Утвержден

Н.С.00001-01 33 01-1-ЛУ

программа «Нейронная сеть ПМХГ»

Программа

Описание кода программы

Н.С.00001-01 33 01-1-ЛУ

Листов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Инв. № подп. | Подп. и дата | Взам. Инв.№ | Инв. № дубл. | Подп. и дата |
|  |  |  |  |  |

2021 г.

программа «Нейронная сеть ПМХГ»

Программа

Описание кода программы

Лист утверждения

Н.С.00001-01 33 01-1-ЛУ

Листов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Инв. № подп. | Подп. и дата | Взам. Инв.№ | Инв. № дубл. | Подп. и дата |
|  |  |  |  |  |

2021 г.

ИНТЕРФЕЙС

**window.h**

Класс "Window" отвечает за интерфейс

class **Window** : public QMainWindow

{

Q\_OBJECT

public:

**Window**(QWidget \*parent = nullptr);

~***Window***();

Функция успешного завершения авторизации

void **Auth**();

Функция неудачного завершения авторизации

void **FailAuth**();

Функция успешного завершения регистрации

void **Reg**();

Функция неудачного завершения авторизации//

void **FailReg**();

private slots:

Нажатие на конопку "Authorization"

void **on\_Authorization\_clicked**();

Нажатие на конопку "Cancel"

void **on\_Cancel\_clicked**();

Нажатие на конопку "OK"

void **on\_OK\_clicked**();

Нажатие на конопку "Registration"

void **on\_Registretion\_clicked**();

Нажатие на кнопку "OK"в окне регистрации

void **on\_buttonBox\_accepted**();

Нажатие на кнопку "Cancel"в окне регистрации//

void **on\_buttonBox\_rejected**();

private:

Ui::Window \*ui;

MyTcpClient \*clientAuth;

};

**Window.cpp**

Нажатие на кнопку "Authorization"

Данная функция реализует действия, которые будут происходить после нажатия кнопки "Authorization".

Начальное окно закрывается, а вместо него открывается окно авторизации.

void Window::**on\_Authorization\_clicked**()

{

ui->winfirst->*setVisible*(false);

ui->winauth->*setVisible*(true);

}

Нажатие на кнопку "Cancel".

Данная функция реализует действия, которые будут происходить после нажатия кнопки "Cancel".

Окно авторизации закрывается, а вместо него открывается начальное окно.

void Window::**on\_Cancel\_clicked**()

{

ui->winauth->*setVisible*(false);

ui->winfirst->*setVisible*(true);

}

Нажатие на кнопку "OK"

Данная функция реализует действия, которые будут происходить после нажатия кнопки "OK".

Аргументы:

QString login - логин пользователя;

QString pass - пароль пользователя.

void Window::**on\_OK\_clicked**()

{

QString login = ui->auth\_login->text();

QString password = ui->auth\_password->text();

Создание нового подключения.

MyTcpClient \*clientAuth = new MyTcpClient;

Условия, при которых авторизация будет успешна:

* login не должен быть пустым;
* pass не должен быть пустым.

if (login !="" and password !="")

{

Формируется сообщение, которое содержит пароль, логин.

После того, как сообщение было сформировано, оно отправляется на сервер.

QString message = "auth&"+login+"&"+password;

clientAuth->slot\_send\_to\_server(message);

}

Если не будет выполнено одно из условий:

* login не должен быть пустым;
* pass не должен быть пустым.

Программа выводит сообщение об ошибке. Текст сообщения: "login or password not entered!!!".

else

{

QMessageBox::critical(this, "ERROR", "login or password not entered!!!");

}

}

Нажатие на кнопку "Registration"

Данная функция реализует действия, которые будут происходить после нажатия кнопки "Registration".

Окно регистрации открывается, а начальное окно закрывается.

void Window::**on\_Registretion\_clicked**()

{

ui->winfirst->*setVisible*(false);

ui->winreg->*setVisible*(true);

}

Нажатие на кнопку "OK"в окне регистрации

Данная функция реализует действия, которые будут происходить после нажатия кнопки "OK".

Аргументы:

* QString login - логин пользователя;
* QString email - почта пользователя;
* QString pass - пароль пользователя;
* QString conpass - пароль пользователя для повторного ввода/

void Window::**on\_buttonBox\_accepted**()

{

QString login = ui->login->text();

QString email = ui->email->text();

QString pass = ui->pass->text();

QString conpass = ui->conpass->text();

Создание нового подключения

MyTcpClient \*clientAuth = new MyTcpClient;

Условия при котором регистрация будет пройдена успешна

* login не должен быть пустым;
* pass не должен быть пустым;;
* conpass не должен быть пустым;
* email не должен быть пустым.

if (login !="" and pass !="" and conpass !="" and email !="" )

{

Условие, при котором регистрация будет пройдена успешна.

Пароль пользователя и пароль, который введен еще раз, должны совпадать.

if (pass == conpass)

{

Формируется сообщение, которое содержит пароль, логин и почту.

После того, как сообщение было сформировано, оно отправляется на сервер.

QString message = "reg&"+login+"&"+pass+"&"+email;

clientAuth->slot\_send\_to\_server(message);

}

Если условие совпадения пароля и пароля, введённого второй раз, нарушена,то выводится сообшение об ошибке. Текст сообщения: "ERROR", "Password and Conpassword unequal!!!".

else

QMessageBox::critical(this, "ERROR", "Password and Conpassword unequal!!!");

}

Если не будет выполнено одно из условий:

* login не должен быть пустым;
* pass не должен быть пустым;
* conpass не должен быть пустым;
* email не должен быть пустым.

То программа выводит ошибку и сообщение: "login, password, conpassword or email not entered!!!".

else

QMessageBox::critical(this, "ERROR", "login, password, conpassword or email not entered!!!");

}

Нажатие на кнопку "Cancel"в окне регистрации

Данная функция реализует действия, которые будут происходить после нажатия кнопки "Cancel".

Окно регитстрации закрывается, а вместо него открывается начальное окно.

void Window::**on\_buttonBox\_rejected**()

{

ui->winreg->*setVisible*(false);

ui->winfirst->*setVisible*(true);

}

Данная функция реализует действия, которые будут происходить после успешного прохождения авторизации.

Окно авторизации закрывается, а окно выбора действия (обучение или просмотр отчетов) открывается.

void Window::**Auth**()

{

ui->winauth->*setVisible*(false);

ui->winsecond->*setVisible*(true);

}

Данная функция реализует действия, которые будут происходить после неудачного прохождения авторизации.

Программа выводит на экран сообщение об ошибке.Текст сообщения: "Incorrect login or password!!!".

void Window::**FailAuth**()

{

QMessageBox::critical(this, "ERROR", "Incorrect login or password!!!");

}

Данная функция реализует действия, которые будут происходить после удачного прохождения регистрации.

Окно регистрации закрывается, а окно выбора действия (обучение или просмотр отчетов) открывается.

void Window::**Reg**()

{

ui->winauth->*setVisible*(false);

ui->winfirst->*setVisible*(true);

}

Данная функция реализует действия, которые будут происходить после неудачного прохождения регистрации.

Программа выводит на экран сообщение об ошибке.

void Window::**FailReg**()

{

QMessageBox::critical(this, "ERROR", "ERROR");

}

СЕРВЕР

НЕЙРОННАЯ СЕТЬ