**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ УЧРЕЖДЕНИЯ ОБРАЗОВАНИЯ “БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”**

**КАФЕДРА ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

Лабораторная работа №12

По дисциплине “Основы алгоритмизации и программирования”

Тема: «Модульное программирование»

**Выполнил:**

Студент группы ИИ-23

Макаревич Н.Р.

**Проверила:**

Гирель Т.Н.

**БРЕСТ 2022**

**Вариант 2**

**Цель работы:** изучить принципы модульного программирования; ознакомиться с основными возможностями межмодульного взаимодействия.

**Задания по варианту:**

В программу, разработанную в лабораторной работе 12, внести следующие изменения и дополнения:

1. Программа должна быть разделена на несколько модулей (например, работа с файлами, работа с интерфейсом, обработка запросов к базе данных и т.п.).

2. Взаимодействие модулей организовать при помощи вызова функций и переменных внешнего типа.

**Решение задания :**

**Main.cpp**

#include <clocale>

#include "menu.h"

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

menu();

return 0;

}

**Menu.h**

#pragma once

#include "FileTXT.h"

void menu();

**menu.cpp**

#include "menu.h"

void menu()

{

short int menu;

bool choise;

int size = 0;

product\* shop = new product[size];

product\*\* index = new product \* [size];

TXTinput(shop, size, index);

do {

cout << endl << "Выберите действие: " << endl;

cout << "1) Добавить продукт через консоль" << endl;

cout << "2) Загрузить оригинальный список" << endl;

cout << "3) Загрузить отсортированный список" << endl;

cout << "4) Вывести весь список на консоль" << endl;

cout << "5) Сортировка по названию" << endl;

cout << "6) Сортировка по цене" << endl;

cout << "7) Удаление продукта" << endl;

cout << "0) Выход" << endl;

cin >> menu;

switch (menu) {

case 1: {

system("cls");

input(shop, size, index);

TXToutput(shop, size, index);

sort(size, index, choise);

sortedOutput(size, index, choise);

break;

}

case 2: {

system("cls");

TXTinput(shop, size, index);

break;

}

case 3: {

system("cls");

sortedInput(shop, size, index);

break;

}

case 4: {

system("cls");

output(size, index);

break;

}

case 5: {

system("cls");

choise = false;

sort(size, index, choise);

sortedOutput(size, index, choise);

output(size, index);

break;

}

case 6: {

system("cls");

choise = true;

sort(size, index, choise);

sortedOutput(size, index, choise);

output(size, index);

break;

}

case 7: {

system("cls");

TXTinput(shop, size, index);

remove(shop, size, index);

TXToutput(shop, size, index);

sort(size, index, choise);

sortedOutput(size, index, choise);

break;

}

case 0:

return;

}

} while (true);

}

**Files.h**

#pragma once

#include <fstream>

#include "Functions.h"

void TXTinput(product\*& shop, int& size, product\*\*& index);

void TXToutput(product shop[], int size, product\*\*& index);

void sortedOutput(int size, product\*\*& index, bool choise);

void sortedInput(product shop[], int size, product\*\*& index);

**files.cpp**

#include "FileTXT.h"

void TXTinput(product\*& shop, int& size, product\*\*& index) {

ifstream fin("outputTXT.txt");

fin >> size;

cout << "Добавлен список с " << size << " продуктами" << endl;

char tempp[1024];

fin.getline(tempp, 1024, '\n');

fin.getline(tempp, 1024, '\n');

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

fin >> shop[i].index;

fin >> shop[i].name;

fin >> shop[i].ammount;

fin >> shop[i].price;

fin >> shop[i].date;

fin >> tempp;

if (strcmp(tempp, "Рыбный") == NULL) {

shop[i].section = Рыбный;

}

else if (strcmp(tempp, "Мясной") == NULL) {

shop[i].section = Мясной;

}

else if (strcmp(tempp, "Хлебный") == NULL) {

shop[i].section = Хлебный;

}

else if (strcmp(tempp, "Растительность") == NULL) {

shop[i].section = Растительность;

}

else if (strcmp(tempp, "Напитки") == NULL) {

shop[i].section = Напитки;

}

fin >> shop[i].smade.ch;

index[i] = &shop[shop[i].index];

}

fin.close();

}

void TXToutput(product shop[], int size, product\*\*& index) {

ofstream fout("outputTXT.txt");

fout << size << " всего." << endl;

fout << setw(5) << left;

fout << "N" << setw(15) << left;

fout << "Название:" << setw(15) << left;

fout << "Кол-во:" << setw(15) << left;

fout << "Цена:" << setw(25) << left;

fout << "Дата поступления:" << setw(20) << left;

fout << "Раздел:" << setw(25) << left;

fout << "Сделано ли в Бел:" << endl;

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

fout << setw(5) << left;

fout << shop[i].index << setw(15) << left;

fout << shop[i].name << setw(15) << left;

fout << shop[i].ammount << setw(15) << left;

fout << shop[i].price << setw(25) << left;

fout << shop[i].date << setw(20) << left;

switch (shop[i].section) {

case Рыбный:

fout << "Рыбный" << setw(25) << left;

break;

case Мясной:

fout << "Мясной" << setw(25) << left;

break;

case Хлебный:

fout << "Хлебный" << setw(25) << left;

break;

case Растительность:

fout << "Растительность" << setw(25) << left;

break;

case Напитки:

fout << "Напитки" << setw(25) << left;

break;

}

fout << shop[i].smade.ch;

fout << endl;

}

fout.close();

}

void sortedOutput(int size, product\*\*& index, bool choise) {

ofstream fout("sorted.txt");

fout << size << " всего." << endl;

fout << setw(5) << left;

fout << "N" << setw(15) << left;

if (!choise) {

fout << "Название:" << setw(15) << left << endl;

}

else {

fout << "Цена:" << setw(25) << left << endl;

}

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

fout << setw(5) << left;

fout << index[i]->index << setw(15) << left;

if (!choise) {

fout << index[i]->name << setw(15) << left;

}

else {

fout << index[i]->price << setw(25) << left;

}

fout << endl;

}

fout.close();

}

void sortedInput(product shop[], int size, product\*\*& index) {

ifstream fin("sorted.txt");

fin >> size;

cout << "Добавлен список с " << size << " продуктами" << endl;

char tempp[1024];

fin.getline(tempp, 1024, '\n');

fin.getline(tempp, 1024, '\n');

int tempInt;

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

fin >> tempInt;

index[i] = &shop[tempInt];

fin >> tempp;

}

fin.close();

}

**Functions.h**

#pragma once

#include "Product.h"

#include <iostream>

#include <iomanip>

using namespace std;

void input(product\*& shop, int& size, product\*\*& index);

void output(int size, product\*\*& index);

void remove(product shop[], int& size, product\*\*& index);

void sort(int size, product\*\*& index, bool& choise);

**functions.cpp**

#include "Functions.h"

void input(product\*& shop, int& size, product\*\*& index)

{

int newSize;

int choiseForSection;

cout << "Сколько продуктов нужно добавить?" << endl;

cin >> newSize;

size += newSize;

product\* temp = new product[size];

product\*\* tempi = new product \* [size];

for (int i = 0; i < size - newSize; i++) {

temp[i] = shop[i];

tempi[i] = index[i];

}

shop = new product[size];

index = new product \* [size];

for (int i = 0; i < size - newSize; i++) {

shop[i] = temp[i];

index[i] = tempi[i];

}

delete[]temp;

delete[]tempi;

for (int i = size - newSize; i < size; i++) {

shop[i].index = i;

cout << "Введите название продукта " << endl;

cin >> shop[i].name;

cout << "Введите кол-во продукта" << endl;

cin >> shop[i].ammount;

cout << "Введите цену продукта" << endl;

cin >> shop[i].price;

cout << "Введите дату поступления продукта" << endl;

cin >> shop[i].date;

cout << "Введите номер раздела продукта (1 - Рыбный, 2 - Мясной, 3 - Хлебный, 4 - Растительность, 5 - Напитки)" << endl;

cin >> choiseForSection;

shop[i].section = (Section)choiseForSection;

cout << "Произведен ли продукт в Беларуси? (1 - да, 0 - нет)" << endl;

cin >> shop[i].smade.itIs;

if (shop[i].smade.itIs)

shop[i].smade.ch = '+';

else shop[i].smade.ch = '-';

index[i] = &shop[i];

system("cls");

}

}

void output(int size, product\*\*& index) {

cout << setw(5) << left;

cout << "N" << setw(15) << left;

cout << "Название:" << setw(15) << left;

cout << "Кол-во:" << setw(15) << left;

cout << "Цена:" << setw(25) << left;

cout << "Дата поступления:" << setw(20) << left;

cout << "Раздел:" << setw(25) << left;

cout << "Сделано ли в Бел:" << endl;

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

cout << setw(5) << left;

cout << index[i]->index << setw(15) << left;

cout << index[i]->name << setw(15) << left;

cout << index[i]->ammount << setw(15) << left;

cout << index[i]->price << setw(25) << left;

cout << index[i]->date << setw(20) << left;

switch (index[i]->section) {

case Рыбный:

cout << "Рыбный" << setw(25) << left;

break;

case Мясной:

cout << "Мясной" << setw(25) << left;

break;

case Хлебный:

cout << "Хлебный" << setw(25) << left;

break;

case Растительность:

cout << "Растительность" << setw(25) << left;

break;

case Напитки:

cout << "Напитки" << setw(25) << left;

break;

}

cout << index[i]->smade.ch;

cout << endl;

}

}

void sort(int size, product\*\*& index, bool& choise) {

if (choise) {

for (int i = 0; i < size - 1; i++) {

for (int j = i; j < size; j++) {

if (index[i]->price < index[j]->price) {

swap(index[i], index[j]);

}

}

}

}

else {

for (int i = 0; i < size - 1; i++) {

for (int j = i; j < size; j++) {

if (strcmp(index[i]->name, index[j]->name) > 0) {

swap(index[i], index[j]);

}

}

}

}

}

void remove(product shop[], int& size, product\*\*& index) {

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

cout << i << ": ";

cout << "Название продукта: " << index[i]->name << endl;

}

cout << "Введите номер продукта который хотите удалить: ";

int num, inum;

cin >> num;

inum = shop[num].index;

for (int i = num; i < size - 1; i++) {

shop[i] = shop[i + 1];

index[i] = &shop[i];

if (shop[i].index > inum) {

shop[i].index--;

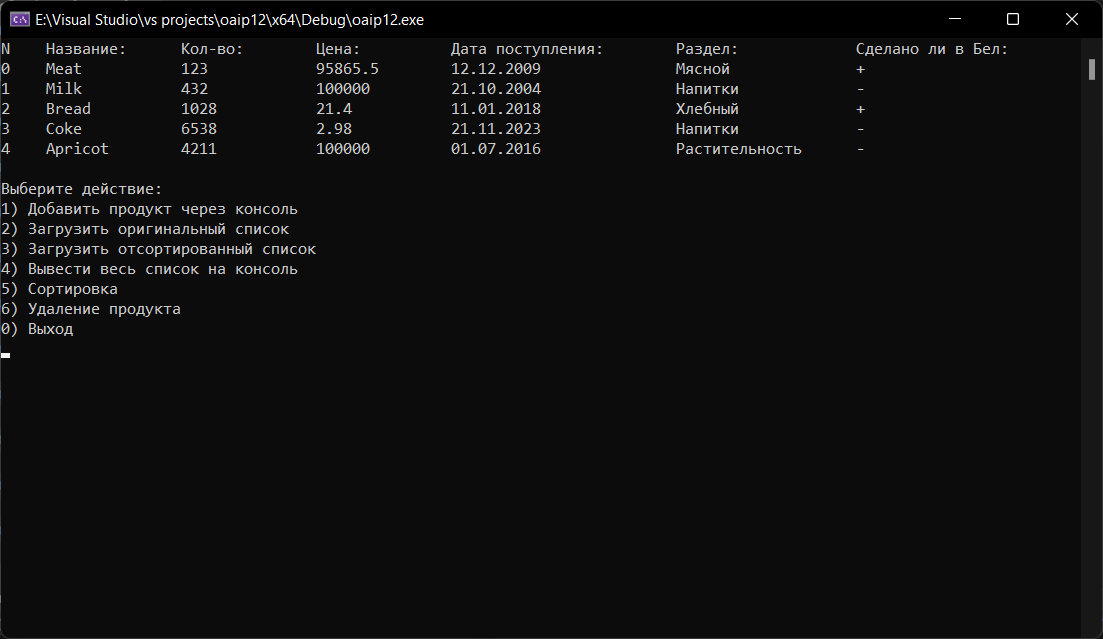
}

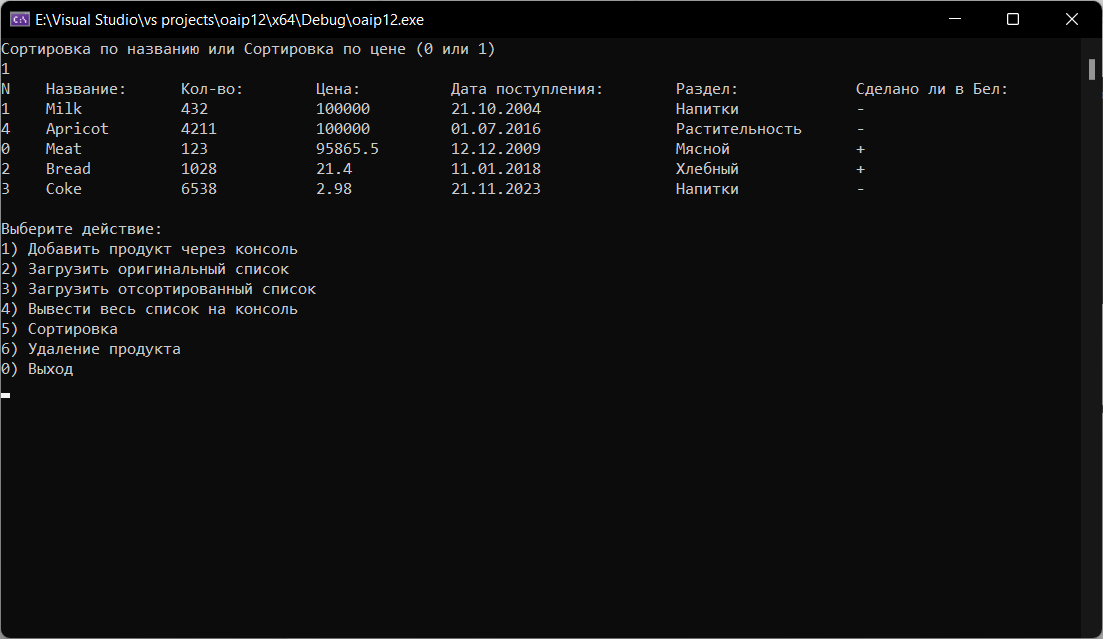
}

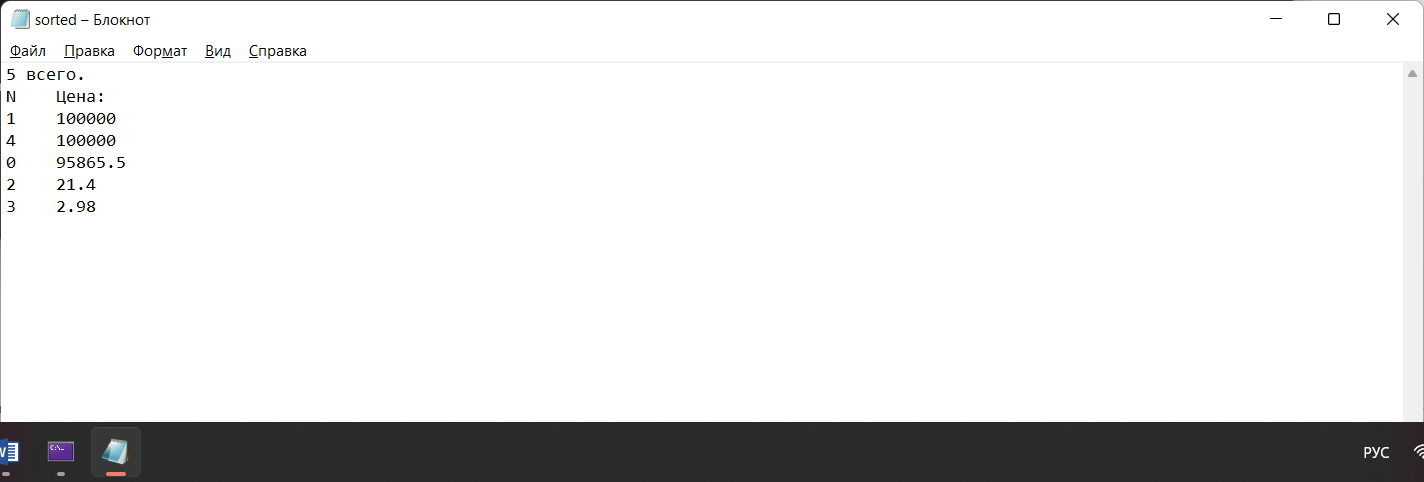
size--;

}

**Результат:**







**Вывод:**  изучил принципы модульного программирования и ознакомилcя с основными возможностями межмодульного взаимодействия.