	<p align="center"> <b>Министерство науки и высшего образования Российской Федерации</b>  <b>Федеральное государственное бюджетное образовательное учреждение</b>  <b>высшего образования</b>  <b>«Московский государственный технический университет</b>  <b>имени Н.Э. Баумана</b>  <b>(национальный исследовательский университет)»</b>  <b>(МГТУ им. Н.Э. Баумана)</b> </p>
---	--

ФАКУЛЬТЕТ «Информатика и системы управления».

КАФЕДРА «Прикладная информатика»

## **ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ**

Студент Гуляева Валерия Алексеевна  
*фамилия, имя, отчество*

Группа ИУ6-24Б Вариант 11.

Тип практики Практика по получению первичных профессиональных умений и навыков (Программирование)

Название предприятия МГТУ им. Н.Э. Баумана

Студент	_____	_____
	<i>подпись, дата</i>	<i>фамилия, и.о.</i>

Руководитель практики	_____	_____
	<i>подпись, дата</i>	<i>фамилия, и.о.</i>

Оценка \_\_\_\_\_

2019 г.

## Оглавление

1.Индивидуальное задание.....	3
2.Цели и задачи. ....	4
3.Основная часть.....	5
3.1.Код приложения.....	5
3.2. Пользовательский интерфейс. Результаты работы программы...	19
3.3. Диаграммы.....	21
4.Заключение.....	24
5.Источники.....	24

## **1.Индивидуальное задание.**

Выполнить объектную декомпозицию, разработать формы интерфейса, диаграмму состояний интерфейса, диаграммы классов интерфейсной и предметной областей, диаграмму последовательностей одной из реализуемых операций. Разработать, протестировать и отладить программу в среде Visual Studio или QT Creator.

Сведения о мотоциклах представлены для каждого их них маркой, годом выпуска, госномером, фамилией владельца. Программа должна в интерактивном режиме формировать файл, добавлять и удалять данные, а также воспринимать каждый из перечисленных запросов и давать на него ответ.

1. Определить фамилии владельцев, имеющих мотоциклы данной марки.
2. определить марки и госномера мотоциклов, принадлежащих данному владельцу.
3. Определить фамилии владельцев, марки и госномера мотоциклов не старше данного года выпуска.
4. Построить график зависимости количества зарегистрированных мотоциклов в зависимости от года выпуска.

## 2.Цели и задачи.

**Целью практикума** является получение навыков создания небольших программных систем с оконными и консольными интерфейсами.

**Задачами практикума** являются:

- более глубокое изучение средств реализации проектов программ на одном из изучаемых универсальных языках программирования высокого уровня;
  - овладение методикой и получение практических навыков проектирования небольших программных систем при структурном и объектном подходах;
  - воспитание внимания, аккуратности, систематичности, а также формирование интереса к изучаемой профессиональной деятельности.
- Выполнение практикума должно способствовать формированию и развитию навыков и умений, обеспечивающих следующие компетенции:
- выделение объектов предметной области, обобщение их в классы, определение связей между классами (ПК-3);
  - проектирование эргономичного обеспечения информационных систем (ПК-5);
  - разработка и отладка компонентов аппаратно-программных комплексов с помощью современных автоматизированных средств проектирования (ПК-7);
  - разработка проектной и эксплуатационной документации на программную и техническую продукцию (ПК-8);
  - выполнение контроля разрабатываемых проектов и технической документации на соответствие стандартам и техническим требованиям (ПК-10);
  - разработка интерфейсов «человек - ЭВМ» (ПК-12). Практикум включает три задания, одно – на структурную декомпозицию и два – на объектную декомпозицию.

### 3.Основная часть.

#### 3.1.Код приложения.

Файл “mainwindow.h”

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H
#include <QMainWindow>

namespace Ui {
class MainWindow;
}

class MainWindow : public QMainWindow
{
    Q_OBJECT

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

private slots:
    void on_pushButtonAdd_clicked();

    void on_pushButtonSearch_clicked();

    void on_pushButtonGraf_clicked();

private:
    Ui::MainWindow *ui;
};
#endif // MAINWINDOW_H
```

Файл “mainwindow.cpp”

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include "adddel.h"
#include "search.h"
#include "graph.h"
#include <QDialog>
#include "stdio.h"
```

```

MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    this->setWindowTitle("Меню");
}

MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::on_pushButtonAdd_clicked()
{
    AddDel w;
    w.setModal(true);
    w.exec();
}

void MainWindow::on_pushButtonSearch_clicked()
{
    Search w;
    w.setModal(true);
    w.exec();
}

void MainWindow::on_pushButtonGraf_clicked()
{
    Graph w;
    w.setModal(true);
    w.exec();
}

```

Файл “adddel.h”

```

#ifndef ADDDEL_H
#define ADDDEL_H

#include <QDialog>

namespace Ui {
class AddDel;
}

```

```

class AddDel : public QDialog
{
    Q_OBJECT

public:
    explicit AddDel(QWidget *parent = nullptr);
    ~AddDel();

private slots:
    void on_pushButtonAdd_clicked();

    void showEvent(QShowEvent *event);

    void on_pushButtonMenu_clicked();

    void on_pushButtonDel_clicked();

    void on_lineEditMar_textChanged(const QString &arg1);

    void on_lineEditNum_textChanged(const QString &arg1);

    void on_lineEditFam_textChanged(const QString &arg1);

    void on_listWidget_clicked(const QModelIndex &index);

private:
    Ui::AddDel *ui;
};
#endif // ADDDEL_H

```

Файл “adddel.cpp”

```

#include "adddel.h"
#include "ui_adddel.h"
#include "string.h"
#include "stdio.h"
FILE *f,*f2;

```

```

AddDel::AddDel(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::AddDel)
{
    ui->setupUi(this);
    this->setWindowTitle("Добавить/удалить запись");
}

```

```

}

AddDel::~AddDel()
{
    delete ui;
}

void AddDel::on_pushButtonAdd_clicked()
{
    QString mar, god, num, fam, all;
    char *ALL;
    f=fopen("moto.txt", "a");
    mar = ui->lineEditMar->text()+" ";
    god = ui->spinBox->text()+" ";
    num = ui->lineEditNum->text()+" ";
    fam = ui->lineEditFam->text()+"\n";
    ui->lineEditMar->clear();
    ui->spinBox->clear();
    ui->lineEditNum->clear();
    ui->lineEditFam->clear();
    all = mar + god + num + fam;
    ui->listWidget->addItem(all);
    all.toLocal8Bit();// перевод из QString в char
    ALL = all.toUtf8().data();// перевод из QString в char
    fputs(ALL, f);
    fclose(f);
    ui->pushButtonAdd->setEnabled(false);
    ui->spinBox->setValue(1990);
}

void AddDel::showEvent(QShowEvent *event)
{
    char str[100];
    ui->listWidget->clear();
    f=fopen("moto.txt", "r");
    rewind(f);
    while (fgets (str, sizeof(str), f) != NULL)
        ui->listWidget->addItem(str);
    fclose(f);
}

void AddDel::on_pushButtonMenu_clicked()
{
    fclose(f);
}

```



```

    this->close();
}

void AddDel::on_pushButtonDel_clicked()
{
    QString del;
    char *DEL, str[100];
    del = ui->listWidget->currentItem()->text();
    delete ui->listWidget->currentItem();
    del.toLocal8Bit(); //перевод из QString в char
    DEL = del.toUtf8().data(); //перевод из QString в char
    f2=fopen("moto2.txt", "w");
    f=fopen("moto.txt", "r");
    rewind(f);
    while (fgets (str, sizeof(str), f) != NULL) // пока не конец файла считываю по
строке
    {
        if (strcmp(str, DEL) != 0) fputs(str, f2); // если находит этот год в строке
    }
    fclose(f);
    fclose(f2);
    f=fopen("moto.txt", "w");
    f2=fopen("moto2.txt", "r");
    while (fgets (str, sizeof(str), f2) != NULL)
    {
        fputs(str, f);
    }
    fclose(f);
    fclose(f2);
}

void AddDel::on_lineEditMar_textChanged(const QString &arg1)
{
    if(ui->lineEditMar->text() != "" && ui->lineEditNum->text() != "" && ui-
>lineEditFam->text() != "") ui->pushButtonAdd->setEnabled(true);
    ui->listWidget->clearSelection();
    ui->pushButtonDel->setEnabled(false);
}

void AddDel::on_lineEditNum_textChanged(const QString &arg1)
{
    if(ui->lineEditMar->text() != "" && ui->lineEditNum->text() != "" && ui-
>lineEditFam->text() != "") ui->pushButtonAdd->setEnabled(true);
    ui->listWidget->clearSelection();
    ui->pushButtonDel->setEnabled(false);
}

```

```

}

void AddDel::on_lineEditFam_textChanged(const QString &arg1)
{
    if(ui->lineEditMar->text()!="" && ui->lineEditNum->text()!="" && ui->lineEditFam->text()!="") ui->pushButtonAdd->setEnabled(true);
    ui->listWidget->clearSelection();
    ui->pushButtonDel->setEnabled(false);
}

void AddDel::on_listWidget_clicked(const QModelIndex &index)
{
    ui->pushButtonDel->setEnabled(true);
}

```

Файл “search.h”

```

#ifndef SEARCH_H
#define SEARCH_H

#include <QDialog>

namespace Ui {
class Search;
}

class Search : public QDialog
{
    Q_OBJECT

public:
    explicit Search(QWidget *parent = nullptr);
    ~Search();

private slots:
    void on_pushButtonSearch1_clicked();

    void on_pushButtonSearch2_clicked();

    void on_pushButtonSearch1_3_clicked();

    void on_pushButtonClear_clicked();

    void on_pushButtonMenu_clicked();

```

```

void on_lineEditMar_textChanged(const QString &arg1);

void on_lineEditFam_textChanged(const QString &arg1);

private:
    Ui::Search *ui;
};

#endif // SEARCH_H

Файл "search.cpp"
#include "search.h"
#include "ui_search.h"
#include <QDebug>
#include "iostream"
using namespace std;

FILE *a;

Search::Search(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::Search)
{
    ui->setupUi(this);
    this->setWindowTitle("Поиск");
}

Search::~Search()
{
    delete ui;
}

void Search::on_pushButtonSearch1_clicked()
{
    QString mar;
    char *MAR, str[100];
    int i=0;
    ui->textEdit->clear();
    mar = ui->lineEditMar->text();
    mar.toLocal8Bit();
    MAR = mar.toUtf8().data();
    a=fopen("moto.txt", "r");
    rewind(a);
    while(fgets(str, sizeof(str), a) != NULL)
    {

```

```

        if (strstr(str, MAR)!=NULL)
        {
            qDebug() << QString::fromLocal8Bit(str); //перевод из char в QString
            ui->textEdit->append(str);
            i=1;
        }
        mar = ui->lineEditMar->text();
        mar.toLocal8Bit();
        MAR = mar.toUtf8().data();
    }
    if (i==0) ui->textEdit->append("no information");
    ui->pushButtonSearch1->setEnabled(false);
    fclose(a);
}

void Search::on_pushButtonSearch2_clicked()
{
    QString fam;
    char *FAM, str[100];
    int i=0;
    ui->textEdit->clear();
    fam = ui->lineEditFam->text();
    fam.toLocal8Bit();
    FAM = fam.toUtf8().data();
    a=fopen("moto.txt", "r");
    rewind(a);
    while(fgets(str, sizeof(str), a)!=NULL)
    {
        if (strstr(str, FAM)!=NULL)
        {
            qDebug() << QString::fromLocal8Bit(str);
            ui->textEdit->append(str);
            i=1;
        }
        fam = ui->lineEditFam->text();
        fam.toLocal8Bit();
        FAM = fam.toUtf8().data();
    }
    if (i==0) ui->textEdit->append("no information");
    fclose(a);
    ui->pushButtonSearch2->setEnabled(false);
}

void Search::on_pushButtonSearch1_3_clicked()
{

```

```

QString god;
char *GOD, str[100], year[5];
int i=0;
ui->textEdit->clear();
god = ui->spinBox->text();
god.toLocal8Bit();
GOD = god.toUtf8().data();
a=fopen("moto.txt", "r");
rewind(a);
int j=atoi(GOD);
sprintf(year, "%d", j);
while(fgets(str, sizeof(str), a) != NULL)
{
    while (j <= 2019)
    {
        if (strstr(str, year) != NULL)
        {
            qDebug() << QString::fromLocal8Bit(str);
            ui->textEdit->append(str);
            i=1;
        }
        j++;
        sprintf(year, "%d", j);
    }
    god = ui->spinBox->text();
    god.toLocal8Bit();
    GOD = god.toUtf8().data();
    j=atoi(GOD);
    sprintf(year, "%d", j);
}
if (i==0) ui->textEdit->append("no information");
fclose(a);
}

void Search::on_pushButtonClear_clicked()
{
    ui->textEdit->clear();
    ui->lineEditFam->clear();
    ui->lineEditMar->clear();
    ui->pushButtonSearch1->setEnabled(false);
    ui->pushButtonSearch2->setEnabled(false);
}

void Search::on_pushButtonMenu_clicked()
{

```

```

    fclose(a);
    this->close();
}

void Search::on_lineEditMar_textChanged(const QString &arg1)
{
    if ((ui->lineEditMar->text())!="") ui->pushButtonSearch1->setEnabled(true);
}

void Search::on_lineEditFam_textChanged(const QString &arg1)
{
    if ((ui->lineEditFam->text())!="") ui->pushButtonSearch2->setEnabled(true);
}

```

Файл “graph.h”

```

#ifndef GRAPH_H
#define GRAPH_H

```

```

#include <QDialog>

```

```

namespace Ui {
class Graph;
}

```

```

class Graph : public QDialog
{
    Q_OBJECT

```

```

public:
    explicit Graph(QWidget *parent = nullptr);
    ~Graph();

```

```

private slots:
    void on_pushButton_clicked();
    void showEvent(QShowEvent *);

    void on_pushButton_2_clicked();

```

```

private:
    Ui::Graph *ui;
};

```

```

#endif // GRAPH_H

```

Файл “graph.cpp”

```

#include "graph.h"
#include "ui_graph.h"
#include <QGraphicsTextItem>
QGraphicsScene *scene;
FILE *b;

Graph::Graph(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::Graph)
{
    ui->setupUi(this);
    this->setWindowTitle("График");
}

Graph::~Graph()
{
    delete ui;
}

void Graph::showEvent(QShowEvent *)
{
    scene = new QGraphicsScene(ui->graphicsView);
    int x=0,y=0,x1=0,y1=225;
    QPen pen(Qt::black);
    while (x<=435)
    {
        scene->addLine(x,y,x1,y1,pen);
        x=x+15;
        x1=x1+15;
    }
    x=0;y=0;x1=435;y1=0;
    while (y<=225)
    {
        scene->addLine(x,y,x1,y1,pen);
        y=y+15;
        y1=y1+15;
    }
    ui->graphicsView->setScene(scene);
    QGraphicsTextItem *textItem = new QGraphicsTextItem("1990");
    textItem->setPos(-18, 225);
    QGraphicsTextItem *textItem1 = new QGraphicsTextItem("1995");
    textItem1->setPos(59, 225);
    QGraphicsTextItem *textItem2 = new QGraphicsTextItem("2000");
    textItem2->setPos(136, 225);
    QGraphicsTextItem *textItem3 = new QGraphicsTextItem("2005");

```

```

textItem3->setPos(210, 225);
QGraphicsTextItem *textItem4 = new QGraphicsTextItem("2010");
textItem4->setPos(285, 225);
QGraphicsTextItem *textItem5 = new QGraphicsTextItem("2015");
textItem5->setPos(360, 225);
QGraphicsTextItem *textItem6 = new QGraphicsTextItem("год");
textItem6->setPos(430, 225);
scene->addItem(textItem);
scene->addItem(textItem1);
scene->addItem(textItem2);
scene->addItem(textItem3);
scene->addItem(textItem4);
scene->addItem(textItem5);
scene->addItem(textItem6);
QGraphicsTextItem *textItem01 = new QGraphicsTextItem("1");
textItem01->setPos(-18, 200);
QGraphicsTextItem *textItem02 = new QGraphicsTextItem("2");
textItem02->setPos(-18, 185);
QGraphicsTextItem *textItem03 = new QGraphicsTextItem("3");
textItem03->setPos(-18, 170);
QGraphicsTextItem *textItem04 = new QGraphicsTextItem("4");
textItem04->setPos(-18, 155);
QGraphicsTextItem *textItem05 = new QGraphicsTextItem("5");
textItem05->setPos(-18, 140);
QGraphicsTextItem *textItem06 = new QGraphicsTextItem("6");
textItem06->setPos(-18, 125);
QGraphicsTextItem *textItem07 = new QGraphicsTextItem("7");
textItem07->setPos(-18, 110);
QGraphicsTextItem *textItem08 = new QGraphicsTextItem("8");
textItem08->setPos(-18, 95);
QGraphicsTextItem *textItem09 = new QGraphicsTextItem("9");
textItem09->setPos(-18, 80);
QGraphicsTextItem *textItem10 = new QGraphicsTextItem("10");
textItem10->setPos(-20, 65);
QGraphicsTextItem *textItem11 = new QGraphicsTextItem("11");
textItem11->setPos(-20, 50);
QGraphicsTextItem *textItem12 = new QGraphicsTextItem("12");
textItem12->setPos(-20, 35);
QGraphicsTextItem *textItem13 = new QGraphicsTextItem("13");
textItem13->setPos(-20, 20);
QGraphicsTextItem *textItem14 = new QGraphicsTextItem("14");
textItem14->setPos(-20, 5);
QGraphicsTextItem *textItem15 = new QGraphicsTextItem("15");
textItem15->setPos(-20, -10);
QGraphicsTextItem *textItem16 = new QGraphicsTextItem("КОЛ-ВО");

```



```

textItem16->setPos(-30,-25);
scene->addItem(textItem01);
scene->addItem(textItem02);
scene->addItem(textItem03);
scene->addItem(textItem04);
scene->addItem(textItem05);
scene->addItem(textItem06);
scene->addItem(textItem07);
scene->addItem(textItem08);
scene->addItem(textItem09);
scene->addItem(textItem10);
scene->addItem(textItem11);
scene->addItem(textItem12);
scene->addItem(textItem13);
scene->addItem(textItem14);
scene->addItem(textItem15);
scene->addItem(textItem16);
}

void Graph::on_pushButton_clicked()
{
    char str[100],year[5]="1990";
    int i=0,i1=0,dx=0,dx1=15;
    b=fopen("moto.txt","r");
    rewind(b);
    int j=atoi(year);
    while(fgets(str,sizeof(str),b)!=NULL)
    {
        if (strstr(str, year)!=NULL) ++i;
    }
    j++;
    sprintf(year,"%d",j);
    rewind(b);
    while (j<=2019)
    {
        while(fgets(str,sizeof(str),b)!=NULL)
        {
            if (strstr(str, year)!=NULL) ++i1;
        }
        QPen pen(Qt::red);
        scene->addLine(dx,225-i*15,dx1,225-i1*15,pen);
        ui->graphicsView->setScene(scene);
        dx=dx+15;
        dx1=dx1+15;
        i=i1;
    }
}

```

```

        i1=0;
        j++;
        sprintf(year,"%d",j);
        rewind(b);
    }
    fclose(b);
}

void Graph::on_pushButton_2_clicked()
{
    fclose(b);
    this->close();
}

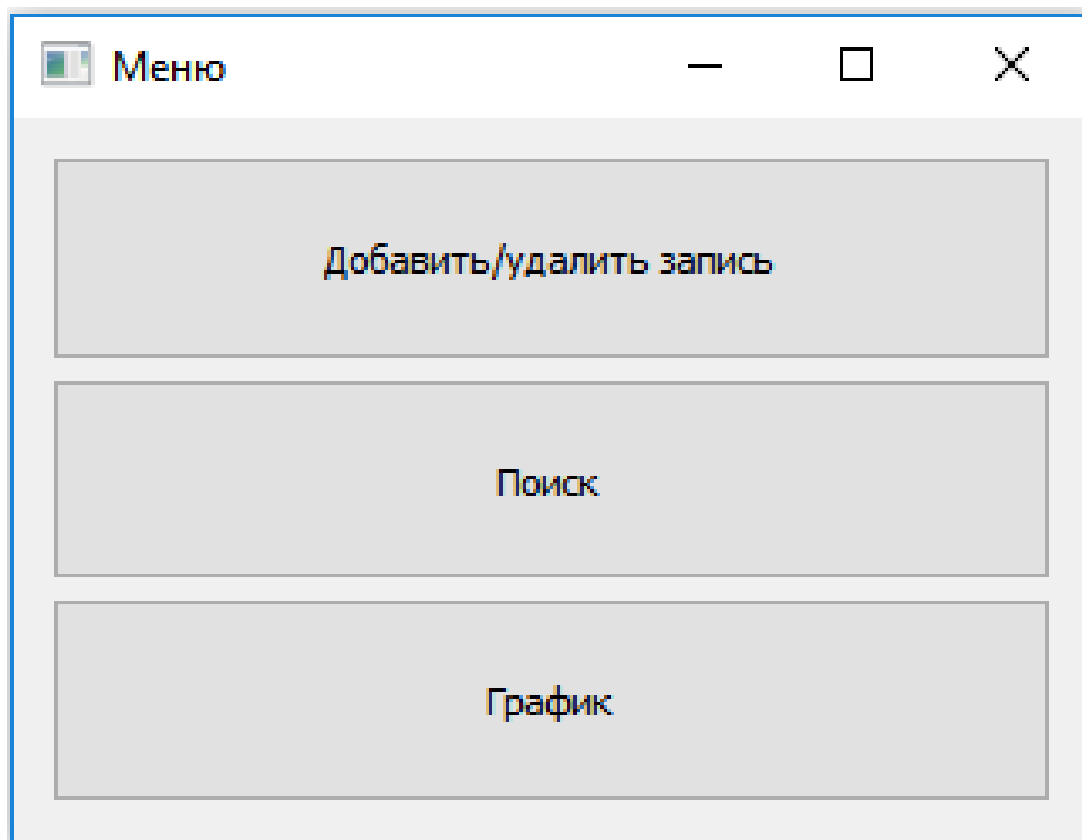
Файл “main.cpp”
#include "mainwindow.h"
#include <QApplication>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.show();

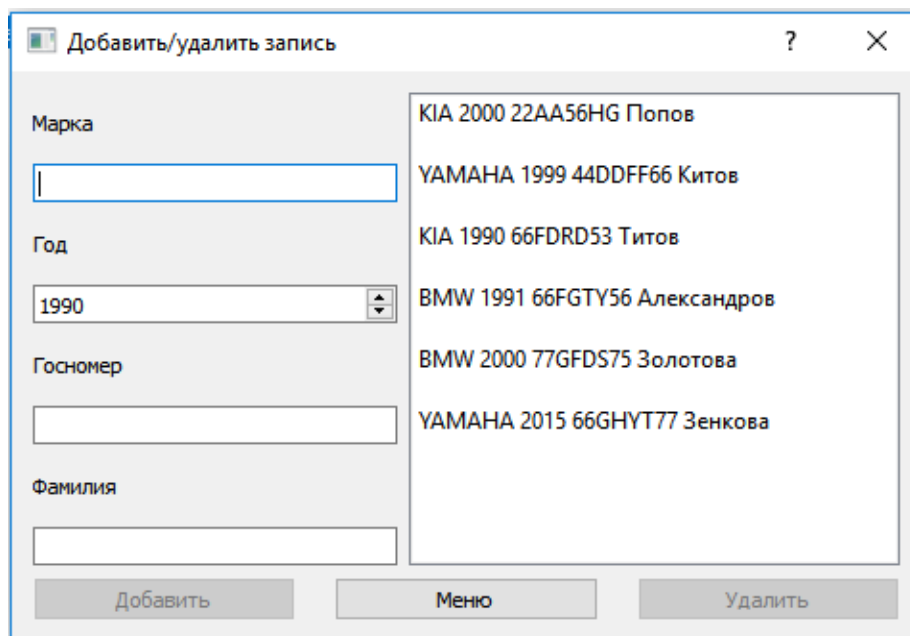
    return a.exec();
}

```

### 3.2. Пользовательский интерфейс. Результаты работы программы.



Скрин 1. Главное меню.



Скрин 2. Окно добавления/удаления записи.

Поиск

Марка

Поиск

Фамилия владельца

Поиск

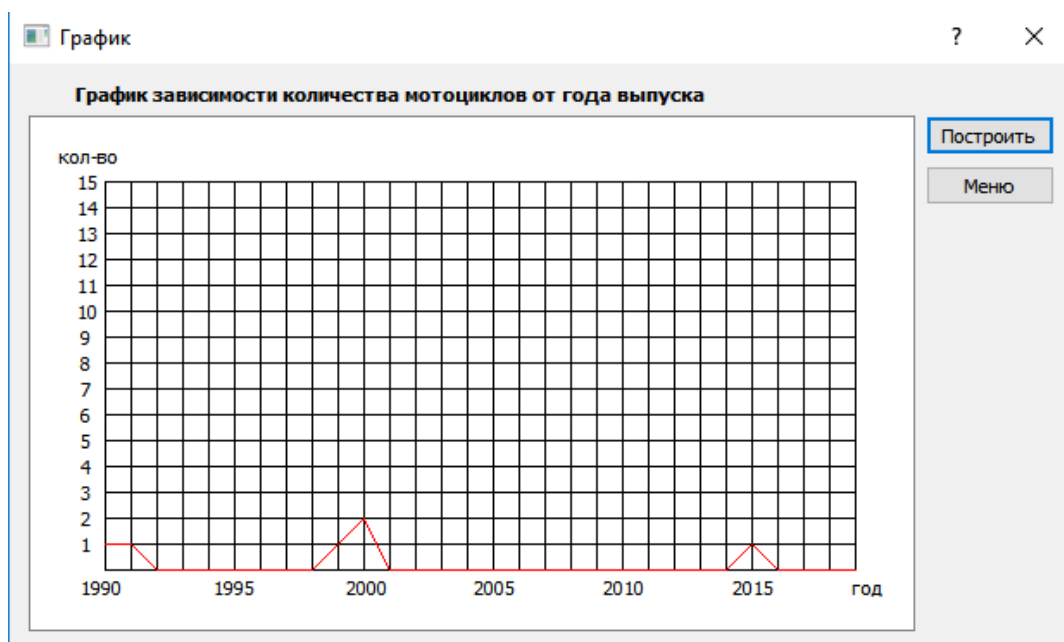
Год(не страше)

Поиск

Результаты поиска

Очистить Меню

Скрин 3. Окно поиска записей.



Скрин 4. Окно графика.

### 3.3. Диаграммы.

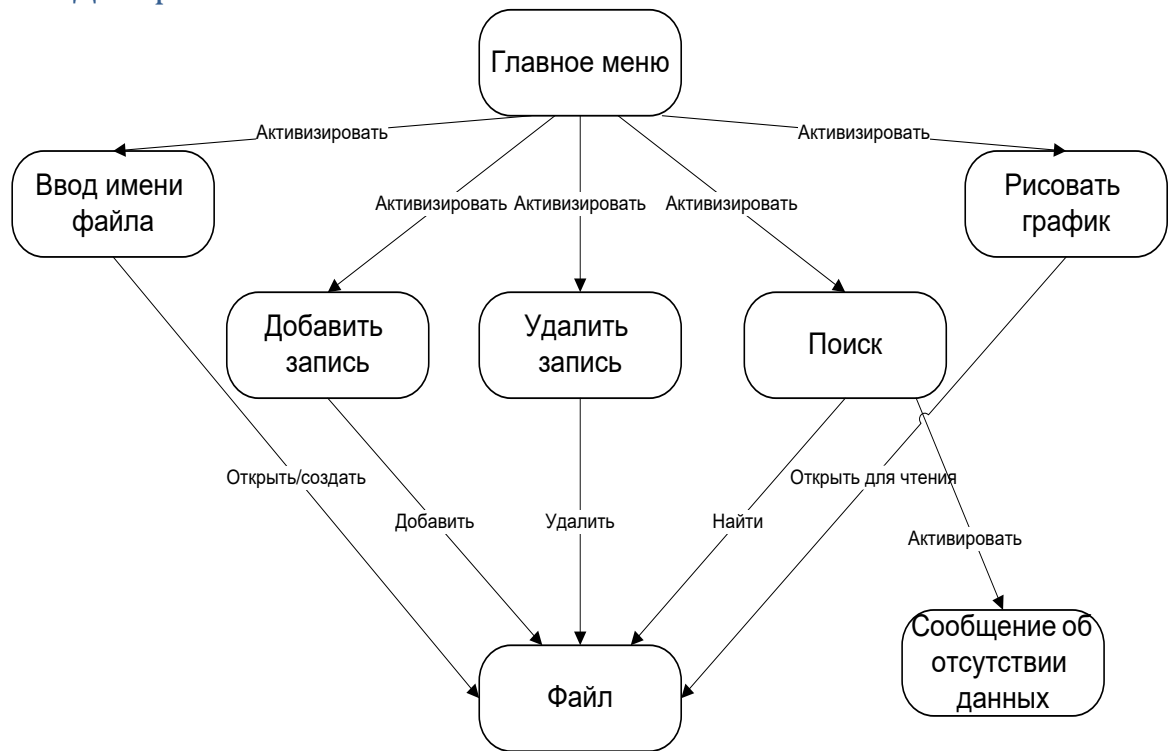


Схема 1. Объектная декомпозиция приложения.

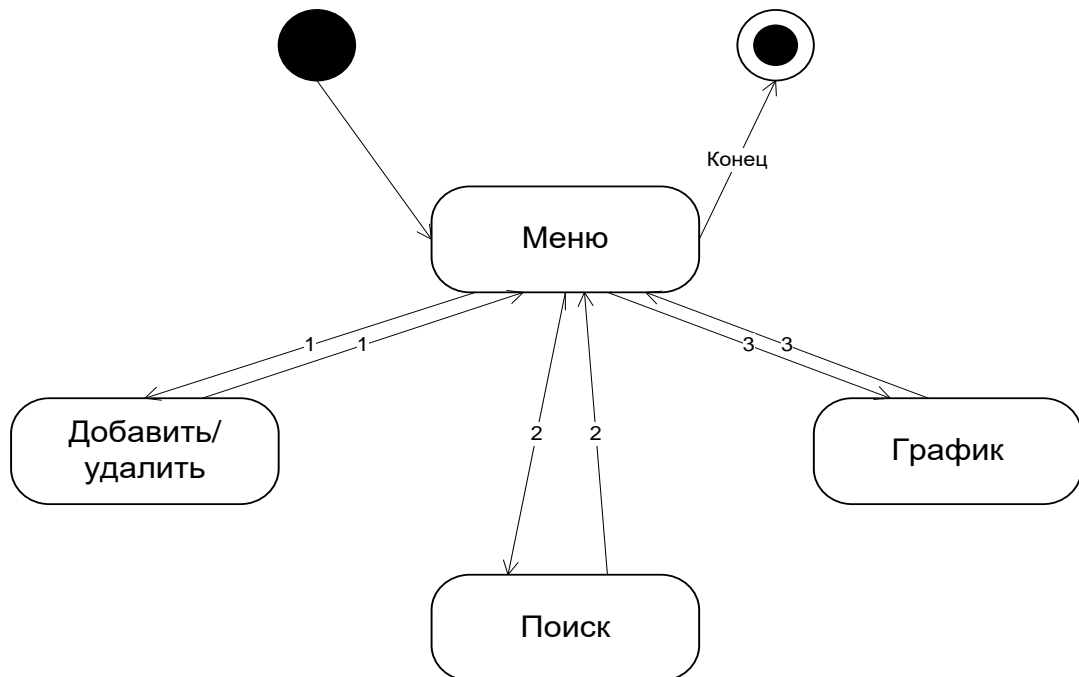


Схема 2. Диаграмма состояний интерфейса.

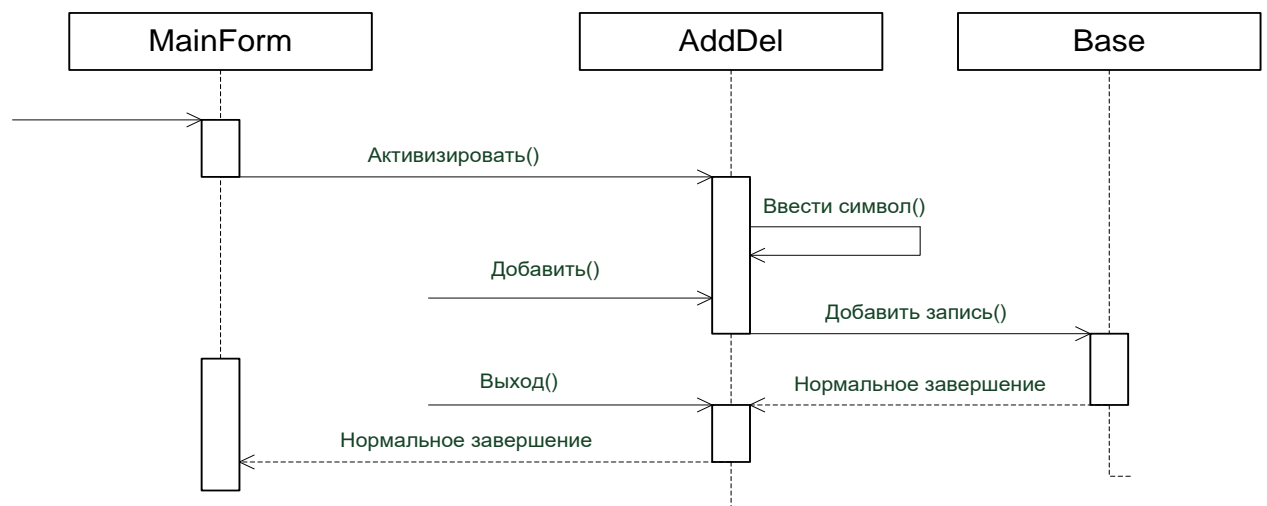


Схема 3. Диаграмма последовательностей функции записать мотоцикл.

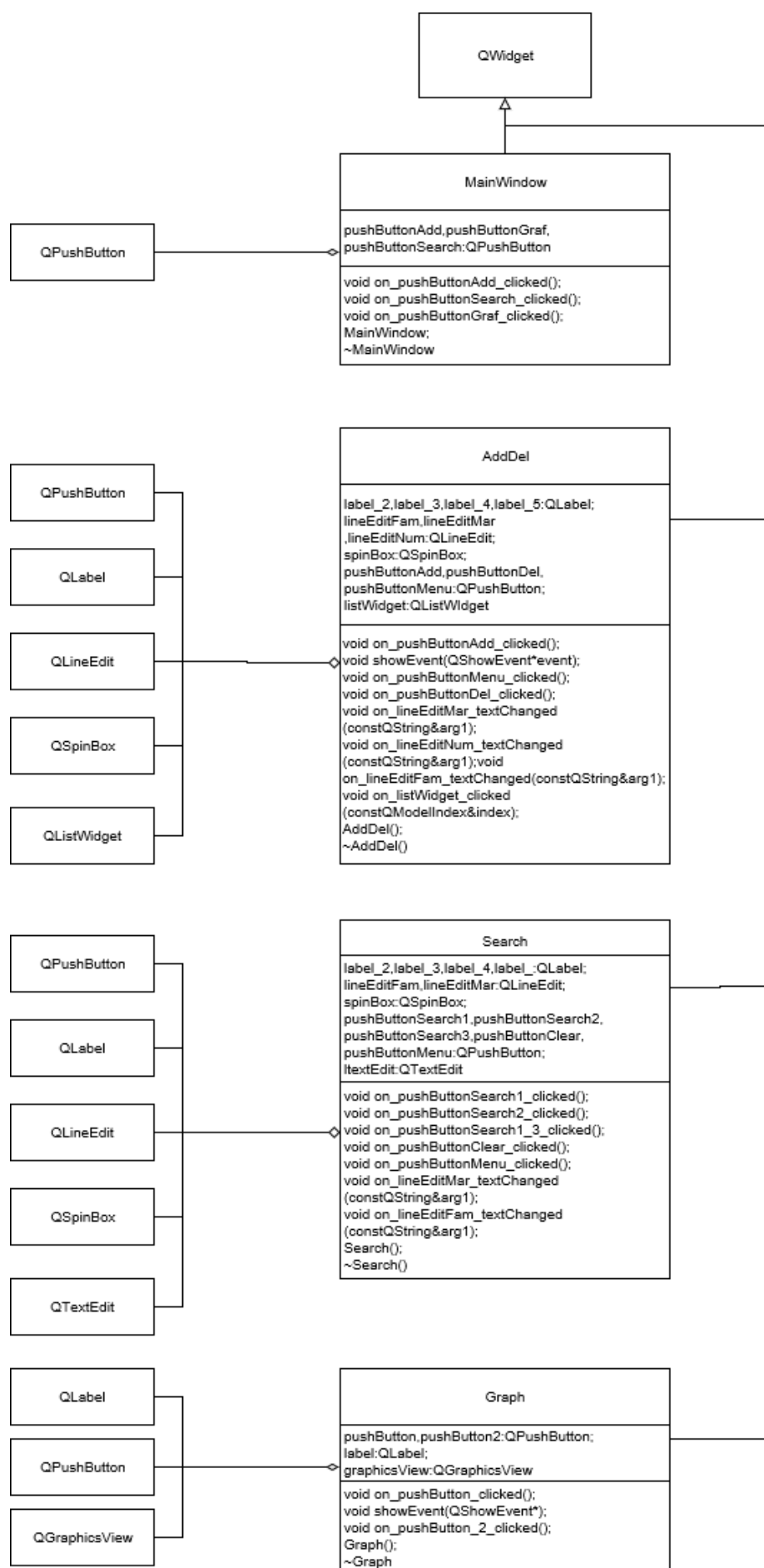


Схема 4. Диаграмма классов интерфейсной и предметной областей.

#### **4.Заключение.**

Выводы:

- научилась разрабатывать программы
- получила знание особенностей программирования
- научилась создавать - проектировать, отлаживать и тестировать –  
небольшие программные системы
- научилась работать с интерфейсом.

#### **5.Источники.**

Учебник «Основы программирования» Г. С. Иванова.

Сайт [ravesli.com](http://ravesli.com)