

УДК 004.42

РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА ДЛЯ АВТОМАТИЗАЦИИ ПРОЦЕССА ПРОВЕРКИ ДОСТУПНОСТИ ИЗДАНИЙ В ЭБС

Пепельшев Д.И., Карелова Р.А.

*ФГАОУ ВО «Уральский федеральный университет имени первого Президента России Б.Н. Ельцина», Нижнетагильский технологический институт (филиал),
Нижний Тагил, e-mail: riya2003@mail.ru*

Информационное обеспечение образовательного процесса в вузе, в том числе укомплектованность его библиотечного фонда, подлежит аккредитационной экспертизе. Указанные в рабочих программах дисциплин электронные издания должны быть доступны студентам на момент изучения дисциплины. Если по каким-то причинам в программе указана литература, которая недоступна, это будет являться нарушением Федерального государственного образовательного стандарта. В статье предложен вариант реализации программного средства, позволяющего автоматизировать проверку доступности изданий в электронно-библиотечных системах, указанных в рабочей программе дисциплины. Описана логика работы такого средства, особенности и инструменты его реализации, приведены примеры листингов на языке Python, иллюстрации интерфейса программы, а также результатов ее работы. Полученное программное средство позволяет пользователю выбрать файл с рабочей программой дисциплины, осуществляет в нем поиск списка литературы, в котором идентифицирует ссылки на электронно-библиотечные системы, а затем проверяет возможность доступа к ним под логином и паролем организации. По результатам работы программы пользователь получает сообщение о том, по каким ссылкам издания доступны для чтения, а по каким – нет.

Ключевые слова: прикладное программирование, программный модуль, разработка программного обеспечения, издания в ЭБС, доступность электронных изданий, мониторинг электронных изданий, Python, PyQt, python-docx

SOFTWARE TOOL DEVELOPMENT FOR AUTOMATIC ACCESS CHECKING TO PUBLICATIONS IN ELECTRONIC LIBRARIES

Pepelyshev D.I., Karelova R.A.

*Ural Federal University named after the First President of Russia B.N. Yeltsin,
Nizhny Tagil Technological Institute, Nizhny Tagil, e-mail: riya2003@mail.ru*

The staffing of the university library fund is subject to an accreditation examination. The electronic editions indicated in the disciplines' work programs should be available to students at the time of the discipline studying. Lack of access to this literature is a violation of the Federal State Educational Standard. The article proposes a variant of a computer program implementation that makes it possible to automate the check of the availability of publications in electronic library systems. There are the logic of the program operation, features and tools for its implementation are described, examples of listings in Python, illustrations of the program interface, results of its work are given. The resulting software tool allows the user to select a file with the discipline's work program, searches for a list of references in it and then checks the possibility of accessing them under the organization's login and password. Based on the results of the program's work, the user receives a message about which links are available for reading, and which are not.

Keywords: applied programming, software module, software development, electronic library system, availability of electronic publications, monitoring of electronic publications, Python, PyQt, python-docx

Информационное обеспечение образовательного процесса в учреждении высшего образования подлежит аккредитационной экспертизе, которая подтверждает соответствие образовательных программ вуза требованиям Федеральных государственных образовательных стандартов (далее – ФГОС). Библиотечный фонд, как одна из главных составляющих такого обеспечения, должен быть укомплектован печатными и/или электронными учебными изданиями [1].

Списки литературы рабочих программ дисциплин (далее – РПД) подлежат проверке. Как правило, перед утверждением РПД сотрудники библиотеки проверяют наличие печатных изданий и доступ к электронным изданиям, указанным в качестве основной и дополнительной литературы в РПД. Та-

кая верификация осуществляется перед утверждением РПД и после их актуализации, в том числе внесения изменений в список литературы.

На момент изучения любой дисциплины, входящей в учебный план направления подготовки, вся указанная в ней литература должна быть доступна обучающимся, и в частности издания, находящиеся на сайтах ЭБС. В противном случае происходит нарушение требований ФГОС [2]. По определенным причинам то или иное издание может стать недоступным для студентов и сотрудников вуза в конкретной ЭБС. Это обуславливает необходимость периодических проверок ссылок на их доступность, что, в свою очередь, требует временных затрат со стороны проверяющих, а также не исключает человеческого фактора.

Информатизация общества обусловила проникновение компьютерных технологий во все сферы жизни человека и позволила автоматизировать процессы во многих предметных областях. Сфера образования, как одна из наиболее быстро реагирующих на изменения в обществе, не является исключением и активно внедряет современные информационные технологии, в том числе и в деятельность библиотек.

В настоящее время библиотеки вузов находятся в поиске средств, которые позволили бы автоматизировать процесс мониторинга учебно-методического обеспечения образовательного процесса [3]. Однако в описаниях применяемых библиотеками программных средств не указаны модули, осуществляющие мониторинг доступности изданий в ЭБС.

Следует отметить, что в настоящее время существуют сервисы для проверки работоспособности ссылок. Рассмотрим подробнее особенности двух наиболее известных и распространенных таких сервисов: *xseo.in* и *Ablebits Link Checker for Microsoft Word*.

Сервис *Xseo.in* (рис. 1) предназначен для поиска ссылок в тексте и проверки работоспособности этих ссылок (проверяется код ответа HTTP). Популярность сервиса обусловлена тем, что он бесплатный. В рамках решаемой нами проблемы к недостаткам данного сервиса можно отнести необходимость ручного поиска и ввода списка литературы из РПД, а, главное, отсутствие проверки статуса издания в ЭБС.

Программный сервис *Ablebits Link Checker for Microsoft Word* (рис. 2) является дополнением к *Microsoft Office Word*

и позволяет напрямую проверять работоспособность ссылок в документе, особым образом отмечая и отображая список «подозрительных» ссылок. По сути, в РПД, содержащей ссылки на литературу в ЭБС, данный сервис сможет проверить, корректна ли ссылка на издание, то есть осуществится ли переход по ней и откроется ли что-то в браузере.

Сервисы проверки работоспособности ссылок не решают проблемы определения доступности издания в ЭБС, так как в таких сервисах не предусмотрен отбор пользователем особенностей отображаемого по ссылке контента. Это, в свою очередь, актуализировало необходимость разработки программного средства, решающего указанную проблему.

Материалы и методы исследования

В данной статье представлен вариант реализации программного решения, которое позволяет автоматизировать мониторинг доступности издания следующим образом.

В выбранном пользователем файле (РПД) осуществляется поиск списка литературы, в котором идентифицируются ссылки на ЭБС, и происходит проверка их статуса. По результатам работы программного средства пользователь получает сообщение о том, по каким ссылкам издания доступны для чтения или не доступны. У программного средства для входа в ЭБС должны быть соответствующие логины и пароли.

В общем виде жизненный цикл проверки показан на UML-диаграмме последовательности (рис. 3).

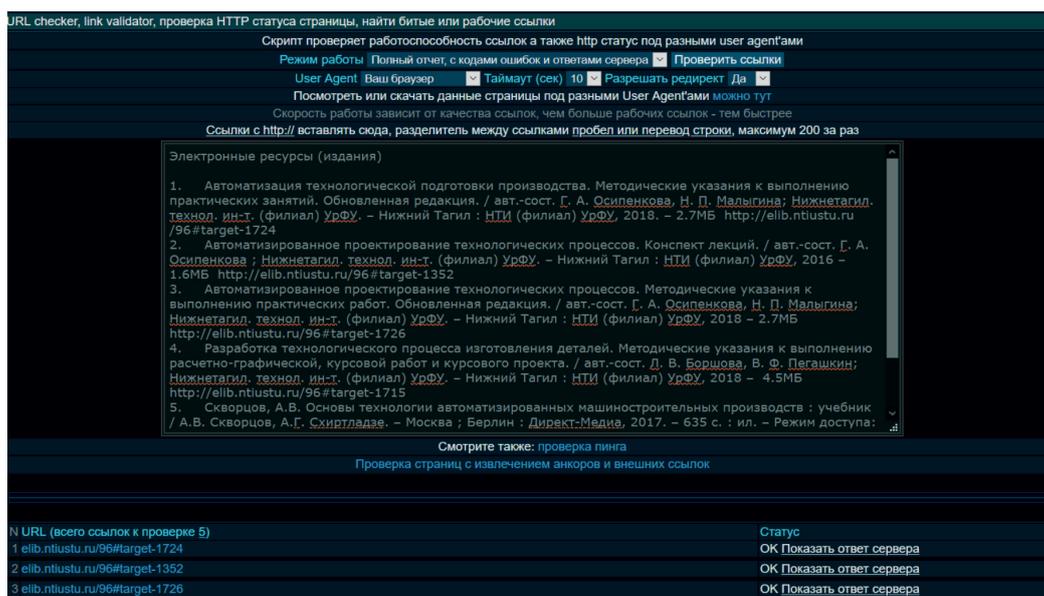


Рис. 1. Интерфейс сервиса *xseo.in*

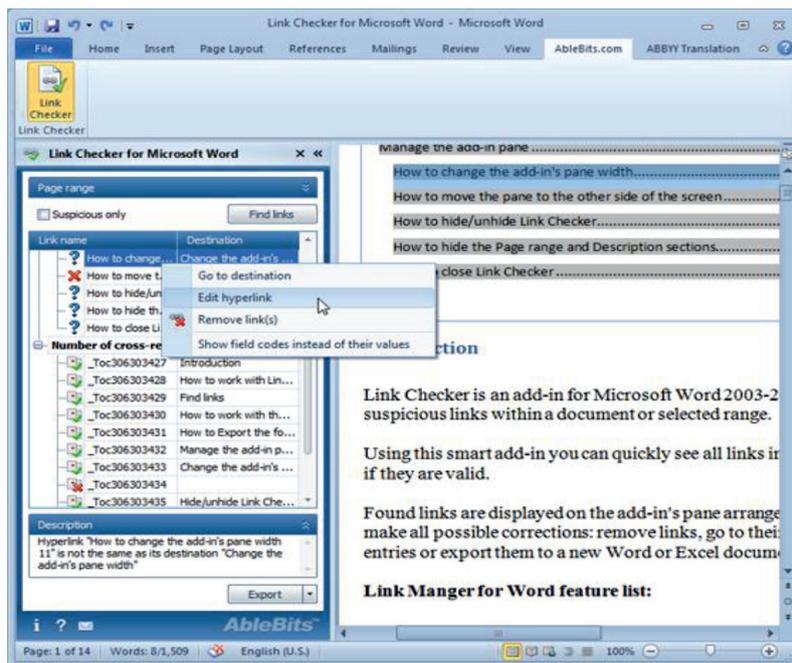


Рис. 2. Интерфейс Ablebits Link Checker for Microsoft Word

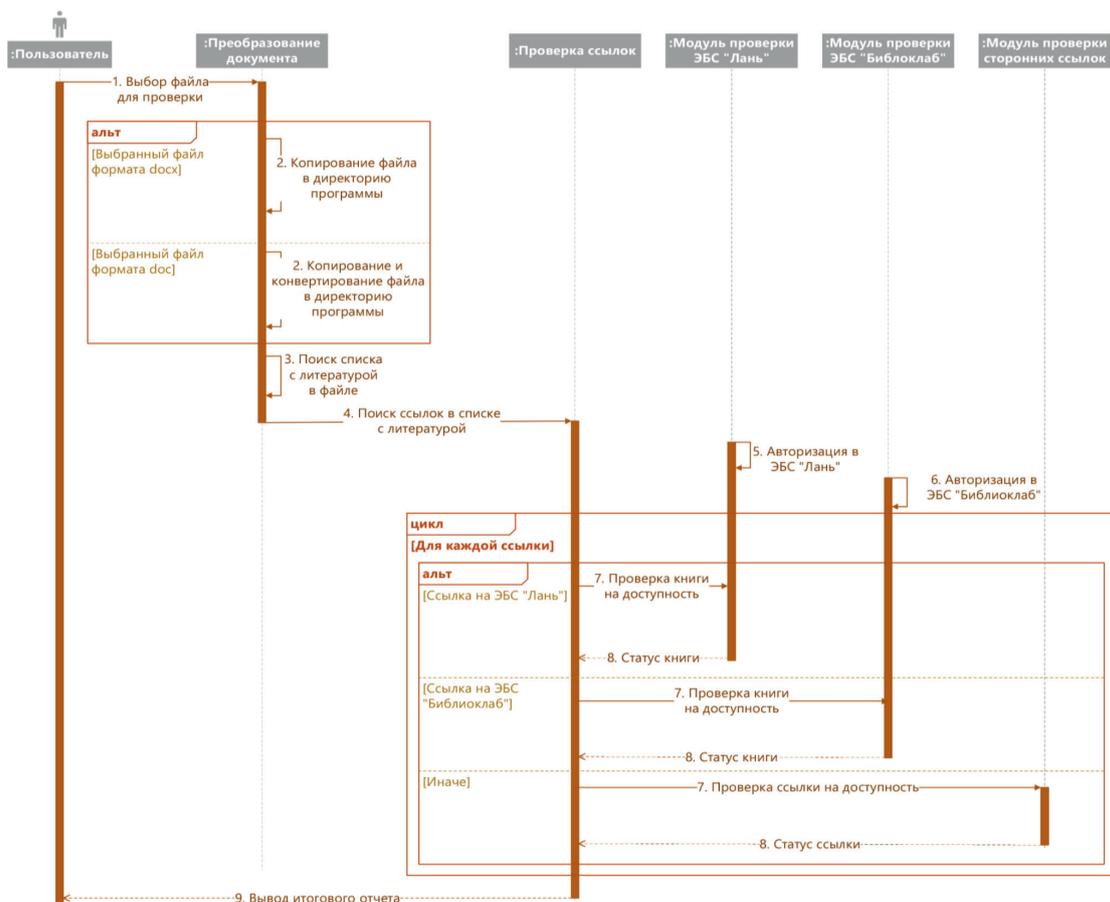


Рис. 3. Диаграмма последовательности

Как видно из диаграммы, выбранный пользователем файл сначала преобразуется. Обратим внимание, что РПД, как правило, в библиотеку для верификации попадают в формате текстового файла.

В качестве языка программирования для реализации программного средства был выбран Python 3.7.9.

Для работы *Python* с документом *Microsoft Word* необходимо преобразовать его в современный формат *docx*, это необходимо сделать по требованию работы с библиотекой *python-docx* [4]. Для этого в программе была предусмотрена функция преобразования (листинг 1).

Листинг 1

Конструктор класса *Doc*

```
class Doc:
    # получение корневой папки с программой
    ProgramInstalDir = "\\".join(os.path.abspath(__file__).split("\\")[:-1])

    def __init__(self, FileName, Login1, Password1, load):
        # получение расширение файла и копирование файла в корневую папку
        Extension = FileName.split(".")[-1]
        FileName = shutil.copyfile(
            FileName, self.ProgramInstalDir+'\\temp.doc')

        if Extension == "docx":
            self.FileName = FileName
        else:
            self.FileName = self.ConvertToDocx(FileName)
```

Как видно из листинга 1, если файл имеет расширение *docx*, то конвертация не требуется, иначе – вызывается метод конвертации (листинг 2).

Листинг 2

Метод *ConvertToDocx*

```
def ConvertToDocx(self, FileName):
    word = win32com.client.Dispatch('Word.Application')
    doc = word.Documents.Add(FileName)
    Target = self.ProgramInstalDir+'\\temp_converted.docx'
    doc.SaveAs(Target, FileFormat=16)
    doc.Close()
    return Target
```

В результате получается файл формата *docx*, с которым можно проводить операции чтения.

Далее в файле необходимо найти список с литературой. Для этого вызывается метод поиска списка (листинг 3), который ищет в документе соответствующие заголовки. Пример того, как выглядит список литературы в РПД, приведен на рис. 4.

9	Нормы времени на операцию.	Методика расчета нормы времени для технологических операций изготовления деталей машиностроения высокой сложности. Установление нормы времени на технологические операции изготовления деталей машиностроения высокой сложности.
10	Расчетно-технологическая карта.	Разработка расчетно-технологической карты (РТК). Последовательность оформления РТК. Связь системы координат станка, детали, инструмента. Правила построения траектории движения инструмента. Особенности расчета траекторий инструмента. Расчет координат опорных точек на контуре детали.
11	Технологическая документация.	Нормативно-технические и руководящие документы по оформлению технологической документации. Основные технологические документы. Разновидности описания технологического процесса. Комплексность документов. Оформление технологической документации на технологические процессы изготовления деталей машиностроения высокой сложности.

2.1.1.4. Язык реализации программы
Программа дисциплины реализуется на государственном языке Российской Федерации.

2.1.2. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Автоматизация машиностроительного производства - В

Электронные ресурсы (издания)

1. Автоматизация технологической подготовки производства. Методические указания к выполнению практических занятий. Обновленная редакция. / авт.-сост. Г. А. Осипенкова, Н. П. Малыгина; Нижнетагил. технол. ин-т. (филиал)

- УрФУ. – Нижний Тагил : НТИ (филиал) УрФУ, 2018. – 2.7МБ <http://elib.ntiustu.ru/96#target-1724>
2. Автоматизированное проектирование технологических процессов. Конспект лекций. / авт.-сост. Г. А. Осипенкова ; Нижнетагил. технол. ин-т. (филиал) УрФУ. – Нижний Тагил : НТИ (филиал) УрФУ, 2016 – 1.6МБ <http://elib.ntiustu.ru/96#target-1352>
3. Автоматизированное проектирование технологических процессов. Методические указания к выполнению практических работ. Обновленная редакция. / авт.-сост. Г. А. Осипенкова, Н. П. Малыгина; Нижнетагил. технол. ин-т. (филиал) УрФУ. – Нижний Тагил : НТИ (филиал) УрФУ, 2018 – 2.7МБ <http://elib.ntiustu.ru/96#target-1726>
4. Разработка технологического процесса изготовления деталей. Методические указания к выполнению расчетно-графической, курсовой работ и курсового проекта. / авт.-сост. Л. В. Боршова, В. Ф. Пегашкин; Нижнетагил. технол. ин-т. (филиал) УрФУ. – Нижний Тагил : НТИ (филиал) УрФУ, 2018 – 4.5МБ <http://elib.ntiustu.ru/96#target-1715>
5. Скворцов, А.В. Основы технологии автоматизированных машиностроительных производств : учебник / А.В. Скворцов, А.Г. Смиртладзе. – Москва ; Берлин : Директ-Медиа, 2017. – 635 с. : ил. – Режим доступа: по подписке. – URL: <http://biblioclub.ru/index.php?page=book&id=469049> (дата обращения: 03.03.2020). – Библиогр. в кн. – ISBN 978-5-4475-8420-7. – DOI 10.23681/469049. – Текст : электронный.

Печатные издания

1. Автоматизация технологических процессов : учеб. пособие для вузов / А. Г. Смиртладзе, С. В. Бочкарев, А. Н. Лыков [и др.]. – Старый Оскол : ТНТ, 2014. – 523 с. : ил. – Глоссарий: с. 402-413. – Библиогр.: с. 414-418 (53 назв.). – Приложение: с. 419-523 8 экз.
2. Основы автоматизации машиностроительного производства: учебник для машиностроит.-спец. ВУЗов/Е. П. Ковальчук, М. Г. Косов, В. Г. Митрофанов [и др.]

Рис. 4. Список литературы в РПД

Метод *SearchLit*

```
def SearchLit(self):
    doc = docx.Document(self.FileName)
    lit_stop = [0, 0]
    self.Main_lit = []
    for par in doc.paragraphs:
        # чтение строк из документа
        if lit_stop[1] == 1:
            break
        elif par.text == "":
            pass
        elif re.search(r'Электронные ресурсы', par.text, flags=re.IGNORECASE):
            # следующие строки будут с литературой
            lit_stop[0] = 1
        elif re.search(r'Печатные издания', par.text, flags=re.IGNORECASE):
            # список литературы закончился
            lit_stop[1] = 1
        elif lit_stop[0] == 1 and lit_stop[1] == 0:
            self.Main_lit.append(par.text)
```

На основании строк, содержащихся в списке литературы, проводится их синтаксический анализ при помощи регулярных выражений. Далее все ссылки помещаются в массив для дальнейшей обработки (листинг 4).

Метод *UrlSearch*

```
def UrlSearch(self, lit_arr, load):
    url_arr = []
    # для отображения прогресса
    proggres_hop = int(20/(len(arr)+0.01))
    for i in lit_arr:
        url = re.search(r'https?:\/\/[\S]+', i, flags=re.IGNORECASE)
        if not url:
            url = re.search(r'www.[\S]+', i, flags=re.IGNORECASE)
        if url:
            # если последний символ точка, то удалить ее
            if url[0][-1] == ".":
                url_arr.append(url[0][:-1])
            else:
                url_arr.append(url[0])
        else:
            url_arr.append("Ссылка не найдена")
        self.Progress += proggres_hop
        load.setValue(self.Progress)
    return url_arr
```

Проверку доступности необходимо выполнять индивидуально для каждой ЭБС, так как все они имеют различный подход к авторизации и проверке доступности изданий. В качестве примера рассмотрим модуль для ЭБС "Лань" с использованием открытого API (листинг 5).

Модуль *ebs_lan*

```
import requests
import json

def Authorization(login, password):
    # Выполняем вход в ЭБС
    url = 'https://e.lanbook.com/api/v2/signin'
    user_agent_val = 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
    (KHTML, like Gecko) Chrome/75.0.3770.142 Safari/537.36'
    # Создаем сессию и указываем наш user-agent
    session = requests.Session()
    r = session.get(url, headers={
        'User-Agent': user_agent_val
    })
```

```

# Указываем referer
session.headers.update({'Referer': url})
session.headers.update({'User-Agent': user_agent_val})
# Осуществляем вход с помощью метода POST с указанием необходимых данных
post_request = session.post(url, {
    'login': login,
    'password': password,
    'type': 'UserCredentials'
})
login1 = json.loads(post_request.text)
if login1["status"] == "ok":
    return session
else:
    return "Ошибка авторизации в ЭБС Лань"

def Check_available(book_id, session):
    user_agent_val = 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/75.0.3770.142 Safari/537.36'
    session.headers.update({'User-Agent': user_agent_val})
    a = session.get(
        'https://e.lanbook.com/api/v2/catalog/book/'+str(book_id)).text
    j = json.loads(a)
    if j["body"]:
        if j["body"]["available"]:
            return "Книга доступна"
        else:
            return "Книга не доступна"
    else:
        return "Книга не найдена"

```

Данный модуль возвращает информацию о доступности издания (то есть возможности его открытия и/или скачивания) для аккаунта, а также о наличии самого издания в электронной библиотеке.

Результаты исследования и их обсуждение

Готовая программа выглядит следующим образом (рис. 5). Графический интерфейс программного средства был создан с использованием *PyQt* [5], так как данная библиотека позволяет довольно просто создавать графический интерфейс для программы.

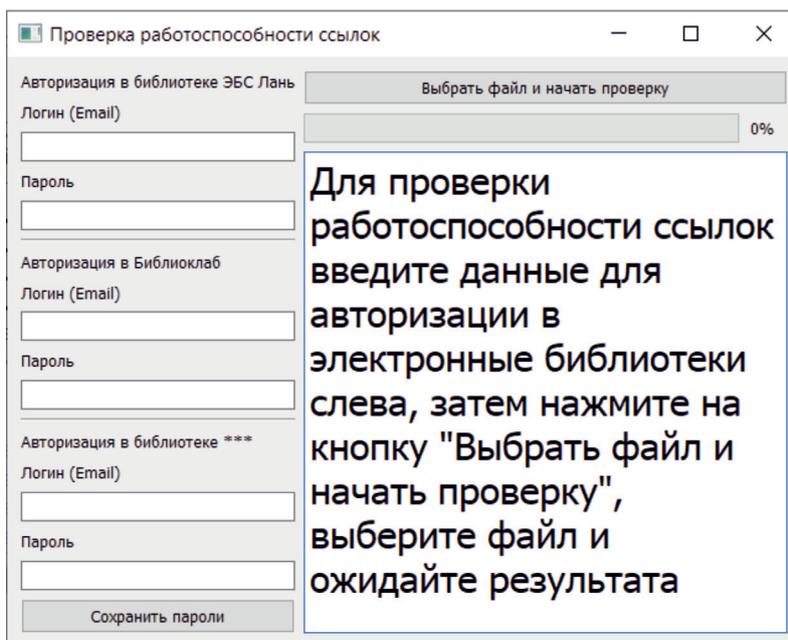


Рис. 5. Графический интерфейс программного средства

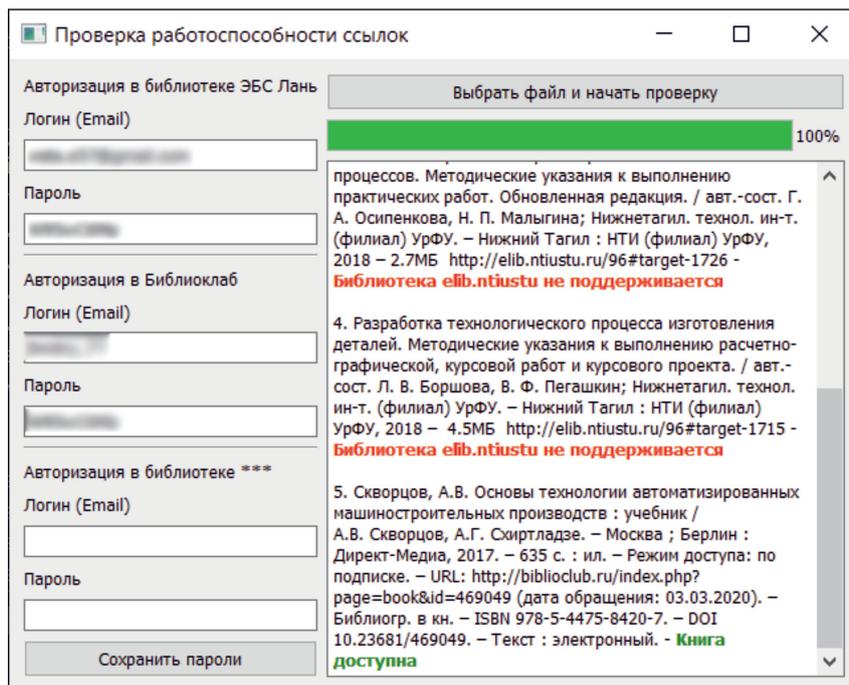


Рис. 6. Отчет о проверке работоспособности ссылок

Для осуществления проверки работоспособности ссылок пользователь вводит данные для авторизации в системах ЭБС, затем нажимает соответствующую кнопку и выбирает текстовый файл. После этого проходит авторизация с помощью модуля *request* [6], затем пользователь получает готовый отчет о проверке (рис. 6).

Заключение

В результате работы получилось программное средство, которое имеет простой интерфейс и позволяет автоматизировать проверку РПД на наличие недоступных ссылок. Спроектированная структура программы позволяет легко расширять количество проверяемых ЭБС с помощью добавления отдельных модулей.

Список литературы

1. Федеральный закон от 29 декабря 2012 г. № 273 – ФЗ «Об образовании в Российской Федерации». [Электронный ресурс]. URL: <https://base.garant.ru/70291362/a573badc-fa856325a7f6c5597efaaedf/> (дата обращения: 22.12.2020).
2. Аккредитационная экспертиза библиотечного фонда // Информационно-аналитический журнал «Университетская книга». [Электронный ресурс]. URL: <http://www.unkniga.ru/biblioteki/vuzbiblio/7698-akkreditatsionnayaekspertiza-bibliotechnogo-fonda.html> (дата обращения: 22.12.2020).
3. Квелидзе-Кузнецова Н.Н., Морозова С.А. Комплексный подход к эффективному обеспечению образовательных программ высшего образования библиотечно-информационными ресурсами // Наука и научная информация. 2019. № 2(1). С. 41–52. DOI: 10.24108/2658-3143-2019-2-1-41-52.
4. python-docx. [Электронный ресурс]. URL: <https://python-docx.readthedocs.io/en/latest/> (дата обращения: 11.12.2020).
5. What is PyQt? // Riverbank Computing. [Электронный ресурс]. URL: <https://www.riverbankcomputing.com/software/pyqt/> (дата обращения: 22.12.2020).
6. Requests: HTTP for Humans. [Электронный ресурс]. URL: <https://requests.readthedocs.io/en/master/> (дата обращения: 22.12.2020).