

## ИНФОРМАТИКА

*Вестник Сыктывкарского университета.  
Серия 1: Математика. Механика. Информатика.  
Выпуск 2 (35). 2020*

УДК 004

### РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ПРОВЕРКИ ССЫЛОК НА ЭЛЕКТРОННЫЕ ИЗДАНИЯ В РАБОЧИХ ПРОГРАММАХ ДИСЦИПЛИН

*Д. А. Ушаков*

В статье описывается разработка программного продукта для автоматизации процесса проверки корректности ссылок на электронные библиотечные системы в рабочих программах учебных дисциплин.

*Ключевые слова:* рабочая программа учебной дисциплины, электронные библиотечные системы, автоматизация, алгоритм.

#### **Введение**

Информатизация образования и развитие информационных систем вузов, помогающих стандартизировать документацию, улучшить организацию внутренних бизнес-процессов и взаимодействие различных подразделений и структур, улучшить сервисный подход к предоставлению услуг преподавателям и студентам, являются крайне интересными и востребованными с практической точки зрения вопросами. В СГУ им. Питирима Сорокина давно ведется разноплановая работа в этом направлении, что представлено, например, в работах [1–3].

Представленная работа посвящена аспектам разработки и совершенствования электронной информационной образовательной среды (ЭИОС) вуза, что является чрезвычайно актуальной задачей на современном этапе развития системы образования.

В соответствии с Федеральным государственным образовательным стандартом и другими регламентирующими документами, включая внутренние нормативные документы вуза, на каждую учебную дисциплину должна ежегодно составляться рабочая программа дисциплины (РПД). Ее структура и содержание определяются рядом документов, включая локальные документы вуза. Ввиду зависимости от локальных

актов вузов и отсутствия приемлемого универсального отраслевого решения автоматизацией процесса составления РПД практически все вузы занимаются самостоятельно, о чем свидетельствует множество предлагаемых моделей и решений, например, в работах [4–6].

В данной работе рассматривается только один из аспектов такой автоматизации. РПД должна содержать, кроме прочего, список основной и дополнительной литературы, включающий в себя как книги на бумажном носителе, так и в электронном виде. Во многих вузах разрешается приводить ссылки на электронные библиотечные системы (ЭБС), круг которых может быть ограничен имеющимися у вуза договорами. Так как создание РПД выполняется преподавателями вручную, то он связан с возможным возникновением ошибок, что обуславливает необходимость проверки корректности списка литературы в РПД. Если проблем с проверкой актуальности книг на бумажном носителе не возникает, то книги из ЭБС могут стать недоступны через некоторое время (например, из-за изменений условий договора с издательством или автором, окончания срока подписки и т. д.). Поэтому возникает необходимость проверки доступности для чтения этих изданий. Такая проверка выполняется оператором вручную. В общем виде алгоритм актуализации может выглядеть следующим образом:

1. Открыть файл РПД.
2. Найти соответствующий раздел, содержащий ссылки (например, «Материально-техническое и учебно-методическое обеспечение дисциплины (модуля)»).
3. Перейти по ссылке из РПД на страницу книги в ЭБС и убедиться в ее корректности или отсутствии издания.
4. Повторять пункт 3, пока не будут проверены все ссылки раздела.

Как видно, такой алгоритм достаточно прост и не вызывает затруднений у сотрудника при его реализации. Но необходимо учитывать большие временные затраты на этот процесс. Если предположить, что один файл РПД будет обрабатываться хотя бы 3 минуты (а как показывает практика, этот процесс занимает гораздо больше времени), то на достаточно большой объем (например, 5000 РПД) может потребоваться порядка 250 часов рабочего времени, что соответствует 30 полным рабочим дням. Это привело к возникновению решения об автоматизации данного процесса путем разработки программы для проверки актуальности электронных изданий в РПД, которая снизила бы нагрузку на сотрудников и существенно уменьшила время проверки РПД. Ее главными функциями должны быть проверка РПД на предмет присутствия некорректных ссылок на ЭБС или их полного отсутствия, а также фор-

мирование и вывод списка таких РПД.

Для достижения этой цели необходимо решить следующие задачи:

1. Определить признаки доступности книги для чтения.
2. Разработать алгоритм проверки ссылок на ЭБС.
3. Разработать удобный способ предоставления отчета о проверке.

В данной работе демонстрируется пример создания рабочего модуля (прототипа) программы, выполняющей вышеперечисленные функции, возможность реализации подобного алгоритма и оценка его преимущества на примере проверки ссылок на одну из популярных ЭБС «Университетская библиотека онлайн» (<http://biblioclub.ru/>).

### Признаки доступности книги для чтения

В первую очередь возникает вопрос о том, как определить, что книга доступна для чтения. Это одна из главных задач, которую должна решать предлагаемая программа.

При использовании ЭБС «Университетская библиотека онлайн» книга доступна для чтения только в том случае, если на странице ее библиографического описания есть кнопка «Читать онлайн». Для других ЭБС признак доступности будет другим. Рассмотрим это на примере (рис. 1, 2).

На рис. 1 приведен фрагмент страницы сайта ЭБС с примером книги, в описании которой присутствует кнопка «Читать онлайн», то есть книга доступна для чтения. Напротив, на рис. 2 эта кнопка отсутствует, значит, просмотреть ее содержимое не получится. Такая книга должна быть исключена или заменена в списке литературы в РПД.

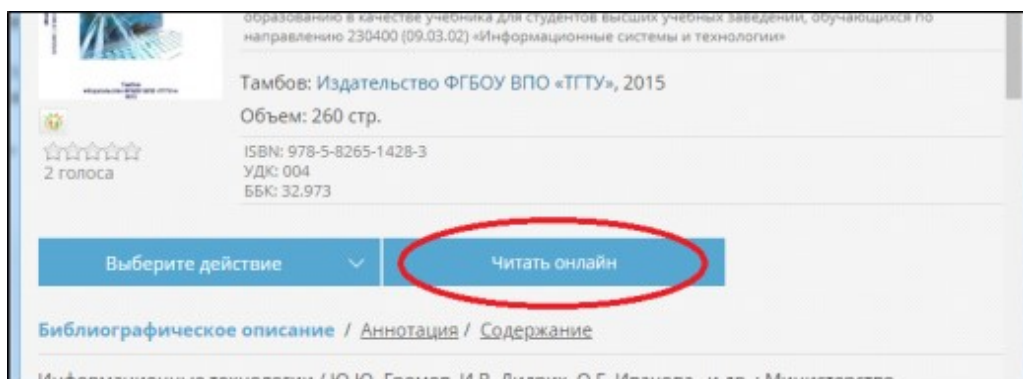


Рис. 1. Книга доступна для чтения

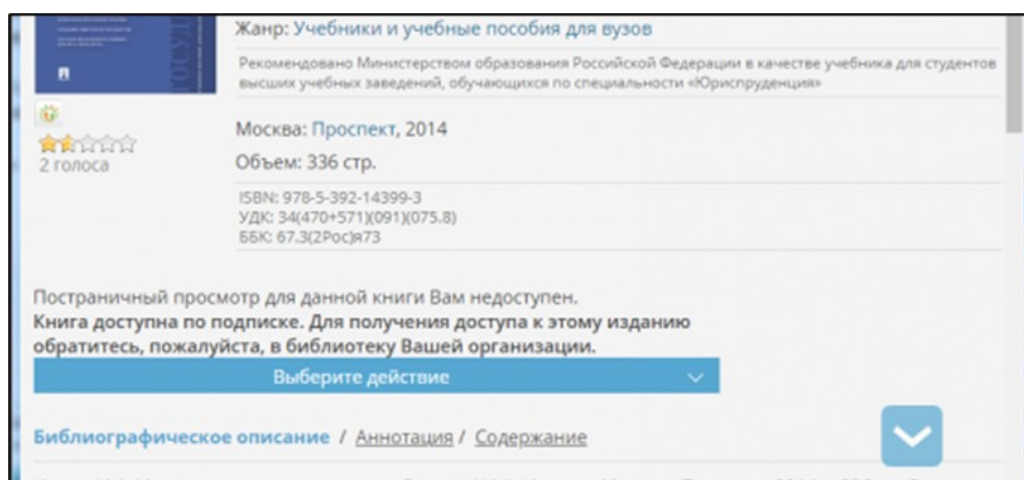


Рис. 2. Книга недоступна для чтения

### Алгоритм проверки ссылок на ЭБС

В общем виде алгоритм, реализованный с использованием языка C# на платформе .NET Framework, будет таким же, как описано выше во введении, но с некоторыми изменениями:

1. Открыть файл РПД.
2. Проанализировать его содержимое на наличие ссылок на ЭБС.
3. Отнести ссылку к той или иной ЭБС (идентифицировать ЭБС).
4. Проверить ссылку путем отправки запроса на сервер и обработки его ответа на предмет наличия признаков доступности книги для чтения.
5. Выполнять пункт 1 до тех пор, пока не будут обработаны все файлы РПД.
6. Сформировать и вывести отчет о результатах проверки.

Требования пункта 4 обусловлены тем, что существует множество ЭБС, которые могут быть использованы в РПД. Это приводит к необходимости поиска признаков, обозначенных в предыдущем разделе, для каждой ЭБС.

Выбор платформы .NET Framework объясняется, во-первых, большими возможностями ее стандартных библиотек, во-вторых, относительной легкостью моделирования графического интерфейса пользователя и, в-третьих, отсутствием необходимости установки дополнительных зависимостей на компьютере пользователя.

Как правило, РПД представлены в виде файлов в формате .doc или .docx, реже в формате pdf, поэтому в данной статье для демонстрации работоспособности предлагаемого метода остановимся только на

этих форматах. Существуют два способа чтения содержимого файла таких форматов: COM Interop [7] и использование сторонних библиотек. Первый вариант кажется наиболее предпочтительным в данном случае, так как становится возможной работа с файлами формата .doc и .docx без дополнительных библиотек. Тем не менее, он требует наличия установленного пакета Microsoft Office. Второй вариант предполагает использование внешних библиотек. Примером таких библиотек для бинарного формата .doc могут служить Spire DOC [7] и портированная на .NET Framework Apache POI [8]. Следует учитывать, что расширенные возможности данных библиотек, как правило, являются платными, а базовых может не хватить для комфортной обработки множества файлов. Для работы с файлами формата .docx, который представляет собой архив с набором XML-файлов, также существует библиотека OpenXML [10]. Однако такой подход связан с определенными дополнительными действиями: необходимо определять формат исходного файла РПД и применять к нему функции из той или иной библиотеки, что делает код приложения громоздким и сложным для отладки. Учитывая все вышесказанное, автором было решено осуществлять доступ к файлам с помощью COM-интерфейса Microsoft Office.

Ссылки на электронные книги извлекаются из раздела «Библиографическое описание» и в общем виде выглядят так: «<http://biblioclub.ru/index.php?page=book&id=xxxxxx>», где «xxxxxx» заменяется соответствующим внутренним идентификационным номером книги. Чтобы получить HTML-разметку страницы с библиографическим описанием книги программно, нужно отправить запрос на сервер и обработать ответ. Для отправки запросов на сервер и приема ответов применяется стандартный класс `HttpRequest`. Ответ приходит в формате HTML в виде строки и нуждается в дополнительной обработке, в ходе которой производится поиск ссылок по шаблону с применением методов стандартной библиотеки. В ходе такой проверки производится поиск HTML-тега с атрибутом «`class="button button-green button-read"`», соответствующим описанному выше признаку доступности книги.

Такие атрибуты а также вид искомого адреса не должны быть зафиксированы в исходном коде конечной версии программы, поскольку могут меняться с течением времени. Поэтому существует необходимость предусмотреть простое добавление, удаление и изменение используемых ЭБС и их атрибутов без перекомпиляции всей программы. С этой целью должен быть предусмотрен интерфейс настройки программы (например, в виде конфигурационного файла, редактируемого напрямую, или графического интерфейса).

### Предоставление отчета о проверке

Важным элементом разработанной программы является отчет по результатам проверки. Он должен предоставляться оператору в понятной для него форме. Результат выводится на экран (рис. 3) с помощью компонента RichTextBox и его содержимое дублируется в текстовый лог-файл, хранящийся в директории с исполняемым файлом программы. На каждую проверку может создаваться отдельный файл лога. В его названии содержатся дата и время начала проверки.

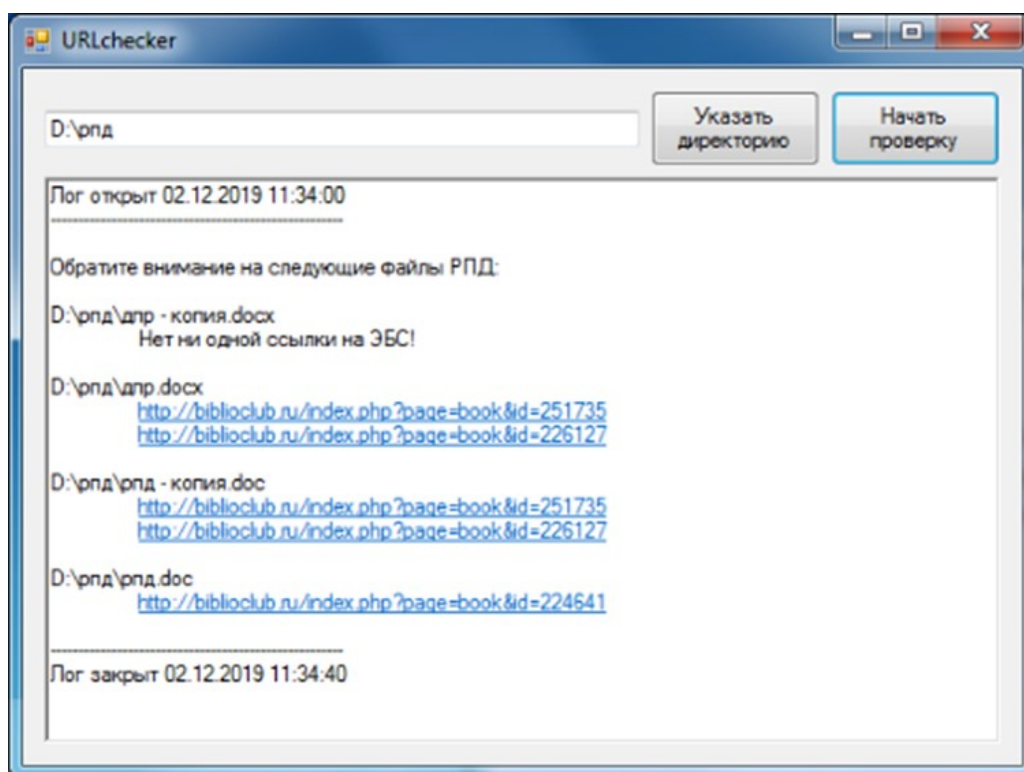


Рис. 3. Окно программы после завершения проверки

Перед проверкой программа выводит сообщение об открытии лога с указанием даты и времени. Далее поочередно выводятся названия файлов РПД, где обнаружены недоступные для чтения книги, с указанием ссылок на них. Все ссылки, выводимые в главном окне, являются активными. Это значит, что при нажатии такая ссылка будет открыта в браузере по умолчанию с целью дальнейшего ее анализа оператором. Кроме того, производится проверка на полное отсутствие ссылок на ЭБС. Проверка завершается указанием даты и времени ее окончания.

### Заключение

Таким образом, представленный программный модуль для проверки актуальности электронных изданий в РПД автоматизирует рутинный и затратный по времени процесс. Благодаря ее использованию на проверку каждой РПД тратится в среднем около 20 секунд, что составит примерно 27 часов непрерывной работы программы при объеме 5000 РПД. Это чуть больше одного рабочего дня, то есть затраты времени сократятся более чем в 30 раз (с учетом выходных дней). Точное время может варьироваться в зависимости от пропускной способности канала связи, количества ссылок в РПД и загруженности серверов ЭБС.

Результаты, сформулированные в статье, могут найти применение не только при реализации системы разработки РПД, но и в других секторах ЭИОС, например проверки корректности ссылок в учебных курсах, что подтверждает востребованность такой системы.

Требования к разработанной программе:

- наличие .NET Framework 3.5 (уже установлен в Windows 7, Windows Server 2008 R2 и последующих версиях);
- наличие Microsoft Office 2007 и выше (для работы с форматом .docx).

Среди возможностей представленного модуля можно выделить следующие:

- работа с файлами РПД в формате .doc и .docx и других (при добавлении соответствующих механизмов доступа к содержимому файлов);
- проверка отсутствия ссылок на ЭБС в РПД;
- проверка доступности для чтения книг ЭБС «Университетская библиотека онлайн» по ссылке из РПД (список ЭБС также может быть расширен в будущем);
- предоставление отчета о проверке;
- долгосрочное хранение отчета о проверке.

## Список литературы

1. Гольчевский Ю. В., Сургуладзе А. М. Опыт разработки и эксплуатации системы онлайн-приема электронных заявлений от абитуриентов // *Открытое образование*. № 6(95). 2012. С. 57–63.
2. Бабенко В. В., Гольчевский Ю. В. Концепция информационного пространства кафедры вуза на основе Web-портала // *Открытое образование*. 2014. № 2(103). С. 46–50.

3. **Гольчевский Ю. В., Виноградов И. М.** Опыт разработки интернет-сервиса расписания учебных занятий // *Информатизация образования и науки*. 2016. № 1. С. 16–25.
4. **Шаламков С. А., Старичкова Ю. В.** Опыт разработки и внедрения модуля автоматизации процесса создания и утверждения программ учебных дисциплин в рамках информационной образовательной среды поддержки основного образовательного процесса // *Вестник Российского университета дружбы народов. Серия: Информатизация образования*. 2015. № 4. С. 67–76.
5. **Марухленко А. Л., Таныгин М. О., Марухленко И. И., Кононов А. С.** Автоматизация системы формирования рабочих программ учебных дисциплин // *Инфокоммуникации и космические технологии: состояние, проблемы и пути решения : материалы 1 Всероссийской научно-практической конференции : в 2 ч. / Юго-Западный государственный университет. Курск, 2017. Ч. 1. С. 202–208.*
6. **Сафонов Е. И., Чернавский А. Ю.** Автоматизация составления рабочих программ учебных дисциплин // *Вестник Югорского государственного университета*. 2019. № 2(53). С. 33–40.
7. COM Interop Part 1: C# Client Tutorial [Электронный ресурс]. URL: <https://msdn.microsoft.com/en-us/ie/aa645736%28v=vs.94%29?f=255&MSPPErr=-2147217396> (дата обращения: 19.04.2020).
8. NPOI [Электронный ресурс]. URL: <https://github.com/dotnetcore/NPOI> (дата обращения: 19.04.2020).
9. Spire.Doc for .NET [Электронный ресурс]. URL: [https://www.e-iceblue.com/Introduce/word-for-net-introduce.html#.Xe\\_QEpMzaUk](https://www.e-iceblue.com/Introduce/word-for-net-introduce.html#.Xe_QEpMzaUk) (дата обращения: 19.04.2020).
10. Страница пакета Open XML SDK 2.5 для Office [Электронный ресурс]. URL: <https://docs.microsoft.com/ru-ru/office/open-xml/open-xml-sdk> (дата обращения: 19.04.2020).



### Summary

**Ushakov D. A.** Development of software for checking links to electronic publications in work programs of academic subjects

The paper describes the automation of checking the links to electronic library systems correctness process in the work programs of academic subjects.

*Keywords: work program of academic subject, electronic library systems, automation, algorithm.*

### References

1. **Golchevskiy Yu. V., Surguladze A. M.** Opyt razrabotki i ekspluatatsii sistemy onlajn priema elektronnykh zayavlenij ot abiturientov (Developing and operating experience in online admission system for prospective university students), *Open Education*, no 6(95), 2012, pp. 57–63.
2. **Babenko V. V., Golchevskiy Yu. V.** Kontseptsiya informacionnogo prostranstva kafedry VUZa na osnove Web-portala (Concept of web-based university department information space), *Open Education*, no 2(103), 2014, pp. 46–50.
3. **Golchevskiy Yu. V., Vinogradov I. M.** Opyt razrabotki Internet-servisa raspisaniya uchebnykh zanyatij (Schedule internet-service developing experience), *Informatization of Education and Science*, no 1, 2016, pp. 16–25.
4. **Shalamkov S. A., Starichkova J. V.** Opyt razrabotki i vnedreniya modulya avtomatizatsii processa sozdaniya i utverzhdeniya programm uchebnykh disciplin v ramkah informacionnoj obrazovatel'noj sredy podderzhki osnovnogo obrazovatel'nogo processa (Experience in development and implementation of the module automate the process of creating and approving programs of study within the information educational environment of support of the basic educational process), *Bulletin of the Peoples' Friendship University of Russia. Series: Informatization of Education*, no 4, 2015, pp. 67–76.
5. **Maruhlenko A. L., Tanygin M. O., Maruhlenko I. I., Kononov A. S.** Avtomatizatsiya sistemy formirovaniya rabochikh programm uchebnykh distsiplin (Automation of the system of working programs of academic disciplines formation), *Infocommunications and space*

*technologies: state, problems and solutions. Materials of the 1st All-Russian Scientific and Practical Conference, Part 1, 2017, Southwestern State University, Kursk, pp. 202–208.*

6. **Safonov E. I., Chernavskiy A. Yu.** Avtomatizatsiya sostavleniya rabochikh programm uchebnykh distsiplin (Automation of compilation of working programs of educational disciplines), *Bulletin of the Yugra State University*, no 2 (53), 2019, pp. 33–40.
7. COM Interop Part 1: C# Client Tutorial. URL: <https://msdn.microsoft.com/en-us/ie/aa645736%28v=vs.94%29?f=255MSPPError=-2147217396> (date of the application: 19.04.2020).
8. NPOI. URL: <https://github.com/dotnetcore/NPOI> (date of the application: 19.04.2020).
9. Spire.Doc for .NET. URL: [https://www.e-iceblue.com/Introduce/word-for-net-introduce.html#.Xe\\_QEpMzaUk](https://www.e-iceblue.com/Introduce/word-for-net-introduce.html#.Xe_QEpMzaUk) (date of the application: 19.04.2020).
10. Open XML SDK 2.5 for Office. URL: <https://docs.microsoft.com/ru-ru/office/open-xml/open-xml-sdk> (date of the application: 19.04.2020).

**Для цитирования:** Ушаков Д. А. Разработка программного обеспечения для проверки ссылок на электронные издания в рабочих программах дисциплин // *Вестник Сыктывкарского университета. Сер. 1: Математика. Механика. Информатика. 2020. Вып. 2 (35). С. 49–58.*

**For citation:** Ushakov D. A. Development of software for checking links to electronic publications in work programs of academic subjects, *Bulletin of Syktyvkar University. Series 1: Mathematics. Mechanics. Informatics*, 2020, 2 (35), pp. 49–58.