

«Национальный открытый институт»

Сибирев В.Н., Рачева Н.В.

**РАЗРАБОТКА, ВНЕДРЕНИЕ И АДАПТАЦИЯ ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ ОТРАСЛЕВОЙ НАПРАВЛЕННОСТИ**

**Методические указания
к выполнению курсовой работы**

Рекомендовано Методической комиссией по качеству
Национального открытого института
для студентов, обучающихся по направлению
09.03.03 «Прикладная информатика»

Санкт-Петербург
2016

УДК 519.2.06 (07)
ББК 32.973
С34

Методические указания к выполнению курсовой работы. разработаны на основе рабочей программы “Прикладная информатика (по отраслям)” в соответствии с требованиями государственных образовательных стандартов среднего профессионального образования.

Курсовая работа выполняется студентами самостоятельно по индивидуальному варианту. Методические указания предназначены для студентов направления подготовки – 09.03.03 «Прикладная информатика» и профиля подготовки – «Прикладная информатика в экономике».

УДК 519.2.06 (07)
ББК 32.973

© Сибирев В.Н. 2016
© Рачева Н.В. 2016
© «Национальный открытый институт» 2016
© ИКЦ 2016

1. Введение

Курсовая работа является завершающим этапом подготовки студентов по дисциплине «Разработка, внедрение и адаптация программного обеспечения отраслевой направленности». Предполагается, что основы программирования были изучены в предыдущих дисциплинах: «Информатика и программирование», «Высокоуровневые методы информатики и программирования». Для успешного самостоятельного выполнения индивидуальных заданий приводится простейший пример оформления курсовой работы с проверенными исходными текстами программных модулей на языке C# в инструментальной среде Visual Studio.

Целью курсовой работы является освоение задач по работе с внешней памятью – сохранение, корректировка, выборка разнотипной информации, размещенной в файлах последовательной организации. Базовой подготовкой студентов для работы над курсовой работой являются практические занятия, где происходит знакомство с отдельными конструкциями интегрированной среды проектирования программного комплекса.

2. Задания на выполнение индивидуальных работ

Требования к оформлению курсовой работы следующие: на титульном листе необходимо написать свою фамилию, имя и отчество, шифр, название дисциплины и фамилию преподавателя. Курсовая работа заключается в составлении технического задания (ТЗ) на разработку ПК по конкретной теме. ТЗ на заказ по изготовлению ПК должен обязательно содержать следующее:

1. Четкую формулировку конечной цели разработки ПК (указать объект и условия применения ПК).
2. Весь перечень входных документов с указанием объема (количества) информации;
3. Весь перечень выходных форм (документов) результатов обработки информации с указанием устройств вывода информации (экран, печать, CD и другие формы сохранения данных);
4. Взаимное согласование сроков и условий выполнения заказа, возможно с разбивкой по этапам во времени.

5. Расчет затрат (ресурсов) на ПК с указанием мер при невыполнении тех или иных условий как заказчиком, так и исполнителем.

В случае, если нет условий выполнить обоснование пунктов 4,5 возможно их отсутствие в формулировке ТЗ.

Перечень индивидуальных заданий по вариантам

1. Предметная область «Гараж»

«Транспорт»

Наименование единицы	Гос. номер	Марка	Год выпуска	Первоначальная стоимость
----------------------	------------	-------	-------------	--------------------------

2. Предметная область «Выставка»

«Зал»

№ зала	Площадь кв.м	ФИО ответственного дежурного	Этаж	Статус помещения
--------	--------------	------------------------------	------	------------------

3. Предметная область «Спорт»

«Тренер»

ФИО тренера	№ группы	№ договора	Дата закрепления	Длительность занятий
-------------	----------	------------	------------------	----------------------

4. Предметная область «Кинопрокат»

«Фильм»

Название фильма	Наименование киностудии	Жанр фильма	Год выпуска	Первоначальная стоимость
-----------------	-------------------------	-------------	-------------	--------------------------

5. Предметная область «Учебная точка»

«Аудитория»

№ аудитории	Площадь кв. м	Категория аудитории	Количество мест	Дата ремонта
-------------	---------------	---------------------	-----------------	--------------

6. Предметная область «Командировки»

«Преподаватель»

ФИО преподавателя	Кафедра	Моб. телефон	Должность	Дата приезда
-------------------	---------	--------------	-----------	--------------

7. Предметная область «Библиотека»

«Книга»

Название книги	ФИО автора	Цена первоначальная	Тип жанра	Год приобретения
----------------	------------	---------------------	-----------	------------------

8. Предметная область «Закрепление дисциплин»

«Кафедра»

Наименование кафедры	Код кафедры	Аббревиатура наименования	Дата образования	№ аудитории
----------------------	-------------	---------------------------	------------------	-------------

9. Предметная область «Склад»

«Товар»

Название товара	Код товара	Единицы измерения	Год выпуска	Первоначальная стоимость
-----------------	------------	-------------------	-------------	--------------------------

10. Предметная область «Туристская путевка»

«Тур. агент»

ФИО тур. агента	№ страхового свидетельства	№ ИНН	№ паспорта	Срок визы
-----------------	----------------------------	-------	------------	-----------

3. Образец технического задания на проектирование ПК

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

на разработку программного комплекса
для АРМа специалиста УЧЕБНОГО ОТДЕЛА
" КОНТИНГЕНТ "

1. ВВЕДЕНИЕ

Необходимость в оперативной информации по текущему состоянию контингента студентов требует наличия надежного источника данных и простого, быстрого доступа к нужным сведениям. Электронный вариант хранения информации позволяет решать подобные задачи на базе программных комплексов различной сложности.

2. ОСНОВАНИЕ ДЛЯ РАЗРАБОТКИ

Разработка программного комплекса производится на основании решения совета ВУЗа № ... от ... , утвержденного ректором

3. НАЗНАЧЕНИЕ РАЗРАБОТКИ

Программный комплекс для АРМа специалиста учебного отдела "КОНТИНГЕНТ" предназначен для автоматизированного учета контингента студентов, включая типовые операции по подготовке и сопровождению документации учебного отдела ВУЗа.

4. ТРЕБОВАНИЯ К ПРОГРАММНОМУ КОМПЛЕКСУ

4.1 Функциональные характеристики.

Программный комплекс должен обеспечить выполнение следующих функций:

а) первоначальное занесение сведений о студентах на основании приказов учебного отдела следующих типов:

- о зачислении;
- о переводе;
- о выпуске;
- о предоставлении академического отпуска;
- об изменении анкетных данных;

(необходимо наличие полного перечня приказов, на основании которых происходит перемещение студента по курсам, группам и т.д.)

б) корректировка сведений о студентах в основной базе данных;

(реквизиты полей базы данных, характеризующих сведения о студентах, с указанием их форматов и диапазоном возможных значений каждого поля);

в) формирование вспомогательных массивов данных

(например, кодификаторы специальностей, нормативные документы и т.п.)

г) формирование и печать форм отчетных документов:

д) выдача справок по форме ... (образцы всех видов запросов и выдаваемых на экран и печать ответов приводятся в приложении)

е) незапланированные запросы по произвольному сочетанию полей и их значений и т.д.

Важным для этого раздела является корректное описание функций в виде их внешних спецификаций - точное описание структуры входных и выходных образцов документов.

4.2 Надежность:

а) программный комплекс не должен содержать ошибок, искажающих внешние спецификации, обнаруживаемые тестированием;

б) программный комплекс не должен переходить в неопределенное состояние при неправильных действиях пользователя за клавиатурой;

в) программный комплекс не должен переходить в неопределенное состояние при недопустимых форматах, значениях или сочетании значений данных, вводимых с

клавиатуры (форматы данных и диапазоны их значений, приведенных в п. 4.1);

г) программный комплекс не должен зависеть или выдавать несоответствующие сообщения при обращении к неподключенным внешним устройствам;

д) выполнение функций 1 и 2 пункта 4.1 должно защищаться паролем, остальные функции не защищаются;

е) программный комплекс должен быть защищен от вирусов;

4.3 Условия эксплуатации:

а) программный комплекс должен быть рассчитан на пользователя, незнакомого с особенностями программирования и управления операционной системой;

б) дополнительные знания, необходимые для работы с программным комплексом, должны быть ограничены ознакомлением с инструкцией;

в) количество манипуляций за экраном должно быть минимальным, интерфейс пользователя следует строить на основе выбора действий из внятного меню;

г) количество информации, которое необходимо воспринять и переработать пользователю для принятия решений по управлению программным комплексом, должно быть минимальным;

д) программный комплекс должен реагировать на недопустимые действия пользователя за экраном или на неразрешенное сочетание данных, либо игнорируя их, либо выдавая звуковое или световое предупреждение;

е) переход между сопряженными по логике работы меню должен быть максимально коротким;

ж) изображение доступности меню пользователя на экране должно быть постоянным и контекстно-зависимым;

з) должно быть обеспечено оперативное получение контекстно-зависимой инструкции пользователя из любой позиции диалога с программным комплексом (служба HELP);

и) время реакции программного комплекса:

- при операциях типа редактирования не более 0.3 сек;

- время перехода на другую функцию не более 0.3 сек;

- время поиска одиночного объекта не более 10 сек;

- формирование групповых справок не более 1 мин;

- время печати документов определяется параметрами

принтера,

к) эстетическое оформление пользовательского интерфейса должно вызывать положительную реакцию пользователя, следует придерживаться следующих принципов:

- цветовая гамма должна быть сдержанной;
- цвета должны нести функциональную нагрузку и не менять ее на протяжении всего диалога;
- сигнальные сообщения системы, связанные с невозможностью или опасностью выполнения действия, окрашивать в красный цвет;
- для заполнения общего экранного фона избегать цветов GREEN (зеленый) и MAGENTA (розовый);
- текст должен легко читаться на фоне меню, избегать печатать синим или коричневым по черному, зеленым по розовому и т.п.

4.4 Технические средства.

Программный комплекс устанавливается на персональном компьютере в операционной среде версий Windows XP, Windows 7 и выше.

5. ДОКУМЕНТАЦИЯ

- инструкция пользователя в электронном виде (служба HELP) и в твердой копии;
- инструкция по сопровождению системы.

6. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

Работа производится в соответствии с календарным планом.

7. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ

Программный комплекс предъявляется исполнителем в виде исходных и загрузочных модулей (возможен вариант и только загрузочных).

Тестирование программного комплекса на различных этапах производится на предмет соответствия реально выполняемых функций внешним спецификациям, приведенным в п. 4.1.

На заключительном этапе осуществляется проверка выполнения всех требований к системе по пунктам 4.1-4.4 и 5. Тестирование программного комплекса выполняется на примерах, подготовленных исполнителем совместно с заказчиком. Форматы и диапазоны данных должны удовлетворять пункту 4.1.

8. ДОПОЛНИТЕЛЬНЫЕ УСЛОВИЯ

В случае изменения состава и форматов документов и логики их обработки по согласованию сторон могут быть изменены сроки выполнения работы и объемы финансирования; изменения оформляются протоколом.

ТЗ составлено: ФИО автора (ов).

4. Образец оформления курсовой работы

Задача - организовать автоматизированную обработку каталога книг ограниченной информации по каждому объекту.

Цель разработки – повысить эффективность работы с библиотечным каталогом.

Функциональные характеристики ПК:

- создать файл следующей структуры (ограниченная информация объекта): название, автор, стоимость печатного издания;

- организовать просмотр содержимого файла;

- организовать добавление записи в созданный файл;

- спроектировать пользовательский интерфейс по форме меню средствами визуальных компонентов.

4. Протестировать по всем заданным функциям.

5. Указать условия эксплуатации.

Меню управления программным комплексом

Программный комплекс предлагается реализовать на языке С# (возможен и другой язык по согласованию с преподавателем).

Пользовательское меню представлено ниже. Представлен вариант возможного управления программным комплексом. Запуск на выполнение отдельных функций организован через шесть командных кнопок (CommandButton).

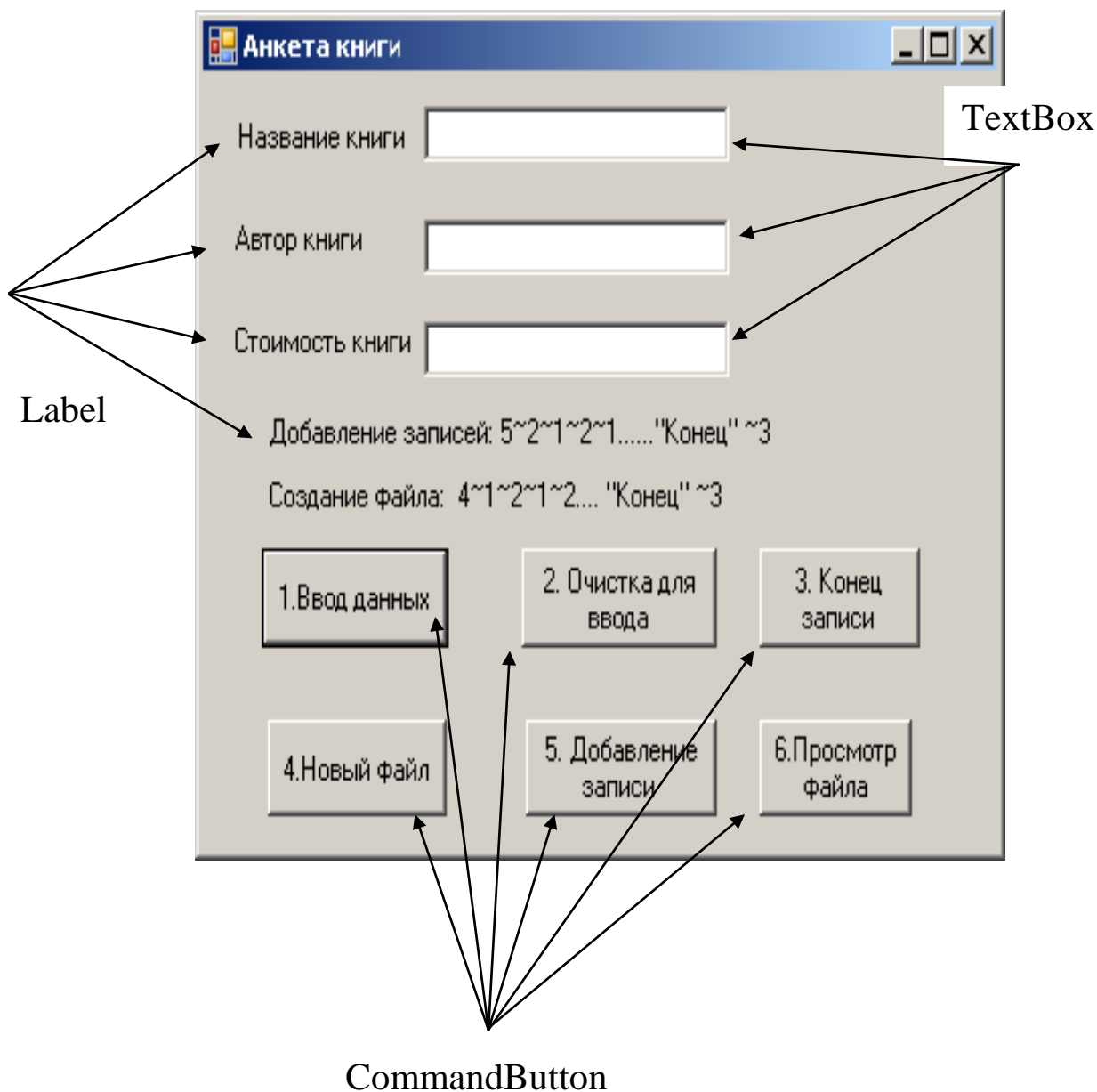
Для создания нового файла следует выполнить следующую последовательность операций:

- создать новый файл (CommandButton4);

- выполнить ввод данных (CommandButton1);

- выполнить очистку полей для ввода данных (CommandButton2);
- по окончании ввода данных на запись необходимо закрыть файл (CommandButton 3).

Пример формы пользовательского интерфейса



Здесь приведен интерфейс с указанием внутренних имен визуальных компонентов. Для ввода/вывода текста в одну строку используется компоненты TextBox. Для расположения поясняющих надписей на экране используется компоненты Label со свойствами Caption, Font. Для запуска действий использован

компоненты `CommandButton`. Используемые свойства – `Caption`, `Font`.

Для добавления записей в существующий файл следует выполнить следующую последовательность операций:

- открыть файл на добавление записей (`CommandButton5`);
- выполнить очистку полей для ввода данных (`CommandButton2`);
- выполнить ввод данных (`CommandButton1`);
- по окончании ввода данных на добавление записей закрыть файл (`CommandButton 3`).

Исходные тексты модулей программного комплекса

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;

// System.IO – устанавливает автор ПК

namespace WindowsFormsApplication5
{
    struct Lib
    {
        public
            Lib(string fname, string favtor, decimal fchena)
            {
                name = fname;
                avtor = favtor;
                chena = fchena;
            }
        public string name;
        public string avtor;
        public decimal chena;
    }
}
```

```
}
```

```
public partial class Form1 : Form
```

```
{
```

```
public BinaryWriter fout;
```

```
// оператор - комментарий «инициализация формы»
```

```
public Form1()
```

```
{
```

```
    InitializeComponent();
```

```
}
```

```
private void button1_Click(object sender, EventArgs e)
```

```
{
```

```
string str;
```

```
decimal k;
```

```
    Lib z = new Lib();
```

```
// проверка на пустое поле
```

```
if(textBox1.Text == "")
```

```
{
```

```
    MessageBox.Show(" Введите название книги");
```

```
return;
```

```
}
```

```
    z.name = textBox1.Text;
```

```
if (textBox2.Text == "")
```

```
{
```

```
    MessageBox.Show(" Введите ФИО автора");
```

```
return;
```

```
}
```

```
    z.avtor = textBox2.Text;
```

```
if (textBox3.Text == "")
```

```
{
```

```
    MessageBox.Show(" Введите стоимость книги");
```

```
return;
```

```
}
```

```
    str = textBox3.Text;
```

```
    k = Convert.ToDecimal(str);
```

```

z.chena=k;
// перенос данных во внешнюю память
try
{
    fout.Write(z.name);
    fout.Write(z.avtor);
    fout.Write(z.chena);
}
catch (IOException exc)
{
Console.WriteLine(exc.Message);
}
    MessageBox.Show(" Запись в файл выполнена");

}

// ===== очистка полей формы
private void button2_Click_1(object sender, EventArgs e)
{
    textBox2.Text = "";
    textBox1.Text = "";
    textBox3.Text = "";
    MessageBox.Show(" Выполнена очистка формы");
return;
}

// ===== создание файла запись
private void button3_Click(object sender, EventArgs e)
{
try
{
    fout = new BinaryWriter
(new FileStream("DDD.txt", FileMode.Create));
}
catch (IOException exc)
{
    Console.WriteLine(exc.Message);
    return;
}
    MessageBox.Show(" Файл открыт для записи");

```

```

    }
    //===== открытие файла на добавление записей
private void button4_Click(object sender, EventArgs e)
    {
        try
        {
            fout = new BinaryWriter
                (new FileStream("DDD.txt", FileMode.Append));
        }
catch (IOException exc)
        {
            Console.WriteLine(exc.Message);
return;
        }
        MessageBox.Show(" Файл открыт для добавления записей");
    }
    //===== закрытие файла
private void button5_Click(object sender, EventArgs e)
    {
        fout.Close();
    }
    // Просмотр записей содержимого файла через форму
private void button6_Click(object sender, EventArgs e)
    {
        Lib r = new Lib();
        string str;
int k = 0;
        BinaryReader f_in;
        try
        {
            f_in = new BinaryReader
                (new FileStream("DDD.txt", FileMode.Open));
        }
catch (IOException exc)
        {
            Console.WriteLine(exc.Message);
return;
        }
try

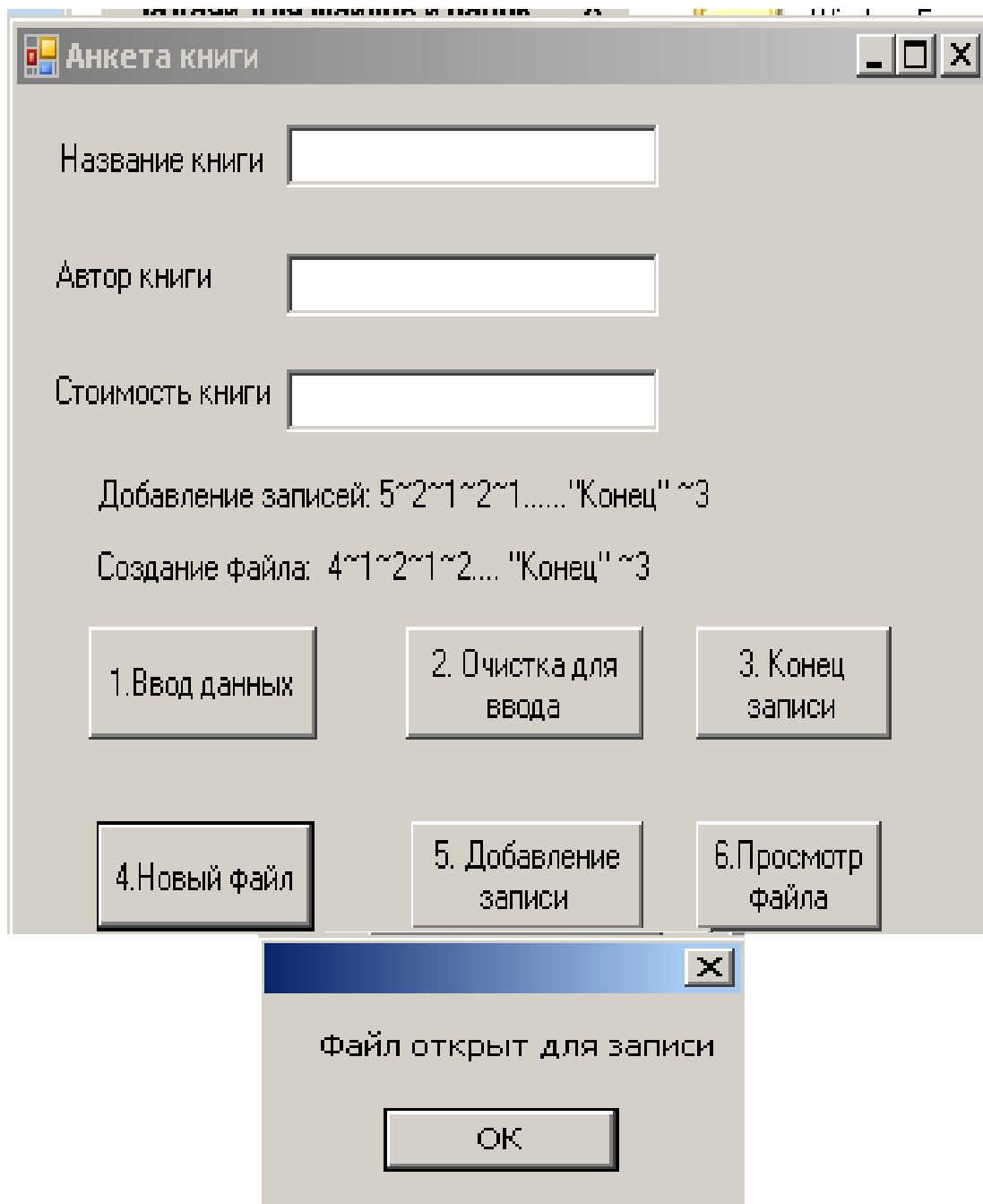
```

```

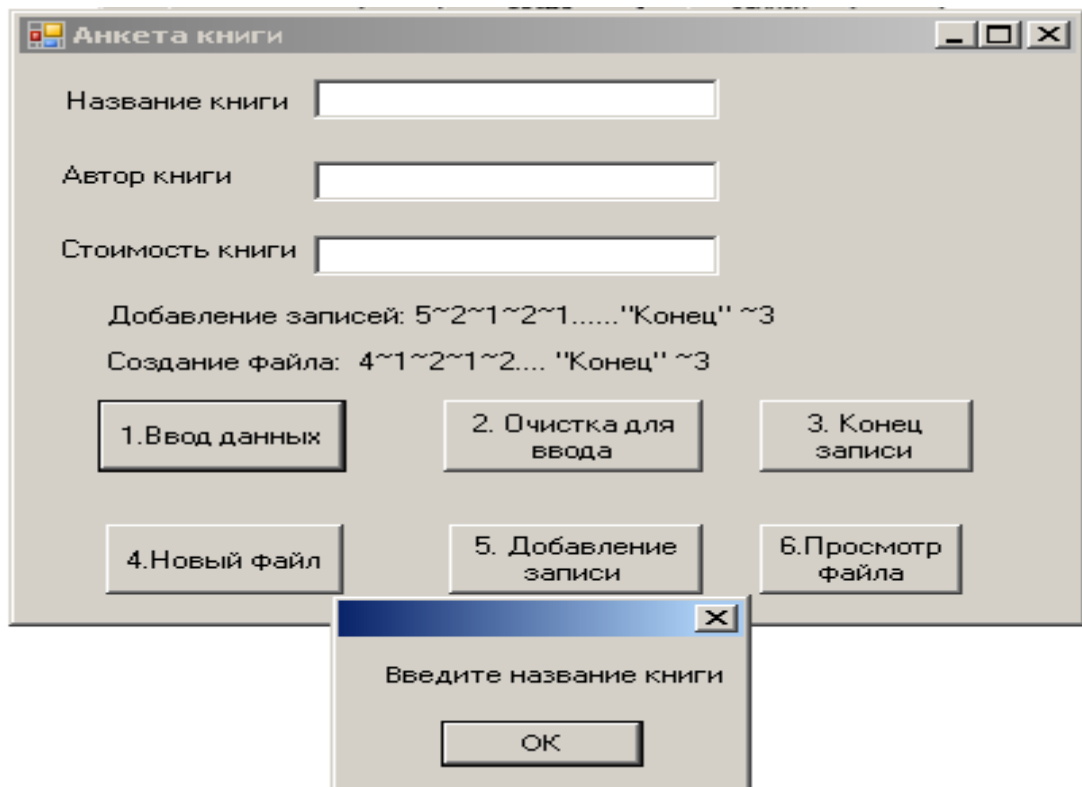
    {
        while ((r.name = f_in.ReadString()) != null)
        {
            r.avtor = f_in.ReadString();
r.chena = f_in.ReadDecimal();
textBox1.Text = r.name;
            textBox2.Text = r.avtor;
            str = Convert.ToString(r.chena);
            textBox3.Text = str;
// MessageBox.Show(r.name); - промежуточный вывод
            // MessageBox.Show(r.avtor);
            // MessageBox.Show(" s="+r.chena);
            k++;
            MessageBox.Show(" Запись №=" + k);
        }
    }
}
catch (IOException exc)
{
    Console.WriteLine(exc.Message);
}
f_in.Close();
MessageBox.Show(" Всего в файле К=" + k + " записей");
}
}
}

```

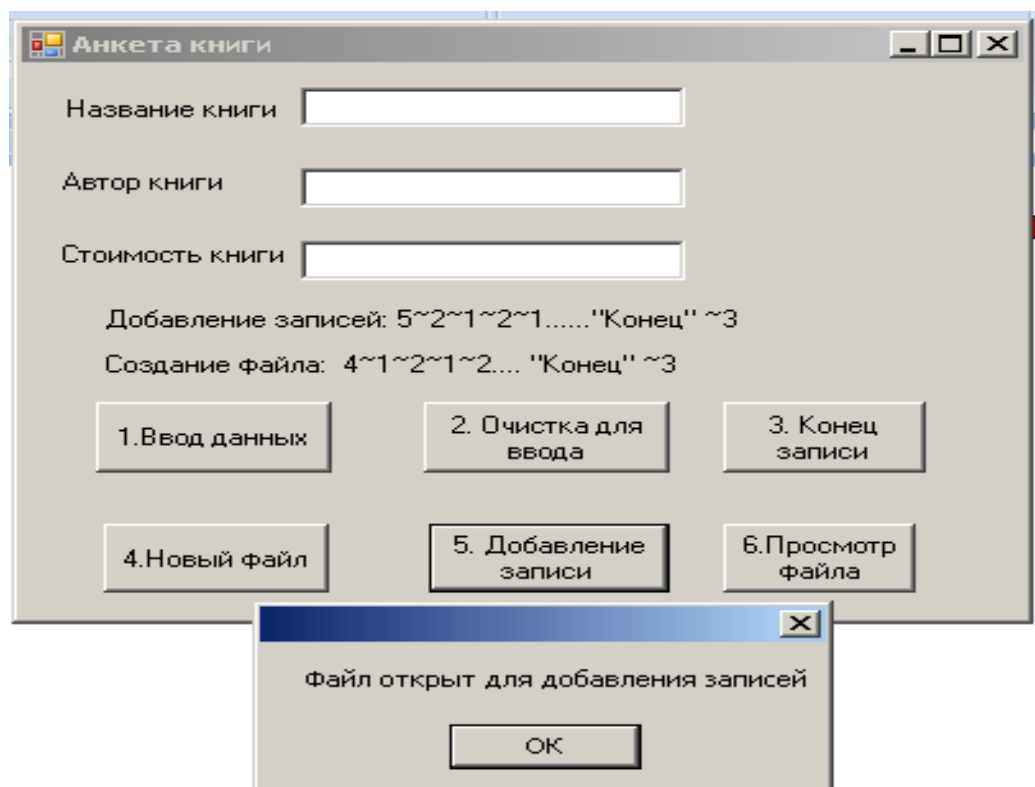
Создание файла



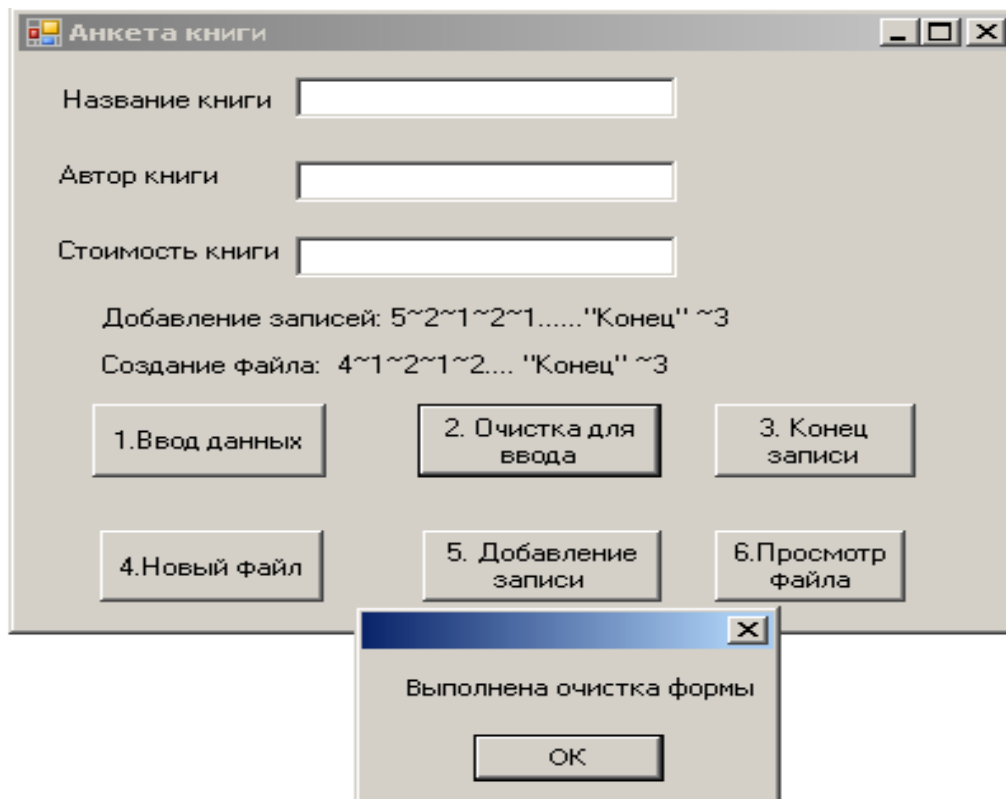
Ввод данных с клавиатуры в оперативную память через форму



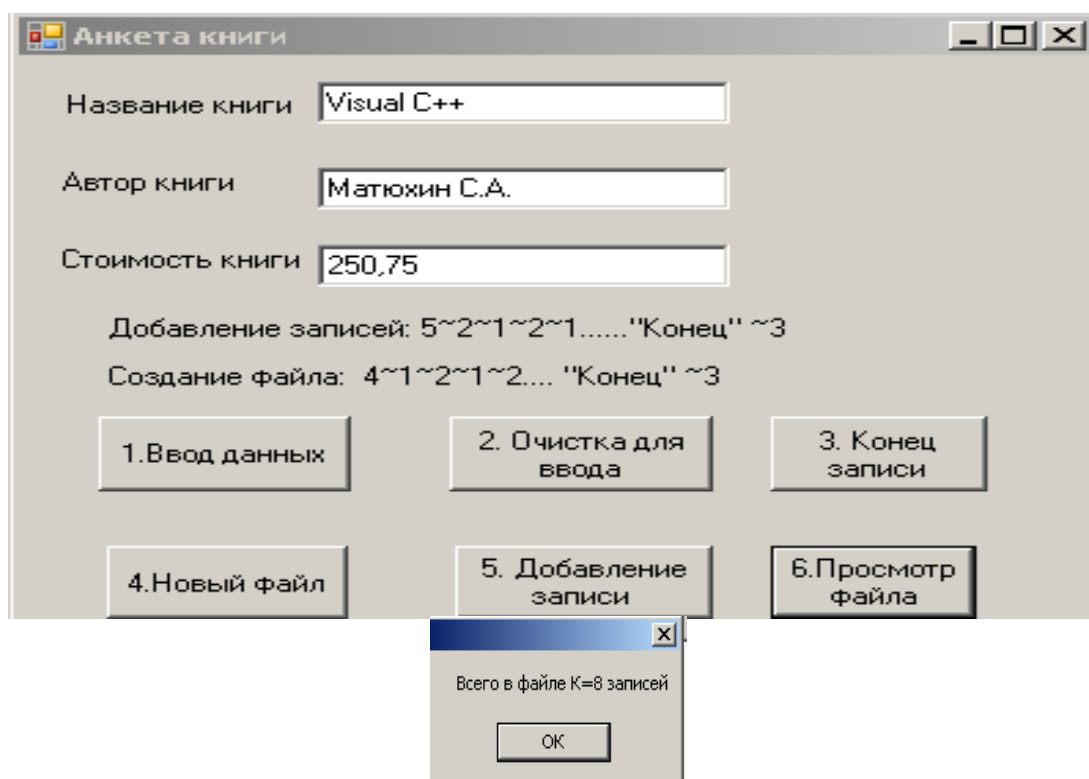
Открытие файла для добавления записи



Выполнение очистки для ввода.



Просмотр содержимого файла



Выводы

Тестирование выполнено на примере порядка 12 записей по заданным функциям, описанным в ТЗ.

Представленный ПК является простейшим примером работы с внешней памятью в интегрированной среде **Visual Studio** на языке **C#** и полностью удовлетворяет техническому заданию. Реализованы все заданные функции. Предложен интуитивно понятный интерфейс в пределах ТЗ.

Создана достаточно удобная структура классов с возможностью дальнейшего развития и добавления функций, доступных пользователю.

Данный проект ПК не исчерпал своих возможностей и при наличии временных ресурсов может быть доведен до уровня коммерческого приложения и использоваться во всех библиотечных структурах.

Информационные источники

1. Фленов, Ф.Е. Библия C#. / Ф.Е. Фленов. – СПб.: БХВ-Петербург, 2009. – 560 с.

2. Шилдт, Герберт. C# 3.0: руководство для начинающих.: пер. с англ. / Герберт Шилдт. – М.: ООО «И.Д. Вильямс», 2009. – 668 с.

3. Программирование: Учебное пособие / Сост.: Е.О.Шумова. Национальный минерально-сырьевой университет «Горный». - СПб, 2013, 87 с. (Электронный вариант)

4. Объектно-ориентированное программирование: Учебное пособие / Сост.: Е.О.Шумова. Национальный минерально-сырьевой университет «Горный». - СПб., 2013, 124 с. (Электронный вариант)

Заключение

В методических указаниях к выполнению курсовой работы рассмотрен конкретный пример программного комплекса, доступный для понимания с начальным знакомством среды **Visual Studio** и языка программирования **C#**.

Чтобы изучить принципы работы с искусственными языками, проектированием программных комплексов в современных условиях быстро развивающихся программных

средств, лучший вариант погружения в программную среду - начать с простейших *примеров-образцов*. Классики науки программирования, в частности, Д. Грис сказал: не спеши создавать «**BIGPROGRAM**», освой «**LITTLEPROGRAM**».

СОДЕРЖАНИЕ

1. Введение	3
2.Задания на выполнение индивидуальных работ.....	4
3.Образец технического задания на проектирование ПК.....	5
4.Образец оформления курсовой работы	9
5.Заключение.....	20