

ПОМОЩНИК В СОЗДАНИИ ГИПЕРССЫЛОК

Белов В.Н., Ковалёв А.И., Новиков С.А.

Арзамасский филиал ННГУ, Арзамас, e-mail: bwn.arz@list.ru

С появлением персональных компьютеров люди возложили часть своей работы на них, автоматизировали часть своей деятельности. Перед нами возникла задача, проанализировав которую, мы пришли к выводу, что необходимо создать программный продукт, который возьмет на себя большую часть работы. Представленная в данной статье программа «List of Hyperlinks Helper» (LHH) нацелена на автоматизацию человеческой деятельности. В частности, данная программа автоматически создает гиперссылки для сайтов. Для начала использования LHH, необходим текстовый файл, в котором находятся все названия файлов, для которых будут реализованы гиперссылки (чаще всего это файлы формата «.pdf»). Кроме этого, программе необходимо указать путь к директории, где лежат файлы для гиперссылок, и, если это требуется, указать место сохранения конечного файла (если путь не указан, программа создаст файл с результатами в той директории, из которой была запущена). Все методы, которые использует List of Hyperlinks Helper, подробно описаны в статье. На данный момент программа находится на стадии разработки, но ее уже можно использовать. LHH была проверена в действии и позволила ускорить процесс создания гиперссылок для сайта в несколько раз. Программа была написана в короткие сроки, потому имеет ряд недоработок и ошибок, которые выявляются в процессе использования. Хотя ошибки, встречающиеся чаще всего, уже были устранены, доработка программы продолжается, и в ходе ее использования планируется минимизировать количество проблем в работе программы. В нашем случае время, потраченное на написание программы, было полностью оправдано ее функциональностью, именно поэтому было принято решение по доработке программы для дальнейшего использования.

Ключевые слова: автоматизация, быстрая разработка, гиперссылки, инновация, открытый программный код, язык программирования C#, HTML, List of Hyperlinks Helper, Visual Studio Express C#, Windows Presentation Foundation (WPF)

LIST OF HYPERLINKS HELPER

Belov V.N., Kovalev A.I., Novikov S.A.

Lobachevsky University (Arzamas branch), Arzamas, e-mail: bwn.arz@list.ru

With the advent of personal computers, people put part of their work on them, has automated part of its activities. We have a task, which, we came to the conclusion that it is necessary to create a software product that will take most of the work. Presented in this article, the program «List of Hyperlinks Helper» (LHH) is aimed at the automation of human activities. In particular, this program automatically creates hyperlinks for your sites. To start using LHH required text file containing all the names of the files that will be implemented hyperlinks (most often the file format «.pdf»). In addition, the program must specify the path to the directory where the files for hyperlinks, and if required, specify the location to save target file (if no path is specified, the program will create a file in the directory from which it was launched). All of the methods that uses a List of Hyperlinks Helper, described in detail in the article. At the moment the program is still under development, but it is already possible to use. LHH was tested in action and helped accelerate the process of creating hyperlinks to the site several times. The program was written in a short time, therefore has a number of flaws and errors that are detected during use. Although the errors that occur most often, have already been eliminated, the improvement of the program continues, and in the course of its use is planned to minimise the number of problems in the program. In our case, the time spent on the writing program was fully justified by its functionality, that is why it was decided to refine the program for future use.

Keywords: automation, rapid development, hyperlinks, innovation, free program code, HTML, List of Hyperlinks Helper, program language C#, Visual Studio Express C#, Windows Presentation Foundation (WPF)

С изобретением персональных компьютеров человек стал перекладывать на них большую часть рутинной работы, начал автоматизацию своей деятельности. Благодаря этому человек начал двигать прогресс вперед большими шагами, так как он перестал тратить огромную часть энергии на уже исследованную им часть рабочей области: он просто автоматизировал ее и перешел к изучению более важных проблем. Это принесло за собой создание информационных сетей глобального масштаба, некоторые из которых являются полностью автоматизированными и требуют лишь наблюдения за бесперебойным функционированием.

В сфере компьютерных технологий такими информационными системами являются написанные людьми программы, которые решают возникающие задачи больших объемов в очень короткие сроки, тем самым облегчая труд работников [1, 5, 15].

Перед нами возникла задача: создать из файлов формата «.pdf» список в виде гиперссылок. Все файлы, которые сможет открыть пользователь, находятся в файловом пространстве внутреннего сайта Арзамасского филиала ННГУ: <http://ep.aspi.edu.ru/>. Для того, чтобы реализовать список, необходимо из панели администратора сайтом создать новые страницы, на которых будут

находиться все списки. После чего на каждой из них вписать код на языке разметки HTML. Ввиду того, что количество файлов было достаточно велико (порядка 2500 экземпляров), для создания ссылок можно использовать визуальный редактор HTML. Для получения одной ссылки в нем необходимо выполнить алгоритм:

1. Копируем полное название файла (вместе с расширением).

2. В визуальном редакторе нажимаем кнопку «вставить гиперссылку».

3. В поле «Текст» вставляем название файла и убираем расширение.

4. В поле «Ссылка» вставляем название файла, не удаляя расширения, и добавляем перед ним путь директории, в которой он находится, и нажимаем кнопку «Вставить» [6, 11, 15].

После выполнения данного алгоритма мы видим на странице сайта гиперссылку, при нажатии на которую в браузере открывается новая вкладка, и в ней пользователь видит PDF-файл, на который сделана эта ссылка. На выполнение данного алгоритма вручную уходит примерно 15–20 секунд. Если принять во внимание, что работа однообразная, уже через 10–15 минут человек, выполняющий ее, начинает путаться в алгоритме, из-за чего время создания каждой ссылки увеличивается вдвое. При самом хорошем раскладе, если на создание каждой ссылки будет уходить по 15 секунд, то на создание 2500 ссылок потребуется примерно 10,5 часов непрерывной рабо-

ты. В реальности на создание всех ссылок ушло бы 4–5 рабочих дней. Поэтому было принято решение создать программу, которая смогла бы взять на себя больший объем работы и позволила бы сократить время создания всех списков в несколько раз. Программе было дано название List of Hyperlinks Helper.

Материалы и методы исследования

Перед использованием программы List of Hyperlinks Helper пользователю необходимо подготовить текстовый файл, в котором построчно записаны полные названия файлов, предназначенных для публикации на сайте. Это можно сделать довольно быстро с помощью программы файловых менеджеров. После этой процедуры пользователю станет доступен файл формата «.txt». Далее пользователю можно запустить и использовать программу.

Описание программы

Программа запрашивает 3 поля, необходимые к заполнению: WAY, IN и OUT. Самое верхнее поле «WAY» – это путь к директории файлов, которые будут отображать в виде списка на странице сайта. В эту строку необходимо ввести путь к папке, в которой находятся нужные файлы формата «pdf». Поле «IN» – это путь к файлу, который пользователь должен был подготовить перед запуском. Поле «OUT» – это путь для сохранения конечного результата. В последнем поле можно выбрать конкретный файл, в который будет записан результат, либо же выбрать директорию, в которой программа создаст файл «out.txt» и запишет в него результат. А с помощью кнопки «Show OUT File» пользователь может просмотреть результат работы программы.

При запуске программы выполняется следующий участок кода:

```
int i = 0; // Объявлена локальная переменная, доступная всем методам
public MainWindow()
{
    InitializeComponent();
    textBox3.Text = Directory.GetCurrentDirectory() + «\out.txt»;
    textBox2.IsEnabled = false; // Отключает редактирование поля «IN»
    textBox3.IsEnabled = false; // Отключает редактирование поля «OUT»

    if (File.Exists(«in.txt») == false) // Проверяет наличие файла «in.txt»
    { // Если файл не найден, то:
        button4.IsEnabled = false; // Отключить кнопку «Result»
        textBox4.Text += «File doesn't find» + «\n»; // Вывести сообщение об ошибке
    }
    else // Иначе, если файл найден
    {
        textBox4.Text += «File find: « + System.IO.Path.GetFileName(textBox2.Text) + «\n»;
        // Сообщить об этом
        textBox2.Text = Directory.GetCurrentDirectory() + «\in.txt»;
        // Вывести путь к файлу в поле «IN»
    }
}
Кнопка «IN».
При нажатии на эту кнопку программа вызывает метод «ofd_in»:
public void ofd_in(object sender, RoutedEventArgs e)
{
    Microsoft.Win32.OpenFileDialog dlg = new Microsoft.Win32.OpenFileDialog();
```

```

dlg.CheckFileExists = true; // Проверка наличия файла
dlg.DefaultExt = «.txt»; // Расширения, доступные по умолчанию
dlg.Filter = «Text documents (.txt)|*.txt»; // Фильтр, отсеивающий .txt -файлы
Nullable<bool> res = dlg.ShowDialog();
if (res == true) // Проверка, выбран ли файл
{
    textBox2.Text = dlg.FileName;
    button4.IsEnabled = true; // Если файл выбран, включить кнопку «Result»
    textBox4.Text += «Selected file: « + System.IO.Path.GetFileName(textBox2.Text) + «\n»;
    // Сообщение о том, какой файл выбран
}
}

```

Кнопка «OUT».

При нажатии на эту кнопку программа вызывает метод «ofd_out»:

```

public void ofd_out(object sender, RoutedEventArgs e)
{
    Microsoft.Win32.SaveFileDialog dlg = new Microsoft.Win32.SaveFileDialog();
    dlg.DefaultExt = «.txt»; // Программа задает разрешение конечного файла
    dlg.FileName = «out»; // Название конечного файла по умолчанию
    Nullable<bool> res = dlg.ShowDialog();
    if (res == true) // Проверка, выбран ли путь конечного файла
    {
        textBox3.Text = dlg.FileName; // Выписывает путь к файлу в строку «OUT»
    }
    textBox4.Text += «Selected directory (OUT)» + «\n»; // Выводит сообщение в диалоговое окно
}

```

Кнопка «Show Out File»:

При нажатии на эту кнопку программа вызывает метод «show»

```

public void show(object sender, RoutedEventArgs e)
{
    Process.Start(«notepad++.exe», textBox3.Text); // Открыть файл из поля «OUT» с помощью
    // программы Notepad++
}

```

Кнопка «Clear».

При нажатии на эту кнопку программа выполняет метод «clear»:

```

private void clear(object sender, RoutedEventArgs e)
{
    textBox1.Text = «»;
    textBox2.Text = «»; // Задает полям «WAY», «IN» и «OUT» пустые значения
    textBox3.Text = «»;
    textBox4.Text += «Pressed \»Clear\»» + «\n»; // Сообщает о нажатии в диалоговое окно
}

```

Кнопка «?» (справка).

При нажатии на эту кнопку программа вызывает метод «help»:

```

public void help(object sender, RoutedEventArgs e)
{
    Process.Start(«WinHlp32.exe», str); // С помощью программы «Справка Windows» открывает справку
    // для программы. Здесь «str» - переменная типа «string», которой присвоен путь к файлу справки («help.chm»);
}

```

Кнопка «Result»

При нажатии на эту кнопку программа вызывает метод «result»:

```

public void result(object sender, RoutedEventArgs e)
{
    // Условия для запуска программы, если какое-либо не выполнено, выводится
    // соответствующая ошибка
    if (string.IsNullOrWhiteSpace(textBox1.Text))
    {
        textBox4.Text += «Way is failed, please write it» + «\n»;
        return;
    }
    if (string.IsNullOrWhiteSpace(textBox2.Text) || !File.Exists(textBox2.Text))
    {
        textBox4.Text += «File isn't selected, please press \»IN\»» + «\n»;
        return;
    }
}

```

```

if (string.IsNullOrEmpty(textBox3.Text) || !File.Exists(textBox2.Text))
{
    textBox4.Text += «Directory isn't selected, please press \»OUT\»» + «\n»;
    return;
}
string line = «», // Объявление необходимых переменных и присваивание значений
a = «<p><a href=\»», // Первая часть будущей ссылки
b = «\» target=\»_blank\»», // Вторая часть
c = «</a></p>»; // Третья часть
TextReader tr = new StreamReader(textBox2.Text, Encoding.UTF8);
List<string> str = new List<string>();
using (TextWriter sw = new StreamWriter(textBox3.Text, false, Encoding.UTF8))
{
    while ((line = tr.ReadLine()) != null)
    { // Если считанная из файла строка не пустая, то выполнить
        str.Add(a + textBox1.Text + line + b + line.Remove(line.Length - 4) + c);
        // Создание конечной ссылки
        sw.WriteLine(str[i]); // Запись ссылки в новую строку конечного файла
        i++;
    }
    textBox4.Text += «Complete!» + «\n»; // Сообщение об успешном выполнении программы
}
}

```

Из программного кода видно, что программа проста в устройстве и имеет всего одну главную функцию – создание файлов с гиперссылками, которые можно использовать на различных сайтах [11, 13, 14, 16].

Результаты исследования и их обсуждения

Но, несмотря на всю простоту программы, в ней существуют незначительные недостатки, которые в свою очередь не уменьшают значимость программы:

- Неправильное чтение символов русского алфавита. Устраняется проблема очень просто: пользователю необходимо пересохранить исходный файл (in.txt) в кодировке UTF-8, тогда программа правильно считает все внутренние символы. Среднее затрачиваемое время: 10 секунд. Нюанс: если в исходном файле находятся названия файлов, в которых присутствуют только цифры и символы латинского алфавита, то процедура пересохранения файла в другой кодировке не требуется.

- Пока что программа работает только с одним форматом файлов – «.pdf». В дальнейшем пользователи смогут вписывать свой формат файлов, с которым они будут работать. Для реализации будет введено поле «Формат файлов», в которое пользователь будет передавать программе строку с расширением, например «. doc», «.djvu» и т.п., а если формат не будет указан, то по умолчанию программа будет работать с расширением «.pdf».

- Отсутствует возможность работы с несколькими файлами. Если предоставить пользователю возможность выбрать не-

сколько файлов сразу, то это компенсирует временные затраты на пересохранение исходного файла, что положительно скажется на работе в целом.

- Отсутствует возможность автоматического создания входного файла «in.txt», что влияет на скорость достижения конечного результата.

Но все эти недоработки не влияют на работоспособность и в скором времени будут устранены. Планируемые модификации программы:

1. Исправление недочетов программного кода: исправление ошибок в работе программы, стабилизация кода, корректировка, внедрение более действенных алгоритмов исполнения основной функции программы.

2. Внедрение новых кнопок управления выходным файлом, в частности, планируется ввести кнопку «Options» («Опции»), при нажатии на которую пользователю будет открываться окно, в котором у него появится возможность внести некоторые визуальные изменения в конечные ссылки (такие как цвет, шрифт, размер шрифта). А также появится возможность изменить текст гиперссылки с названия файлов (по умолчанию) на нумерацию с указанного номера, и некоторые другие возможности.

3. Создание другого приложения на основании разработанной программы, использование описанных методов в схожих по функциональности приложениях.

4. Создание справки для пользователя. Работа по созданию этого элемента уже началась, но полноценной справки, которая будет полезна пользователю, пока создать

не удалось. Будущая структура данного элемента представляется таковой: на главной странице справки пользователю будут доступны ссылки на описание каждого элемента программы. При нажатии на одну из ссылок, пользователь перейдет к описанию этого элемента, какую функцию этот элемент выполняет, и какой участок кода или метод выполняется при его активации.

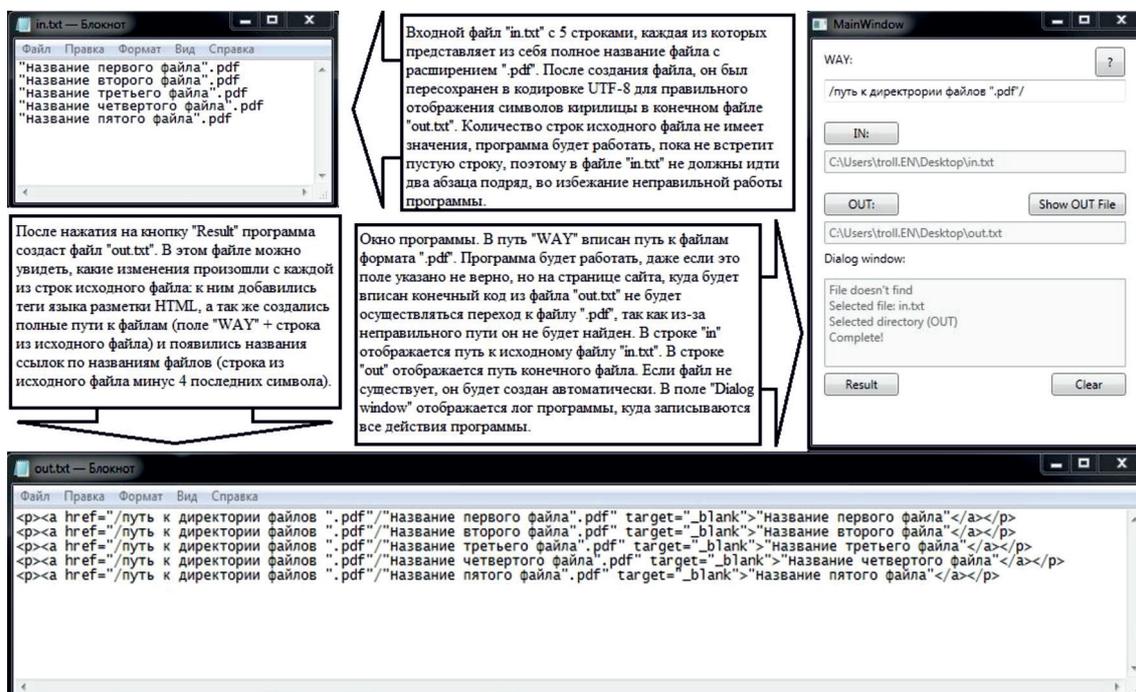
5. Будет встроена возможность работать сразу с несколькими файлами. Предполагаемый алгоритм внедрения данной функции и изменения в программе: 1. Удаление текстового поля «IN» и замена его на счетчик выбранных пользователем файлов; 2. Описание метода, запоминающего пути к выбранным файлам и добавляющим их в массив класса «list»; 3. Замена некоторых констант на переменные соответствующих типов, которые будут изменяться для каждого файла; 4. Изменение функций некоторых кнопок программы в соответствии с новыми ее возможностями (например, кнопка Show OUT File не будет открывать конечный файл, а будет открывать директиву, где будут находиться все конечные файлы).

6. Планируется создать дополнение, которое будет автоматически создавать исходный файл «in.txt», и встроить его в программу List of Hyperlinks Helper. Предполагаемый план разработки: в окне

программы появится кнопка «Create «in.txt»». При нажатии на эту кнопку пользователь должен будет выбрать директорию, где находятся файлы, предназначенные для одной из страниц сайта. После чего нажимает «Ok», и все названия файлов, находящихся в выбранной директории, записываются в файл «in.txt».

7. Программе будет дана возможность считывать системное время и приписывать его к сообщениям, выводимым пользователю. Таким образом, пользователь сможет узнать, за какой период времени он выполняет работу и, например, заполняет одну страницу сайта [2, 7, 12, 18, 20].

Внедрение подобных изменений напрямую повлияет на интерфейс программы, что повлечёт за собой новый дизайн. Поменяется расположение кнопок, расположение текстовых окон программы, будут изменены некоторые функции кнопок и переписана справка, функциональность программы повысится. Написание новых методов (статических) позволит общаться с пользователем через окно программы «Dialog window», что позволит упростить понимание ее пользователями и даст возможность использовать программу без просмотра справки и прочтения описания, что для многих является большим плюсом [3, 6, 17].



Скриншот с описанием работы программы

Для наглядного представления работы программы продемонстрирован скриншот с описанием работы программы: как должно выглядеть содержание файла «in.txt», как выглядит интерфейс программы на данный момент, и как будет выглядеть выходной файл «out.txt» в зависимости от содержания исходного файла. А также на скриншоте представлено краткое описание действий, которые должен выполнить пользователь, и что после этого будет (рисунок).

Заключение

Автоматизация рутинной работы – важный шаг в развитии человеческого общества. Именно поэтому с изобретением компьютеров более популярными профессиями стали программист и системный администратор. Программисты автоматизируют работу, связанную с компьютерами, то есть все области работы человечества, так как компьютеры используются сейчас везде. А системные администраторы способны правильно внедрить в информационные системы, которые они поддерживают, написанные программистами приложения автоматизации деятельности.

Автоматизация таких, казалось бы, незначительных задач, как автоматическое создание ссылок, – это еще один шаг на пути к созданию компьютерного сознания, которое будет неотличимо от человеческого, это еще один шаг на пути развития человеческого общества.

В нашем случае время, потраченное на написание данной программы, оправдало себя: мы выполнили работу в более короткие сроки, чем если бы создавали каждую ссылку вручную. Кроме этого, планируется также использовать данное приложение в будущем для заполнения внутреннего сайта АФ ННГУ примерно 15 тысячами ссылок [9, 10].

Как видим, разработанный помощник в создании гиперссылок позволяет автоматизировать деятельность и способствует повышению эффективности работы.

Список литературы

1. Белов В.Н., Ковалев А.И. Некоторые аспекты использования электронных ключей в подходах защиты информации // Математический вестник педвузов и университетов Волго-Вятского региона. – 2015. – № 17. – С. 318–325.
2. Воронина В.В. Типовые алгоритмы и их реализация на языке С# // Методические указания для проведения учебной практики для студентов направления «Прикладная информатика», УлГТУ. – Ульяновск, 2013.
3. Записная книжка программиста-новичка, С#, SQL, PHP и все-все-все // Самоучитель по С# для начинающих. – [Электронный ресурс] URL: <http://nullpro.info/2013/samouchitel-po-c-dlya-nachinayushhix-01-osnovy-yazyka-peremennye-logika-cikly/> (дата обращения: 22.07.2016).

4. Застела М.Ю., Ямпурин Н.П. Защита информации в сетях передачи данных // учебное пособие для студентов высших учебных заведений: [в 2 ч.]. Арзамасский филиал ННГУ им. Н.И. Лобачевского. – Арзамас, 2012.

5. Ковалев А.И. Защита пользователей OS Windows с помощью USB-flash накопителей // Информационные технологии и прикладная математика. Межвузовский сборник аспирантских и студенческих научных работ. – Арзамас, 2016. – № 6. – С. 75–82.

6. Напалков С.В., Сазанов А.А., Широков Л.В. // Web-комплексы и их приложения // В сборнике: Web-технологии в образовательном пространстве: проблемы, подходы, перспективы. Сборник статей участников Международной научно-практической конференции. Под общей редакцией С.В. Арюткиной, С.В. Напалкова. – 2015. – С. 125–130.

7. Парадеев Д.С. Методы разработки программного обеспечения // Исследования в области естественных наук. – 2014. – № 10 (34). – С. 25–27.

8. Понимание XAML [Электронный ресурс]; URL: <http://habrhabr.ru/post/141069> (дата обращения: 22.07.2016).

9. Сазанов А.А. Исследование многосерверной корпоративной информационной системы // Информационные технологии и прикладная математика. Межвузовский сборник студенческих и аспирантских работ. Арзамасский филиал ННГУ. – 2014. – С. 112–119.

10. Трухманов В.Б. Математические модели в экономике и их анализ с помощью компьютерных средств // В сборнике: Педагогические технологии математического творчества. Сборник статей участников международной научно-практической конференции. Редакционная коллегия: под общей редакцией М.И. Зайкина, С.В. Арюткина (ответственный редактор), С.В. Напалков, Т.В. Романова. – 2011. – С. 338–342.

11. Фишки XAML-разработчика: динамический Grid. – [Электронный ресурс] URL: <http://habrhabr.ru/post/276077/> (дата обращения: 22.07.2016).

12. Хабрхабр Сортировка в .NET. – [Электронный ресурс] URL: <https://habrhabr.ru/post/188012/> (дата обращения: 22.07.2016).

13. Широков Л.В., Яблонский Д.В., Глухова А.Ф. Дифференциальные уравнения и уравнения в частных производных // Учебное пособие для студентов вузов. Государственное образовательное учреждение высшего профессионального образования «Арзамасский государственный педагогический институт им. А.П. Гайдара». – Арзамас, 2006.

14. EasyWebSkipts.net – Веб-программирование [Электронный ресурс]. – JavaScript Slider // URL: <http://easywebscripts.net/javascript/slider.php> (дата обращения: 22.07.2016).

15. Htmlbook – Самоучитель по HTML [Электронный ресурс]; URL: <http://htmlbook.ru/> (дата обращения: 22.07.2016).

16. MSDN Library // Практическое руководство. Создание приложения WPF C# // URL: [http://msdn.microsoft.com/ru-ru/library/bb655895\(v=vs.90\).aspx](http://msdn.microsoft.com/ru-ru/library/bb655895(v=vs.90).aspx) (дата обращения: 22.07.2016).

17. MSDN Library // Руководство по программированию на C#. [Электронный ресурс] URL: <https://msdn.microsoft.com/ru-ru/library/67ef8sbd.aspx> [дата обращения: 30.06.2016].

18. Wikipedia – Свободная энциклопедия // XAML. – [Электронный ресурс] URL: <http://ru.wikipedia.org/wiki/XAML> (дата обращения: 22.07.2016).

19. Wikipedia – Свободная энциклопедия // Автоматизация. – [Электронный ресурс] URL: <https://ru.wikipedia.org/wiki/Автоматизация> (дата обращения: 22.07.2016).

20. Wikipedia – Свободная энциклопедия [Электронный ресурс] – C Sharp // URL: https://ru.wikipedia.org/wiki/C_Sharp (дата обращения: 22.07.2016).

References

1. Belov V.N., Kovalev A.I. Nekotorye aspekty ispol'zovaniya jelektronnyh kljuচেj v podhodah zashhity informacii // Matematicheskij vestnik pedvuzov i universitetov Volgo-Vjatskogo regiona. 2015. no. 17. pp. 318–325.

2. Voronina V.V. Tipovye algoritmy i ih realizacija na jazyke C# // Metodicheskie ukaza-nija dlja provedenija uchebnoj praktiki dlja studentov napravlenija «Prikladnaja informatika», UIGTU. Uljanovsk, 2013.
3. Zapisnaja knizhka programista-novichka, C#, SQL, PHP i vse-vse-vse // Samouchitel po C# dlja nachinajushih. [Elektronnyj resurs] URL: <http://nullpro.info/2013/samouchitel-po-c-dlya-nachinayushix-01-osnovy-yazyka-peremennye-logika-cikly/> (data obrashhenija: 22.07.2016).
4. Zastela M.Ju., Jampurin N.P. Zashhita informacii v setjah peredachi dannyh // uchebnoe posobie dlja studentov vysshih uchebnyh zavedenij: [v 2 ch.]. Arzamasskij filial NNGU im. N.I. Lobachevskogo. Arzamas, 2012.
5. Kovalev A.I. Zashhita polzovatelej OS Windows s pomoshhju USB-flash nakopitelej // Informacionnye tehnologii i prikladnaja matematika. Mezhhuzovskij sbornik aspirantskih i studencheskih nauchnyh rabot. Arzamas, 2016. no. 6. pp. 75–82.
6. Napalkov S.V., Sazanov A.A., Shirokov L.V. // Web-kompleksy i ih prilozhenija // V sbornike: Web-tehnologii v obrazovatelnom prostranstve: problemy, podhody, perspektivy. Sbornik statej uchastnikov Mezhdunarodnoj nauchno-prakticheskoj konferencii. Pod obshej redakciej S.V. Arjutkinoj, S.V. Napalkova. 2015. pp. 125–130.
7. Paradeev D.S. Metody razrabotki programmnogo obezpechenija // Issledovanija v oblasti estestvennyh nauk. 2014. no. 10 (34). pp. 25–27.
8. Ponimanie XAML [Elektronnyj resurs]; URL: <http://habrhabr.ru/post/141069> (data obrashhenija: 22.07.2016).
9. Sazanov A.A. Issledovanie mnogoservernoj korporativnoj informacionnoj sistemy // Informacionnye tehnologii i prikladnaja matematika. Mezhhuzovskij sbornik studencheskih i aspirantskih rabot. Arzamasskij filial NNGU. 2014. pp. 112–119.
10. Truhmanov V.B. Matematicheskie modeli v jekonomike i ih analiz s pomoshhju kompjuternyh sredstv // V sbornike: Pedagogicheskie tehnologii matematicheskogo tvorcestva sbornik statej uchastnikov mezhdunarodnoj nauchno-prakticheskoj konferencii. Redakcionnaja kollegija: pod obshej redakciej M.I. Zajkina, S.V. Arjutkina (otvetstvennyj redaktor), S.V. Napalkov, T.V. Romanova. 2011. pp. 338–342.
11. Fishki XAML-razrabotchika: dinamicheskij Grid. [Elektronnyj resurs] URL: <http://habrhabr.ru/post/276077/> (data obrashhenija: 22.07.2016).
12. Habrhabr Sortirovka v .NET. [Elektronnyj resurs] URL: <https://habrhabr.ru/post/188012/> (data obrashhenija: 22.07.2016).
13. Shirokov L.V., Jablonskij D.V., Gluhova A.F. Differencialnye uravnenija i uravnenija v chastnyh proizvodnyh // Uchebnoe posobie dlja studentov vuzov. Gosudarstvennoe obrazovatelnoe uchrezhdenie vysshego professionalnogo obrazovanija «Arzamasskij gosudarstvennyj pedagogicheskij institut im. A.P. Gajdara». Arzamas, 2006.
14. EasyWebSkripts.net Veb-programmirovanie [Elektronnyj resurs]. JavaScript Slider // URL: <http://easywebscripts.net/javascript/slider.php> (data obrashhenija: 22.07.2016).
15. Htmlbook Samouchitel po HTML [Elektronnyj resurs]; URL: <http://htmlbook.ru/> (data obrashhenija: 22.07.2016).
16. MSDN Library // Prakticheskoe rukovodstvo. Sozdanie prilozhenija WPF C# // URL: [http://msdn.microsoft.com/ru-ru/library/bb655895\(v=vs.90\).aspx](http://msdn.microsoft.com/ru-ru/library/bb655895(v=vs.90).aspx) (data obrashhenija: 22.07.2016).
17. MSDN Library // Rukovodstvo po programmirovaniu na C#. [Elektronnyj resurs] URL: <https://msdn.microsoft.com/ru-ru/library/67ef8sbd.aspx> [data obrashhenija: 30.06.2016].
18. Wikipedia Svobodnaja jenciklopedija // XAML. [Elektronnyj resurs] URL: <http://ru.wikipedia.org/wiki/XAML> (data obrashhenija: 22.07.2016).
19. Wikipedia Svobodnaja jenciklopedija // Avtomatizacija. [Elektronnyj resurs] URL: <https://ru.wikipedia.org/wiki/Автоматизация> (data obrashhenija: 22.07.2016).
20. Wikipedia Svobodnaja jenciklopedija [Elektronnyj resurs] C Sharp // URL: https://ru.wikipedia.org/wiki/C_Sharp (data obrashhenija: 22.07.2016).