Hesed

version 0.1

Generated by Doxygen 1.8.7

Mon Jul 21 2014 19:56:56

Contents

Chapter 1

Namespace Documentation

1.1 Ui Namespace Reference

Chapter 2

Class Documentation

2.1 Person::comp Struct Reference

The comp struct is used to sort the info.

Public Member Functions

- bool operator() (info a, info b)

  Needed to sort info.
- comp ()

  Default constructor.
- void select (int s)

  To selec the desired option.

Public Attributes

- int sel

  Stores the selected ordering type.

2.1.1 Detailed Description

The comp struct is used to sort the info.

Definition at line 99 of file person.h.

2.1.2 Constructor & Destructor Documentation

2.1.2.1 Person::comp::comp ( )

Default constructor.

Definition at line 126 of file person.h.

```
127         {
128             sel = 0;
129         }
```

2.1.3 Member Function Documentation

2.1.3.1 bool Person::comp::operator() (info *a,* info *b*)

Needed to sort info.

Definition at line 108 of file person.h.

```
109         {
110             if(sel == 0) return a.ID < b.ID;
111             else if (sel == 1) return a.name < b.name;
112             else if (sel == 2) return a.surname < b.surname;
113             else if (sel == 3) return a.NIF < b.NIF;
114             else if (sel == 4) return a.birthDate < b.birthDate;
115             else if (sel == 5) return a.cNIF < b.cNIF;
116             else if (sel == 6) return a.adress < b.adress;
117             else if (sel == 7) return a.phone < b.phone;
118             else if (sel == 8) return a.beginHelp < b.beginHelp;
119             else if (sel == 9) return a.endHelp < b.endHelp;
120             else if (sel == 10) return a.numberHelp < b.numberHelp;
121             return a.famMembers < b.famMembers;
122         }
```

2.1.3.2 void Person::comp::select (int *s*)

To selec the desired option.

Definition at line 133 of file person.h.

```
133 { sel = s; }
```

2.1.4 Member Data Documentation

2.1.4.1 int Person::comp::sel

Stores the selected ordering type.

Definition at line 103 of file person.h.

The documentation for this struct was generated from the following file:

- person.h

2.2 Person::info Struct Reference

Struct to store basic people information.

Public Attributes

- QString name

  Name.
- QString surname

  Surname.
- QString NIF

  Personal NIF (Número de Identificación Fiscal)
- QString cNIF

  Couple NIF (Número de Identificación Fiscal)
- QString adress

Personal home adress.

- QString phone

  Personal phone.

- QDate birthDate

  Date of birth.

- QDate beginHelp

  First day recieved help.

- QDate endHelp

  Finishing help date.

- int numberHelp

  Number of times that somebody has been helped.

- int famMembers

  Amount of members in the home family.

- unsigned int ID

  Identification number for internal use.

Friends

- QDataStream & operator>> (QDataStream &in, info &i)

  Definition of the operator '>>' to the struct.

- QDataStream & operator<< (QDataStream &out, info &i)

  Definition of the operator '<<' to the struct.

2.2.1 Detailed Description

Struct to store basic people information.

Stored info is:

- Name
- Surname
- NIF
- Birth Date
- Couple NIF
- Adress
- Phone number
- Beginning help date
- End help date
- Times helped
- Family members
- Identification Number

Definition at line 37 of file person.h.

2.2.2 Friends And Related Function Documentation

2.2.2.1 QDataStream& operator>> (QDataStream & *in,* info & *i*) [friend]

Definition of the operator '>>' to the struct.

Definition at line 78 of file person.h.

```
78                                                              {
79             in >> i.adress >> i.beginHelp >> i.birthDate >> i.cNIF
80                >> i.endHelp >> i.famMembers >> i.ID >> i.name >> i.NIF
81                >> i.numberHelp >> i.phone >> i.surname;
82             return in;
83         }
```

2.2.2.2 QDataStream& operator<< (QDataStream & *out,* info & *i*) [friend]

Definition of the operator '<<' to the struct.

Definition at line 88 of file person.h.

```
88                                                              {
89             out << i.adress << i.beginHelp << i.birthDate << i.cNIF
90                 << i.endHelp << i.famMembers << i.ID << i.name << i.NIF
91                 << i.numberHelp << i.phone << i.surname;
92             return out;
93         }
```

2.2.3 Member Data Documentation

2.2.3.1 Person::info::name

Name.

Definition at line 52 of file person.h.

2.2.3.2 Person::info::surname

Surname.

Definition at line 52 of file person.h.

2.2.3.3 Person::info::NIF

Personal NIF (Número de Identificación Fiscal)

Definition at line 52 of file person.h.

2.2.3.4 Person::info::cNIF

Couple NIF (Número de Identificación Fiscal)

Definition at line 52 of file person.h.

2.2.3.5 Person::info::adress

Personal home adress.

Definition at line 52 of file person.h.

2.2.3.6 Person::info::phone

Personal phone.

Definition at line 52 of file person.h.

2.2.3.7 Person::info::birthDate

Date of birth.

Definition at line 61 of file person.h.

2.2.3.8 Person::info::beginHelp

First day recieved help.

Definition at line 61 of file person.h.

2.2.3.9 Person::info::endHelp

Finishing help date.

Definition at line 61 of file person.h.

2.2.3.10 Person::info::numberHelp

Number of times that somebody has been helped.

Definition at line 68 of file person.h.

2.2.3.11 Person::info::famMembers

Amount of members in the home family.

Definition at line 68 of file person.h.

2.2.3.12 Person::info::ID

Identification number for internal use.

Definition at line 73 of file person.h.

The documentation for this struct was generated from the following file:

- person.h

2.3 MainWindow Class Reference

Class to show basic information to user.

Public Member Functions

- MainWindow (QWidget ∗parent=0)

  Default constructor.
- ∼MainWindow ()

- void loadData ()

  Function that loads data when exists de Data folder.
- void addPath (const QDir &data)

  Function to add the Data folder path.

Private Attributes

- QDir data
- Ui::MainWindow * ui
- Person info

2.3.1 Detailed Description

Class to show basic information to user.

Definition at line 15 of file mainwindow.h.

2.3.2 Constructor & Destructor Documentation

2.3.2.1 MainWindow::MainWindow (QWidget * *parent* = 0) [explicit]

Default constructor.

Parameters

parent	Default parameter

Definition at line 9 of file mainwindow.cpp.

```
9                                       :QMainWindow(parent),
10     ui(new Ui::MainWindow)
11 {
12     ui->setupUi(this);
13 }
```

2.3.2.2 MainWindow::~MainWindow ( )

Definition at line 15 of file mainwindow.cpp.

```
16 {
17     delete ui;
18 }
```

2.3.3 Member Function Documentation

2.3.3.1 void MainWindow::loadData ( )

Function that loads data when exists de *Data* folder.

Definition at line 24 of file mainwindow.cpp.

```
25 {
26     info.read(data);
27     info.addTable(ui->tableView);
28     info.show();
29 }
```

2.3.3.2 void MainWindow::addPath (const QDir & *data*)

Function to add the *Data* folder path.

Parameters

data	Contains de PATH to de *Data* folder.

Definition at line 31 of file mainwindow.cpp.

```
32 {
33     this->data = data;
34 }
```

2.3.4 Member Data Documentation

2.3.4.1 QDir MainWindow::data [private]

Definition at line 49 of file mainwindow.h.

2.3.4.2 Ui::MainWindow* MainWindow::ui [private]

Definition at line 50 of file mainwindow.h.

2.3.4.3 Person MainWindow::info [private]

Definition at line 51 of file mainwindow.h.

The documentation for this class was generated from the following files:

- mainwindow.h
- mainwindow.cpp

2.4 Person Class Reference

The Person Class contains all people basic data.

Classes

- struct comp

  The comp struct is used to sort the info.
- struct info

  Struct to store basic people information.

Public Member Functions

- Person ()

  Default constructor.
- Person (const Person &p)

  Copy constructor.
- void read (QDir data)

  Function to read all stored data.
- void write (QDir data)

  Function to store all data to a file.
- void sort (int type)

  Function to sort the data with the specified metod.

- void addTable (QTableView *t)

  Adds the pointer to a QTableView.
- void show ()

  Function to show all stored data.

Private Attributes

- list< info > all

  List to store all data.
- unsigned int maxID

  Contains de maximum ID that has ever been assigned.
- QTableView * table

2.4.1 Detailed Description

The Person Class contains all people basic data.

Definition at line 16 of file person.h.

2.4.2 Constructor & Destructor Documentation

2.4.2.1 Person::Person ( )

Default constructor.

Definition at line 7 of file person.cpp.

```
8 {
9     maxID = 0;
10 }
```

2.4.2.2 Person::Person (const Person & *p*)

Copy constructor.

Definition at line 12 of file person.cpp.

```
13 {
14     maxID = p.maxID;
15     all = p.all;
16 }
```

2.4.3 Member Function Documentation

2.4.3.1 void Person::read (QDir *data*)

Function to read all stored data.

Parameters

data	Path to *Data* folder

Precondition

True

Postcondition

All data stored in *datafile.db* has been read and stored to implicit parameter.

Definition at line 22 of file person.cpp.

```
23 {
24     QFile file(data.filePath("datafile.db"));
25     if(not file.open(QIODevice::ReadWrite)) {
26         qWarning() << "Cannot create the file" << file.fileName();
27     }
28     QDataStream in(&file);
29     in.setByteOrder(QDataStream::LittleEndian);
30     in >> maxID;
31     while(not in.atEnd()) {
32         info aux;
33         in >> aux;
34         all.push_back(aux);
35     }
36     file.close();
37 }
```

2.4.3.2 void Person::write (QDir *data*)

Function to store all data to a file.

Parameters

data	Path to *Data* folder

Precondition

True

Postcondition

All existing data has been stored in *datafile.db* overwriting existing data

Definition at line 39 of file person.cpp.

```
40 {
41     QFile file(data.filePath("datafile.db"));
42     if (not file.open(QIODevice::ReadWrite)) {
43         qWarning() << "Cannot create the file" << file.fileName();
44     }
45     sort(0);
46     QDataStream out(&file);
47     out.setByteOrder(QDataStream::LittleEndian);
48     out << maxID;
49     list<info>::iterator it = all.begin();
50     while(it != all.end()) {
51         out << *it;
52         ++it;
53     }
54
55 }
```

2.4.3.3 void Person::sort (int *type*)

Function to sort the data with the specified metod.

Order

- **0** -> Identification Number (ID)
- **1** -> Name (name)

- **2** -> Surname (surname)
- **3** -> NIF (NIF)
- **4** -> Birth Date(birthDate)
- **5** -> Couple NIF (cNIF)
- **6** -> Adress (adress)
- **7** -> Phone number (phone)
- **8** -> Beginning help date (beginHelp)
- **9** -> End help date (endHelp)
- **10** -> Times helped (numberHelp)
- **11** -> Family members (famMembers)

Parameters

type	Variable to specify the desired metod to sort the data

Precondition

True

Postcondition

All data has been stored as type specifies

Definition at line 57 of file person.cpp.

```
58 {
59     if (not all.empty() and type >= 0) {
60         comp a;
61         a.select(type);
62         all.sort(a);
63     }
64     else qWarning() << "Error while trying to sort";
65 }
```

2.4.3.4 void Person::addTable (QTableView * *t*)

Adds the pointer to a QTableView.

Parameters

t	Pointer to a QTabletView

Precondition

t is not NULL

Postcondition

The pointer has been added

Definition at line 67 of file person.cpp.

```
67 { table = t; }
```

2.4.3.5 void Person::show ( )

Function to show all stored data.

Definition at line 73 of file person.cpp.

```
74 {
75     QStandardItemModel *model = new QStandardItemModel();
76     QStringList titols;
77     titols << "DNI" << "Nom" << "Cognoms" << "Data Fi ajuda" << "Nº Ajudat"
78            << "Membres";
79     model->setHorizontalHeaderLabels(titols);
80
81     table->setModel(model);
82 }
```

2.4.4 Member Data Documentation

2.4.4.1 list<info> Person::all [private]

List to store all data.

Definition at line 139 of file person.h.

2.4.4.2 unsigned int Person::maxID [private]

Contains de maximum ID that has ever been assigned.

Definition at line 143 of file person.h.

2.4.4.3 QTableView* Person::table [private]

Definition at line 144 of file person.h.

The documentation for this class was generated from the following files:

- person.h
- person.cpp

Chapter 3

File Documentation

3.1 main.cpp File Reference

Include dependency graph for main.cpp:

Functions

- int main (int argc, char ∗argv[])

3.1.1 Function Documentation

3.1.1.1 int main (int *argc*, char ∗ *argv[]*)

Definition at line 8 of file main.cpp.

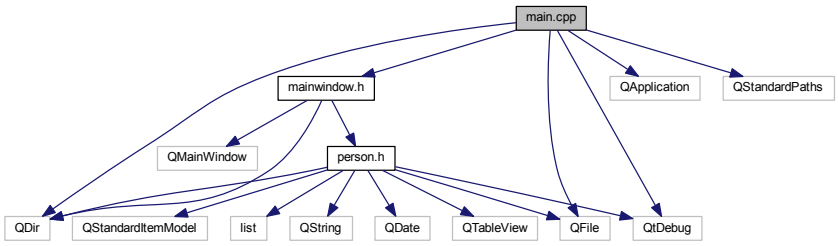
```
9 {
10     QApplication a(argc, argv);
11     MainWindow w;
12
13     QString path(QStandardPaths::writableLocation(QStandardPaths::DataLocation));
14     qDebug() << path;
15     QDir data(path);
16     w.addPath(data);
17     if (data.exists()) {
18         qDebug() << "Trobat";
19         w.loadData();
20     }
21     else {
22         qDebug() << "No s'ha trobat, creant directori";
23         data.mkpath(path);
24     }
25
26     w.showMaximized();
27
28     return a.exec();
29 }
```

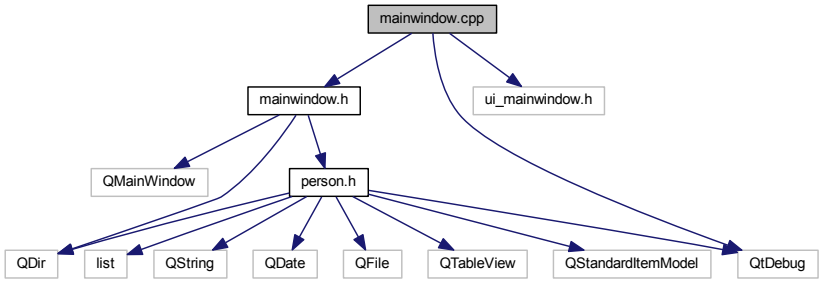
3.2 mainwindow.cpp File Reference

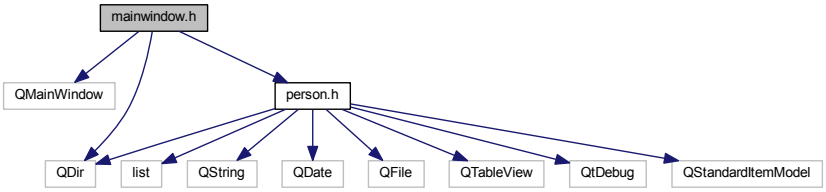
Include dependency graph for mainwindow.cpp:

3.3 mainwindow.h File Reference

Include dependency graph for mainwindow.h:


main.cpp
mainwindow.h
QApplication
QStandardPaths
QMainWindow
person.h
QDir
QStandardItemModel
list
QString
QDate
QTableView
QFile
QtDebug


mainwindow.cpp
mainwindow.h
ui_mainwindow.h
QMainWindow
person.h
QDir
list
QString
QDate
QFile
QTableView
QStandardItemModel
QtDebug


mainwindow.h
QMainWindow
person.h
QDir
list
QString
QDate
QFile
QTableView
QtDebug
QStandardItemModel

Classes

- class MainWindow

  Class to show basic information to user.

Namespaces

- Ui

3.4 person.cpp File Reference

Include dependency graph for person.cpp:

3.5 person.h File Reference

Include dependency graph for person.h:

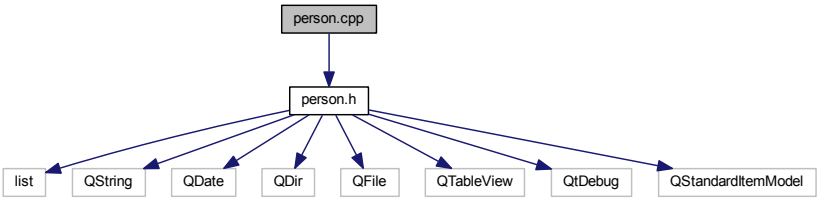
Classes

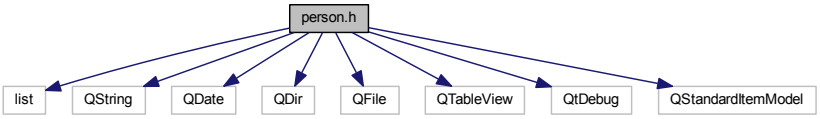
- class Person

  The Person Class contains all people basic data.
- struct Person::info

  Struct to store basic people information.
- struct Person::comp

  The comp struct is used to sort the info.


person.cpp
person.h
list
QString
QDate
QDir
QFile
QTableView
QtDebug
QStandardItemModel


person.h
list
QString
QDate
QDir
QFile
QTableView
QtDebug
QStandardItemModel

Index