

```

#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include <QTimer>
#include <QDebug>
#include "status.h"
#include "fejlenum.h"
#include "manudialog.h"
#include <time.h>

namespace Ui {
class MainWindow;
}

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    explicit MainWindow(QWidget *parent = 0);
    ~MainWindow();

    int getSecondsSinceLastGuiUpdate();

    //Viser en informationsdialog hvis programmet ikke kunne lukkes ned som
    følge af kommunikationsfejl med SM.
    void displayCloseFailure();

public slots:
    //Bliver kaldt når gui'en skal opdateres. Den modtager parameterent temp som
    er af typen status-struct. Ud fra denne struct
    //hives de værdier ud, som skal vises på GUI'en. Dette gøres ved at kalde de
    funktioner der er tilknyttet de anvendte widgets og er således en del af Qt-
    frameworket. Når værdierne er opdateret vises det som en aktivitet i
    aktivitetsloggen
    void updateGuiSlot(status temp);

    //Kaldes hvis SM forbindelsen til styringsmodulet ændres fra forbindelse
    til mistet forbindelse eller omvendt. Det udløser en aktivitet i
    aktivitetsloggen. Derudover skiftes lyset på gui'en. Parameteren state er den
    status som forbindelsen har ændret sig til.
    void showSMConnectionChange(bool state);

    //Kaldes når brugeren har ændret i indstillingen til den manuelle hældning.
    Som parametre modtages hvilken side man ønsker at skibet skal hælde til (int
    side), hvor meget det skal hælde (double level) samt de to informationer samlet
    i en enum, Level samlet.
    //Funktionen emitter signalet "manuChanged(Level temp). Det sætter også
    klasseattributterne manuelSide, manuelLevel samt autoActivated til deres rette
    værdier.
    //Til sidst kaldes funktionen displayManuText(true)
    void manuChangedHandle(Level samlet, int side, double level);

    //Kaldes hver gang signalet DataServer::serverUpdateStatus() udsendes.
    Funktionen undersøger om forbindelsen har ændret sig ved at sammenligne med
    attributen serverConnection.
    //Hvis forbindelsen har ændret sig udsendes dette som en aktivitet. Lyset
    ændres også således at det passer ved hjælp af setLight(DATA, serverConnection.
    Hvis vi modtager "true" vil last_DATA_Update opdateres til således til den
    nuværende værdi af sekunder siden epoch.
    void showServerUpdateStatus(bool state);

    //Kaldes hvert sekund. Funktionen opdaterer antallet af sekunder siden
    sidste overførelse af data til serveren eller til SM. Når tiden er længere end
    tiden mellem hver opdatering vil dette tal skifte til rødt.
    void showTimeSinceLastUpdate();

```

```

    //Viser eller skjuler teksten med indstillingen af manuel hældning afhængig
    af parameteren show.
    void displayManuText(bool show);

signals:
    //udsendes når der er blevet trykket på knappen "activateAutoClicked"
    void activateAutoClicked();

    //Udsendes når der er blevet ændret en manuel indstilling.
    void manuChanged(Level grader);

    //Udsendes når programmet er blevet lukket ned.
    void programTerminated();

private slots:
    void btn_CloseApplication_clicked();
    //Kaldes når luk-knappen på GUI'en er blevet trykket. Udsender signalet
    programTerminated()

    //Kaldes når der bliver trykket på knappen for manuel hældning. Viser
    dialogen "manuDialog".
    void on_btn_manu_clicked();

    //Aktiver automatisk styring og deaktiverer den manuelle.
    void on_btn_auto_clicked();

private:
    //Sætter lyset i forhold til parametrene. lys er en enum der bestemmer
    hvilket element lyset skal ændres for. state er om lyset skal være tændt eller
    ej
    void setLight(lys id, bool state);

    //Tester om alarmerne har ændret sig. Sentence er starten af den sætning der
    skrives i aktivitetsloggen. critical_point er det kritiske punkt for det emne
    der arbejdes på. Value er den værdi den har. Earlier-state er hvilket stadie
    alarmen var i tidligere.
    void alarmChangerCheck(QString sentence, int critical_point, int value, bool
    &earlier_state);

    //Konverterer en integer baseret på Level-enumeratoren om til en side og en
    vinkel.
    void levelEnumConverterAndSetter(int level);

    Ui::MainWindow *ui;
    QTimer *clockTimer;
    manuDialog *mD;

    bool autoActivated;

    //Manuelt indstillet af brugeren
    double manuelLevel;
    QString manuelSide;

    //ALARMER
    bool SMConnection, serverConnection, styrbordVandtankNiveauAlarm,
    bagbordVandtankNiveauAlarm, levelAlarm;

    //Tiden ved sidste opdatering
    int last_GUI_Update;
    int last_DATA_Update;
};

#endif // MAINWINDOW_H

```