**Міністерство освіти і науки, МОЛОДІ І СПОРТУ України**

**Національний технічний університет України**

**“Київський політехнічний інститут”**

**IПСА**

**СП**

Пояснювальна записка

до курсової роботи на тему:

Електронний бiблiотечний каталог

**Студент групи** ДА-82м Юрченко Володимир Вiкторович

(шифр групи) (прізвище, ім’я, по батькові) (підпис)

**Керівник проекту** к. т. н. доцентКиселев Г. Д

(вчені ступінь та звання, прізвище, ініціали) (підпис)

|  |  |
| --- | --- |
| **Руководитель:** | **Исполнитель:** |
| **Кисельов Г. Д.** | **ст. Юрченко В.В**.  **гр. ДА-82м** |
| Допущен к защите | **Зач. книжка**  **№ ДА-8220** |
| **Защищено с оценкой** |  |

Київ – 2012

Додаток 1

ННК «ІПСА» НТУУ “Київський Політехнічний Інститут”

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(назва вищого навчального закладу)

Системного проектування

Кафедра\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Сучасні парадігми програмування і моделювання програмних систем

6.0501

Дисципліна\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Спеціальність\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

9

ДА – 82м\_\_\_

5

Курс\_\_\_\_\_\_\_\_\_\_Група\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Семестр\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ЗАВДАННЯ**

***на курсову роботу студента***

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_ Юрченка Володимира Вiкторовича \_\_\_\_\_\_­­­­­­\_\_\_\_\_\_\_\_\_\_\_

1. Тема курсової роботи \_\_\_ Електронний бiблiотечний каталог\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

до 25.12.2012 р.

1. Строк здачі студентом закінченого проекту (роботи)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. Вихідні дані до виконання роботи\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

Студент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис)

(підпис)

Керівник\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ доц. Кисельов Г.Д.

(прізвище, ім’я, по батькові)

«15» вересня 2012р.

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| п/п | Назва этапів курсової роботи | Термін виконання єтапів роботи | Відмітки |
|  |  |  |  |
| 1. |  |  |  |
|  |  |  |  |
|  |  |  |  |
| 2. |  | . |  |
|  |  |  |  |
|  |  |  |  |
| 3. |  |  |  |
|  |  |  |  |
| 4. |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Студент Юрченко В.В.

(підпис)

(підпис)

Керівник доц. Кисельов Г.Д.

«15» вересня 2011р.

**Содержание**

[Введение 5](#_Toc343845388)

[1. Задание и функции программы 6](#_Toc343845389)

[2. Построение концептуальной модели интерфейса 6](#_Toc343845390)

[2.1. Идентификация актеров 6](#_Toc343845391)

[2.2. Идентификация элементов Use Case 6](#_Toc343845392)

[2.3. Описания элементов Use Case 7](#_Toc343845393)

[2.3.1. Сценарии для элемента Use Case «Использование главного окна» 7](#_Toc343845394)

[2.3.2. Развитие описания элемента Use Case «Использование форм для специализированных действий» 8](#_Toc343845395)

[3. Проектирование функциональных элементов и межмодульных интерфейсов 11](#_Toc343845396)

[3.1. Выделение основных режимов работы 11](#_Toc343845397)

[3.2. Выбор и группировка пунктов меню 11](#_Toc343845398)

[4. Проектирование внешнего дизайна и структуры диалогов 12](#_Toc343845399)

[5. Проектирование дополнительных элементов управления и средств информационной поддержки пользователя 13](#_Toc343845400)

[5.1. Определение режимов, которые запускаются с помощью горячих клавиш и самих горячих клавиш 13](#_Toc343845401)

[5.2. Разработка скролл-баров 13](#_Toc343845402)

[6. Реализация графического интерфейса 15](#_Toc343845403)

[6.1. Выбор языка программирования 15](#_Toc343845404)

[6.2. Выбор операционной среды и определение ограничений на аппаратуру 15](#_Toc343845405)

[6.3. Реализация графического интерфейса модулей программы 15](#_Toc343845406)

[6.4. Архитектура приложения 17](#_Toc343845407)

[7. Реализация программы 19](#_Toc343845408)

[7.1. Выбор и обоснование парадигмы проектирования 19](#_Toc343845409)

[7.2. Программная реализация (листинг) 19](#_Toc343845410)

[Выводы 26](#_Toc343845411)

[Список использованной литературы 27](#_Toc343845412)

# Введение

Современные ПК обеспечивают пользователей вычислительной мощьность и объемами памяти, которые еще несколько лет назад были в разпоряжение исключительно профессиональных программистов. Однако пользователи этих ПК не особенно стремятся становиться компьютерными профессионалами. Они пришли в мир ПК из различных сфер деятельности: среди них учителя и преподаватели, бухгалтеры, врачи, юристы, банковские служащие.

Все больше услугами компьютеров пользуются ученики и студенты в своей учебной деятельности. Для всех этих и многих других категорий пользователей в отношении ПК характерно есть одна черта: они склонны воспринимать компьютер как техническое средство, просто помогает им выполнять определенную работу, и не стремятся понимать, как и почему работает ПК, как построены программы и т.д.., однако надеются, что с помощью компьютера в своей деятельности, выполнять свои задачи быстрее и надежнее, с минимальными затратами на овладение этой техникой.

На современном этапе развития программных средств ПК широком и успешном их внедрению способствует учета запросов массового пользователя по компьютерных информационных технологий. Создавая программные системы, разработчики должны учитывать эти запросы, то есть прежде предусмотреть приспособления программных средств системы к особенностям человеческого существа учитывая приоритет потребностей в общении с компьютером. Систематическое внедрение такого подхода стимулировать рост массовости компьютерных применений.

Напротив, программное обеспечение, которое заставляет пользователя приспосабливаться к каким-то, иногда даже хорошо не обоснованных ограничений, создает неудобства в работе, наверняка не найдет массового применения и будет отвергнуто.

# Задание и функции программы

Разработать оконный интерфейс пользователя для электронного библиотечного каталога.

Электронный библиотечный каталог (ЭБК) — совокупность программных и аппаратных средств по обеспечению деятельности библиотеки по заказу, каталогизации, поиску, выдаче книг, решения различных задач по отчётности и книгообеспеченности читателей и др. как в локальной вычислительной сети, так и через web-сопряжение.

Для ЭКБ реализовывается графический интерфейс пользователя (GUI).

# Построение концептуальной модели интерфейса

# Идентификация актеров

Актерами для ЭКБявляются **Пользователь-читатель, Сотрудники отдела комплектования, Сотрудники отдела выдачи***.*

Внешнее окружение имеет вид, представленный на рис. 1.

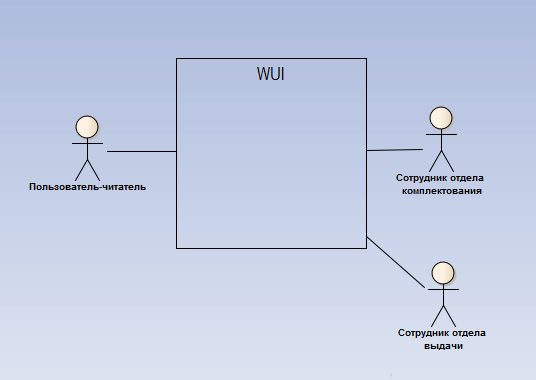


Рис 1. Внешнее окружение WUI

# Идентификация элементов Use Case

В WUI могут быть выделены несколько элементов Use Case:

* Использование главного окна
* Использование форм для специализированных действий

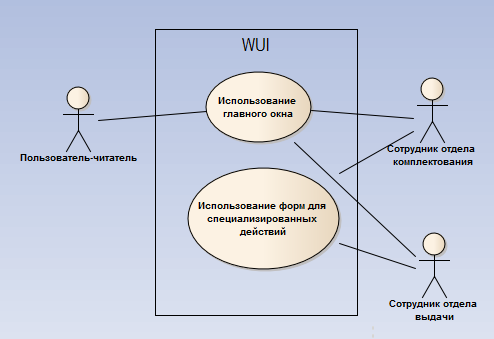


Рис 2. Диаграмма Use Case для ЭКБ

# Описания элементов Use Case

* Описание элемента Use Case «Использование главного окна»:

Действия начинаются любым из актеров. Главное окно является узловым пунктом работы системы. Пользователь-читатель может работать с текстовыми полями и кнопками главного окна. Сотрудники имеют дополнительную возможность с помощью меню и кнопок вызывать вторичные окна (формы) для выполнения функции из набора модулей ЭКБ.

* Описание элемента Use Case «Использование форм для специализированных действий»:

Действия начинаются Сотрудником. Обеспечивается возможность работы с простыми окнами. Каждая форма содержит списки полей и кнопок для выполнения конкретной специализированной задачи.

# Сценарии для элемента Use Case «Использование главного окна»

В элементе Use Case «Использование главного окна» заданы такие сценарии:

1. *Сценарий «Поиск издания».*

После заполнения необходимых текстовых полей и нажатия кнопки результат поиска выводится в таблицу найденных книг основного окна.

1. *Сценарий «Заказ издания».*

После выбора необходимого издания в таблице с помощью кнопки вызывается диалоговое окно «Параметры заказа».

1. *Сценарий «Вход».*

При выборе пункта меню открывается форма входа с текстовыми полями «Логин» и «Пароль». После успешного входа открывается специализированная форма, иначе возврат к главному окну.

# Развитие описания элемента Use Case «Использование форм для специализированных действий»

В элементе Use Case «Использование форм для специализированных действий» объединены сценарии использования форм книговыдачи и комплектования:

1. *Сценарий «Добавить запись».*

После заполнения всех необходимых текстовых полей (3шт) необходимо нажать кнопку «Добавить запись», чтобы ввести новую информацию в ЭКБ.

1. *Сценарий «Редактировать запись».*

С помощью кнопки вызывается диалоговое окно с текстовыми полями, предназначенными для ввода новой информации, которая заменяет введенные в форме данные об издании.

1. *Сценарий «Открыть карточку читателя».*

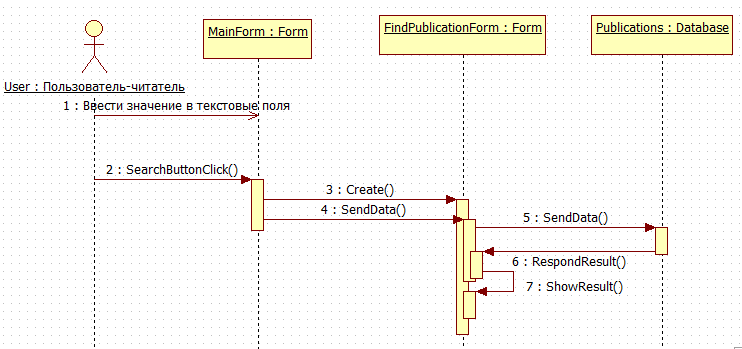
Посредством заполнения соответствующих полей и нажатием кнопки в элемент типа список загружается информация об изданиях, находившихся, или находящихся у читателя.

1. *Сценарий «Выдать издание».*

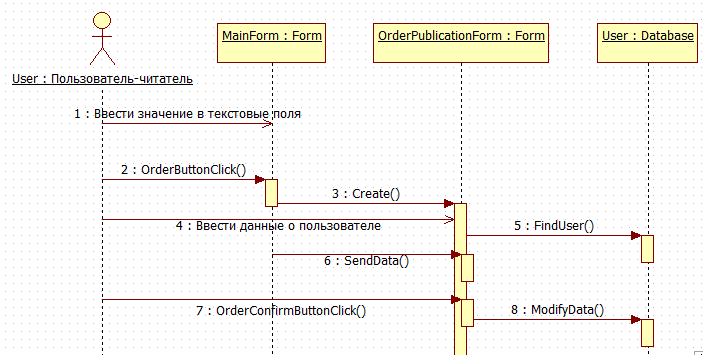
После заполнения текстовых полей в таблице в карточку читателя добавляется новая запись.

1. *Сценарий «Подтвердить возврат издания».*

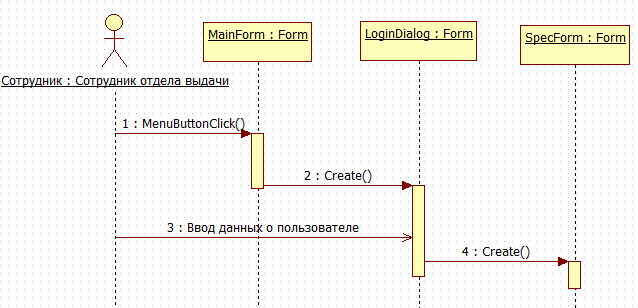
Запись о возвращении издания пользователем осуществляется удалением строки из таблицы.



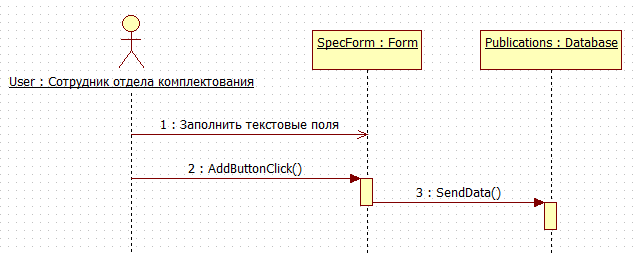
*Сценарий «Поиск издания».*



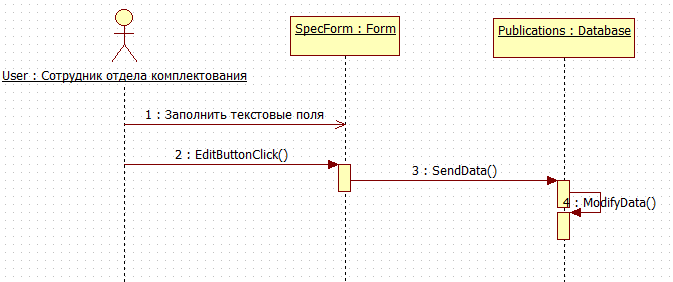
*Сценарий «Заказ издания».*



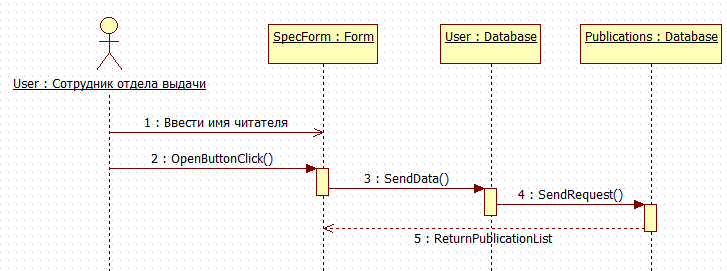
*Сценарий «Вход»*



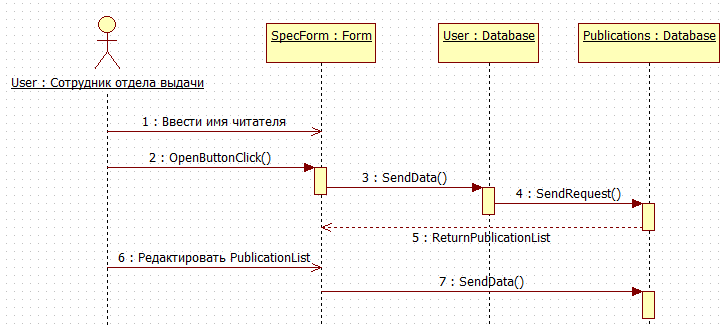
*Сценарий «Добавить запись»*



*Сценарий «Редактировать запись»*



*Сценарий «Открыть карточку читателя»*

**

*Сценарий «Выдать издание»*

# Проектирование функциональных элементов и межмодульных интерфейсов

# Выделение основных режимов работы

Программа может работать в 3-х режимах:

1. Режим поиска/заказа издания
2. Режим управления записями об изданиях
3. Режим управления учетными картами пользователей

# Выбор и группировка пунктов меню

В первом режиме программа находится по умолчанию – этот режим реализуется главным окном программы.

Для входа в остальные режимы пользователю-сотруднику библиотеки необходимо воспользоваться одним из пунктов меню «Вход».

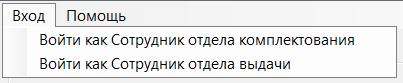


Рис 3. Иерархия пунктов меню главного окна

Пиктографические пункты меню в данной программе не применяются, поскольку не способствуют более интуитивному пониманию функций меню.

Для других окон применение меню не было сочтено сообразным, поскольку функции, реализованные в специализированных формах, осуществляются взаимодействием с кнопками или таблицами.

# Проектирование внешнего дизайна и структуры диалогов

Для каждого пункта меню были разработаны специализированные окна-диалоги. Для доступа к специальным функциям программы пользователя нужно ввести свои логин и пароль в диалоговое окно.

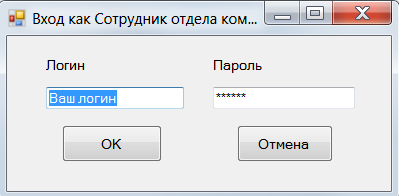


Рис 4. Внешний вид диалогового окна авторизации

В диалоговом окне используются подсказки для указания типа данных, которые необходимо ввести в конкретное текстовое поле.

После нажатия кнопки ОК открывается специализированная форма, а основное окно скрывается, но не закрывается. После закрытия специализированного окна программа возвращается к основному окну для того, чтобы пользователь мог продолжать выполнять другие функции.

Для пункта меню «Помощь» также разработано диалоговое окно, которое содержит только надпись-сообщение об авторе программы.

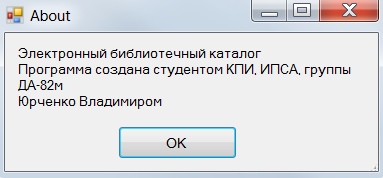


Рис 5. Диалоговое окно "О программе"

Для создания информативных диалоговых окон также используется шаблон с надписью и кнопкой подтверждения:

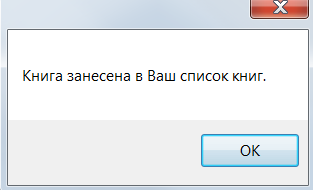


Рис 6. Информативное диалоговое окно

# Проектирование дополнительных элементов управления и средств информационной поддержки пользователя

# Определение режимов, которые запускаются с помощью горячих клавиш и самих горячих клавиш

С помощью горячих клавиш пользователь может быстро войти в режим Сотрудника отдела комплектования или Сотрудника отдела выдачи.

Вход в главное меню осуществляется нажатием комбинации клавиш Alt + 1.

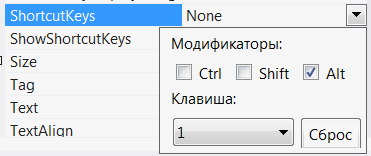


Рис 7. Создание горячих клавиш

Дальнейшие действия производятся другими сочетаниями клавиш:

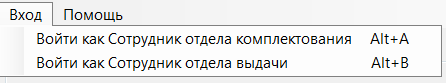


Рис 8. Горячие клавиши для доступа к различным режимам

# Разработка и определение текста всплывающих подсказок

Для облегчения понимания пользователем предназначения кнопок программы были разработаны всплывающие подсказки.

private void button2\_MouseHover(object sender, EventArgs e)

{

toolTip1.SetToolTip(button2, "Выберите книгу в списке найденных изданий и нажмите кнопку \"Заказать\", чтобы внести книгу в свою карточку");

}

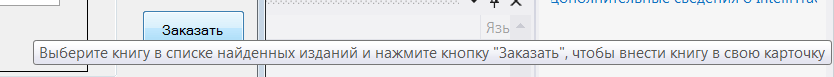


Рис 9. Пример работы всплывающей подсказки

# Разработка скролл-баров

Использование скролл-баров (линеек прокрутки) определяется на этапе создания элемента управления.

В данном примере показан процесс установки скролл-бара для элемента типа «таблица».

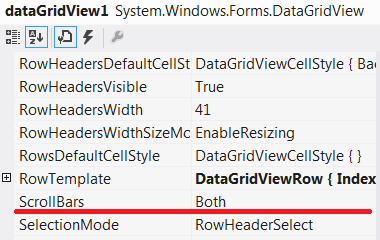


Рис 10. Создание линеек прокрутки для таблицы в главном окне

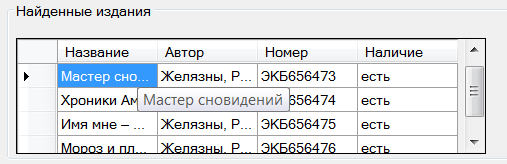


Рис 11. Пример использования полей прокрутки

# Реализация графического интерфейса

# Выбор языка программирования

Был выбран язык программирования С# ввиду простоты реализации графического интерфейса на данном языке с помощью среды Microsoft Visual Studio 2012 и технологии Windows Forms.

# Выбор операционной среды и определение ограничений на аппаратуру

Запуск программы предполагается на конфигурации оборудования не ниже приведенных требований:

* процессор - Intel Pentium III,
* оперативная память (RAM) - 512 Мб,
* свободное место на жестком диске - 4 ГБ.
* операционная система семейства Windows.

# Реализация графического интерфейса модулей программы

Внешний вид главного окна показан на рисунке ниже:

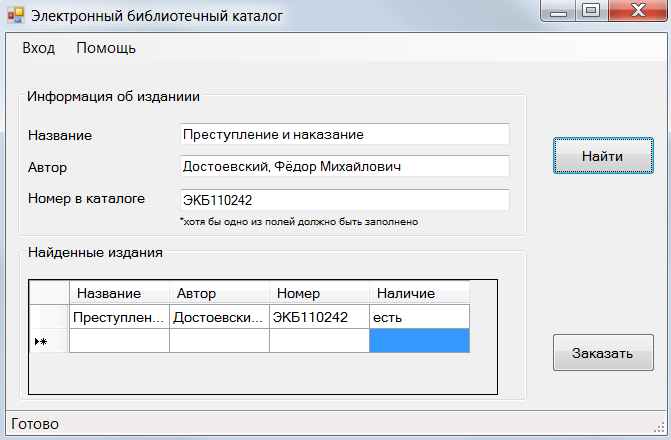


Рис 12. Основное окно программы

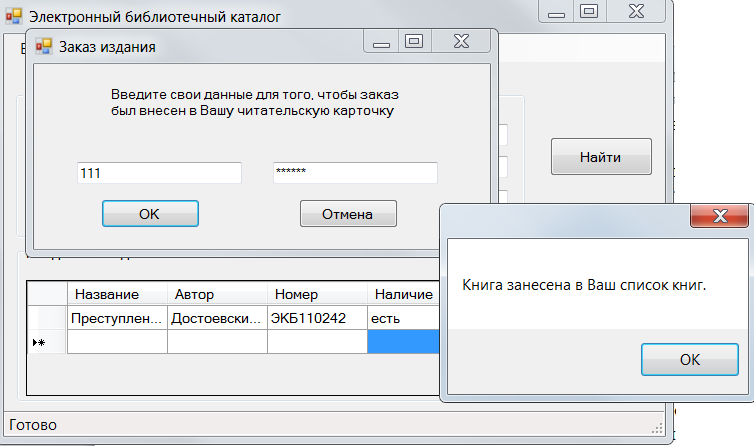


Рис 13. Сценарий заказа издания

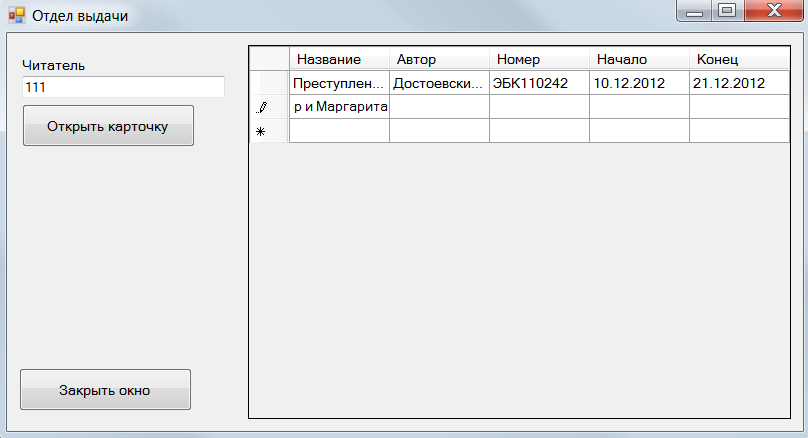


Рис 14. Сценарий редактирования читательской карты пользователя

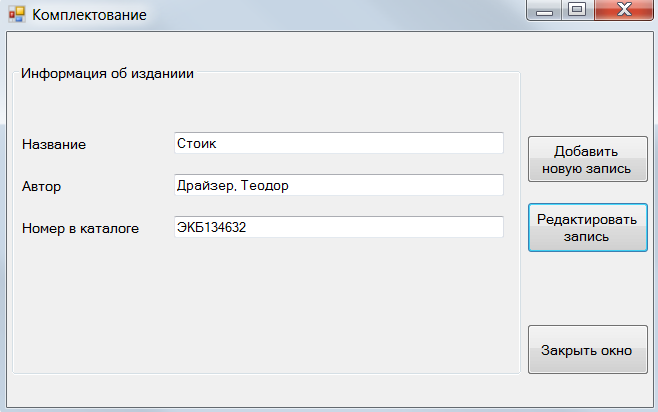


Рис 15. Окно управления информацией об изданиях

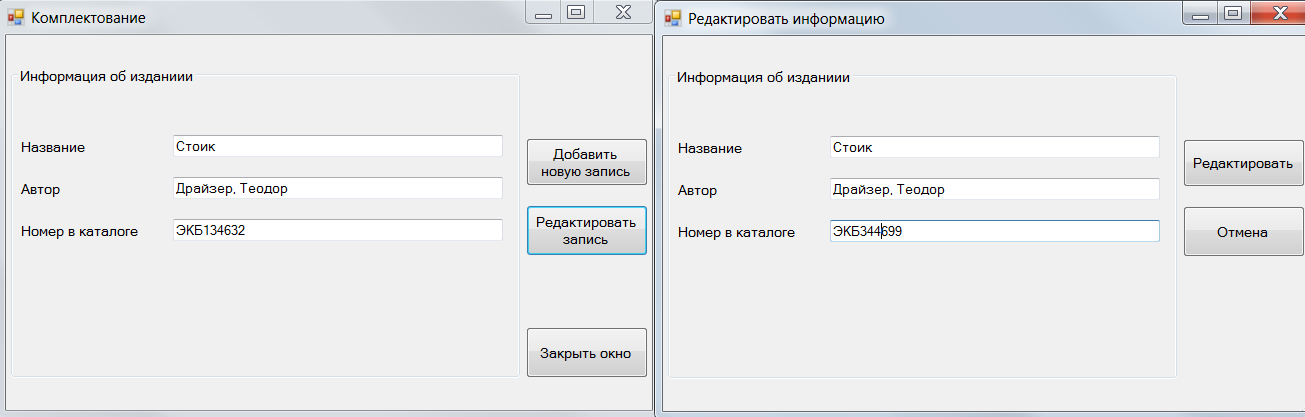


Рис 16. Диалог изменения записи об издании

# Архитектура приложения

Работа по созданию классов (и включению их в диаграмму классов) требует изучения содержания всех диаграмм последовательности. Проводится она в три этапа.

На первом этапе выявляются и именуются классы. Для этого просматривается каждая диаграмма последовательности. Любой объект в этой диаграмме должен принадлежать конкретному классу.

На втором этапе выявляются операции классов. На диаграмме последовательности такая операция соответствует стрелке (и имени) сообщения, указывающей на линию жизни объекта класса.

На третьем этапе определяются отношения ассоциации между классами — они обеспечивают пересылки сообщений между соответствующими объектами.

В данной программе были реализованы и использованы следующие классы:

* Form — класс, объектами которого являются простые окна;
* Database — класс, объектами которого являются некоторые структуры, подобные базам данных, спискам.
* Menu — класс, объектами которого являются окна меню. Класс является стандартным элементом интерфейса Windows;
* Button — класс, объектом которого является кнопка.
* Label — объект этого класса является надписью, основным средством для передачи сообщений пользователю. При этом текст сообщения пользователь не может редактировать;
* TextBox — редактируемое поле. Является средством введения информации пользователем.
* DataGridView — объект этого класса есть таблица. Также используется для ввода структурированной информации.

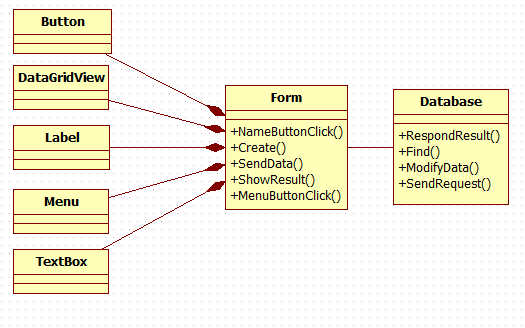


Рис 17. Начальная диаграмма классов WUI

# Реализация программы

# Выбор и обоснование парадигмы проектирования

Выбор был остановлен на событийно-ориентированной парадигме проектирования ввиду использования таких инструментов, как MS Visual Studio, позволяющих в наиболее короткие сроки создать приложение с графическим интерфейсом с данной парадигмой.

Событи́йно-ориенти́рованное программи́рование (англ. event-driven programming; в дальнейшем СОП) — парадигма программирования, в которой выполнение программы определяется событиями — действиями пользователя (клавиатура, мышь), сообщениями других программ и потоков, событиями операционной системы (например, поступлением сетевого пакета).

СОП можно также определить как способ построения компьютерной программы, при котором в коде (как правило, в головной функции программы) явным образом выделяется главный цикл приложения, тело которого состоит из двух частей: выборки события и обработки события.

Событийно-ориентированное программирование, как правило, применяется в трех случаях:

* при построении пользовательских интерфейсов (в том числе графических);
* при создании серверных приложений в случае, если по тем или иным причинам нежелательно порождение обслуживающих процессов;
* при программировании игр, в которых осуществляется управление множеством объектов.

# Программная реализация (листинг)

Ниже приведен фрагмент программы, который отвечает за обработку событий от действий пользователя в главном окне 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;

namespace ParKP

{

public partial class MainForm : Form

{

public MainForm()

{

InitializeComponent();

}

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

{

ComplectDialog complectDialog = new ComplectDialog(this);

complectDialog.Show();

}

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

{

IssueDialog issueDialog = new IssueDialog(this);

issueDialog.Show();

}

private void button1\_Click(object sender, EventArgs e)

{

bool already\_in\_list = false;

for (int i = 0; i < dataGridView1.RowCount; i++)

{

if ((dataGridView1.Rows[i].Cells[0].Value != null) &&

(dataGridView1.Rows[i].Cells[1].Value != null) &&

(dataGridView1.Rows[i].Cells[2].Value != null) &&

(dataGridView1.Rows[i].Cells[0].Value.ToString() == textBox1.Text) &&

(dataGridView1.Rows[i].Cells[1].Value.ToString() == textBox2.Text) &&

(dataGridView1.Rows[i].Cells[2].Value.ToString() == textBox3.Text))

{

already\_in\_list = true;

}

}

if (!already\_in\_list)

{

dataGridView1.Rows.Add(textBox1.Text, textBox2.Text, textBox3.Text, "есть");

}

}

private void button2\_Click(object sender, EventArgs e)

{

if (textBox1.Text != "" && textBox2.Text != "" && textBox3.Text != "")

{

OrderPublicationDialog opDialog = new OrderPublicationDialog();

opDialog.ShowDialog();

}

}

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

{

About about = new About();

about.ShowDialog();

}

}

}

Ниже приведен фрагмент кода, который генерируется автоматически при создании элемента графического интерфейса. Тем не менее, этот код может быть изменен вручную или с помощью графических средств проектирования MS Visual Studio 2012.

private void InitializeComponent()

{

this.menuStrip1 = new System.Windows.Forms.MenuStrip();

this.входToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();

this.loginComplectToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();

this.loginIssueToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();

this.helpToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();

this.aboutToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();

this.groupBox1 = new System.Windows.Forms.GroupBox();

this.label4 = new System.Windows.Forms.Label();

this.textBox3 = new System.Windows.Forms.TextBox();

this.label3 = new System.Windows.Forms.Label();

this.textBox2 = new System.Windows.Forms.TextBox();

this.label2 = new System.Windows.Forms.Label();

this.textBox1 = new System.Windows.Forms.TextBox();

this.label1 = new System.Windows.Forms.Label();

this.button1 = new System.Windows.Forms.Button();

this.button2 = new System.Windows.Forms.Button();

this.groupBox2 = new System.Windows.Forms.GroupBox();

this.dataGridView1 = new System.Windows.Forms.DataGridView();

this.Column1 = new System.Windows.Forms.DataGridViewTextBoxColumn();

this.Column2 = new System.Windows.Forms.DataGridViewTextBoxColumn();

this.Column3 = new System.Windows.Forms.DataGridViewTextBoxColumn();

this.Column4 = new System.Windows.Forms.DataGridViewTextBoxColumn();

this.statusStrip1 = new System.Windows.Forms.StatusStrip();

this.toolStripStatusLabel1 = new System.Windows.Forms.ToolStripStatusLabel();

this.menuStrip1.SuspendLayout();

this.groupBox1.SuspendLayout();

this.groupBox2.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.dataGridView1)).BeginInit();

this.statusStrip1.SuspendLayout();

this.SuspendLayout();

//

// menuStrip1

//

this.menuStrip1.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {

this.входToolStripMenuItem,

this.helpToolStripMenuItem});

this.menuStrip1.Location = new System.Drawing.Point(0, 0);

this.menuStrip1.Name = "menuStrip1";

this.menuStrip1.Size = new System.Drawing.Size(661, 28);

this.menuStrip1.TabIndex = 0;

this.menuStrip1.Text = "menuStrip1";

//

// входToolStripMenuItem

//

this.входToolStripMenuItem.DropDownItems.AddRange(new System.Windows.Forms.ToolStripItem[] {

this.loginComplectToolStripMenuItem,

this.loginIssueToolStripMenuItem});

this.входToolStripMenuItem.Name = "входToolStripMenuItem";

this.входToolStripMenuItem.ShortcutKeys = ((System.Windows.Forms.Keys)((System.Windows.Forms.Keys.Alt | System.Windows.Forms.Keys.D1)));

this.входToolStripMenuItem.Size = new System.Drawing.Size(54, 24);

this.входToolStripMenuItem.Text = "Вход";

//

// loginComplectToolStripMenuItem

//

this.loginComplectToolStripMenuItem.Name = "loginComplectToolStripMenuItem";

this.loginComplectToolStripMenuItem.ShortcutKeys = ((System.Windows.Forms.Keys)((System.Windows.Forms.Keys.Alt | System.Windows.Forms.Keys.A)));

this.loginComplectToolStripMenuItem.Size = new System.Drawing.Size(442, 24);

this.loginComplectToolStripMenuItem.Text = "Войти как Сотрудник отдела комплектования";

this.loginComplectToolStripMenuItem.Click += new System.EventHandler(this.loginComplectToolStripMenuItem\_Click);

//

// loginIssueToolStripMenuItem

//

this.loginIssueToolStripMenuItem.Name = "loginIssueToolStripMenuItem";

this.loginIssueToolStripMenuItem.ShortcutKeys = ((System.Windows.Forms.Keys)((System.Windows.Forms.Keys.Alt | System.Windows.Forms.Keys.B)));

this.loginIssueToolStripMenuItem.Size = new System.Drawing.Size(442, 24);

this.loginIssueToolStripMenuItem.Text = "Войти как Сотрудник отдела выдачи";

this.loginIssueToolStripMenuItem.Click += new System.EventHandler(this.loginIssueToolStripMenuItem\_Click);

//

// helpToolStripMenuItem

//

this.helpToolStripMenuItem.DropDownItems.AddRange(new System.Windows.Forms.ToolStripItem[] {

this.aboutToolStripMenuItem});

this.helpToolStripMenuItem.Name = "helpToolStripMenuItem";

this.helpToolStripMenuItem.ShortcutKeys = ((System.Windows.Forms.Keys)((System.Windows.Forms.Keys.Alt | System.Windows.Forms.Keys.D2)));

this.helpToolStripMenuItem.Size = new System.Drawing.Size(81, 24);

this.helpToolStripMenuItem.Text = "Помощь";

//

// aboutToolStripMenuItem

//

this.aboutToolStripMenuItem.Name = "aboutToolStripMenuItem";

this.aboutToolStripMenuItem.ShortcutKeys = ((System.Windows.Forms.Keys)((System.Windows.Forms.Keys.Alt | System.Windows.Forms.Keys.H)));

this.aboutToolStripMenuItem.Size = new System.Drawing.Size(222, 24);

this.aboutToolStripMenuItem.Text = "О программе";

this.aboutToolStripMenuItem.Click += new System.EventHandler(this.aboutToolStripMenuItem\_Click);

//

// groupBox1

//

this.groupBox1.Controls.Add(this.label4);

this.groupBox1.Controls.Add(this.textBox3);

this.groupBox1.Controls.Add(this.label3);

this.groupBox1.Controls.Add(this.textBox2);

this.groupBox1.Controls.Add(this.label2);

this.groupBox1.Controls.Add(this.textBox1);

this.groupBox1.Controls.Add(this.label1);

this.groupBox1.Location = new System.Drawing.Point(12, 54);

this.groupBox1.Name = "groupBox1";

this.groupBox1.Size = new System.Drawing.Size(509, 150);

this.groupBox1.TabIndex = 5;

this.groupBox1.TabStop = false;

this.groupBox1.Text = "Информация об изданиии";

//

// label4

//

this.label4.AutoSize = true;

this.label4.Font = new System.Drawing.Font("Microsoft Sans Serif", 5.8F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));

this.label4.Location = new System.Drawing.Point(159, 127);

this.label4.Name = "label4";

this.label4.Size = new System.Drawing.Size(245, 13);

this.label4.TabIndex = 11;

this.label4.Text = "\*хотя бы одно из полей должно быть заполнено";

//

// textBox3

//

this.textBox3.Location = new System.Drawing.Point(162, 102);

this.textBox3.Name = "textBox3";

this.textBox3.Size = new System.Drawing.Size(330, 22);

this.textBox3.TabIndex = 10;

//

// label3

//

this.label3.AutoSize = true;

this.label3.Location = new System.Drawing.Point(7, 102);

this.label3.Name = "label3";

this.label3.Size = new System.Drawing.Size(125, 17);

this.label3.TabIndex = 9;

this.label3.Text = "Номер в каталоге";

//

// textBox2

//

this.textBox2.Location = new System.Drawing.Point(162, 68);

this.textBox2.Name = "textBox2";

this.textBox2.Size = new System.Drawing.Size(330, 22);

this.textBox2.TabIndex = 8;

//

// label2

//

this.label2.AutoSize = true;

this.label2.Location = new System.Drawing.Point(7, 71);

this.label2.Name = "label2";

this.label2.Size = new System.Drawing.Size(47, 17);

this.label2.TabIndex = 7;

this.label2.Text = "Автор";

//

// textBox1

//

this.textBox1.Location = new System.Drawing.Point(162, 36);

this.textBox1.Name = "textBox1";

this.textBox1.Size = new System.Drawing.Size(330, 22);

this.textBox1.TabIndex = 6;

//

// label1

//

this.label1.AutoSize = true;

this.label1.Location = new System.Drawing.Point(7, 39);

this.label1.Name = "label1";

this.label1.Size = new System.Drawing.Size(72, 17);

this.label1.TabIndex = 5;

this.label1.Text = "Название";

//

// button1

//

this.button1.Location = new System.Drawing.Point(546, 103);

this.button1.Name = "button1";

this.button1.Size = new System.Drawing.Size(103, 39);

this.button1.TabIndex = 6;

this.button1.Text = "Найти";

this.button1.UseVisualStyleBackColor = true;

this.button1.Click += new System.EventHandler(this.button1\_Click);

//

// button2

//

this.button2.Location = new System.Drawing.Point(546, 300);

this.button2.Name = "button2";

this.button2.Size = new System.Drawing.Size(103, 39);

this.button2.TabIndex = 7;

this.button2.Text = "Заказать";

this.button2.UseVisualStyleBackColor = true;

this.button2.Click += new System.EventHandler(this.button2\_Click);

//

// groupBox2

//

this.groupBox2.Controls.Add(this.dataGridView1);

this.groupBox2.Location = new System.Drawing.Point(13, 210);

this.groupBox2.Name = "groupBox2";

this.groupBox2.Size = new System.Drawing.Size(508, 158);

this.groupBox2.TabIndex = 8;

this.groupBox2.TabStop = false;

this.groupBox2.Text = "Найденные издания";

//

// dataGridView1

//

this.dataGridView1.BackgroundColor = System.Drawing.SystemColors.Control;

this.dataGridView1.ColumnHeadersHeightSizeMode = System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;

this.dataGridView1.Columns.AddRange(new System.Windows.Forms.DataGridViewColumn[] {

this.Column1,

this.Column2,

this.Column3,

this.Column4});

this.dataGridView1.EditMode = System.Windows.Forms.DataGridViewEditMode.EditProgrammatically;

this.dataGridView1.Location = new System.Drawing.Point(9, 36);

this.dataGridView1.Name = "dataGridView1";

this.dataGridView1.RowTemplate.Height = 24;

this.dataGridView1.Size = new System.Drawing.Size(470, 116);

this.dataGridView1.TabIndex = 5;

//

// Column1

//

this.Column1.HeaderText = "Название";

this.Column1.Name = "Column1";

//

// Column2

//

this.Column2.HeaderText = "Автор";

this.Column2.Name = "Column2";

//

// Column3

//

this.Column3.HeaderText = "Номер";

this.Column3.Name = "Column3";

//

// Column4

//

this.Column4.HeaderText = "Наличие";

this.Column4.Name = "Column4";

//

// statusStrip1

//

this.statusStrip1.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {

this.toolStripStatusLabel1});

this.statusStrip1.Location = new System.Drawing.Point(0, 377);

this.statusStrip1.Name = "statusStrip1";

this.statusStrip1.Size = new System.Drawing.Size(661, 25);

this.statusStrip1.TabIndex = 9;

this.statusStrip1.Text = "Готово";

//

// toolStripStatusLabel1

//

this.toolStripStatusLabel1.Name = "toolStripStatusLabel1";

this.toolStripStatusLabel1.Size = new System.Drawing.Size(57, 20);

this.toolStripStatusLabel1.Text = "Готово";

//

// MainForm

//

this.AutoScaleDimensions = new System.Drawing.SizeF(8F, 16F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.ClientSize = new System.Drawing.Size(661, 402);

this.Controls.Add(this.statusStrip1);

this.Controls.Add(this.groupBox2);

this.Controls.Add(this.button2);

this.Controls.Add(this.button1);

this.Controls.Add(this.groupBox1);

this.Controls.Add(this.menuStrip1);

this.MainMenuStrip = this.menuStrip1;

this.Name = "MainForm";

this.Text = "Электронный библиотечный каталог";

this.menuStrip1.ResumeLayout(false);

this.menuStrip1.PerformLayout();

this.groupBox1.ResumeLayout(false);

this.groupBox1.PerformLayout();

this.groupBox2.ResumeLayout(false);

((System.ComponentModel.ISupportInitialize)(this.dataGridView1)).EndInit();

this.statusStrip1.ResumeLayout(false);

this.statusStrip1.PerformLayout();

this.ResumeLayout(false);

this.PerformLayout();

}

# Выводы

В результате выполнения курсовой работы был разработан пользовательский интерфейс электронного библиотечного каталога. Были изучены и использованы основные принципы построения пользовательского интерфейса с применением средств C# в среде Microsoft Visual Studio 2012.

Были применены такие техники, как всплывающие подсказки, горячие клавиши, полосы прокрутки, использованы стандартные элементы графического интерфейса, созданы свои макеты форм и диалоговых окон.

# Список использованной литературы

1. С.А. Орлов. Технологии разработки программного обеспечения: Учебник. – СПб.: Питер, 2002. – 464 с.
2. Hans-Erik Eriksson, Magnus Penker. UML Toolkit. – John Willey & Sons, Inc., 1998, p.397.
3. Дэвид Флорэнгэн. Java in a Nutsheell. - Kиев: BHV, 1998/
4. Грегори Р. Эндрюс. Основы многопоточного, параллельного и распределенного программирования. – М.: Издательский дом «Вильямс», 2003. – 512с.
5. Т.В. Ковалюк. «Основи програмування». Київ, BHV, 2005р.
6. Баженова И.Ю., Сухомлин В.А. Введение в программирование <http://www.intuit.ru/department/pl/plintro/>
7. Непейвода Н.Н. Стили и методы программирования <http://www.intuit.ru/department/se/progstyles/>
8. Храмцов П.,Б., Брик С.А., Русак А.М., Сурин А.И. Введение в HTML <http://www.intuit.ru/department/internet/htmlintro/>
9. Городняя Л.В. Основы функционального программирования <http://www.intuit.ru/department/pl/funcpl/>
10. Алексеев В.Е., Таланов В.А. Структуры данных и модели вычислений <http://www.intuit.ru/department/algorithms/dscm/>
11. <http://www.intuit.ru/department/se/ooad/>Основы объектно-ориентированного проектирования
12. Барский А.Б. Параллельное программирование <http://www.intuit.ru/department/se/parallprog/>
13. Верещагин Н.К., Шень А.Х. Языки и исчисления <http://www.intuit.ru/department/calculate/lancalc/>
14. Леоненков А.В. Нотация и семантика языка UML <http://www.intuit.ru/department/pl/umlbasics/>
15. Котляров В.П. <http://www.intuit.ru/department/se/testing/> Основы тестирования программного обеспечения
16. Кулямин В.В. Компонентный подход в программировании <http://www.intuit.ru/department/se/compprog/>