функция для отправки запроса на соединение другому пиру если статус текущего пользователя - приватный.
- void ParseAllUsersData (QString Response)

  Парсер списка пользователей, полученного от сервера
- QString Encrypt (QString &Message, QString Key)

  Функция зашифровки сообщения с заданным ключом Функция шифрует сообщение по алгоритму "Кузнечик" (ГОСТ Р 34.12-2015) с заданным ключом. Подробнее ознакомиться с реализацией можно перейдя по ссылке в источниках.
- QString Decrypt (QString &Message, QString Key)

  Функция расшифровки сообщения с заданным ключом Функция дешифрует сообщение по алгоритму "Кузнечик" (ГОСТ Р 34.12-2015) с заданным ключом. Подробнее ознакомиться с реализацией можно перейдя по ссылке в источниках.
- void GenKeyParams ()

  Функция для генерации параметров ключей (Prime, Generator, PrivNumb, PublicNumb) и самой пары ключей.
- CryptoPP::SecByteBlock IncomingSessionKeyGen (const QString &Username, CryptoPP::Integer prime, CryptoPP::Integer generator, CryptoPP::SecByteBlock publicNumb)

  Функция для генерации параметров и выработки сеансового ключа с пиром, приславшим свои публичные данные для обмена ключами.
- void GettingAgreement (const QString &Username, CryptoPP::SecByteBlock publicNumb)

  Функция для согласования общего сеансового ключа
- void SendGeneratedPublicKey (const QString &UserName, CryptoPP::SecByteBlock publicKey)

  Функция отправки публичного ключа заданному пиру
- Peer * SearchPeerByName (const QString &Name)

  Функция поиска пира в структуре по имени.
- int IndexPeerByName (const QString &Name)

  Функция возврата индекса пира в структуре по имени.

### Закрытые данные

- CryptoPP::DH dh

  Объект класса DH, необходимый для реализации алгоритма Диффи-Хеллмана. Подробнее можно узнать в официальной документации библиотеки CryptoPP.
- CryptoPP::Integer Prime

  Параметр необходимый для генерации пары ключей. Подробнее можно узнать в официальной документации библиотеки CryptoPP.
- CryptoPP::Integer Generator

  Параметр необходимый для генерации пары ключей. Подробнее можно узнать в официальной документации библиотеки CryptoPP.
- CryptoPP::SecByteBlock PrivNumb

  Публичное сгенерированное с помощью вышеуказанных параметров число. Подробнее можно узнать в официальной документации библиотеки CryptoPP.
- CryptoPP::SecByteBlock PublicNumb

  Приватное сгенерированное с помощью вышеуказанных параметров число. Подробнее можно узнать в официальной документации библиотеки CryptoPP.
- CryptoPP::SecByteBlock MySecretKey

  Приватный ключ. Подробнее можно узнать в официальной документации библиотеки CryptoPP.
- QVector< Peer > Peers

  Структура для хранения списка известных пиров
- std::unique_ptr< QTcpSocket > ServerSocket

  Указатель на сокет инициализированный адресом и портом информационного сервера
- std::unique_ptr< QTcpServer > ThisListenSocket

  Указатель на слушающий сокет текущего пользователя
- QString Destination

  Имя адресата
- QString NickName

  Имя текущего пользователя
- QString ServerIP

  IPv4 адрес сервера
- int ServerPort

  Порт информационного сервера
- bool ConnectedToServer = false

  Флаг, сигнализирующий о успешном присоединении к серверу. Если соединения нет - поиск недоступен.
- bool Private = false

  Флаг, сигнализирующий о статусе пользователя.
- Ui::ClientWindow * ui

### 3.1.1 Подробное описание

Класс, предоставляющий интерфейс пользователя

### 3.1.2 Конструктор(ы)

#### 3.1.2.1 ClientWindow()

```
ClientWindow::ClientWindow (
                int Port,
                QString address,
                QWidget * parent = 0 )
```

Конструктор, принимающий на вход порт и адрес информационного сервера

Аргументы

| | | |
|---|---|---|
| in | Port | Порт |
| in | address | IPv4 адрес |

### 3.1.3 Методы

#### 3.1.3.1 ConnectToPeer()

```
void ClientWindow::ConnectToPeer (
                const QString & IP,
                int Port,
                const QString & UserName )   [private]
```

Функция для отправки запроса на соединение другому пиру

Функция инициализирует новый сокет переданными данными (IP, Port), создается новый объект класса Peer, использующийся для хранения данных о пользователе, заносит объект в структуру для хранения списка "знакомых" пиров, после чего этому пиру автоматически отправляется запрос на соединение. Получив такой запрос, пир автоматически сохранит пользователя отправившего этот запрос в своей структуре и сможет отправлять ему сообщения. Таким образом пиры будут "знать" друг друга без помощи информационного сервера

Аргументы

| | | |
|---|---|---|
| in | IP | IPv4 адрес пира, которому будет отправляться запрос. |
| in | Port | Порт |
| in | UserName | Имя пира |

Возвращает

Ничего не возвращает

#### 3.1.3.2 Decrypt()

```
QString ClientWindow::Decrypt (
                QString & Message,
                QString Key )   [private]
```

Функция расшифровки сообщения с заданным ключом Функция дешифрует сообщение по алгоритму "Кузнечик" (ГОСТ Р 34.12-2015) с заданным ключом. Подробнее ознакомиться с реализацией можно перейдя по ссылке в источниках.

Аргументы

| | | |
|---|---|---|
| in | Message | Сообщение, которое необходимо дешифровать. |
| in | Key | Ключ для дешифровки /return Возвращает строку - расшифрованное сообщение |

#### 3.1.3.3 Encrypt()

QString ClientWindow::Encrypt (
QString & Message,
QString Key ) [private]

Функция зашифровки сообщения с заданным ключом Функция шифрует сообщение по алгоритму "Кузнечик" (ГОСТ Р 34.12-2015) с заданным ключом. Подробнее ознакомиться с реализацией можно перейдя по ссылке в источниках.

Аргументы

| | | |
|---|---|---|
| in | Message | Сообщение, которое необходимо зашифровать. |
| in | Key | Ключ для зашифровки |

Возвращает

Возвращает строку - зашифрованное сообщение

#### 3.1.3.4 GenKeyParams()

void ClientWindow::GenKeyParams ( ) [private]

Функция для генерации параметров ключей (Prime, Generator, PrivNumb, PublicNumb) и самой пары ключей.

Подробнее о параметрах можно прочесть в описании алгоритма Диффи-Хеллмана.

Возвращает

Ничего не возвращает

#### 3.1.3.5 GettingAgreement()

void ClientWindow::GettingAgreement (
const QString & Username,
CryptoPP::SecByteBlock publicNumb ) [private]

Функция для согласования общего сеансового ключа

С помощью полученного от пира его публичного ключа, сеансовый ключ согласуется и заносится в структру этого пира для последующей шифровки и дешифровки сообщений.

Аргументы

| | | |
|---|---|---|
| in | Username | Имя пира |
| in | publicNumb | Публичный ключ, полученный пиром, приславшим запрос, при выработке сеансового ключа. |

Возвращает

Ничего не возвращает

#### 3.1.3.6 IncomingSessionKeyGen()

```
CryptoPP::SecByteBlock ClientWindow::IncomingSessionKeyGen (
        const QString & Username,
        CryptoPP::Integer prime,
        CryptoPP::Integer generator,
        CryptoPP::SecByteBlock publicNumb )   [private]
```

Функция для генерации параметров и выработки сеансового ключа с пиром, приславшим свои публичные данные для обмена ключами.

Вызывается при получении запроса от пира на обмен ключами. Все необходимые данные он предоставляет в запросе. Подробнее о необходимых параметрах можно прочесть в описании алгоритма Диффи-Хеллмана. После генерации ключ заносится в структуру этого пира для последующей шифровки и дешифровки сообщений.

Аргументы

| | | |
|---|---|---|
| in | Username | Имя пира |
| in | prime | Prime параметр |
| in | generator | Generator параметр |
| in | publicNumb | Публичный ключ, полученный пиром, приславшим запрос, при выработке ключа. |

Возвращает

Возвращает сеансовый ключ для данного пира

#### 3.1.3.7 IndexPeerByName()

```
int ClientWindow::IndexPeerByName (
        const QString & Name )   [private]
```

Функция возврата индекса пира в структуре по имени.

Аргументы

| | | |
|---|---|---|
| in | Name | Имя искомого пира |

Возвращает

Возвращает индекс пира

#### 3.1.3.8 onRead

void ClientWindow::onRead ( ) [slot]

Слот, отвечающий за получение входящих данных

Передает данные в Resolver, который вернет информацию о формате пришедших данных. После определения формата (сообщение, ответ от сервера, запрос на обмен ключами и т.п.) - функция передает данные на обработку

#### 3.1.3.9 ParseAllUsersData()

void ClientWindow::ParseAllUsersData (
QString Response ) [private]

Парсер списка пользователей, полученного от сервера

Выполняет парсинг полученной на вход строки, после чего заносит каждого пользователя в структуру. Вызывается при входе в сеть, либо при обновлении всего списка [in] Response Строка, содержащая информацию обо всех пользователях в сети

Возвращает

Ничего не возвращает

#### 3.1.3.10 Resolver()

int ClientWindow::Resolver (
const QString & Data ) [private]

Функция, предназначенная для распознавания типа данных, переданных в неё

Функция определяет приписанный к началу данных флаг, который указывает на тип данных (сообщение, ответ от сервера, различные запросы от других пиров).

Аргументы

| | | |
|---|---|---|
| in | Data | Данные, тип которых необходимо определить |

Возвращает

Число - код, соответствующий типу пришедших данных.

#### 3.1.3.11 SearchPeerByName()

Peer* ClientWindow::SearchPeerByName (
const QString & Name ) [private]

Функция поиска пира в структуре по имени.

Аргументы

| | | |
|---|---|---|
| in | Name | Имя искомого пира |

Возвращает

Возвращает указатель на объект пира

#### 3.1.3.12 SendConnectRequest()

void ClientWindow::SendConnectRequest (
const QString & PeerName ) [private]

Функция для отправки запроса на соединение другому пиру если статус текущего пользователя - приватный.

Запрос на соединение не должен отправляться от приватного пира без необходимости, чтобы не раскрывать его. Если такая необходимость появляется - запрос отправляется, тогда адресат будет знать приватного пользователя.

Аргументы

| | | |
|---|---|---|
| in | UserName | Имя пира (необходимо для поиска его в своей структуре, так как приватный пир имеет информацию обо всех публичных, соответственно они есть в его структуре, остается их только найти) |

Возвращает

Ничего не возвращает

#### 3.1.3.13 SendGeneratedPublicKey()

void ClientWindow::SendGeneratedPublicKey (
const QString & UserName,
CryptoPP::SecByteBlock publicKey ) [private]

Функция отправки публичного ключа заданному пиру

Аргументы

| | | |
|---|---|---|
| in | UserName | Имя пира |
| in | publicKey | Публичный ключ для отправки |

Возвращает

Ничего не возвращает

#### 3.1.3.14 SendMessageToPeer()

void ClientWindow::SendMessageToPeer (
const QString & PeerName,
QString Message ) [private]

Функция для отправки сообщения пиру

Функция ищет нужного пира в списке пиров и отправляет ему введенное в окне ввода сообщение

Аргументы

| | | |
|---|---|---|
| in | PeerName | Имя адресата |

Возвращает

Ничего не возвращает

Объявления и описания членов класса находятся в файле:

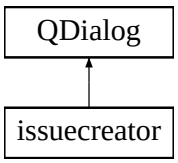
- /Users/antonukhanov/P2P-Chat/Client/clientwindow.h

## 3.2 Класс issuecreator

Класс, предоставляющий интерфейс для создания Github Issues.

#include <issuecreator.h>

Граф наследования:issuecreator:




## Открытые члены

- issuecreator (QWidget *parent=0)
  Конструктор по умолчанию
- issuecreator (const QString &selectedText, QWidget *parent=0)
  Конструктор для случая с выделенным текстом.
- bool IsInternetConnected ()
  Проверка интернет-соединения. Проверка соединения с помощью стандартных средств Qt.
- void WriteTokenToFile (const std::string &Token)
  Запись в config-файл токена.
- QString ReadTokenFromFile ()
  Чтение токена из config-файла.
- QString ParseToken (QString Data)
  Парсинг полученных данных от Github и получение токена из них.
- QString GetGithubToken (const QString &Login, const QString &Pass)
  Отправка запроса на получение Github токена.
- QNetworkRequest DoIssueRequest ()
  Задает параметры http-запроса для создания Github Issue.
- QString DoIssueJSON ()
  Создает JSON с парамятрами для создания Github Issue.

## Закрытые слоты

- void on_Send_clicked ()
  Слот, отвечающий за отправку запроса на Github с целью создания Issue.

## Закрытые данные

- bool isTokenWrited = false
  Флаг, сигнализирующий о наличии токена в файле config.
- QString Token
  Токен
- QString IssueDescription
  Описание проблемы
- Ui::issuecreator * ui

### 3.2.1 Подробное описание

Класс, предоставляющий интерфейс для создания Github Issues.

### 3.2.2 Конструктор(ы)

#### 3.2.2.1 issuecreator()

```
issuecreator::issuecreator (
            const QString & selectedText,
            QWidget * parent = 0 )
```

Конструктор для случая с выделенным текстом.

Весь выделенный текст в окне сообщений будет перенесен в панель ввода Description.

### 3.2.3 Методы

#### 3.2.3.1 DoIssueJSON()

```
QString issuecreator::DoIssueJSON ( )
```

Создает JSON с парамятрами для создания Github Issue.

Возвращает

Строка - JSON.

#### 3.2.3.2 DoIssueRequest()

```
QNetworkRequest issuecreator::DoIssueRequest ( )
```

Задает параметры http-запроса для создания Github Issue.

Возвращает

Объект QNetworkRequest.

#### 3.2.3.3 GetGithubToken()

```
QString issuecreator::GetGithubToken (
            const QString & Login,
            const QString & Pass )
```

Отправка запроса на получение Github токена.

Аргументы

| | | |
|---|---|---|
| in | Login | Логин Github. |
| in | Pass | Пароль от аккаунта Github. |

#### 3.2.3.4 IsInternetConnected()

bool issuecreator::IsInternetConnected ( )

Проверка интернет-соединения. Проверка соединения с помощью стандартных средств Qt.

Возвращает

False если отсутствует интернет-соединение. Иначе - true.

#### 3.2.3.5 ParseToken()

QString issuecreator::ParseToken (
QString Data )

Парсинг полученных данных от Github и получение токена из них.

Аргументы

| | | |
|---|---|---|
| in | Data | Данные от Github. |

Возвращает

Строка - токен.

#### 3.2.3.6 ReadTokenFromFile()

QString issuecreator::ReadTokenFromFile ( )

Чтение токена из config-файла.

Возвращает

Строка - токен.

#### 3.2.3.7 WriteTokenToFile()

void issuecreator::WriteTokenToFile (
const std::string & Token )

Запись в config-файл токена.

Аргументы

| | | |
|---|---|---|
| in | Token | Токен для записи в файл. |

Объявления и описания членов класса находятся в файле:

- /Users/antonukhanov/P2P-Chat/Client/issuecreator.h

## 3.3 Класс Peer

Класс, предоставляющий интерфейс для хранения данных о каждом "знакомом" пире

#include <peer.h>

### Открытые члены

- Peer ()
  Конструктор по умолчанию
- Peer (QString Name, std::shared_ptr< QTcpSocket > Socket)
  Конструктор, принимающий имя пира и указатель на сокет, инициализированный его адресом
- Peer (QString Name, std::shared_ptr< QTcpSocket > Socket, QString Key)
  Конструктор, принимающий имя пира, указатель на сокет(инициализированный его адресом) и сеансовый ключ для этого пира
- void SetSessionKey (QString Key)
  Set-функция для инициализации сеансового ключа этого пира

### Открытые атрибуты

- QString PeerName
  Имя пира
- std::shared_ptr< QTcpSocket > PeerSocket
  Указатель на сокет, инициализированный адресом пира
- QString SessionKey
  Сеансовый ключ для этого пира
- QVector< QString > MessagesHistory
  История сообщений с пиром

### 3.3.1 Подробное описание

Класс, предоставляющий интерфейс для хранения данных о каждом "знакомом" пире

Объявления и описания членов класса находятся в файле:

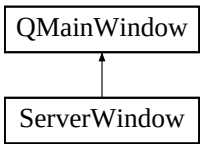
- /Users/antonukhanov/P2P-Chat/Client/peer.h

## 3.4 Класс ServerWindow

Класс, предоставляющий интерфейс сервера

#include <serverwindow.h>

Граф наследования:ServerWindow:




### Открытые члены

- ServerWindow (QWidget *parent=0)
- int Resolver (const QString &Data)

  Функция, предназначенная для распознавания типа данных, переданных в неё

### Закрытые слоты

- void on_Starting_clicked ()

  Слот, включающий сервер при нажатии кнопки Start.
- void on_Stopping_clicked ()

  Слот, выключающий сервер при нажатии кнопки Stop.
- void ConnectClient ()

  Слот, отвечающий за обработку входящих соединения и передачу их в слот ListeningClient()
- void ListeningClient ()

  Слот, "слушающий" приходящие данные от пользователя

### Закрытые члены

- void DeleteOfflineUser (const QString &Name)

  Функция для удаления из базы данных вышедшего из сети пользователя
- void UpdateUsersList ()

  Функция для обновления списка пользователей в экране
- void AddNewUser (const QString &NickName, const QString &Status, const QString &Address)

  Функция, предназначенная для добавления пользователя в базу данных
- void SendAllUsers (QTcpSocket *ClientSocket)

  Функция, предназначенная для отправки списка всех пиров пользователю
- QString SearchUser (const QString &Username)

  Функция, предназначенная для поиска пользователя в базе данных
- bool IsUniqueUser (const QString &Username)

  Функция, предназначенная для проверки имени пользователя на уникальность

### Закрытые данные

- QString ServerAddress = "0.0.0.0"
- int Port = 9090
- bool ServerOn = false
- Ui::ServerWindow * ui

  Переменная, отвечающая за состояние сервера (включен или выключен)
- std::unique_ptr< QTcpServer > tcpServer
- QSqlDatabase DataBase

  Указатель на слушающий сокет сервера
- QMap< QString, QTcpSocket * > Users

  Объект базы данных

## 3.4.1 Подробное описание

Класс, предоставляющий интерфейс сервера

## 3.4.2 Методы

### 3.4.2.1 AddNewUser()

```
void ServerWindow::AddNewUser (
                const QString & NickName,
                const QString & Status,
                const QString & Address )   [private]
```

Функция, предназначенная для добавления пользователя в базу данных

Аргументы

| | | |
|---|---|---|
| in | NickName | Имя пользователя |
| in | Status | Статус (приватный или публичный) |
| in | Address | IP-адрес |

### 3.4.2.2 IsUniqueUser()

```
bool ServerWindow::IsUniqueUser (
                const QString & Username )   [private]
```

Функция, предназначенная для проверки имени пользователя на уникальность

Аргументы

| | | |
|---|---|---|
| in | UserName | Имя пользователя |

#### 3.4.2.3 Resolver()

int ServerWindow::Resolver (
const QString & Data )

Функция, предназначенная для распознавания типа данных, переданных в неё

Функция определяет приписанный к началу данных флаг, который указывает на тип данных (сообщение, ответ от сервера, различные запросы от других пиров).

Аргументы

| | | |
|---|---|---|
| in | Data | Данные, тип которых необходимо определить |

Возвращает

Число - код, соответствующий типу пришедших данных.

#### 3.4.2.4 SearchUser()

QString ServerWindow::SearchUser (
const QString & Username ) [private]

Функция, предназначенная для поиска пользователя в базе данных

Аргументы

| | | |
|---|---|---|
| in | UserName | Имя пользователя |

#### 3.4.2.5 SendAllUsers()

void ServerWindow::SendAllUsers (
QTcpSocket * ClientSocket ) [private]

Функция, предназначенная для отправки списка всех пиров пользователю

Аргументы

| | | |
|---|---|---|
| in | ClientSocket | Сокет, инициализированный адресом пользователя |

Объявления и описания членов класса находятся в файле:

- /Users/antonukhanov/P2P-Chat/InfoServer/serverwindow.h

# Предметный указатель