

```
#include "addressbook.h"
#include "ui_addressbook.h"

AddressBook::AddressBook(QWidget *parent)
    : QWidget(parent), ui(new Ui::AddressBook)
{
    ui->setupUi(this);

    nameEdit = new QLineEdit;
    nameEdit = ui->nameEdit;
    nameEdit->setReadOnly(true);

    addressText = new QTextEdit;
    addressText = ui->addressText;
    addressText->setReadOnly(true);

    addButton = new QPushButton;
    addButton = ui->addButton;

    submitButton = new QPushButton;
    submitButton = ui->submitButton;
    submitButton->hide();

    cancelButton = new QPushButton;
    cancelButton = ui->cancelButton;
    cancelButton->hide();

    nextButton = new QPushButton;
    nextButton = ui->nextButton;
    nextButton->setEnabled(false);

    previousButton = new QPushButton;
    previousButton = ui->previousButton;
    previousButton->setEnabled(false);

    removeButton = new QPushButton;
    removeButton = ui->removeButton;
    removeButton->setEnabled(false);

    editButton = new QPushButton;
    editButton = ui->editButton;
    editButton->setEnabled(false);

    connect(addButton, SIGNAL(clicked()), this,
            SLOT(addContact()));
    connect(submitButton, SIGNAL(clicked()), this,
            SLOT(submitContact()));
    connect(cancelButton, SIGNAL(clicked()), this,
            SLOT(cancel()));
    connect(nextButton, SIGNAL(clicked()), this,
            SLOT(next()));
    connect(previousButton, SIGNAL(clicked()), this,
            SLOT(previous()));
    connect(editButton, SIGNAL(clicked()), this,
            SLOT(editContact()));
    connect(removeButton, SIGNAL(clicked()), this,
            SLOT(removeContact()));

    setWindowTitle(tr("Simple Address Book"));
}

AddressBook::~AddressBook()
```

```
{
    delete ui;
}

void AddressBook::addContact()
{
    oldName = nameEdit->text();
    oldAddress = addressText->toPlainText();

    nameEdit->clear();
    addressText->clear();

    updateInterface(AddingMode);
}

void AddressBook::submitContact()
{
    QString name = nameEdit->text();
    QString address = addressText->toPlainText();

    if (name == "" || address == "") {
        QMessageBox::information(this, tr("Empty Field"),
            tr("Please enter a name and address.));
        return;
    }

    if (currentMode == AddingMode) {
        if (!contacts.contains(name)) {
            contacts.insert(name, address);
            QMessageBox::information(this, tr("Add Successful"),
                tr("\'%1\' has been added to your address book.")).arg(name));
        } else {
            QMessageBox::information(this, tr("Add Unsuccessful"),
                tr("Sorry, \''1\' is already in your address book.")).arg(name));
            return;
        }
    } else if (currentMode == EditingMode) {
        if (oldName != name) {
            if (!contacts.contains(name)) {
                QMessageBox::information(this, tr("Edit Successful"),
                    tr("\'%1\' has been edited in your address
book.")).arg(oldName));
                contacts.remove(oldName);
                contacts.insert(name, address);
            } else {
                QMessageBox::information(this, tr("Edit Unsuccessful"),
                    tr("Sorry, \''1\' is already in your address
book.")).arg(name));
                return;
            }
        } else if (oldAddress != address) {
            QMessageBox::information(this, tr("Edit Successful"),
                tr("\'%1\' has been edited in your address book.")).arg(name));
            contacts[name] = address;
        }
    }

    updateInterface(NavigationMode);
}
```

```
}

void AddressBook::cancel()
{
    nameEdit->setText(oldName);
    addressText->setText(oldAddress);
    updateInterface(NavigationMode);
}

void AddressBook::next()
{
    QString name = nameEdit->text();
    QMap<QString, QString>::iterator i = contacts.find(name);

    if (i != contacts.end())
        i++;
    if (i == contacts.end())
        i = contacts.begin();

    nameEdit->setText(i.key());
    addressText->setText(i.value());
}

void AddressBook::previous()
{
    QString name = nameEdit->text();
    QMap<QString, QString>::iterator i = contacts.find(name);

    if (i == contacts.end()) {
        nameEdit->clear();
        addressText->clear();
        return;
    }

    if (i == contacts.begin())
        i = contacts.end();

    i--;
    nameEdit->setText(i.key());
    addressText->setText(i.value());
}

void AddressBook::editContact()
{
    oldName = nameEdit->text();
    oldAddress = addressText->toPlainText();

    updateInterface(EditingMode);
}

void AddressBook::removeContact()
{
    QString name = nameEdit->text();
    QString address = addressText->toPlainText();

    if (contacts.contains(name)) {

        int button = QMessageBox::question(this,
            tr("Confirm Remove"),
            tr("Are you sure you want to remove \"%1\"?").arg(name),
            QMessageBox::Yes | QMessageBox::No);
    }
}
```

```
        if (button == QMessageBox::Yes) {

            previous();
            contacts.remove(name);

            QMessageBox::information(this, tr("Remove Successful"),
                                     tr("\'%1\' has been removed from your address book.").arg(name));
        }
    }

    updateInterface(NavigationMode);
}

void AddressBook::updateInterface(Mode mode)
{
    currentMode = mode;

    switch (currentMode) {

    case AddingMode:
    case EditingMode:

        nameEdit->setReadOnly(false);
        nameEdit->setFocus(Qt::OtherFocusReason);
        addressText->setReadOnly(false);

        addButton->setEnabled(false);
        editButton->setEnabled(false);
        removeButton->setEnabled(false);

        nextButton->setEnabled(false);
        previousButton->setEnabled(false);

        submitButton->show();
        cancelButton->show();
        break;

    case NavigationMode:

        if (contacts.isEmpty()) {
            nameEdit->clear();
            addressText->clear();
        }

        nameEdit->setReadOnly(true);
        addressText->setReadOnly(true);
        addButton->setEnabled(true);

        int number = contacts.size();
        editButton->setEnabled(number >= 1);
        removeButton->setEnabled(number >= 1);
        nextButton->setEnabled(number > 1);
        previousButton->setEnabled(number > 1);

        submitButton->hide();
        cancelButton->hide();
        break;
    }
}
```