Министерство образования и науки РФ

Государственное образовательное учреждение

Высшего образования

«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых» (ВлГУ)

Кафедра «Физика и прикладная математика» (ВлГУ)

Курсовая работа

По дисциплине «Прикладное программирование»

«Программный модуль для учёта книжного фонда в библиотеке»

Выполнил:

Ст. ПКсп-115

Приняла: Ст. преподаватель  
Шишкина М. В.

Владимир 2017 г.

**Содержание.**

[**Введение.** 3](#_Toc447841424)

[**Глава1. Постановка задачи** 4](#_Toc447841425)

[**Глава 2. Выбор системы программирования** 5](#_Toc447841426)

[**Глава 3. Описание процесса проектирования приложения** 6](#_Toc447841427)

[**3.1. Взаимосвязь программных модулей** 6](#_Toc447841428)

[**3.1.1. Функция добавления новой книги.** 6](#_Toc447841429)

[**3.1.2. Удаление книги из списка.** 7](#_Toc447841430)

[**3.1.3. Функция меняющие параметры в списке, если книгу забрали.** 7](#_Toc447841431)

[**3.1.4. Функция меняющие параметры в списке, если книгу вернули.** 8](#_Toc447841432)

[**3.1.5. Функция поиска на наличие книги.** 9](#_Toc447841433)

[**3.2. Схема работы программы** 10](#_Toc447841434)

[**3.3. Проектирование интерфейса.** 10](#_Toc447841436)

[**Глава 4. Инструкция для пользователя по работе с приложением** 12](#_Toc447841437)

[**Заключение.** 13](#_Toc447841438)

[**Список использованных информационных источников:** 14](#_Toc447841439)

[**Приложение 1** 15](#_Toc447841440)

# **Введение.**

В наш бурный век, когда все сферы жизни постоянно усовершенствуются и развиваются, трудно найти дом, офис или предприятие, в которых не было бы компьютера. Трудно найти и человека, даже среди детей дошкольного возраста, который не имел бы самых простых навыков работы с этой умной машиной. То, что еще несколько десятков лет назад казалось чем-то непонятным, неизвестным, сложным и почти фантастическим, теперь стало обычным и доступным даже ребенку. Многие уже не представляют своей работы, учебы и повседневной жизни без компьютера. Одна из самых утомляющих работ, которую может облегчить компьютер – ручное заполнение тысяч одинаковых форм, содержащих, к примеру, информацию о студентах. В большинстве государственных и частных учреждений, например, в библиотеке учет книг ведется с помощью ПО. В данной работе я рассмотрю все этапы создания программного модуля “Картотека для учёта литературы в библиотеке”.

# **Глава1. Постановка задачи**

В теме моей курсовой работы требуется создать приложение “Картотека для учёта литературы в библиотеке”, которое бы использовалось в библиотеках и облегчало учет книг. Приложение должно иметь возможность добавлять новые книги, удалять книги из списка, следить находится ли книга на данный момент в библиотеке, если нет, то показывать у кого она находится.

# **Глава 2. Выбор системы программирования**

Для выполнения поставленной задачи, в качестве системы программирования, я выбрал C#. Потому что, С# полностью обьектно- ориентирован и поддерживает абсолютно все классы и пространства имен платформы .NET Framework. С# отлично подходит для прикладного программирования под платформу Windows.

# **Глава 3. Описание процесса проектирования приложения**

# **3.1. Взаимосвязь программных модулей**

Итак, первым делом необходимо определиться с функционалом, которым будет обладать модуль. Во-первых, мы должны иметь возможность добавлять книги в список. Во-вторых, мы должны, иметь доступ к этому списку. В-третьих, в этом списке должны быть параметры, указывающие на наличие определенной книги в библиотеке и возможность этот параметр редактировать. Информация о книге содержит: Название книги, Автор, Наличие в библиотеке, если на руках, то у кого и дату выдачи.

Форма была разделена на 2 вкладки:

* Работа с книжным фондом.
* Информация о наличии книг.

# **3.1.1. Функция добавления новой книги.**

После нажатия кнопки управление передается обработчику события “btnAdd\_Click”. Программа создает новый элемент в listView1, далее выдается сообщение о том, что данные добавлены, и программа переходит в функцию записи элементов listView1 в файл, что переписать изменения. Ниже представлен фрагмент кода:

private void btnAdd\_Click(object sender, EventArgs e)

{

ListViewItem ListItem = new ListViewItem(textBox1.Text);

ListItem.SubItems.Add(textBox2.Text);

ListItem.SubItems.Add("В наличии");

ListItem.SubItems.Add("Находится в библиотеке");

ListItem.SubItems.Add("Находится в библиотеке");

listView1.Items.Add(ListItem);

MessageBox.Show("Данные добавлены!", "Библиотека");

save();

}

# **3.1.2. Удаление книги из списка.**

После нажатия одной из кнопок управление передается обработчику события “button\_Click” нажатой кнопки. Далее идет поиск книги которую требуется удалить, либо удаляются все книги. После программа переходит в функцию записи элементов listView1 в файл, что переписать изменения. Ниже представлен фрагмент кода:

private void btnDelFirst\_Click(object sender, EventArgs e)

{

if (listView1.Items.Count > 0)

listView1.Items.Remove(listView1.Items[0]);

MessageBox.Show("Первая книга удалена! ", "Библиотека");

save();

}

private void btnDelName\_Click(object sender, EventArgs e)

{ int i=0;

string m = textBox8.Text;

foreach (ListViewItem item in this.listView1.Items)

{

if (item.SubItems[0].Text == m)

{

listView1.Items.Remove(listView1.Items[i]);

}

i++;

}

MessageBox.Show("Книга " + m + " удалена! ", "Библиотека");

save();

}

private void btnDelAll\_Click(object sender, EventArgs e)

{

listView1.Items.Clear();

MessageBox.Show("Все книги удалены! ", "Библиотека");

save();

}

# **3.1.3. Функция меняющие параметры в списке, если книгу забрали.**

После нажатия кнопки управление передается обработчику события “button2\_Click”. Программа проверят написано ли что-то в textBox4.Text и textBox3.Text, если нет, то программа выходит из этого обработчика события. Далее осуществляется поиск по элементам, когда программа находит книгу, требуемую пользователем, она заменит значения о наличии книги. Потом программа переходит в функцию записи элементов listView1 в файл, что переписать изменения. Ниже представлен фрагмент кода:

private void btnBookOut\_Click(object sender, EventArgs e)

{

if (textBox4.Text == string.Empty)

return;

if (textBox3.Text == string.Empty)

return;

string date = maskedTextBox1.Text;

foreach (ListViewItem item in this.listView1.Items)

{

if ((item.SubItems[0].Text == textBox4.Text) && (item.SubItems[1].Text == textBox3.Text))

{

item.SubItems[2].Text = "На руках";

item.SubItems[3].Text = textBox5.Text;

item.SubItems[4].Text = date;

}

}

save();

}

# **3.1.4. Функция меняющие параметры в списке, если книгу вернули.**

После нажатия кнопки управление передается обработчику события “button3\_Click”. Программа проверят написано ли что-то в textBox7.Text и textBox6.Text, если нет, то программа выходит из этого обработчика события. Далее осуществляется поиск по элементам, когда программа находит книгу, требуемую пользователем, она заменит значения о наличии книги. Потом программа переходит в функцию записи элементов listView1 в файл, что переписать изменения. Ниже представлен фрагмент кода:

private void btnBookRe\_Click(object sender, EventArgs e)

{

{

if (textBox7.Text == string.Empty)

return;

if (textBox6.Text == string.Empty)

return;

foreach (ListViewItem item in this.listView1.Items)

{

if ((item.SubItems[0].Text == textBox7.Text) && (item.SubItems[1].Text == textBox6.Text))

{

item.SubItems[2].Text = "В наличии";

item.SubItems[3].Text = "Находится в библиотеке";

item.SubItems[4].Text = "Находится в библиотеке";

}

}

}

save();

}

# **3.1.5. Функция поиска на наличие книги.**

После нажатия кнопки управление передается обработчику события “button7\_Click”. Программа инициализирует логическую переменную и придает ей значение false. Создаются строковые переменные m и l в них заносятся значения из textBox10.Text и textBox9.Text соответственно. Далее осуществляется перебор всех элементов ListViewItem на поиск нужной книги и значения item.SubItems[2].Text == “В наличии”, если книга отвечает критериям поиска логическая переменная принимает значение true. В зависимости от значения лог.переменной выводится сообщение о наличии книги в библиотеке. Ниже представлен фрагмент кода:

private void btnSeach\_Click(object sender, EventArgs e)

{

bool seach = false;

string m = textBox10.Text, l = textBox9.Text;

foreach (ListViewItem item in this.listView1.Items)

{

if ((item.SubItems[0].Text == m) && (item.SubItems[1].Text == l) && (item.SubItems[2].Text == "В наличии"))

{ seach = true; }

}

if(seach == true)

MessageBox.Show("Книга - " + m + ", Автора - " + l +"; В наличии ", "Библиотека");

else

MessageBox.Show("Книга - " + m + ", Автора - " + l + "; Находится на руках ", "Библиотека");

}

# **3.2. Схема работы программы**

# 

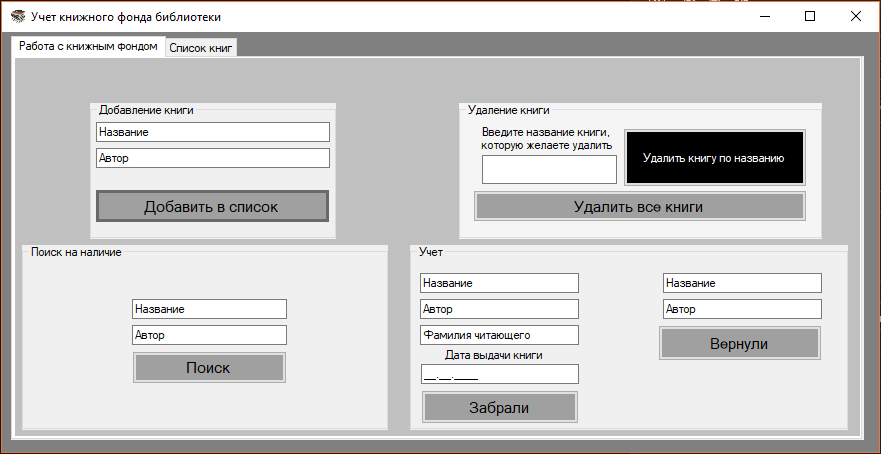
Библиотекарь

Читатель

# **3.3. Проектирование интерфейса.**

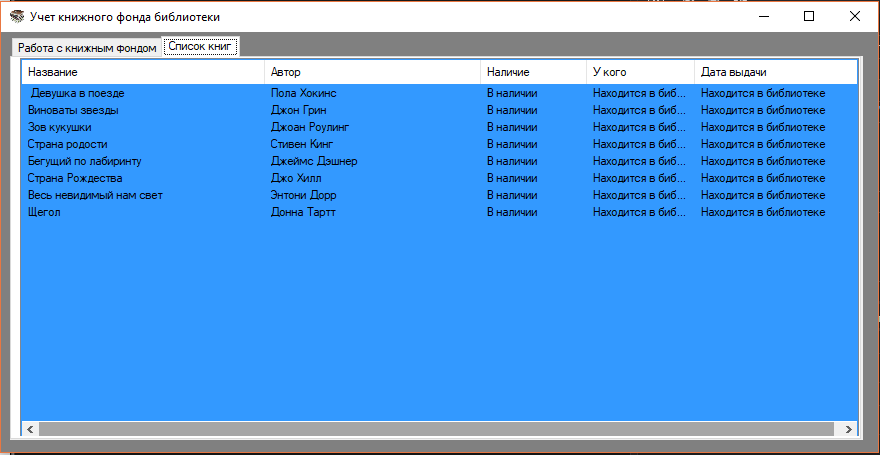
Проектирование интерфейса – не самая простая задач. Интерфейс должен быть интуитивен и понятен не только программисту, создавшему его, но и конечному пользователю, для которого собственно проект и разрабатывается. В ходе разработки интерфейс несколько раз менялся, пока не принял, на мой взгляд, достаточно приятный и понятный для стороннего пользователя вид:

На представленном ниже окне формы пользователь заполняет информацию, о книге которую хочет добавить в список книг, либо удалить книгу из списка, производится учет книг. Пользователь вводит информацию, о книге которую забрали, либо вернули. Пользователь заполняет информацию о книге и осуществляется поиск.



*Рис. 1. Интерфейс программы. Первая вкладка.*

На представленном ниже окне формы располагается список книг, содержащий информацию о них.



*Рис. 2. Интерфейс программы. Вторая вкладка.*

# **Глава 4. Инструкция для пользователя по работе с приложением**

После запуска программы, вы создаете список книг, которые находятся в библиотеке, чтобы в дальнейшем с ним работать. На вкладке “Работа с книжным фондом” можно ввести учет книг, если книгу взяли, вы вводите информацию о книге, фамилию читателя и дату выдачи книги на руки, вся эта информация заменяется в списке книг. Если книгу вернули, вам достаточно ввести название и автора книги и нажать кнопку “вернули”, книга снимется с учета, по информации о книге можно узнать, находится ли она на данный момент в библиотеке или нет, так же можно добавить, либо удалить книгу. На вкладке “Список книг” библиотекарь может ознакомиться с списком книг находящихся в библиотеке, а так же с их состоянием (в библиотеке они, или нет, если нет, то у кого и когда книгу забрали).

# **Заключение.**

В ходе выполнения этой работы был разработан программный модуль, предоставляющий полный функционал картотеки для учёта литературы в библиотеки, который полностью соответствует поставленной задаче. Я научился работать с файлами и объектом listView. Также были закреплены мои знания объектно-ориентированного языка С#.

# **Список использованных информационных источников:**

1. Язык программирования C#. Классика Computer Science. 4-е изд.

Авторы: А. Хейлсберг, М.Торгерсен, С.Вилтамут, П. Голд

1. [www.habrahabr.ru/hub/csharp/](http://www.habrahabr.ru/hub/csharp/) - Хабрахабр, информация по языку C#
2. [https://msdn.microsoft.com/](https://msdn.microsoft.com/ru-RU/library/mt472912(v=vs.110).aspx) - Microsoft Developer Network
3. "Полный справочник по C#" Автор: Герберт Шилдт
4. "Советы программистам" Автор: А. Климов
5. Язык программирования C# 5.0 и платформа .NET 4.5 Автор: Эндрю Троелсен..

# **Приложение 1**

Код программы:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Diagnostics;

using System.IO;

using System.Xml.Linq;

using System.Globalization;

namespace BiblioFond

{

public partial class Form1 : Form

{

string UsersDirectory = Environment.CurrentDirectory;

public Form1()

{

InitializeComponent();

parseItems();

}

bool isWrite = false;

private void save()

{

isWrite = true;

StringBuilder listViewContent = new StringBuilder();

foreach (ListViewItem item in this.listView1.Items)

{

foreach (ListViewItem.ListViewSubItem subItem in item.SubItems)

{

listViewContent.Append(subItem.Text);

listViewContent.Append('|');

}

listViewContent.Append(Environment.NewLine);

}

using (TextWriter tw = new StreamWriter(UsersDirectory + "\\biblio.txt"))

{

tw.WriteLine(listViewContent.ToString());

tw.Close();

}

}

private void parseItems()

{

listView1.Items.Clear();

using (StreamReader sReader = new StreamReader("biblio.txt"))

{

string line;

while ((line = sReader.ReadLine()) != null)

{

string[] values = line.Split('|');

listView1.Items.Add(new ListViewItem(values));

}

}

}

private void закрытьToolStripMenuItem\_Click(object sender, EventArgs e)

{

Close();

}

private void btnAddBook\_Click(object sender, EventArgs e)

{

ListViewItem ListItem = new ListViewItem(textBox1.Text);

ListItem.SubItems.Add(textBox2.Text);

ListItem.SubItems.Add("В наличии");

ListItem.SubItems.Add("Находится в библиотеке");

ListItem.SubItems.Add("Находится в библиотеке");

listView1.Items.Add(ListItem);

MessageBox.Show("Данные добавлены!", "Библиотека");

save();

}

private void textBox1\_MouseDown(object sender, MouseEventArgs e)

{

textBox1.Clear();

}

private void textBox2\_MouseDown(object sender, MouseEventArgs e)

{

textBox2.Clear();

}

private void textBox4\_MouseDown(object sender, MouseEventArgs e)

{

textBox4.Clear();

}

private void textBox3\_MouseDown(object sender, MouseEventArgs e)

{

textBox3.Clear();

}

private void textBox5\_MouseDown(object sender, MouseEventArgs e)

{

textBox5.Clear();

}

private void textBox7\_MouseDown(object sender, MouseEventArgs e)

{

textBox7.Clear();

}

private void textBox6\_MouseDown(object sender, MouseEventArgs e)

{

textBox6.Clear();

}

private void textBox10\_MouseDown(object sender, MouseEventArgs e)

{

textBox10.Clear();

}

private void textBox9\_MouseDown(object sender, MouseEventArgs e)

{

textBox9.Clear();

}

private void btnBookOut\_Click(object sender, EventArgs e)

{

if (textBox4.Text == string.Empty)

return;

if (textBox5.Text == string.Empty)

return;

string date = maskedTextBox1.Text;

foreach (ListViewItem item in this.listView1.Items)

{

if ((item.SubItems[0].Text == textBox4.Text) && (item.SubItems[1].Text == textBox5.Text))

{

item.SubItems[2].Text = "На руках";

item.SubItems[3].Text = textBox6.Text;

item.SubItems[4].Text = date;

}

}

save();

}

private void btnBookRe\_Click(object sender, EventArgs e)

{

{

if (textBox7.Text == string.Empty)

return;

if (textBox8.Text == string.Empty)

return;

foreach (ListViewItem item in this.listView1.Items)

{

if ((item.SubItems[0].Text == textBox7.Text) && (item.SubItems[1].Text == textBox8.Text))

{

item.SubItems[2].Text = "В наличии";

item.SubItems[3].Text = "Находится в библиотеке";

item.SubItems[4].Text = "Находится в библиотеке";

}

}

}

save();

}

private void btnDelFirst\_Click(object sender, EventArgs e)

{

if (listView1.Items.Count > 0)

listView1.Items.Remove(listView1.Items[0]);

MessageBox.Show("Первая книга удалена! ", "Библиотека");

save();

}

private void btnDelName\_Click(object sender, EventArgs e)

{

int i = 0;

string m = textBox3.Text;

foreach (ListViewItem item in this.listView1.Items)

{

if (item.SubItems[0].Text == m)

{

listView1.Items.Remove(listView1.Items[i]);

}

i++;

}

MessageBox.Show("Книга " + m + " удалена! ", "Библиотека");

save();

}

private void btnDelAll\_Click(object sender, EventArgs e)

{

listView1.Items.Clear();

MessageBox.Show("Все книги удалены! ", "Библиотека");

save();

}

private void btnSeach\_Click(object sender, EventArgs e)

{

bool seach = false;

string m = textBox10.Text, l = textBox9.Text;

foreach (ListViewItem item in this.listView1.Items)

{

if ((item.SubItems[0].Text == m) && (item.SubItems[1].Text == l) && (item.SubItems[2].Text == "В наличии"))

{ seach = true; }

}

if (seach == true)

MessageBox.Show("Книга - " + m + ", Автора - " + l + "; В наличии ", "Библиотека");

else

MessageBox.Show("Книга - " + m + ", Автора - " + l + "; Находится на руках ", "Библиотека");

}

private void Form1\_FormClosing(object sender, FormClosingEventArgs e)

{

if (isWrite)

{

StreamReader sReader = new StreamReader("biblio.txt");

string str = sReader.ReadToEnd();

sReader.Close();

TextWriter tw = new StreamWriter(UsersDirectory + "\\biblio.txt");

tw.Write(str.Remove(str.Length - 4, 4));

tw.Close();

}

}

private void textBox8\_MouseDown(object sender, MouseEventArgs e)

{

textBox8.Clear();

}

}

}