ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ "САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М. А. БОНЧ-БРУЕВИЧА"

Факультет инфокоммуникационных сетей и систем Кафедра программной инженерии и вычислительной техники

# КУРСОВАЯ РАБОТА

**по дисциплине «Объектно-ориентированное программирование» ВАРИАНТ 9**

Выполнил: студент 2 курса группы ИКПИ-15

Ильницкий И.Н.

Санкт-Петербург 2022

**Оглавление**

[1. ПОСТАНОВКА ЗАДАЧИ……………………………………………………………….3](#_Toc121960665)

[2. ОБОСНОВАНИЕ ТИПА РАЗРАБАТЫВАЕМОГО КОНТЕЙНЕРА………………....4](#_Toc121960666)

[3. РАЗРАБОТКА СТРУКТУРЫ ПРОГРАММЫ………………………………………….4](#_Toc121960667)

[4. РАЗРАБОТКА ОТДЕЛЬНЫХ МОДУЛЕЙ……………………………………………..5](#_Toc121960668)

[5. РАЗРАБОТКА СОБСТВЕННЫХ КЛАССОВ………………………………………….5](#_Toc121960669)

[6. РАЗРАБОТКА ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ…………………………………....7](#_Toc121960674)

[7. ИНСТРУКЦИЯ ПОЛЬЗОВАТЕЛЮ…………………………………………………….8](#_Toc121960675)

[8. ЗАКЛЮЧЕНИЕ…………………………………………………………………………..15](#_Toc121960683)

9[. СПИСОК ЛИТЕРАТУРЫ……………………………………………………………….15](#_Toc121960684)

[10. ПРИЛОЖЕНИЕ………………………………………………………………………....15](#_Toc121960685)

# ПОСТАНОВКА ЗАДАЧИ

Создать базу данных, работающую на основе контейнера, который следует разработать в процессе выполнения курсовой работы. В качестве прототипа разрабатываемого контейнера можно использовать контейнеры vector и list из библиотеки STL. Для работы с контейнером следует предусмотреть итератор. Остальные требования к контейнеру определяются студентом самостоятельно.

Разрабатываемая база данных должна быть создана с использованием C++ и библиотеки Qt6. Среда программирования Qt Creator.

В качестве элементов контейнеров должны использоваться объекты класса, характеризующий самолет.

* База данных должна обеспечивать выполнение следующих операций:
* Создание базы данных,
* Объединение баз данных,
* Добавления и удаления записей,
* Редактирования записей,
* Просмотра базы данных,
* Поиск данных,
* Запись на диск,
* Чтение с диска.

Интерфейсная часть программы должна содержать следующие компоненты:

* Окно «О программе»,
* Главное меню,
* Всплывающее меню,
* Строку состояния,
* Панель инструментов,
* Подсказки по командам меню и панели инструментов.

# ОБОСНОВАНИЕ ТИПА РАЗРАБАТЫВАЕМОГО КОНТЕЙНЕРА

В данной курсовой работе необходимо использовать контейнер для хранения объектов. Причем контейнер необходимо разработать самостоятельно, а не подключить, например из стандартной библиотеки. В качестве прототипа разрабатываемого контейнера я решил использовать контейнер vector из библиотеки STL. Выбор данного контейнера объясняется простотой своей разработки и организации, а также возможностью произвольного доступа к своим элементам, чего нет у контейнера типа list. Контейнер list обладает только последовательным доступом.

Разрабатываемый контейнер должен обладать следующими функциями**:**

* Создание пустого контейнера; размера n; на основе массива; а также копии уже имеющегося контейнера того же типа.
* Получение элемента по индексу;
* Добавление элементов в конец занятого массива;
* Вставка элемента в произвольную позицию;
* Удаление элемента по индексу;
* В классе контейнере предусмотрена возможность итерации при помощи begin() и end().

# 3. РАЗРАБОТКА СТРУКТУРЫ ПРОГРАММЫ

Программа должна состоять из графического интерфейса и кода, который остается «за кадром». Интерфейс составляют: элемент отображение таблицы TableWidget, строка поиска по таблице LineEdit, панель инструментов, главное меню и статусная строка. А также диалоговые окна: изменение записей в таблицу NewForm, окно вывода сообщений: о создателе, помощь, ошибка.

Фоновую часть составляют классы: контейнер List, содержимое контейнера – класс, содержащих описание самолетов.

Таким образом, программу составляют модули:

* main.cpp
* mainwindow.h mainwindow.cpp
* newform.h newform.cpp
* List.h
* airplane.h

# 4. РАЗРАБОТКА ОТДЕЛЬНЫХ МОДУЛЕЙ

List – класс-контейнер выбранного типа

MainWindow – класс для работы с интерфейсом программы

NewForm – класс-диалоговое окно для добавления и редактирования элементов

Airplane – класс, хранящий в себе поля выбранного типа элементов ( рейса самолета)

# 5. РАЗРАБОТКА СОБСТВЕННЫХ КЛАССОВ

# 5.1 LIST

Шаблонный класс, предназначенный для работы с элементами пользовательского типа

Класс содержит поля:

* array – содержит данные (в нашем случае объекты класса Airplane)
* int size – количество хранящихся объектов
* int capacity – размер массива

Контейнер предусматривает следующие операции:

* get – получение элемента по индексу
* getSize – получение размера вектора
* set – установка элемента по индексу
* remove – удаление элемента
* add – добавление элемента в конец
* addAll – добавление стороннего вектора позицию текущего
* begin – получение указателя на первый элемент списка
* end – получение указателя на последний элемент списка
* toArray – преобразование вектора в статический массив

5.2 AIRPLANE

Класс, характеризующий самолет (авиарейс)

Содержит поля:

* QString number – номер самолета
* QString direction – направление
* QString depTime – время отправления
* QString seats – количество мест
* QString type - тип

В классе имеются методы доступа к представленным полям, а также установки их значений.

# 5.3 MAINWINDOW

# Класс для работы с интерфейсом пользователя

# Содержит поля:

* Ui::MainWindow \*ui – для обращения к элементам окна
* QTableWidgetItem \* protoitem – прототип элемента таблицы
* List<Airplane> list – база данных(вектор), содержащая объекты класса Airplane
* QString fileName – имя открытого в данный момент файла
* bool isModified – флаг, отражающий наличие изменений в базе

Методы:

* updateTable – отображает данные, находящиеся в контейнере
* loadFile – загрузить данные из выбранного файла
* saveFile – сохранить данные в файл
* on\_helpAbout\_triggered – нажатие на кнопку «О программе»
* on\_fileOpen\_triggered – нажатие на кнопку «Открыть файл»
* on\_fileSave\_triggered – нажатие на кнопку «Сохранить»
* on\_fileMerge\_triggered – нажатие на кнопку «Объединить файлы»
* on\_fileSaveAs\_triggered – нажатие на кнопку «Сохранить как»
* on\_fileNew\_triggered – нажатие на кнопку «Создать базу данных»
* on\_newRow\_triggered – нажатие на кнопку «Новая запись»
* on\_editRow\_triggered – нажатие на кнопку «Редактировать запись»
* on\_deleteRow\_triggered – нажатие на кнопку Удалить запись»
* on\_lineSearch\_textChanged – поиск по базе данных

5.4 NewForm

Класс, который управляет диалоговым окном добавления и редактирования элементов

Содержит поля:

* Ui::NewForm \* ui – для обращения к элементам окна

Методы:

* getData – метод, возвращающий объект класса Airpane
* setPlane – если окно вызвано нажатием на кнопку «редактировать», этот метод устанавливает стандартные значения во все LineEdit

# 6. РАЗРАБОТКА ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ

# 

Главное меню, панель инструментов, таблица, строка поиска и статусная строка

(пустая)

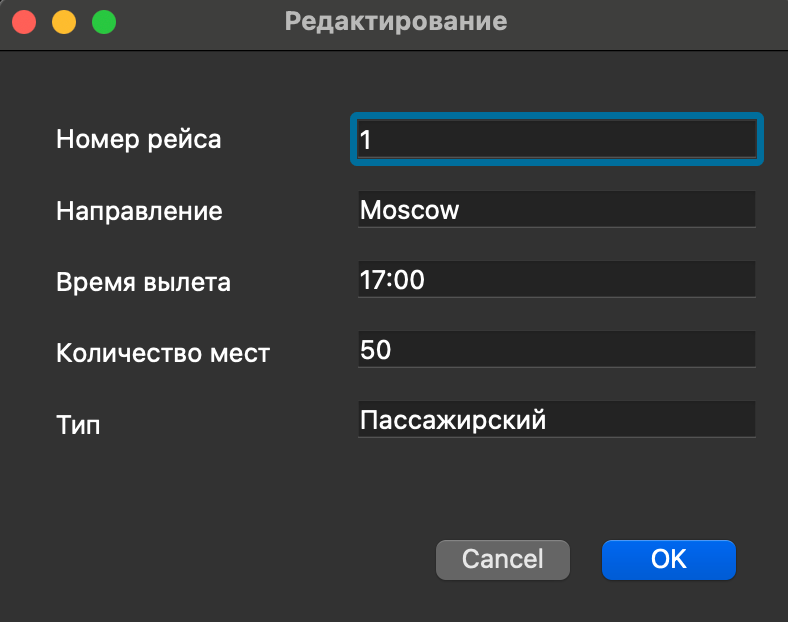
В главном меню доступны операции

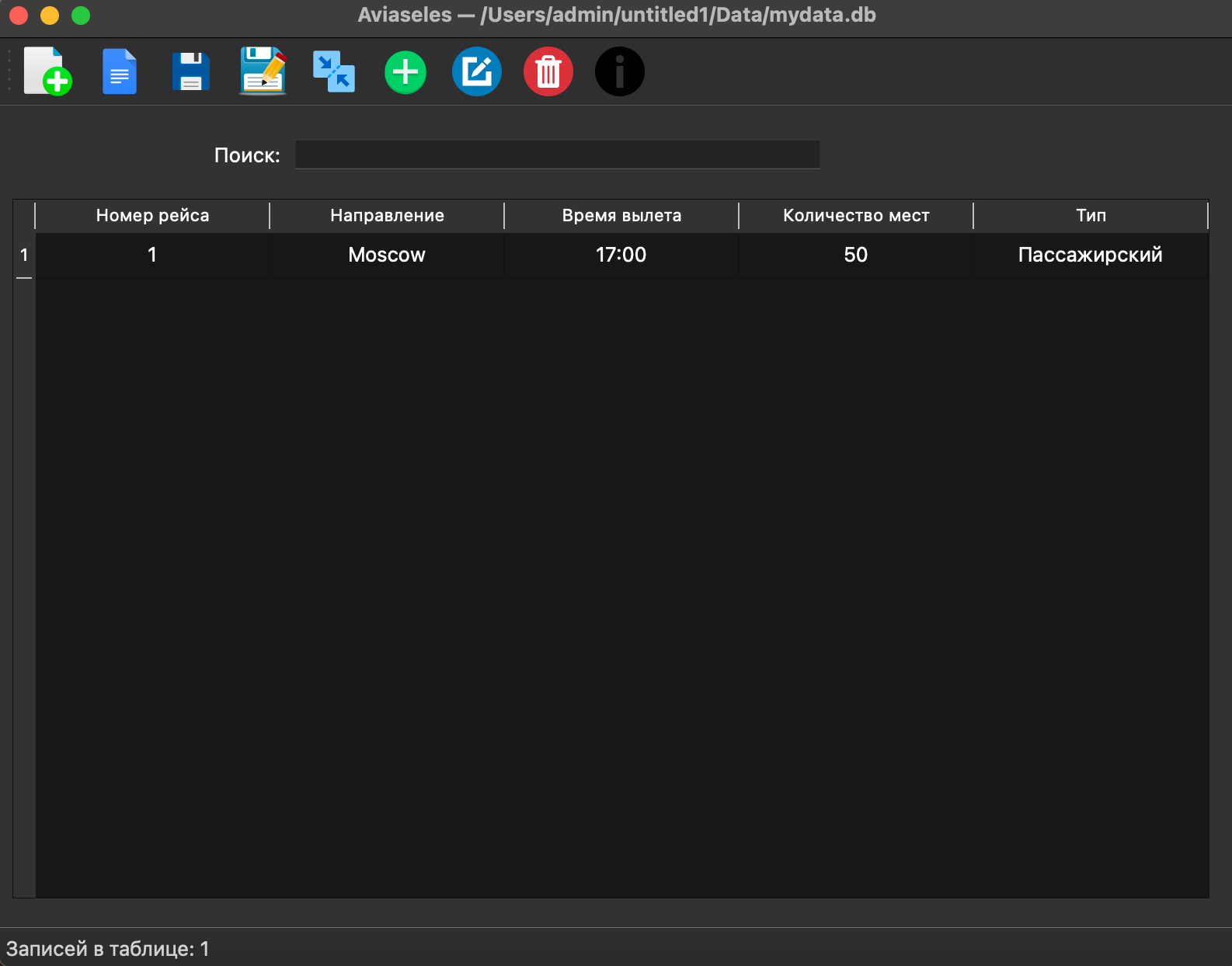
* Создания нового файла
* Открытие
* Сохранение
* Объединение
* Добавление нового элемента
* Удаление
* Редактирование
* Справка

Под панелью инструментов расположена панель поиска, по середине таблицы вывода

# 7. ИНСТРУКЦИЯ ПОЛЬЗОВАТЕЛЮ

1. Добавление записиC:\Users\vanya\AppData\Local\Microsoft\Windows\INetCache\Content.Word\newRow.png**.**





Диалог добавления записи и отображение результата в главном окне в таблице и в

статусной строке

# Редактирование записи C:\Users\vanya\AppData\Local\Microsoft\Windows\INetCache\Content.Word\edit.png.

# 

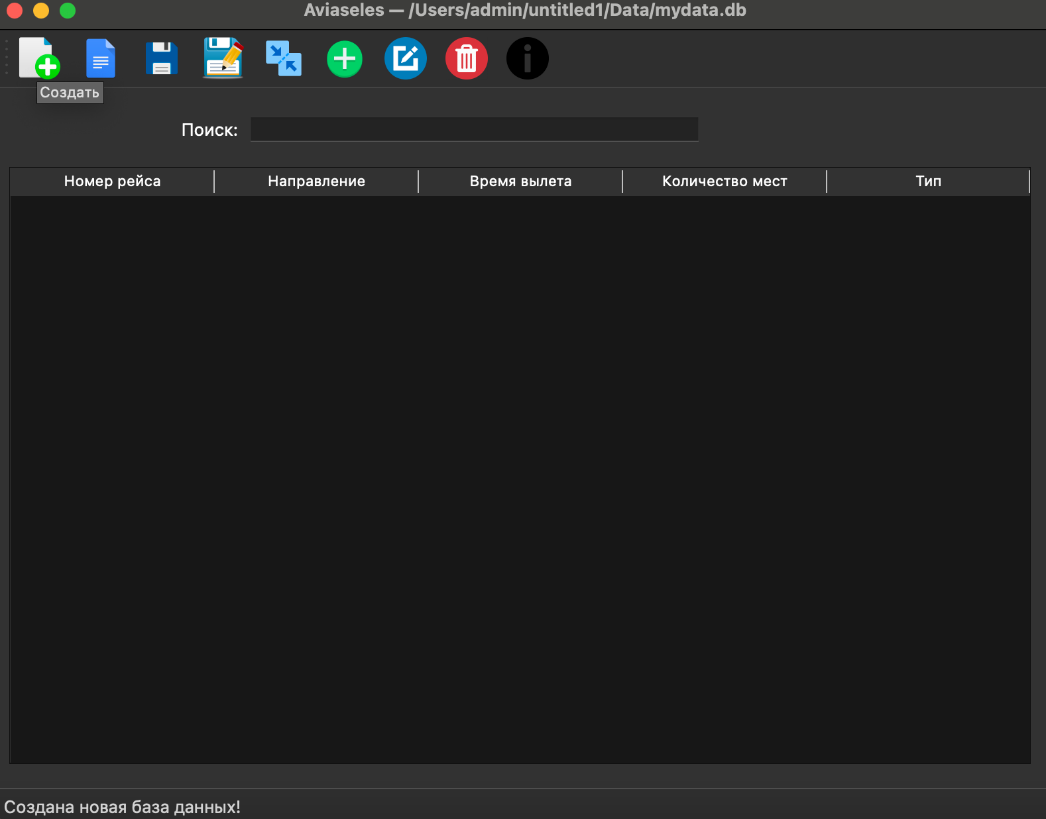
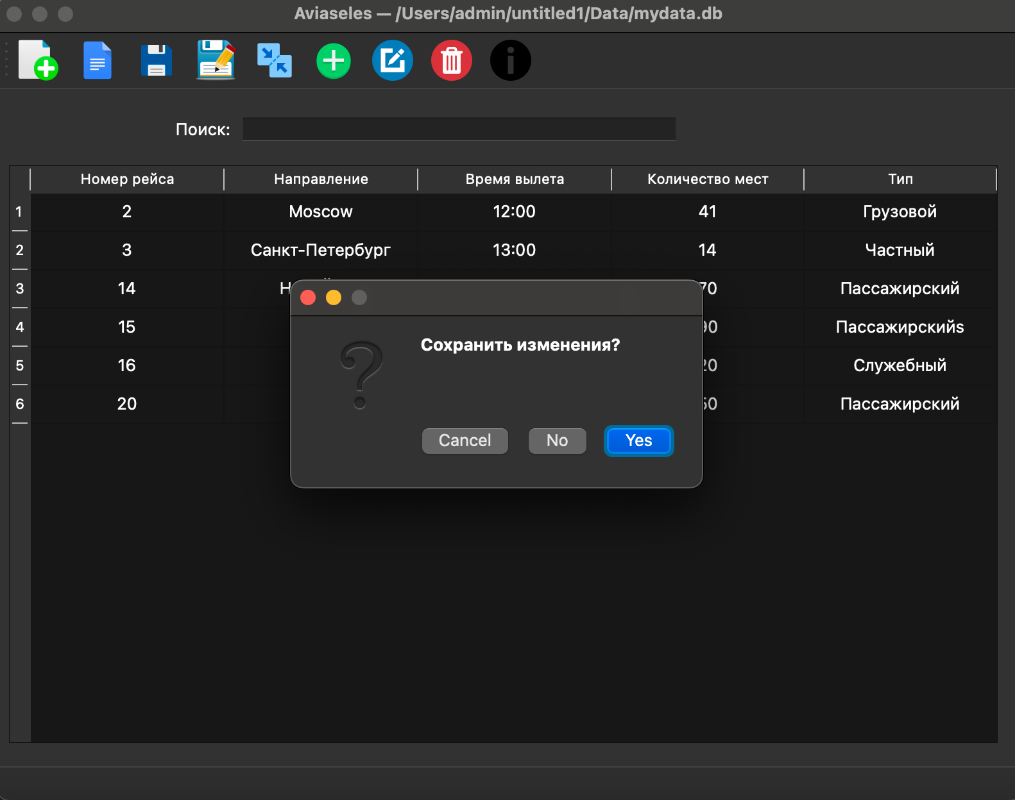
Ди Диалог редактирования записи и отображение результата в главном окне в таблице.

# Удаление записи C:\Users\vanya\AppData\Local\Microsoft\Windows\INetCache\Content.Word\delete3.jpeg.

# 

Удаление записи с подтверждением.

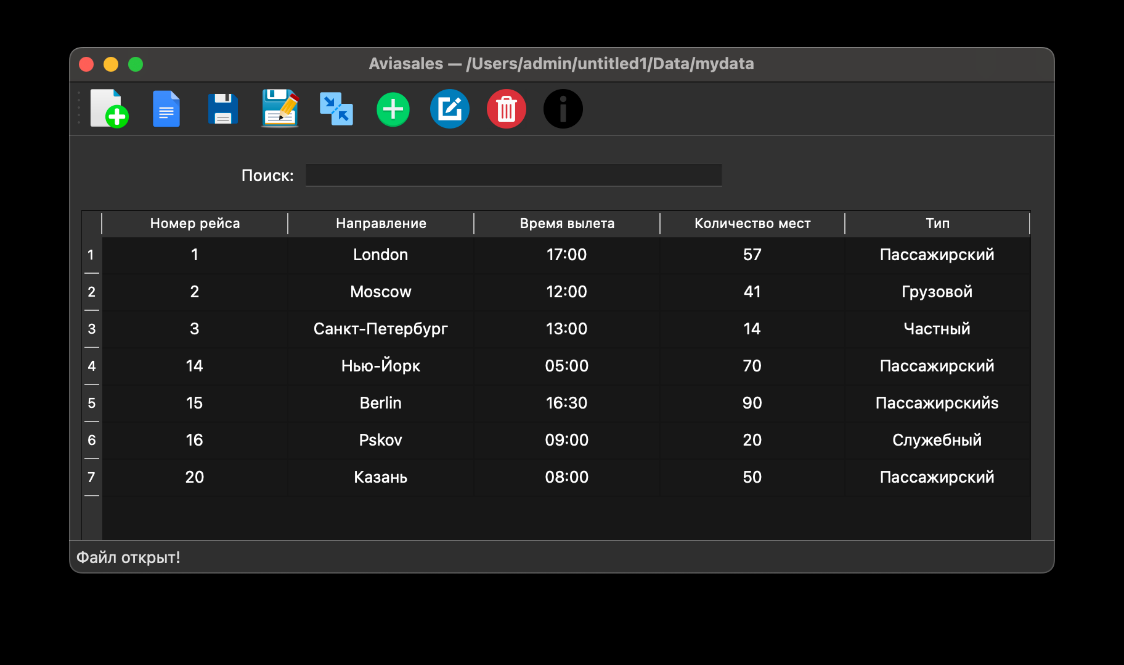
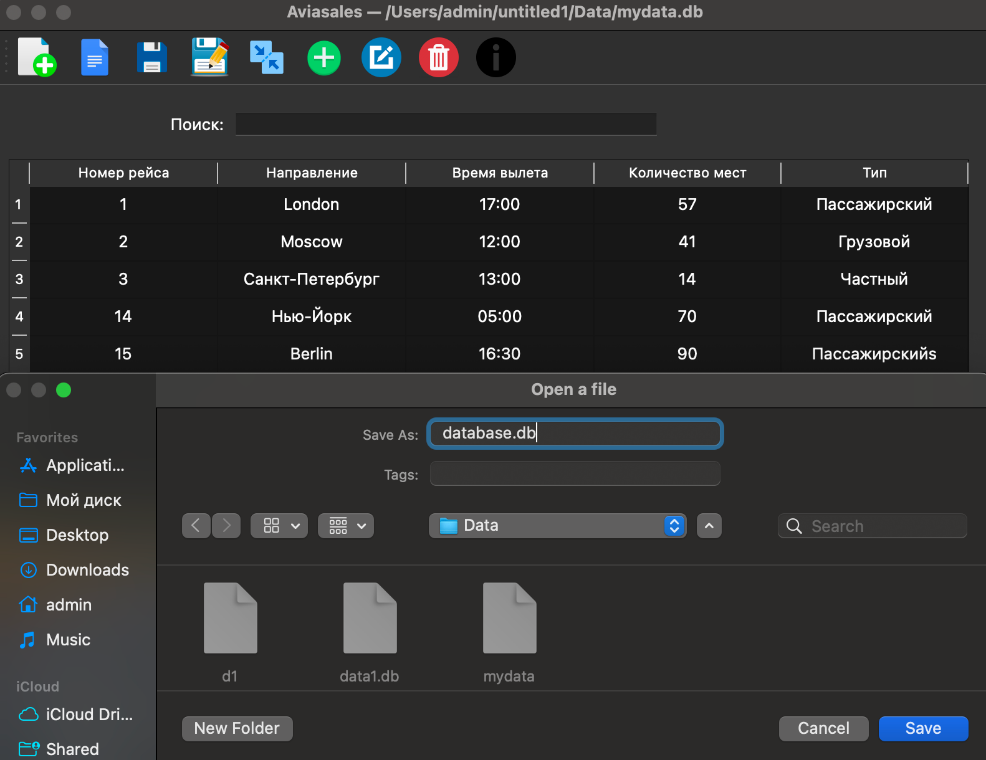
# Создание новой базы данных C:\Users\vanya\AppData\Local\Microsoft\Windows\INetCache\Content.Word\addicon.png.



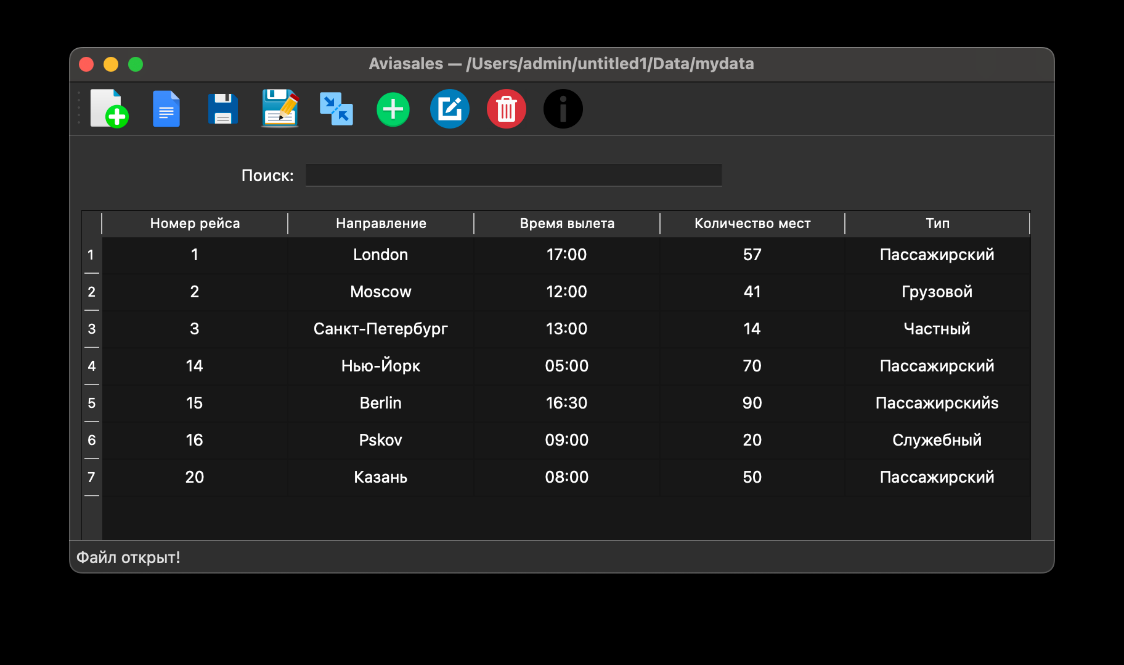
Создание нового файла с подтверждением.

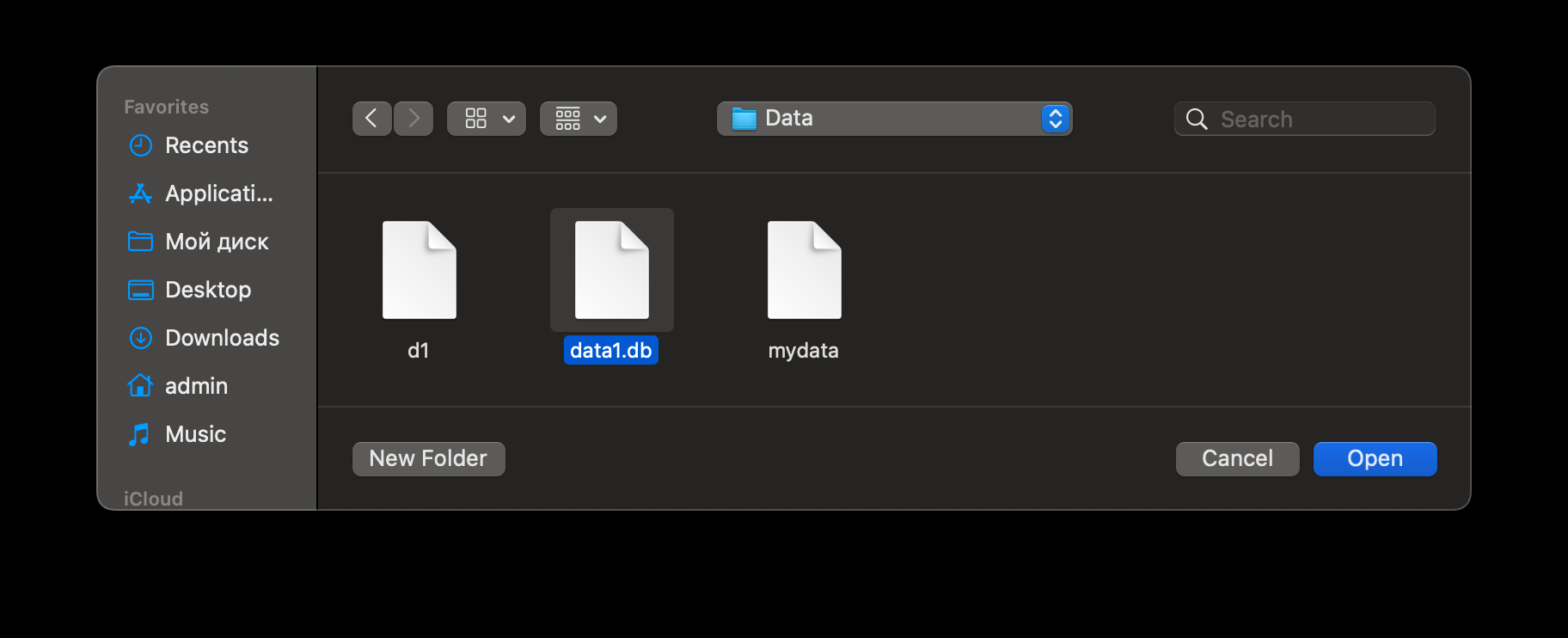
Подтверждение не отображается, если база данных пуста и никакой файл не открыт.

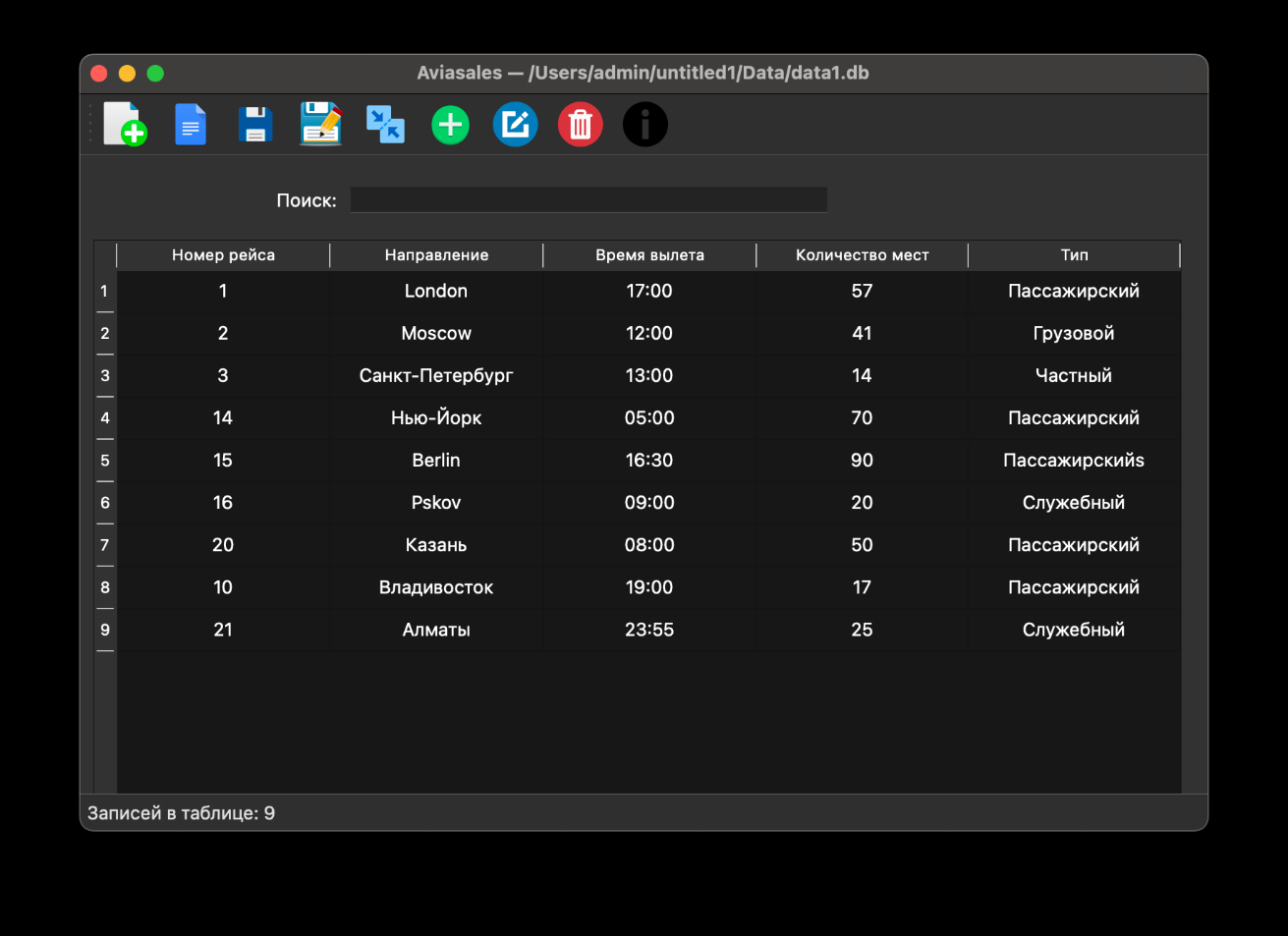
# Сохранение и открытие базы данных C:\Users\vanya\AppData\Local\Microsoft\Windows\INetCache\Content.Word\save-download-icon-10.pngC:\Users\vanya\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Save-as-icon.pngC:\Users\vanya\AppData\Local\Microsoft\Windows\INetCache\Content.Word\newFile.png



Открытие файла с данными и сохранение базы данных в выбранный файл

1.  Объединение баз данных**.** 



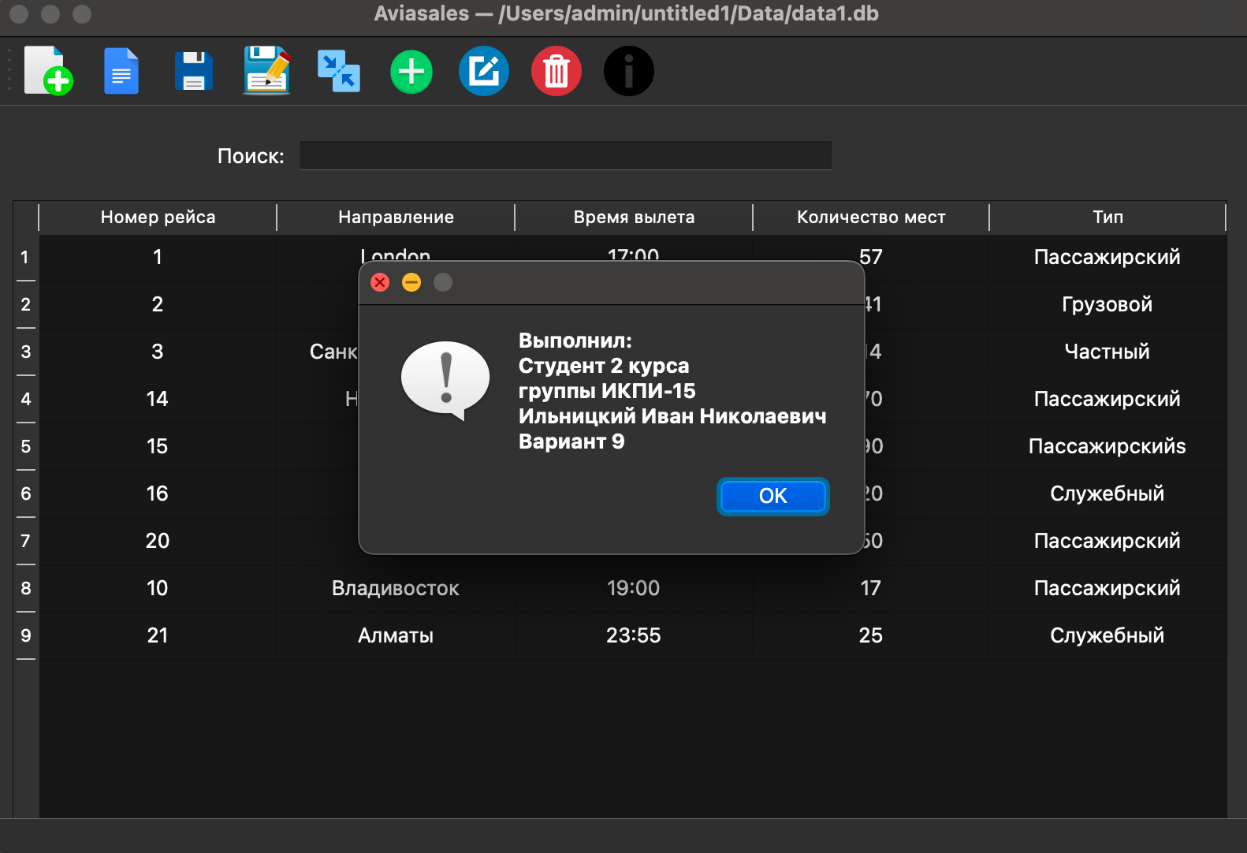


Объединение текущей базы данный с базой данных из выбранного файла

# Поиск.

Поиск по базе данных (соответствующие ячейки будут подсвечены)

1. О программе .



# ЗАКЛЮЧЕНИЕ

Программа выполняет все поставленные задачи: работа с базами данных (открытие, сохранение, добавление записи, удаление записей, редактирование записи и т. д.), работа с вектором самолетов (авиарейсов), интуитивно- понятное отображение всех возможностей программы.

В ходе выполнения курсовой работы была освоена работа с основными средствами языка C++, библиотеки QT и редактором QT Creator.

# СПИСОК ЛИТЕРАТУРЫ

1. С. В. Козин. Лекции по дисциплине "Объектно-ориентированное программирование".
2. Официальная документация QT Creator.
3. М. Шлее — QT 5.3. Профессиональное программирование на C++. 2015.
4. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений (3-е издание, 2008)

# 10. ПРИЛОЖЕНИЕ

**10.1 Файл CMakeList.txt**

cmake\_minimum\_required(VERSION 3.5)

project(untitled1 VERSION 0.1 LANGUAGES CXX)

set(CMAKE\_INCLUDE\_CURRENT\_DIR ON)

set(CMAKE\_AUTOUIC ON)

set(CMAKE\_AUTOMOC ON)

set(CMAKE\_AUTORCC ON)

set(CMAKE\_CXX\_STANDARD 17)

set(CMAKE\_CXX\_STANDARD\_REQUIRED ON)

find\_package(QT NAMES Qt6 Qt5 REQUIRED COMPONENTS Widgets)

find\_package(Qt${QT\_VERSION\_MAJOR} REQUIRED COMPONENTS Widgets)

set(PROJECT\_SOURCES

main.cpp

mainwindow.cpp

mainwindow.h

mainwindow.ui

airplane.cpp

airplane.h

List.h

resource.qrc

newform.h newform.cpp newform.ui

)

if(${QT\_VERSION\_MAJOR} GREATER\_EQUAL 6)

qt\_add\_executable(untitled1

MANUAL\_FINALIZATION

${PROJECT\_SOURCES}

)

# Define target properties for Android with Qt 6 as:

# set\_property(TARGET untitled1 APPEND PROPERTY QT\_ANDROID\_PACKAGE\_SOURCE\_DIR

# ${CMAKE\_CURRENT\_SOURCE\_DIR}/android)

# For more information, see https://doc.qt.io/qt-6/qt-add-executable.html#target-creation

else()

if(ANDROID)

add\_library(untitled1 SHARED

${PROJECT\_SOURCES}

)

# Define properties for Android with Qt 5 after find\_package() calls as:

# set(ANDROID\_PACKAGE\_SOURCE\_DIR "${CMAKE\_CURRENT\_SOURCE\_DIR}/android")

else()

add\_executable(untitled1

${PROJECT\_SOURCES}

)

endif()

endif()

target\_link\_libraries(untitled1 PRIVATE Qt${QT\_VERSION\_MAJOR}::Widgets)

set\_target\_properties(untitled1 PROPERTIES

MACOSX\_BUNDLE\_GUI\_IDENTIFIER my.example.com

MACOSX\_BUNDLE\_BUNDLE\_VERSION ${PROJECT\_VERSION}

MACOSX\_BUNDLE\_SHORT\_VERSION\_STRING ${PROJECT\_VERSION\_MAJOR}.${PROJECT\_VERSION\_MINOR}

MACOSX\_BUNDLE TRUE

WIN32\_EXECUTABLE TRUE

)

install(TARGETS untitled1

BUNDLE DESTINATION .

LIBRARY DESTINATION ${CMAKE\_INSTALL\_LIBDIR})

if(QT\_VERSION\_MAJOR EQUAL 6)

qt\_finalize\_executable(untitled1)

endif()

**10.2 Файлы .cpp**

* **Main.cpp**

#include "mainwindow.h"

#include <QApplication>

#include <QResource>

int main(int argc, char \*argv[])

{

QApplication a(argc, argv);

a.setWindowIcon(QIcon(":/img/appicon1.icns"));

MainWindow w;

w . setWindowIcon(QIcon("img/appicon.icns"));

w.show();

QResource::registerResource("/resource.qrc");

return a.exec();

}

* **MainWindow.cpp**

#include "mainwindow.h"

#include "./ui\_mainwindow.h"

#include <QMessageBox>

#include <QTableWidgetItem>

#include <QFileDialog>

#include <QCloseEvent>

#include "QTextStream"

#include <fstream>

#include <string>

#include <QFile>

#include <iostream>

#include "List.h"

#include "airplane.h"

#include "newform.h"

MainWindow::MainWindow(QWidget \*parent)

: QMainWindow(parent)

, ui(new Ui::MainWindow)

{

protoitem = new QTableWidgetItem();

protoitem->setTextAlignment(Qt::AlignCenter);

ui->setupUi(this);

setWindowTitle("Aviasales");

isModified=false;

QHeaderView\* header = ui->tableWidget->horizontalHeader();

header->setSectionResizeMode(QHeaderView::Stretch);

ui->tableWidget->setSelectionBehavior(QAbstractItemView::SelectRows);

ui->tableWidget->setEditTriggers(QAbstractItemView::NoEditTriggers);

}

MainWindow::~MainWindow()

{

delete ui;

}

void MainWindow::updateTable() {

ui->tableWidget->setRowCount(list.getSize());

int row = 0;

auto iter = list.begin();

while (iter!=list.end()) {

QTableWidgetItem \*item1 = protoitem->clone();

QTableWidgetItem \*item2 = protoitem->clone();

QTableWidgetItem \*item3 = protoitem->clone();

QTableWidgetItem \*item4 = protoitem->clone();

QTableWidgetItem \*item5 = protoitem->clone();

item1->setText(iter->getNumber());

item2->setText(iter->getDirection());

item3->setText(iter->getDepTime());

item4->setText(iter->getSeats());

item5->setText(iter->getType());

ui->tableWidget->setItem(row,0,item1);

ui->tableWidget->setItem(row,1,item2);

ui->tableWidget->setItem(row,2,item3);

ui->tableWidget->setItem(row,3,item4);

ui->tableWidget->setItem(row,4,item5);

row++;

iter++;

}

ui->tableWidget->setStatusTip("Записей в таблице: "+QString::number(ui->tableWidget->rowCount()));

}

void MainWindow::loadFile(QString fileName) {

QFile file(fileName);

if(!file.open(QIODevice::ReadOnly)) {

QMessageBox::information(0, "error", file.errorString());

}

QTextStream in(&file);

setWindowTitle("Aviasales — "+fileName);

while(!in.atEnd()) {

QString line = in.readLine();

QStringList cols = line.split(",");

if (cols.size()!=5) continue;

Airplane plane = Airplane(cols[0],cols[1],cols[2],cols[3],cols[4]);

list.add(plane);

}

file.close();

updateTable();

ui->statusbar->showMessage("Данные загружены!");

}

void MainWindow::saveFile(QString fileName1) {

QFile file(fileName1);

if (file.open(QIODevice::ReadWrite)) {

QTextStream stream(&file);

auto iter = list.begin();

while (iter!=list.end()) {

QString str = iter->getNumber()+","+iter->getDirection()+","+iter->getDepTime()+","+iter->getSeats()+","+iter->getType()+"\n";

stream << str;

iter++;

}

}

file.close();

isModified = false;

}

void MainWindow::on\_helpAbout\_triggered()

{

QMessageBox::information(this,"Info","Выполнил:\nСтудент 2 курса\nгруппы ИКПИ-15\nИльницкий Иван Николаевич\nВариант 9");

}

void MainWindow::on\_fileOpen\_triggered()

{

if (isModified) {

QMessageBox::StandardButton resBtn = QMessageBox::question( this, "k",

tr("Сохранить изменения?\n"),

QMessageBox::Cancel | QMessageBox::No | QMessageBox::Yes,

QMessageBox::Yes);

if (resBtn == QMessageBox::Yes) {

on\_fileSave\_triggered();

}

else if (resBtn==QMessageBox::Cancel) return;

}

list.clear();

QString fileName1 = QFileDialog::getOpenFileName(this,"Open a file",QDir::homePath());

if (fileName1.size()!=0) {

loadFile(fileName1);

fileName = fileName1;

ui->statusbar->showMessage("Файл открыт!");

}

}

void MainWindow::on\_fileSave\_triggered()

{

if (fileName.size()!=0) {

saveFile(fileName);

}

else {

on\_fileSaveAs\_triggered();

}

ui->statusbar->showMessage("Данные сохранены!");

}

void MainWindow::on\_fileMerge\_triggered()

{

loadFile(QFileDialog::getOpenFileName(this,"Open a file",QDir::homePath()));

isModified = true;

ui->statusbar->showMessage("Базы объединены!");

}

void MainWindow::on\_newRow\_triggered()

{

NewForm newForm = NewForm(this);

newForm.setModal(true);

if (newForm.exec() == QDialog::Accepted) {

Airplane plane = newForm.getData();

list.add(plane);

updateTable();

isModified = true;

ui->statusbar->showMessage("Создана новая запись!");

}

}

void MainWindow::on\_editRow\_triggered()

{

NewForm newForm = NewForm(this);

newForm.setModal(true);

int currRow = ui->tableWidget->currentRow();

if (currRow==-1) {

QMessageBox messageBox;

messageBox.critical(0,"Error","Выберите строку для редактирования!");

messageBox.setFixedSize(500,200);

return;

}

Airplane plane = list.get(currRow);

newForm.setPlane(plane);

if (newForm.exec() == QDialog::Accepted) {

Airplane plane1 = newForm.getData();

list.set(currRow,plane1);

updateTable();

isModified = true;

}

}

void MainWindow::on\_deleteRow\_triggered()

{

int currRow = ui->tableWidget->currentRow();

QMessageBox::StandardButton reply;

if (currRow!=-1) {

reply = QMessageBox::question(this, "Test", "Вы хотите удалить запись о рейсе №"+list.get(currRow).getNumber()+"?",QMessageBox::Yes|QMessageBox::No);

if (reply == QMessageBox::Yes) {

list.remove(currRow);

updateTable();

isModified = true;

}

}

}

void MainWindow::on\_fileSaveAs\_triggered()

{

QString fileName = QFileDialog::getSaveFileName(this,"Open a file",QDir::homePath());

if (fileName.size()!=0) {

saveFile(fileName);

}

}

void MainWindow::on\_lineSearch\_textChanged(const QString &arg1)

{

ui->tableWidget->setCurrentCell(-1, -1);

if (arg1 == "") {

return;

}

ui->tableWidget->setSelectionMode(QAbstractItemView::SelectionMode::MultiSelection);

auto find\_items = ui->tableWidget->findItems(arg1, Qt::MatchContains);

int len = find\_items.size();

for (int i = 0; i < len; ++i) {

auto item = find\_items.at(i);

item->setSelected(true);

}

ui->tableWidget->setSelectionMode(QAbstractItemView::SelectionMode::SingleSelection);

ui->statusbar->showMessage("Найдено " + QString::number(len) + " ячеек");

}

void MainWindow::on\_fileNew\_triggered()

{

if (fileName.size()>0&&isModified) {

QMessageBox::StandardButton resBtn = QMessageBox::question( this, "k",

tr("Сохранить изменения?\n"),

QMessageBox::Cancel | QMessageBox::No | QMessageBox::Yes,

QMessageBox::Yes);

if (resBtn==QMessageBox::Yes) {

saveFile(fileName);

}

else if (resBtn == QMessageBox::Cancel) {

return;

}

}

list.clear();

fileName = "";

updateTable();

ui->statusbar->showMessage("Создана новая база данных!");

}

void MainWindow::closeEvent (QCloseEvent \*event)

{

if (isModified==false) return;

QMessageBox::StandardButton resBtn = QMessageBox::question( this, "k",

tr("Сохранить изменения?\n"),

QMessageBox::Cancel | QMessageBox::No | QMessageBox::Yes,

QMessageBox::Yes);

if (resBtn == QMessageBox::Yes) {

if (fileName.size()==0) {

on\_fileSaveAs\_triggered();

}

else saveFile(fileName);

event->accept();

} else if (resBtn == QMessageBox::No){

event->accept();

}

else {

event->ignore();

}

}

* **Airplane.cpp**

#include "airplane.h"

Airplane::Airplane() {

number = QString();

direction = QString();

depTime = QString();

type = QString();

seats = QString();

}

Airplane::Airplane(Airplane & src) {

number = src.number;

direction = src.direction;

depTime = src.depTime;

seats = src.seats;

type = src.type;

}

Airplane::Airplane(QString number1,QString direction1,QString depTime1,QString seats1,QString type1) {

number = number1;

direction = direction1;

depTime = depTime1;

seats = seats1;

type = type1;

}

QString Airplane::getNumber() {

return number;

}

QString Airplane::getDirection() {

return direction;

}

QString Airplane::getDepTime() {

return depTime;

}

QString Airplane::getType() {

return type;

}

QString Airplane::getSeats() {

return seats;

}

void Airplane::setNumber(QString number1) {

number = number1;

}

void Airplane::setDirection(QString direction1) {

direction = direction1;

}

void Airplane::setDepTime(QString deptime1) {

depTime = deptime1;

}

void Airplane::setSeats(QString seats1) {

seats = seats1;

}

void Airplane::setType(QString type1) {

type = type1;

}

Airplane::~Airplane() {

}

* Newform.cpp

#include "newform.h"

#include "ui\_newform.h"

#include <QString>

NewForm::NewForm(QWidget \*parent) :

QDialog(parent),

ui(new Ui::NewForm)

{

ui->setupUi(this);

setWindowTitle("Редактирование");

}

NewForm::~NewForm()

{

delete ui;

}

Airplane NewForm::getData() {

Airplane result;

result.setNumber(ui->number->text());

result.setDirection(ui->direction->text());

result.setDepTime(ui->depTime->text());

result.setSeats(ui->seats->text());

result.setType(ui->type->text());

return result;

}

void NewForm::setPlane(Airplane &plane) {

ui->number->setText(plane.getNumber());

ui->direction->setText(plane.getDirection());

ui->depTime->setText(plane.getDepTime());

ui->seats->setText(plane.getSeats());

ui->type->setText(plane.getType());

}

**10.3 Заголовочные файлы .h**

* **Mainwindow.h**

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

#include <QTableWidgetItem>

#include <QString>

#include "List.h"

#include "airplane.h"

QT\_BEGIN\_NAMESPACE

namespace Ui { class MainWindow; }

QT\_END\_NAMESPACE

class MainWindow : public QMainWindow

{

Q\_OBJECT;

QTableWidgetItem \*protoitem;

List<Airplane> list;

QString fileName;

bool isModified;

public:

MainWindow(QWidget \*parent = nullptr);

~MainWindow();

void updateTable();

void updateTable(QString arg);

void loadFile(QString fileName);

void saveFile(QString fileName);

private slots:

void on\_helpAbout\_triggered();

void on\_fileOpen\_triggered();

void on\_fileSave\_triggered();

void on\_fileMerge\_triggered();

void on\_newRow\_triggered();

void on\_editRow\_triggered();

void on\_deleteRow\_triggered();

void on\_fileSaveAs\_triggered();

void on\_lineSearch\_textChanged(const QString &arg1);

void on\_fileNew\_triggered();

void closeEvent(QCloseEvent \*event);

private:

Ui::MainWindow \*ui;

};

#endif // MAINWINDOW\_H

* Airplane.h

#ifndef AIRPLANE\_H

#define AIRPLANE\_H

#include <QString>

#include <QTime>

#include <QTableWidgetItem>

class Airplane

{

private:

QString number;

QString direction;

QString depTime;

QString seats;

QString type;

public:

Airplane(QString number1,QString direction1,QString depTime1,QString seats1,QString type1);

Airplane();

Airplane (Airplane & src);

QString getNumber();

QString getDirection();

QString getDepTime();

QString getSeats();

QString getType();

void setNumber(QString number1);

void setDirection(QString direction1);

void setDepTime(QString depTime1);

void setSeats(QString seats1);

void setType(QString type1);

};

#endif // AIRPLANE\_H

* Newform.h

#ifndef NEWFORM\_H

#define NEWFORM\_H

#include <QDialog>

#include "airplane.h"

namespace Ui {

class NewForm;

}

class NewForm : public QDialog

{

Q\_OBJECT

public:

explicit NewForm(QWidget \*parent = nullptr);

~NewForm();

Airplane getData();

void setPlane(Airplane & plane);

private slots:

private:

Ui::NewForm \*ui;

};

#endif // NEWFORM\_H

**10.4 Файлы графического интерфейса**

* **Файл mainwindow.ui**

<?xml version="1.0" encoding="UTF-8"?>

<ui version="4.0">

<class>MainWindow</class>

<widget class="QMainWindow" name="MainWindow">

<property name="geometry">

<rect>

<x>0</x>

<y>0</y>

<width>800</width>

<height>600</height>

</rect>

</property>

<property name="windowTitle">

<string>MainWindow</string>

</property>

<widget class="QWidget" name="centralwidget">

<widget class="QTableWidget" name="tableWidget">

<property name="geometry">

<rect>

<x>10</x>

<y>60</y>

<width>771</width>

<height>451</height>

</rect>

</property>

<property name="statusTip">

<string/>

</property>

<column>

<property name="text">

<string>Номер рейса</string>

</property>

</column>

<column>

<property name="text">

<string>Направление</string>

</property>

</column>

<column>

<property name="text">

<string>Время вылета</string>

</property>

</column>

<column>

<property name="text">

<string>Количество мест</string>

</property>

</column>

<column>

<property name="text">

<string>Тип</string>

</property>

</column>

</widget>

<widget class="QWidget" name="layoutWidget">

<property name="geometry">

<rect>

<x>140</x>

<y>20</y>

<width>391</width>

<height>23</height>

</rect>

</property>

<layout class="QHBoxLayout" name="horizontalLayout">

<item>

<widget class="QLabel" name="label">

<property name="text">

<string>Поиск:</string>

</property>

</widget>

</item>

<item>

<widget class="QLineEdit" name="lineSearch"/>

</item>

</layout>

</widget>

</widget>

<widget class="QMenuBar" name="menubar">

<property name="geometry">

<rect>

<x>0</x>

<y>0</y>

<width>800</width>

<height>22</height>

</rect>

</property>

<widget class="QMenu" name="menu">

<property name="title">

<string>Файл</string>

</property>

<addaction name="fileNew"/>

<addaction name="fileOpen"/>

<addaction name="fileSave"/>

<addaction name="fileSaveAs"/>

<addaction name="fileMerge"/>

</widget>

<widget class="QMenu" name="menuInfo">

<property name="title">

<string>Справка</string>

</property>

<addaction name="helpAbout"/>

</widget>

<widget class="QMenu" name="menu\_2">

<property name="title">

<string>Изменить</string>

</property>

<addaction name="editRow"/>

<addaction name="deleteRow"/>

<addaction name="newRow"/>

</widget>

<addaction name="menu"/>

<addaction name="menu\_2"/>

<addaction name="menuInfo"/>

</widget>

<widget class="QStatusBar" name="statusbar"/>

<widget class="QToolBar" name="toolBar">

<property name="windowTitle">

<string>toolBar</string>

</property>

<attribute name="toolBarArea">

<enum>TopToolBarArea</enum>

</attribute>

<attribute name="toolBarBreak">

<bool>false</bool>

</attribute>

<addaction name="fileNew"/>

<addaction name="fileOpen"/>

<addaction name="fileSave"/>

<addaction name="fileSaveAs"/>

<addaction name="fileMerge"/>

<addaction name="newRow"/>

<addaction name="editRow"/>

<addaction name="deleteRow"/>

<addaction name="helpAbout"/>

</widget>

<action name="fileOpen">

<property name="icon">

<iconset resource="resource.qrc">

<normaloff>:/new/img/newFile.png</normaloff>:/new/img/newFile.png</iconset>

</property>

<property name="text">

<string>Открыть</string>

</property>

<property name="statusTip">

<string>Открыть существующую базу данных</string>

</property>

<property name="shortcut">

<string>Ctrl+O</string>

</property>

</action>

<action name="fileSave">

<property name="icon">

<iconset resource="resource.qrc">

<normaloff>:/new/img/save-download-icon-10.png</normaloff>:/new/img/save-download-icon-10.png</iconset>

</property>

<property name="text">

<string>Сохранить</string>

</property>

<property name="statusTip">

<string>Сохранить базу данных</string>

</property>

<property name="shortcut">

<string>Ctrl+S</string>

</property>

</action>

<action name="helpAbout">

<property name="icon">

<iconset resource="resource.qrc">

<normaloff>:/new/img/about.png</normaloff>

<activeon>img/about.png</activeon>:/new/img/about.png</iconset>

</property>

<property name="text">

<string>О программе</string>

</property>

<property name="statusTip">

<string>Информация о программе</string>

</property>

<property name="shortcut">

<string>F1</string>

</property>

</action>

<action name="fileMerge">

<property name="icon">

<iconset resource="resource.qrc">

<normaloff>:/new/img/merge-files.png</normaloff>:/new/img/merge-files.png</iconset>

</property>

<property name="text">

<string>Объединить</string>

</property>

<property name="statusTip">

<string>Добавить базу данных к существующей</string>

</property>

<property name="shortcut">

<string>Ctrl+M</string>

</property>

</action>

<action name="editRow">

<property name="icon">

<iconset resource="resource.qrc">

<normaloff>:/new/img/edit.png</normaloff>:/new/img/edit.png</iconset>

</property>

<property name="text">

<string>Редактировать</string>

</property>

<property name="statusTip">

<string>Редактировать запись</string>

</property>

<property name="shortcut">

<string>Ctrl+E</string>

</property>

</action>

<action name="deleteRow">

<property name="icon">

<iconset resource="resource.qrc">

<normaloff>:/new/img/delete3.jpeg</normaloff>:/new/img/delete3.jpeg</iconset>

</property>

<property name="text">

<string>Удалить</string>

</property>

<property name="statusTip">

<string>Удалить запись</string>

</property>

<property name="shortcut">

<string>Backspace</string>

</property>

</action>

<action name="newRow">

<property name="icon">

<iconset resource="resource.qrc">

<normaloff>:/new/img/newRow.png</normaloff>:/new/img/newRow.png</iconset>

</property>

<property name="text">

<string>Новая запись</string>

</property>

<property name="statusTip">

<string>Новая запись</string>

</property>

<property name="shortcut">

<string>Ctrl+A</string>

</property>

</action>

<action name="fileNew">

<property name="icon">

<iconset resource="resource.qrc">

<normaloff>:/new/img/addicon.png</normaloff>:/new/img/addicon.png</iconset>

</property>

<property name="text">

<string>Создать</string>

</property>

<property name="statusTip">

<string>Создать новую базу данных</string>

</property>

<property name="shortcut">

<string>Ctrl+N</string>

</property>

</action>

<action name="fileSaveAs">

<property name="icon">

<iconset resource="resource.qrc">

<normaloff>:/new/img/Save-as-icon.png</normaloff>:/new/img/Save-as-icon.png</iconset>

</property>

<property name="text">

<string>Сохранить как</string>

</property>

<property name="statusTip">

<string>Сохранить базу данных как</string>

</property>

<property name="shortcut">

<string>Ctrl+Shift+S</string>

</property>

</action>

</widget>

<resources>

<include location="resource.qrc"/>

</resources>

<connections/>

</ui>

* **Файл newform.ui**

<?xml version="1.0" encoding="UTF-8"?>

<ui version="4.0">

<class>NewForm</class>

<widget class="QDialog" name="NewForm">

<property name="geometry">

<rect>

<x>0</x>

<y>0</y>

<width>400</width>

<height>300</height>

</rect>

</property>

<property name="windowTitle">

<string>Dialog</string>

</property>

<widget class="QDialogButtonBox" name="buttonBox">

<property name="geometry">

<rect>

<x>20</x>

<y>240</y>

<width>351</width>

<height>32</height>

</rect>

</property>

<property name="orientation">

<enum>Qt::Horizontal</enum>

</property>

<property name="standardButtons">

<set>QDialogButtonBox::Cancel|QDialogButtonBox::Ok</set>

</property>

</widget>

<widget class="QWidget" name="">

<property name="geometry">

<rect>

<x>180</x>

<y>30</y>

<width>201</width>

<height>171</height>

</rect>

</property>

<layout class="QVBoxLayout" name="verticalLayout\_2">

<item>

<widget class="QLineEdit" name="number"/>

</item>

<item>

<widget class="QLineEdit" name="direction"/>

</item>

<item>

<widget class="QLineEdit" name="depTime"/>

</item>

<item>

<widget class="QLineEdit" name="seats"/>

</item>

<item>

<widget class="QLineEdit" name="type"/>

</item>

</layout>

</widget>

<widget class="QWidget" name="">

<property name="geometry">

<rect>

<x>30</x>

<y>30</y>

<width>131</width>

<height>171</height>

</rect>

</property>

<layout class="QVBoxLayout" name="verticalLayout">

<item>

<widget class="QLabel" name="label">

<property name="text">

<string>Номер рейса</string>

</property>

</widget>

</item>

<item>

<widget class="QLabel" name="label\_2">

<property name="text">

<string>Направление</string>

</property>

</widget>

</item>

<item>

<widget class="QLabel" name="label\_3">

<property name="text">

<string>Время вылета</string>

</property>

</widget>

</item>

<item>

<widget class="QLabel" name="label\_4">

<property name="text">

<string>Количество мест</string>

</property>

</widget>

</item>

<item>

<widget class="QLabel" name="label\_5">

<property name="text">

<string>Тип</string>

</property>

</widget>

</item>

</layout>

</widget>

</widget>

<resources/>

<connections>

<connection>

<sender>buttonBox</sender>

<signal>accepted()</signal>

<receiver>NewForm</receiver>

<slot>accept()</slot>

<hints>

<hint type="sourcelabel">

<x>248</x>

<y>254</y>

</hint>

<hint type="destinationlabel">

<x>157</x>

<y>274</y>

</hint>

</hints>

</connection>

<connection>

<sender>buttonBox</sender>

<signal>rejected()</signal>

<receiver>NewForm</receiver>

<slot>reject()</slot>

<hints>

<hint type="sourcelabel">

<x>316</x>

<y>260</y>

</hint>

<hint type="destinationlabel">

<x>286</x>

<y>274</y>

</hint>

</hints>

</connection>

</connections>

</ui>