|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.03 Прикладная информатика**

**ОТЧЕТ ОБ УЧЕБНОЙ ПРАКТИКЕ №3**

|  |  |
| --- | --- |
| Тип практики | Проектно-технологическая практика |

|  |  |
| --- | --- |
| Название  предприятия | НУК ИУ МГТУ им. Н.Э. Баумана |

|  |  |  |  |
| --- | --- | --- | --- |
| Студент группы ИУ6-25 Б |  |  | Дулина И.А. |
|  |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |
| Руководитель практики |  |  |  |
|  |  | (Подпись, дата) | (И.О. Фамилия) |

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*2023 г.*

**Оглавление**

Техническое задание3

Код программы3

Примеры работающей версии программы35

Структурная декомпозиция38

Заключение41

**Задание 3. Создание программной системы c Qt интерфейсом на С++**

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

О каждом товаре на овощной базе известны следующие сведения: наименование товара, поставщик, количество в наличии (кг), цена (за кг). 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, какого товара на базе больше всего.

2. Определить, сколько килограмм товара каждого наименования и поставщика можно купить на заданную сумму (показать все комбинации «товар, поставщик – масса»).

3. Определить, у какого поставщика выгоднее всего брать заданный товар.

4. Построить гистограмму, демонстрирующую среднюю цену за килограмм каждого товара.

**Цель:** получение навыков создания программной системы на языке С++ с помощью среды QtCreator

*Код модуля basefile.h:*

#ifndef BASEFILE\_H

#define BASEFILE\_H

#include <QFile>

struct Spis{

QString nam; QString pro, am, pr;

Spis\* p, \*s;

};

struct Spisok{

QString name; int k, allpr, count;//количество, сумма всех цен, количество производителей

//(2 последних - ддля гистограммы и средней цены

Spisok \*p;

};

struct recType{

QString name, prov, amount, price;

};

class baseFile

{

QFile \*f;

bool k1,//в запросе присутствует фамилия

k2,//в запросе присутствует имя

k3,//найдена фамилия

k4,//найдено имя

ff;//найдена запись

public:

recType r;

baseFile();

~baseFile();

bool addRec(recType r);

bool delRec(recType r);

bool readRec();

bool findFirst(const recType r1);

bool findNext(const recType r1);

bool FromFirst();

};

#endif // BASEFILE\_H

*Код модуля basefile.cpp:*

#include "basefile.h"

#include <QMessageBox>

baseFile::baseFile()

{

f=new QFile("base.txt");

if(!f->exists()) // если файл не существует, то

{ // формирмируем сообщение

QMessageBox msg(QMessageBox::Critical, "Файл не найден",

"Файл base.txt не создан",

QMessageBox::Ok,0);

msg.exec();// выводим сообщение на экран

}

f->open(QFile::ReadWrite); // открываем файл для ввода-вывода

}

baseFile::~baseFile() // деструктор

{

f->close(); // закрываем файл

delete f; // освобождаем память под указатель

}

bool baseFile::addRec(recType r)

{

f->seek(f->size()); // переходим на конец файла

QDataStream out(f); // связываем с файлом поток вывода

out<<r.name<<r.prov<<r.amount<<r.price;// выводим данные в файл

return true;

}

bool baseFile::delRec(recType r1){

f->seek(0);

QFile\* f1=new QFile("trash.txt");

f1->open(QFile::ReadWrite);

k1=(r1.name==""); // устанавливаем два ключа поиска, присутствует имя

k2=(r1.prov=="");//присутсвует фамилия

f->reset();

f1->reset();

bool fff = FromFirst();

while(fff)

{

k3=(r1.name==r.name); //строим еще два ключа поиска

k4=(r1.prov==r.prov);

QDataStream out1(f1);

if ((!k1 && !k2 && (!k3 || !k4))||

(!k1 && k2 && !k3)||

(k1 && !k2 && !k4)){

out1<<r.name<<r.prov<<r.amount<<r.price;

int k=sizeof(r);

k=f1->size();

}

fff=readRec();

}

f->resize(0);

f1->reset();

fff=true;

while (fff){

QDataStream in1(f1); // связываем с файлом поток ввода

if (in1.atEnd()||r.name=="")fff=false;

else

{

in1>>r.name>>r.prov>>r.amount>>r.price;

addRec(r);

}

}

f1->resize(0);

f1->close();

delete f1;

return true;

}

bool baseFile::readRec()//для таблицы

{

QDataStream in(f); // связываем с файлом поток ввода

if (in.atEnd())return false;

else

{

in>>r.name>>r.prov>>r.amount>>r.price;

return true;

}

}

bool baseFile::findFirst(const recType r1)//первое совпадение

{

k1=(r1.name==""); // устанавливаем два ключа поиска, присутствует имя

k2=(r1.prov=="");//присутсвует фамилия

ff=false; // устанавливаем ключ поиска «запись не найдена»

f->reset();

bool fff = readRec();

while(fff &&(!ff))

{

k3=(r1.name==r.name); //строим еще два ключа поиска

k4=(r1.prov==r.prov);

if ((!k1 && !k2 && k3 && k4)||

(!k1 && k2 && k3)||//найдено имя

(k1 && !k2 && k4)) //найдена фамилия

ff=true; // ключ поиска «запись найдена»

else fff=readRec();

}

return ff; // возвращаем ключ поиска

}

bool baseFile::findNext(const recType r1)

{

ff=false; // ключ поиска «запись не найдена»

bool fff = readRec();

while((!ff) && fff)

{

k3=(r1.name==r.name);//строим еще два ключа поиска

k4=(r1.prov==r.prov);

if ((!k1 && !k2 && k3 && k4)|| //если имя и пров. не пустые и они равны нужному

(!k1 && k2 && k3)||

(k1 && !k2 && k4))

ff=true; // ключ поиска «запись найдена»

else fff=readRec();

}

return ff; // возвращаем ключ поиска

}

bool baseFile::FromFirst(){

bool ff;

f->seek(0);

ff=readRec();

return ff;

}

*Код модуля tableform.h:*

#ifndef TABLEFORM\_H

#define TABLEFORM\_H

#include <QObject>

#include <QWidget>

#include <basefile.h>

#include <QTableWidget>

#include <QPushButton>

class TableForm:public QWidget

{

QTableWidget \*table;

QPushButton \*Exitbtn;

public:

TableForm();

void showAll();//показать все записи

void showRow(int i,recType r);

void showResults(recType r1);

};

#endif // TABLEFORM\_H

*Код модуля tableform.cpp:*

#include "mainform.h"

#include "tableform.h"

#include <QHBoxLayout>

#include <QVBoxLayout>

#include <QMessageBox>

TableForm::TableForm()

{

this->setWindowTitle("Все записи");

QStringList strlist;

strlist<<("Имя товара")<<"Производитель"<<"Количество"<<"Цена";

table=new QTableWidget(20, 4, this);

table->setHorizontalHeaderLabels(strlist);

QHBoxLayout\*layout=new QHBoxLayout();

Exitbtn=new QPushButton("Меню");

layout->addWidget(Exitbtn);

QVBoxLayout\*layout1=new QVBoxLayout();

layout1->addWidget(table);

layout1->addLayout(layout);

setLayout(layout1);

connect(Exitbtn, SIGNAL(clicked(bool)), this, SLOT(close()));

}

void TableForm::showRow(int i,recType r)

{

QTableWidgetItem \*item; // элемент таблицы

item = new QTableWidgetItem(); // создаем элемент

item->setFlags(Qt::NoItemFlags);//запрещаем выделение

item->setText(r.name); // записываем текст

table->setItem(i,0,item);// привязываем элемент к таблице

item = new QTableWidgetItem();// создаем элемент

item->setFlags(Qt::NoItemFlags); //запрещаем выделение

item->setText(r.prov);

table->setItem(i,1,item); // привязываем элемент

item = new QTableWidgetItem();// создаем элемент

item->setFlags(Qt::NoItemFlags); //запрещаем выделение

item->setText(r.amount); // записываем текст

table->setItem(i,2,item); // привязываем элемент

item = new QTableWidgetItem();// создаем элемент

item->setFlags(Qt::NoItemFlags); //запрещаем выделение

item->setText(r.price); // записываем текст

table->setItem(i,3,item);

}

void TableForm::showAll()

{

baseFile base;

if (!base.FromFirst())

{ // если файл пустой , то создаем сообщение

QMessageBox msg(QMessageBox::Critical,

"Нет данных",

"База пуста",

QMessageBox::Ok,0);

msg.exec(); // выводим сообщение

}

else

{ // иначе - выводим таблицу по строкам

showRow(0,base.r);

int i=0;

while (base.readRec())

showRow(++i,base.r);

table->setRowCount(i+1);

resize(600,400);

show();

}

base.FromFirst();

}

void TableForm::showResults(recType r1)

{

baseFile base;

//base->reset();

if (!base.findFirst(r1))

{ // если данные не найдены, то создаем сообщение

QMessageBox msg(QMessageBox::Critical,

"Нет данных",

"Данные не найдены",

QMessageBox::Ok,0);

msg.exec();

}

else

{ // иначе - выводим результаты по строкам

showRow(0,base.r);

int i=0;

while (base.findNext(r1))

showRow(++i,base.r);

table->setRowCount(i+1);

resize(350,200);

show();

}

}

*Код модуля themostform.h:*

#ifndef THEMOSTFORM\_H

#define THEMOSTFORM\_H

#include <QWidget>

#include <QObject>

#include <basefile.h>

#include <QPushButton>

#include <QLineEdit>

class themostForm:public QWidget

{

Q\_OBJECT//!!!!!работает с сигналами и слотами

QPushButton\* countbtn;

QPushButton\* exitbtn;

QLineEdit\* resulte;

public:

themostForm();

public slots:

void prCount();

};

#endif // THEMOSTFORM\_H

*Код модуля themostform.cpp:*

#include "themostform.h"

#include "QVBoxLayout"

themostForm::themostForm()

{

this->setWindowTitle("Какого товара больше всего?");

countbtn=new QPushButton("Посчитать");

exitbtn=new QPushButton("Меню");

resulte=new QLineEdit;

resulte->setReadOnly(true);

QVBoxLayout\* layout=new QVBoxLayout();

layout->addWidget(countbtn);

layout->addWidget(resulte);

layout->addWidget(exitbtn);

setLayout(layout);

resize(300,150);

resulte->clear();

connect(countbtn, SIGNAL(clicked(bool)), this, SLOT(prCount()));

connect(exitbtn, SIGNAL(clicked(bool)), this, SLOT(close()));

}

void themostForm::prCount(){

int max = 0;

QString str;

baseFile base;

bool fff = base.FromFirst();

if (fff==false) resulte->setText("0");

else {

Spisok \*first=new Spisok;

first->name=base.r.name;

str=base.r.amount;

first->k=str.toInt();

//first->allpr=base.r.price.toInt();

///first->count=1;

first->p=nullptr;

Spisok\*r, \*r1;

while (fff=base.readRec(), str=base.r.amount, fff){

r=first;

bool find=false;

while (r!=nullptr&&find==false){

if (base.r.name==r->name) {

r->k+=str.toInt();

//r->allpr+=base.r.price.toInt();

//r->count+=1;

find=true;}

else{

if(r->p!=nullptr)r=r->p;

else{

r1=new Spisok;

r1->name=base.r.name;

r1->k=str.toInt();

//r->allpr=base.r.price.toInt();

//r->count=1;

r1->p=nullptr;

r->p=r1;

r=r1;

find=true;}

}

}

}

r=first;r1=first;

while (r!=nullptr){

if (r->k>max){

str=r->name;

max=r->k;

}

r=r->p;

delete r1;

r1=r;

}

resulte->setText(str);

}

}

*Код модуля muchform.h:*

#ifndef MUCHFORM\_H

#define MUCHFORM\_H

#include <QObject>

#include <QWidget>

#include <QPushButton>

#include <QLineEdit>

#include <QLabel>

#include <basefile.h>

class muchForm:public QWidget

{

Q\_OBJECT

int j=0;

Spis\*first;

QPushButton\* prevbtn;

QPushButton\* inputbtn;

QPushButton\* sucbtn;

QPushButton\* exitbtn;

QLabel\* inputl;

QLineEdit\* inpute;

QLabel\* namel;

QLineEdit\* namee;

QLabel\* provl;

QLineEdit\* prove;

QLabel\* pricel;

QLineEdit\* pricee;

QLabel\* massl;

QLineEdit\* masse;

friend Spis\* createsp();

public:

muchForm();

public slots:

void printinf();

void printsuc();

void printprev();

void closeform();

};

Spis\* createsp();

#endif // MUCHFORM\_H

*Код модуля muchform.cpp:*

#include "muchform.h"

#include <QGridLayout>

#include <QVBoxLayout>

#include <QHBoxLayout>

#include <QMessageBox>

muchForm::muchForm()

{

this->setWindowTitle("Сколько кг можно купить на сумму");

prevbtn = new QPushButton("Предыдущий");

inputbtn= new QPushButton("Вывести");

sucbtn=new QPushButton("Следующий");

exitbtn=new QPushButton("Меню");

inputl=new QLabel("Введите сумму");

inpute=new QLineEdit;

namel=new QLabel("Товар");

namee=new QLineEdit;

namee->setReadOnly(true);

provl=new QLabel("Поставщик");

prove=new QLineEdit;

prove->setReadOnly(true);

pricel=new QLabel("Цена за 1 кг");

pricee=new QLineEdit;

pricee->setReadOnly(true);

massl=new QLabel("Масса");

masse=new QLineEdit;

masse->setReadOnly(true);

prevbtn->setEnabled(false);

QGridLayout\*layout=new QGridLayout();

layout->addWidget(namel, 1, 1);

layout->addWidget(namee, 1, 2);

layout->addWidget(provl, 2, 1);

layout->addWidget(prove, 2, 2);

layout->addWidget(pricel, 3, 1);

layout->addWidget(pricee, 3, 2);

layout->addWidget(massl, 4, 1);

layout->addWidget(masse, 4, 2);

layout->addWidget(prevbtn, 5, 1);

layout->addWidget(inputbtn, 5, 2);

layout->addWidget(sucbtn, 5, 3);

layout->addWidget(exitbtn, 6, 2);

QHBoxLayout\*layouttop=new QHBoxLayout();

layouttop->addWidget(inputl);

layouttop->addWidget(inpute);

QVBoxLayout\*all=new QVBoxLayout();

all->insertLayout(0, layouttop, 0);

all->insertLayout(1, layout, 0);

setLayout(all);

connect(exitbtn, SIGNAL(clicked(bool)), this, SLOT(closeform()));

connect(prevbtn, SIGNAL(clicked(bool)), this, SLOT(printprev()));

connect(sucbtn, SIGNAL(clicked(bool)), this, SLOT(printsuc()));

connect(inputbtn, SIGNAL(clicked(bool)), this, SLOT(printinf()));

}

void muchForm::closeform(){

Spis\*r=first;

while (r=first, first!=nullptr){

first=first->p;

delete r;

}

close();

}

void muchForm::printsuc(){

j+=1;

prevbtn->setEnabled(true);

printinf();

}

void muchForm::printprev(){

j-=1;

sucbtn->setEnabled(true);

printinf();

}

void muchForm::printinf(){

int summa=inpute->text().toInt(), h;

first=createsp(); Spis \*r;

r=first;

for (int i=0; i<j; i++) r=r->s;

namee->setText(r->nam);

prove->setText(r->pro);

pricee->setText(r->pr);

h=summa/r->pr.toInt();

if (h<=r->am.toInt()){

QString str=QString::number(h);

masse->setText(str);

}

else masse->setText(r->am);

if (r->s==nullptr) sucbtn->setEnabled(false);

if (r->p==nullptr) prevbtn->setEnabled(false);

}

Spis\* createsp(){

baseFile base;

bool fff = base.FromFirst();

Spis\*first, \*r, \*r1;

if (fff==false){QMessageBox msg(QMessageBox::Critical, "Файл не найден",

"Файл base.txt не создан",

QMessageBox::Ok,0);

msg.exec();

}

else {

first=new Spis;

first->nam=base.r.name;

first->pro=base.r.prov;

first->am=base.r.amount;

first->pr=base.r.price;

first->p=nullptr;

first->s=nullptr;

r=first;

while (fff=base.readRec(), fff){

r1=new Spis;

r1->nam=base.r.name;

r1->pro=base.r.prov;

r1->am=base.r.amount;

r1->pr=base.r.price;

r1->p=r;

r1->s=nullptr;

r->s=r1;

r=r1;

}

}

r=first;

return first;

}

*Код модуля benproform.h:*

#ifndef BENPROFORM\_H

#define BENPROFORM\_H

#include <QObject>

#include <QWidget>

#include <QPushButton>

#include <QLineEdit>

#include <QLabel>

#include <basefile.h>

class BenProForm:public QWidget

{

Q\_OBJECT

QPushButton\* inputbtn;

QPushButton\* exitbtn;

QLabel\* inputNl;//имя товара

QLineEdit\* inputNe;

QLabel\* outputPl;//имя производителя

QLineEdit\* outputPe;

public:

BenProForm();

public slots:

void printBen();

};

#endif // BENPROFORM\_H

*Код модуля benproform.cpp:*

#include "benproform.h"

//#include "muchform.h"

#include <QGridLayout>

#include <QVBoxLayout>

#include <QHBoxLayout>

BenProForm::BenProForm()

{

this->setWindowTitle("Поиск самого выгоодного поставщикф");

inputbtn= new QPushButton("Вывести");

exitbtn=new QPushButton("Меню");

inputNl=new QLabel("Введите наименование товара");

inputNe=new QLineEdit;

outputPl=new QLabel("Самый выгодный поставщик");

outputPe=new QLineEdit;

inputNe->clear();

outputPe->clear();

inputNe->setFocus();

outputPe->setReadOnly(true);

QGridLayout\*layout=new QGridLayout();

layout->addWidget(inputNl, 1, 1);

layout->addWidget(inputNe, 1, 2);

layout->addWidget(outputPl, 2, 1);

layout->addWidget(outputPe, 2, 2);

layout->addWidget(inputbtn, 3, 1);

layout->addWidget(exitbtn, 3, 2);

setLayout(layout);

connect(exitbtn, SIGNAL(clicked(bool)), this, SLOT(close()));

connect(inputbtn, SIGNAL(clicked(bool)), this, SLOT(printBen()));

}

void BenProForm::printBen(){

QString str=inputNe->text(), result;int min=1000;

baseFile base;

bool fff=base.FromFirst();

while(fff){

if (base.r.name==str&&base.r.price.toInt()<min){

result=base.r.prov;

min=base.r.price.toInt();}

fff=base.readRec();

}

outputPe->setText(result);

}

*Код модуля gystform.h:*

#ifndef GYSTFORM\_H

#define GYSTFORM\_H

#include <QObject>

#include <QWidget>

#include <QPushButton>

#include <QVBoxLayout>

#include <QHBoxLayout>

#include <QGridLayout>

#include "basefile.h"

#include <QtCharts/QtCharts>

class gystForm:public QWidget

{

Q\_OBJECT

bool twice=false;

QPushButton\* exitbtn;

QPushButton\* printbtn;

QChart\* Gystog;

QChartView\* chartview;

QBarSeries \*series;

QValueAxis \*axisY;

QBarCategoryAxis \*axisX;

QBarSet\*srpr;

QStringList categories;//имена продуктов

public:

gystForm();

public slots:

void printGyst();

//void closeform();

};

#endif // GYSTFORM\_H

*Код модуля gystform.cpp:*

#include "gystform.h"

gystForm::gystForm()

{

this->setWindowTitle("Гистограмма");

exitbtn=new QPushButton("Меню");

printbtn=new QPushButton("Вывести");

Gystog=new QChart();

chartview = new QChartView(Gystog);

axisY= new QValueAxis();

axisX= new QBarCategoryAxis();

series= new QBarSeries();

srpr=new QBarSet("Средняя цена");

QHBoxLayout\* layout=new QHBoxLayout();

layout->addWidget(printbtn);

layout->addWidget(exitbtn);

QVBoxLayout\*layout1=new QVBoxLayout();

layout1->addWidget(chartview);

layout1->addLayout(layout);

setLayout(layout1);

resize(600, 400);

connect(exitbtn, SIGNAL(clicked(bool)), this, SLOT(close()));

connect(printbtn, SIGNAL(clicked(bool)), this, SLOT(printGyst()));

}

void gystForm::printGyst(){

baseFile base;float max=0;

srpr->remove(0, categories.size());

axisX->clear();

categories.resize(0);

bool fff = base.FromFirst();

if (fff==false) {QMessageBox msg(QMessageBox::Critical, "Файл не найден",

"Файл base.txt не создан",

QMessageBox::Ok,0);

msg.exec();

}

else {

Spisok \*first=new Spisok;

first->name=base.r.name;

QString str=base.r.amount;

//first->k=str.toInt();

first->allpr=base.r.price.toInt();

first->count=1;

first->p=nullptr;

Spisok\*r, \*r1;

while (fff=base.readRec(), str=base.r.amount, fff){

r=first;

bool find=false;

while (r!=nullptr&&find==false){

if (base.r.name==r->name) {

//r->k+=str.toInt();

r->allpr+=base.r.price.toInt();

r->count+=1;

find=true;}

else{

if(r->p!=nullptr)r=r->p;

else{

r1=new Spisok;

r1->name=base.r.name;

//r1->k=str.toInt();

r1->allpr=base.r.price.toInt();

r1->count=1;

r1->p=nullptr;

r->p=r1;

r=r1;

find=true;

}

}

}

}

r=first;r1=first;

while (r!=nullptr){

srpr->append(r->allpr/r->count);

categories<<r->name;

if (r->allpr/r->count>max) max=r->allpr/r->count;

r=r->p;

delete r1;

r1=r;

}

}

series->append(srpr);

//-

Gystog->addSeries(series);

//про y

axisY->setRange(0,max);

Gystog->addAxis(axisY, Qt::AlignLeft);

series->attachAxis(axisY);

//про x

axisX->append(categories);

Gystog->addAxis(axisX, Qt::AlignBottom);

series->attachAxis(axisX);

//--

Gystog->legend()->setVisible(true);

Gystog->legend()->setAlignment(Qt::AlignBottom);

chartview->setRenderHint(QPainter::Antialiasing);

twice=true;

}

*Код модуля mainform.h:*

#ifndef MAINFORM\_H

#define MAINFORM\_H

#include "gystform.h"

#include <QWidget>

#include <QPushButton>

#include <QLabel>

#include <QLineEdit>

#include <TableForm.h>

#include <themostform.h>

#include <muchform.h>

#include <benproform.h>

#include <gystForm.h>

class mainForm : public QWidget

{

Q\_OBJECT

QPushButton\* Exitbtn;

QPushButton\* addbtn;

QPushButton\* delbtn;

QPushButton\* mostbtn;

QPushButton\* muchbtn;

QPushButton\* benbtn;

QPushButton\* gystbtn;

QPushButton\* tablebtn;

QLabel\* namel;

QLineEdit\* namee;

QLabel\* provl;

QLineEdit\* prove;

QLabel\* amountl;

QLineEdit\* amounte;

QLabel\* pricel;

QLineEdit\* pricee;

TableForm winTable;

themostForm winMost;

muchForm winMuch;

BenProForm winBen;

gystForm winGyst;

public:

mainForm();

public slots:

void addRecord();

void delRecord();

void mostAmount();

void muchBuy();

void benProvider();

void printGyst();

void printTable();

};

#endif // MAINFORM\_H

*Код модуля mainform.cpp:*

#include "mainform.h"

#include "basefile.h"

#include <QVBoxLayout>

#include <QHBoxLayout>

#include <QGridLayout>

mainForm::mainForm()

{

this->setWindowTitle("Овощная база");

Exitbtn= new QPushButton("Выйти");

mostbtn=new QPushButton("Какого товара больше всего?");

muchbtn=new QPushButton("Сколько кг можно купить");

benbtn=new QPushButton("Самый выгодный пооставщик");

gystbtn=new QPushButton("Гистограмма");

namel=new QLabel("Имя товара");

namee=new QLineEdit;

provl=new QLabel("Поставщик");

prove=new QLineEdit;

amountl=new QLabel("Количество в наличии (кг)");

amounte=new QLineEdit;

pricel=new QLabel("Цена (за 1 кг)");

pricee=new QLineEdit;

addbtn=new QPushButton("Добавить");

delbtn=new QPushButton("Удалить");

tablebtn= new QPushButton("Таблица");

namee->setFocus();

QVBoxLayout\* right = new QVBoxLayout ();

right->addWidget(mostbtn);

right->addWidget(muchbtn);

right->addWidget(benbtn);

right->addWidget(gystbtn);

right->addWidget(Exitbtn);

QHBoxLayout\* adde =new QHBoxLayout();

adde->addWidget(addbtn);

adde->addWidget(delbtn);

QGridLayout\*left = new QGridLayout();

left->addWidget(namel, 1, 1);

left->addWidget(provl, 2, 1);

left->addWidget(amountl, 3, 1);

left->addWidget(pricel, 4, 1);

left->addWidget(namee, 1, 2);

left->addWidget(prove, 2, 2);

left->addWidget(amounte, 3, 2);

left->addWidget(pricee, 4, 2);

left->addLayout(adde, 5, 2);

left->addWidget(tablebtn, 6, 2);

QHBoxLayout\* layout2=new QHBoxLayout();

layout2->insertLayout(0, right, 0);

layout2->insertStretch(1, 0);

layout2->insertLayout(2, left, 0);

setLayout(layout2);

resize(600, 300);//размеры окна

connect(Exitbtn, SIGNAL(clicked(bool)), this, SLOT(close()));

connect(addbtn, SIGNAL(clicked(bool)),this,SLOT(addRecord()));

connect(delbtn, SIGNAL(clicked(bool)),this,SLOT(delRecord()));

//другие формы:

connect(mostbtn, SIGNAL(clicked(bool)),this,SLOT(mostAmount()));

connect(muchbtn, SIGNAL(clicked(bool)),this,SLOT(muchBuy()));

connect(benbtn, SIGNAL(clicked(bool)),this,SLOT(benProvider()));

connect(gystbtn, SIGNAL(clicked(bool)),this,SLOT(printGyst()));

connect(tablebtn, SIGNAL(clicked(bool)),this,SLOT(printTable()));

}

void mainForm::addRecord(){

baseFile base;

recType r;

r.name=namee->text();

r.prov=prove->text();

r.amount=amounte->text();

r.price=pricee->text();

namee->clear();

prove->clear();

amounte->clear();

pricee->clear();

base.addRec(r);

namee->setFocus();

}

void mainForm::delRecord(){

baseFile base;

recType r;

r.name=namee->text();

r.prov=prove->text();

//r.amount=amounte->text();

//r.price=pricee->text();

namee->clear();

prove->clear();

amounte->clear();

pricee->clear();

base.delRec(r);

namee->setFocus();

}

//показать другие формы

void mainForm::mostAmount(){

winMost.show();

}

void mainForm::muchBuy(){

winMuch.show();

}

void mainForm::benProvider(){

winBen.show();

}

void mainForm::printGyst(){

winGyst.show();

}

void mainForm::printTable(){

winTable.showAll();

}

*Код модуля main.cpp:*

#include "mainform.h"

#include <QApplication>

#include <QLocale>

#include <QTranslator>

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

{

QApplication a(argc, argv);

QTranslator translator;

const QStringList uiLanguages = QLocale::system().uiLanguages();

for (const QString &locale : uiLanguages) {

const QString baseName = "practice3\_" + QLocale(locale).name();

if (translator.load(":/i18n/" + baseName)) {

a.installTranslator(&translator);

break;

}

}

mainForm w;

w.show();

return a.exec();

}

Рисунок 1-6 – работающая версия программы:

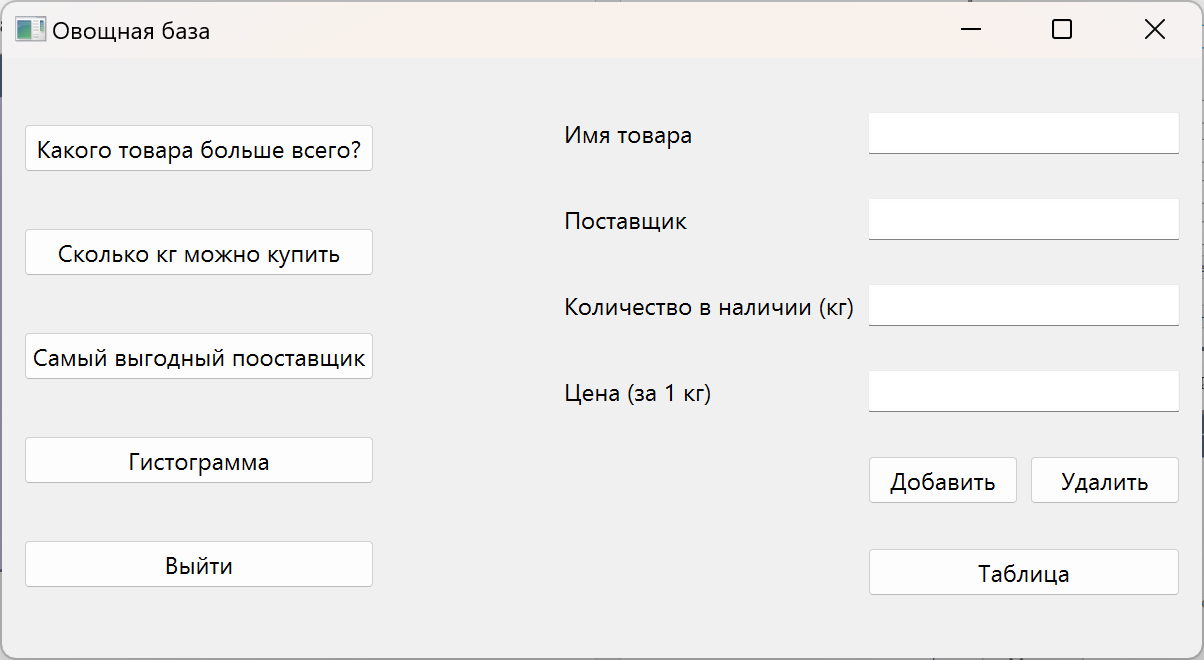


Рис. 1

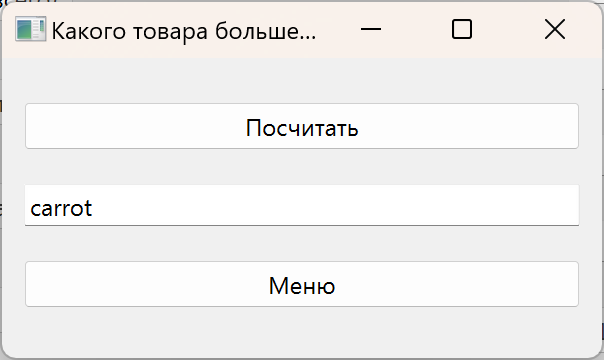


Рис.2

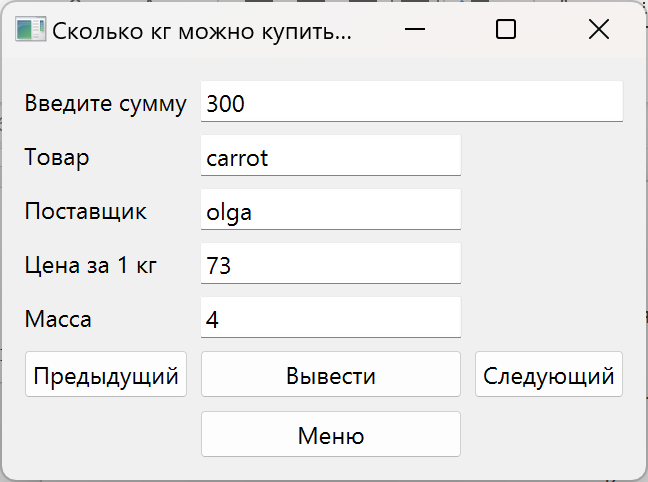


Рис.3

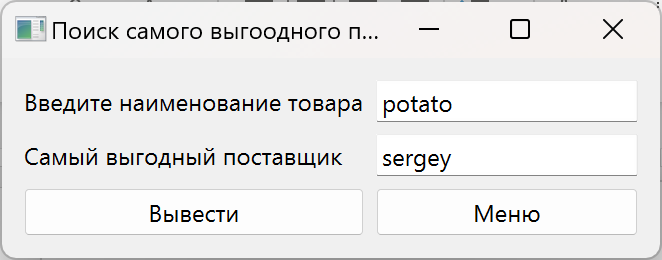


Рис.4

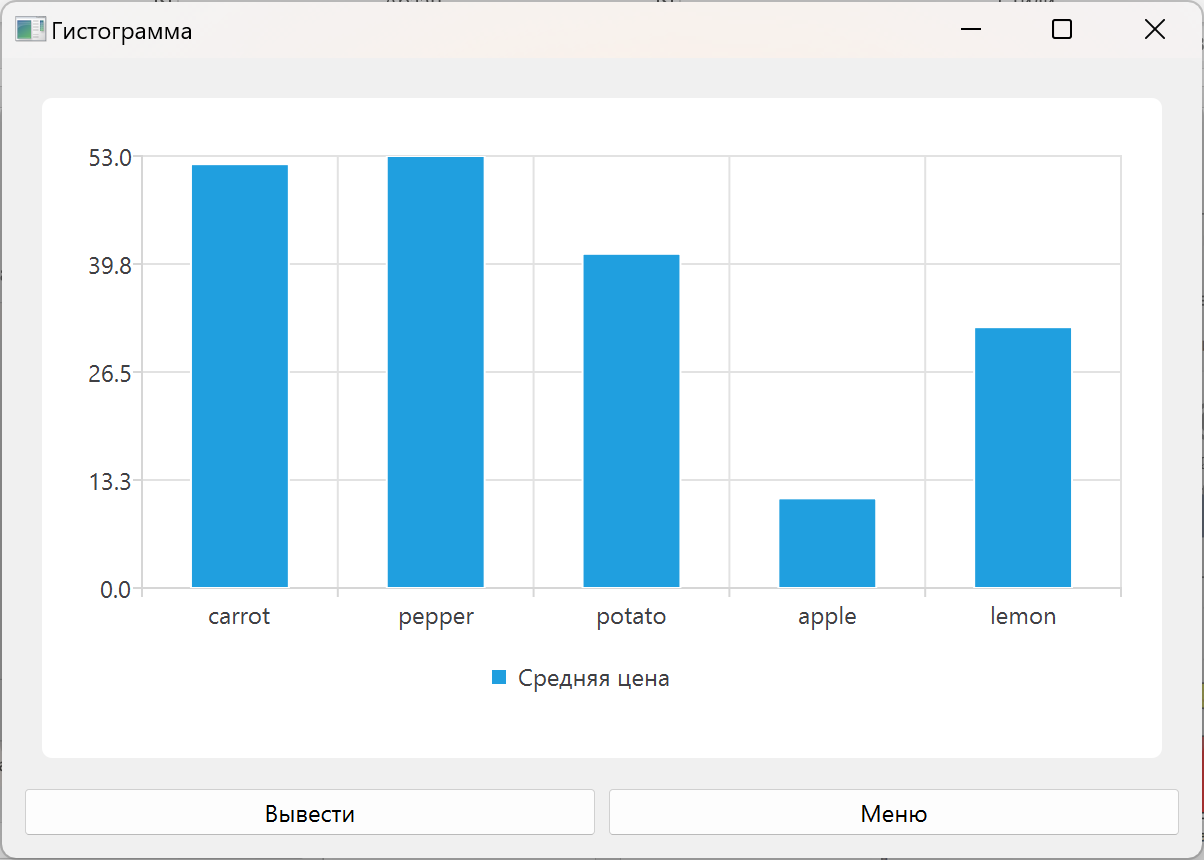


Рис.5

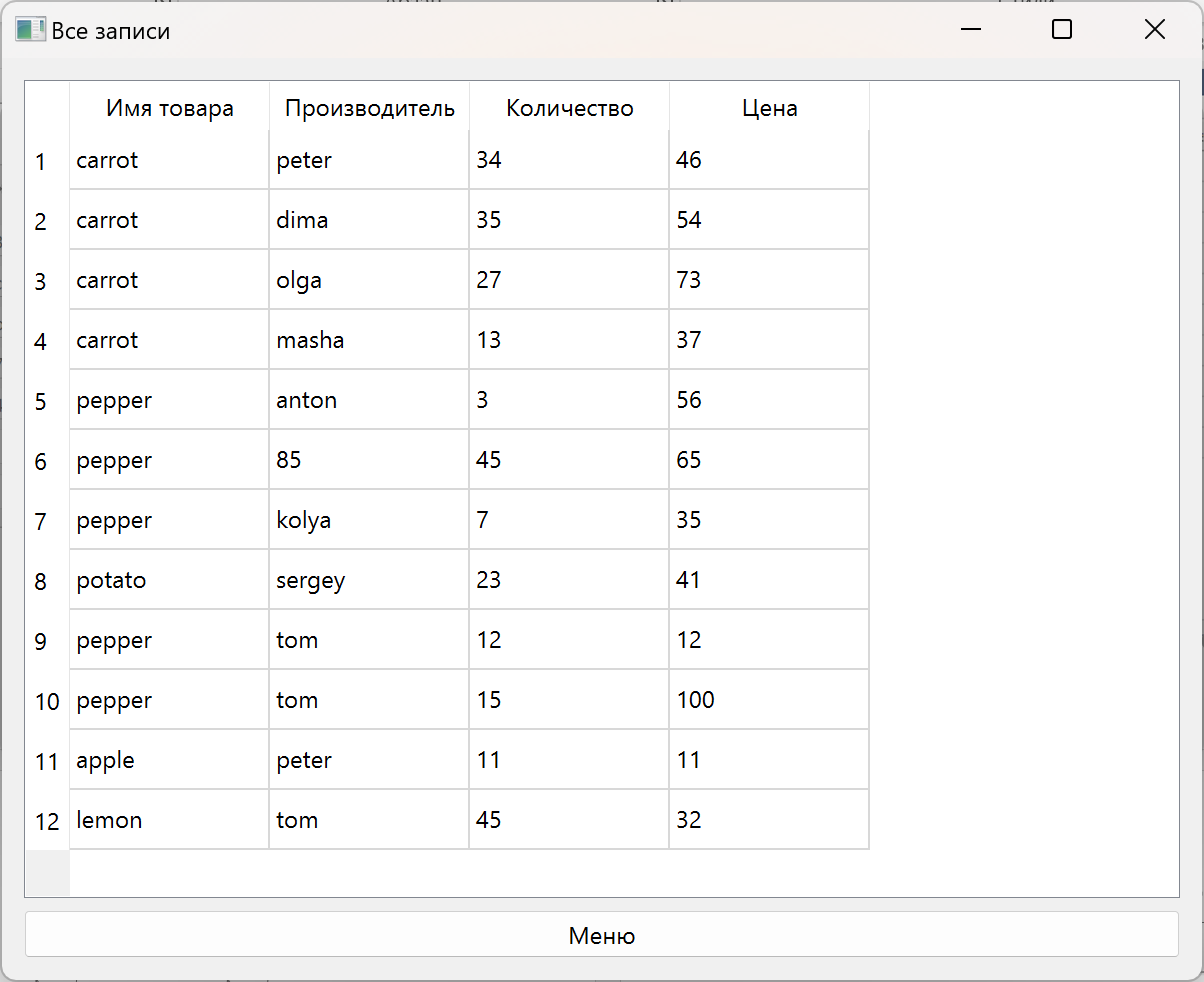


Рис.6

 Рисунок 7 – диаграмма состояний интерфейса



Рисунок 8 – объектная декомпозиция приложения



Рисунок 9 – диаграмма класса mainForm



Рисунок 10 – диаграмма класса themostForm



Рисунок 11 – диаграмма класса muchForm



Рисунок 12 – диаграмма класса BenProForm



Рисунок 13 – диаграмма класса TableForm



Рисунок 14 – диаграмма класса gystForm



Рисунок 15 – диаграмма класса baseFile



Рисунок 16 – диаграмма последовательностей действий при выполнении функции «Добавление записей»

**Вывод:** в соответствии с условием задачи было создано многооконное приложение, работающее с базой данных.