|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  «ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ  И.С. ТУРГЕНЕВА» | | | | |
| Физико-математический факультет  Кафедра алгебры и математических методов в экономике | | | | |
| ОТЧЕТ  по производственной практике | | | | |
| на материалах ФГБОУ ВО «Орловский государственный университет имени И.С. Тургенева» наименование профильной организации  Студент Креханов Герман Алексеевич | | | | |
|  |  | | | (Ф.И.О.) |
| Группа 71 ПИФМ\_\_\_\_\_\_\_\_\_\_ | | | | |
|  |  | | |  |
| Институт/факультет физико-математический | | | | |
|  | |  | |  |
| Направление (специальность) | | | 09.03.03 Прикладная информатика | | |
| Руководитель практики от университета | | | | Зубкова Л.Н. |
|  | | | | (Ф.И.О.) |
| Руководитель практики от профильной  организации | | | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | | | | (Ф.И.О.) |
|  | | | | М.П. |
| Оценка защиты \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Орел 2020 г. | | | | |

**Содержание**

[Общие сведения о производственной практике 3](#_Toc54280302)

[Введение 4](#_Toc54280303)

[1 Описание предметной области 6](#_Toc54280304)

[1.1 Инструменты разработки для автоматизации систем сопровождения научного мероприятия. 6](#_Toc54280305)

[1.2 Обзор модулей Python используемых в разработке 12](#_Toc54280306)

[2 Результаты выполнения практического задания 14](#_Toc54280307)

[2.1 Реализация модулей технического секретаря и экспертов. 14](#_Toc54280308)

[Заключение 20](#_Toc54280309)

[Список литературы 21](#_Toc54280310)

# Общие сведения о производственной практике

**Период прохождения практики:** 08.09.2020 – 23.10.2020 г.

**Место прохождения практики:** ФГБОУ ВО «Орловский государственный университет имени И.С. Тургенева», кафедра алгебры и математических методов в экономике

**Руководитель практики от университета:** к.п.н., доцент Зубкова Л.Н.

**Групповой руководитель практики:** к.п.нЛебедева Елена Николаевна

**Аннотация:** в отчете проводится анализ инструментов автоматизации и выбирается самый подходящий из них, рассматриваются необходимые библиотеки, представлено решение поставленных практикой задач, а также результат их выполнения.

**Ключевые слова:** Python, проведение конференции, сборник статей

# Введение

За последние годы отмечается значительное увеличение объемов различных научных собраний, выставок, а так же научных конференций. Этим обуславливается потребность в автоматизации различных задач, связанных с обеспечением регистрации участников, проверки их работ, проведения самой конференции, создания сборника статей, после окончания мероприятия.

В настоящее время используется стандартный подход, для проведения конференции: регистрация участников с проверкой их работ на соответствие заявленной теме, выступление организаторов конференции, работа по секциям, кофе-брейк в середине работы конференции и фуршет или банкет по ее окончании, иногда проводится культурная программа, в самом конце публикуется сборник научных трудов.

**Цель практики:**

Целью практической работы является реализация приложения для проведения научной конференции с помощью языка программирования Python.

**Задачи практики:**

1. Систематизировать теоретические сведения о проведении научной конференции;

2. Рассмотреть алгоритмы и инструменты, предназначенные для решения отдельных задач по обработке полнотекстовых документов, а так же решения с базами данных;

3. Разработать с помощью языка программирования Python проектное решения для программной реализации системы проведения научной конференции;

4. Описать программную реализацию регистрации экспертов, создание сборника статей;

5. Подвести итоги по результатам проделанной работы.

**Основные результаты практики:**

Рассмотрены алгоритмы и инструменты для обработки полнотекстовых документов, разработано проектное решение, описана программная реализация регистрации экспертов, создание сборника статей, подведены итоги проделанной работы

**Краткое описание структуры и содержания отчета**

Отчет включает введение, две главы, заключение, список литературы.

Во введении обоснована актуальность темы практического задания, определены объект, предмет исследования, сформулированы цель, основные задачи, описана структура работы.

Первая глава состоит из двух пунктов. В первом пункте рассмотрены возможности языка программирования Python. Во втором пункте рассматриваются библиотеки и расширения для решения поставленных задач.

Вторая глава включает описание результатов программной реализации создания приложения для проведения научной конференции.

В заключении приведены основные результаты работы.

Список литературы содержит перечень источников, используемых при написании работы.

**1 Описание предметной области**

**1.1 Инструменты разработки для автоматизации систем сопровождения научного мероприятия.**

На сегодняшний день существует множество инструментов для разработки и автоматизации различных систем. В процессе практической работы были рассмотрены 3 средства разработки программных решений по автоматизации систем сопровождения научного мероприятия. Это языки программирования С#, Python и Delphi.

C# — современный объектно-ориентированный язык программирования. C# относится к широко известному семейству языков C, и покажется хорошо знакомым любому, кто работал с C, C++, Java или JavaScript.   
 C#  — это объектно- и компонентно-ориентированный язык программирования. C# предоставляет языковые конструкции для непосредственной поддержки такой концепции работы. Благодаря этому C# подходит для создания и применения программных компонентов [1].

Вот лишь несколько функций языка C#, обеспечивающих надежность и устойчивость приложений. Сборка мусора автоматически освобождает память, занятую недостижимыми неиспользуемыми объектами. Обработка исключений предоставляет структурированный и расширяемый подход к обнаружению ошибок и восстановлению после них.  Синтаксис запросов создает общий шаблон для работы с данными из любого источника. Поддержка языков для асинхронных операций предоставляет синтаксис для создания распределенных систем. Сопоставление шаблонов предоставляет синтаксис для простого разделения данных из алгоритмов в современных распределенных системах. В C# действует единая система типов. Все типы C#, включая типы-примитивы, такие как int и double, наследуют от одного корневого типа object. Все типы используют общий набор операций, а значения любого типа можно хранить, передавать и обрабатывать схожим образом. Более того, C# поддерживает как определяемые пользователями ссылочные типы, так и типы значений. C# позволяет динамически выделять объекты и хранить упрощенные структуры в стеке.

В C# особое внимание уделяется управлению версиями для обеспечения совместимости программ и библиотек при их изменении. Вопросы управления версиями существенно повлияли на такие аспекты разработки C#, как раздельные модификаторы virtual и override, правила разрешения перегрузки методов и поддержка явного объявления членов интерфейса.  
 У языка программирования C# на сегодняшний день множество поклонников, среди них большое количество компаний и стартапов создающих инди-игры и 3D-игры [2].

К многочисленным преимуществам этого языка относят:

* поддержку подавляющего большинства продуктов Microsoft;
* для небольших компаний и некоторых индивидуальных разработчиков бесплатными являются такие инструменты, Visual Studio, облако Azure, Windows Server, Parallels Desktop для Mac Pro и многие другие;
* большое количество синтаксического сахара, представляющего собой специальные конструкции, разработанные для понимания и написания кода. Они не имеют значения при компиляции;
* порог вхождения у языка C# низкий. Его синтаксис имеет много схожего с другими языками программирования, благодаря чему облегчается переход для программистов. Язык C# считается наиболее понятным и подходящим для новичков;
* после покупки Xamarin на C# можно писать программы и приложения для таких операционных систем, как iOS, Android, MacOS и Linux;  
  имеется целое сообщество из опытных программистов;
* сегодня в любом регионе России имеется много вакантных мест на должность C#-программиста [2].

Кроме многочисленных плюсов эта программа имеет некоторые недостатки. Среди них следует выделить:

* приоритетная ориентированность на Windows платформу;
* язык бесплатен только для небольших фирм, индивидуальных программистов, стартапов и учащихся. Крупной компании покупка лицензионной версии этого языка обойдется в круглую сумму;
* в языке осталась возможность использования оператора безусловного перехода.

Необходимо отметить, что язык C# не представляет собой сложности для новичков, так как его легко изучить и понять.

Delphi можно назвать достаточно многофункциональной и сильной средой разработки приложений, RAD-оболочкой, в которой собраны последние подходы к программированию. Библиотеку VCL, на основе которой написана оболочка Delphi, называют в какой-то степени революционной [3]. Если сравнить Delphi и C++ нельзя не заметить, что эффективность работы кода, который написан на языке Pascal, на пару процентов отличается от кода, сгенерированного на языке C++. Можно сделать вывод, что потенциалы языка почти не ограничены.

Одним из основных преимуществ Delphi перед его аналогами заключается в достаточно быстром процессе разработки программных продуктов, которые, в свою очередь, включают в себя довольно замысловатый интерфейс. Еще можно отметить наличие обширного набора компонентов, обеспечивающих работу с базами данных. Упорядоченный набор данных работает с разными базами данных, как с локальными, так и с промышленными. Например, с такими, как, Oracle или MS SQL Server [4]. Так же возможность управления базами данных на логическом уровне, не используя низкоуровневые запросы к драйверам можно назвать значительным плюсов Delphi в этой области применения.

1. Удобная среда разработки, в которой существует быстрый функциональный отладчик, который, в свою очередь, достаточно доступен в любой момент времени.

2. Наличие справочной системы, в виде контекстного меню, что позволяет изучать язык.

3. Достаточно мощная скорость компиляции и выполнения откомпилированных программ.

4. В данной среде имеется возможность работы со вставками, написанными на языке Assembler.

5. Нет нужды в сторонних библиотеках.

6. Компонентная модель программирования. Использование данной модели дает возможность использовать компоненты, которые разработаны прочими программистами, без использования исходного кода. Применение компонентной модели позволяет сократить время разработчика.

7. Достаточно резвый браузер классов и вывод авто подсказки авто завершения кода.

Помимо такого обширного спектра плюсов, существуют и некоторые недочеты, хоть их и мало. Однако в некоторых случаях их выделяют и как достоинства.

Основным недостатком, можно назвать нединамическое добавление библиотеки VCL и компонентов к исполняемому файлу. Можно выделить, что присоединение VCL осуществляется и инерционно, однако с каждой новой разработкой разработчик будет вынужден увеличивать еще и VCL.

Еще один недочет заключается в том, что в форме Form все данные о форме, а именно: атрибуты, настройки элементов, значения, установленные по умолчанию, находятся в exe-файле. Проанализировав исходный код, можно прийти к выводу, что при генерировании формы происходит практически синтаксический разбор данных инициализации, что, в свою очередь, тормозит ее.

Сам Object Pascal тоже можно обозначить как недостаток. Конечно простота, эффективность и легкость в изучении — это хорошо, однако ему не помешала бы взгляд наличие мощных средств языка C++. Например, таких как: стандарты, перемещение операторов, объектного образца модели. Кроме всего прочего, достаточно сильно ощущается тенденция к разрастанию exe-файлов, которые генерируются Delphi. То есть, большинство маленьких проектов, которые разработаны в Delphi 4, при перекомпиляции в Delphi 5 увеличились на 40-70 Кбайт, при всем при этом, не обретя новой функциональности.

Исходя из выше перечисленного, можно сделать вывод, что, несмотря на достаточную конкуренцию, и обширное количество аналогов в сфере разработки программ Delphi отнюдь не уступает и находит свое применение на рынке информационных систем.

Python – это универсальный современный язык программирования высокого уровня, к преимуществам которого относят высокую производительность программных решений и структурированный, хорошо читаемый код. Синтаксис Python максимально облегчен, что позволяет выучить его за сравнительно короткое время. Ядро имеет очень удобную структуру, а широкий перечень встроенных библиотек позволяет применять внушительный набор полезных функций и возможностей [5]. Python может использоваться для написания прикладных приложений, а также разработки WEB-сервисов.

Python может поддерживать широкий перечень стилей разработки приложений, в том числе, очень удобен для работы с ООП и функционального программирования.

Особенности программирования на Python:

1. Простой язык, легкий и доступный в изучении.

У Python читаемый синтаксис. Гораздо проще читать и писать программы на Python по сравнению с другими языками, такими как: C++, Java, C#. Python делает программирование интересным и позволяет сфокусироваться на решении, а не синтаксисе.

1. Бесплатный и с открытым кодом.

Можно свободно использовать и распространять программное обеспечение, написанное на Python, даже для коммерческого использования. Вносить изменения в исходный код Python. Над Python работает большое сообщество, постоянно совершенствуя язык в каждой новой версии.

1. Портативность.

Можно перемещать и запускать программы на Python из одной платформы на другую без каких-либо изменений. Код работает практически на всех платформах, включая Windows, Mac OS X и Linux.

1. Масштабируемый и встраиваемый.

Пользователь может с легкостью комбинировать фрагменты кода на C/C++ и других языках вместе с кодом Python. Это повысит производительность приложения, а также дает возможность написания скриптов, создание которых на других языках требует больше настроек и времени.

1. Высокоуровневый, интерпретируемый язык.

В отличии от C/C++, пользователю не нужно беспокоиться о таких сложных задачах, как “сборка мусора” или управление памятью. Так же, когда идет запуск кода Python, происходит автоматическое преобразование кода в язык, который понимает компьютер. Не нужно думать об операциях более низкого уровня.

1. Стандартные библиотеки для решения общих задач.

Python укомплектован рядом стандартных библиотек, что облегчает жизнь программиста, так как нет необходимости писать весь код самостоятельно.

1. Большое количество дополнительных библиотек

Большинство библиотек, созданных для Python являются бесплатными. У пользователя будет большой выбор для загрузки и использования библиотек для решения самых разных задач, в том числе, и для решения задачи интеллектуальной обработки текстовых данных как в случае данной курсовой работы. Так же имеются библиотеки для создания интерфейсов, что упрощает программистам работу с визуализацией [6].

После некоторого опыта использования этих языков было принято решение использовать язык программирования Python, потому что он имеет простой синтаксис, у него есть множество библиотек для работы и анализа различных данных, а так он же широко используется в web-разработке.

**1.2 Обзор модулей Python используемых в разработке**

Библиотека fpdf – это библиотека для генерации PDF-документов под Python, портированная с PHP [7]. По сравнению с другими PDF-библиотеками, fpdf является простой, маленькой и универсальной, с большими возможностями и немложна в освоении. В частности, можно создавать PDF файл сразу картинками и дополнительными вложениями.

Tkinter – в Python есть довольно много GUI фреймворков, однако только Tkinter встроен в стандартную библиотеку языка. У Tkinter есть несколько преимуществ. Он кроссплатформенный, поэтому один и тот же код можно использовать на Windows, macOS и Linux.

Визуальные элементы отображаются через собственные элементы текущей операционной системы, поэтому приложения, созданные с помощью Tkinter, выглядят так, как будто они принадлежат той платформе, на которой они работают.

Хотя Tkinter является популярным GUI фреймворком на Python, у него есть свои недостатки. Один из них заключается в том, что графические интерфейсы, созданные с использованием Tkinter, выглядят устаревшими. Если вам нужен современный, броский интерфейс, то Tkinter может оказаться не совсем тем, для этого есть PyQt5 который развивается сильнее в данном плане.

Тем не менее, в плане использования, Tkinter является относительно легким по сравнению с другими библиотеками. Это отличный выбор для создания GUI приложений в Python, особенно если современный облик не в приоритете для программы, а большую роль играет функциональность и кроссплатформенная скорость.

Re – В Python для работы с регулярными выражениями есть модуль re. Для использования его нужно импортировать: import re.  
Чаще всего регулярные выражения используются для:

* поиска в строке;
* разбиения строки на подстроки;
* замены части строки.

Docx – модуль используемый для работы с файлами Microsoft Office Word. Позволяет прочитывать и редактировать данные файлы [7].

Smtplib – модуль позволяющий работать с электронной почтой. Если вам нужно получать уведомления по почте из вашего кода, отправлять пользователям письма-подтверждения, когда те создают аккаунт, или отсылать письма. Отправка писем вручную — это весьма трудоемкая задача, однако ее легко автоматизировать при помощи Python [7].

Sqlite3 – встроенный модуль для создания, редактирования и обращения к СУБД.

Zipfile – модуль для работы с архивами.

QtDesigner – модуль для создания графических интерфейсов

**2 Результаты выполнения практического задания**

**2.1 Реализация модулей технического секретаря и экспертов.**

При запуске программы нас встречает главный экран, на котором видны список открытых конференций и окно авторизации.

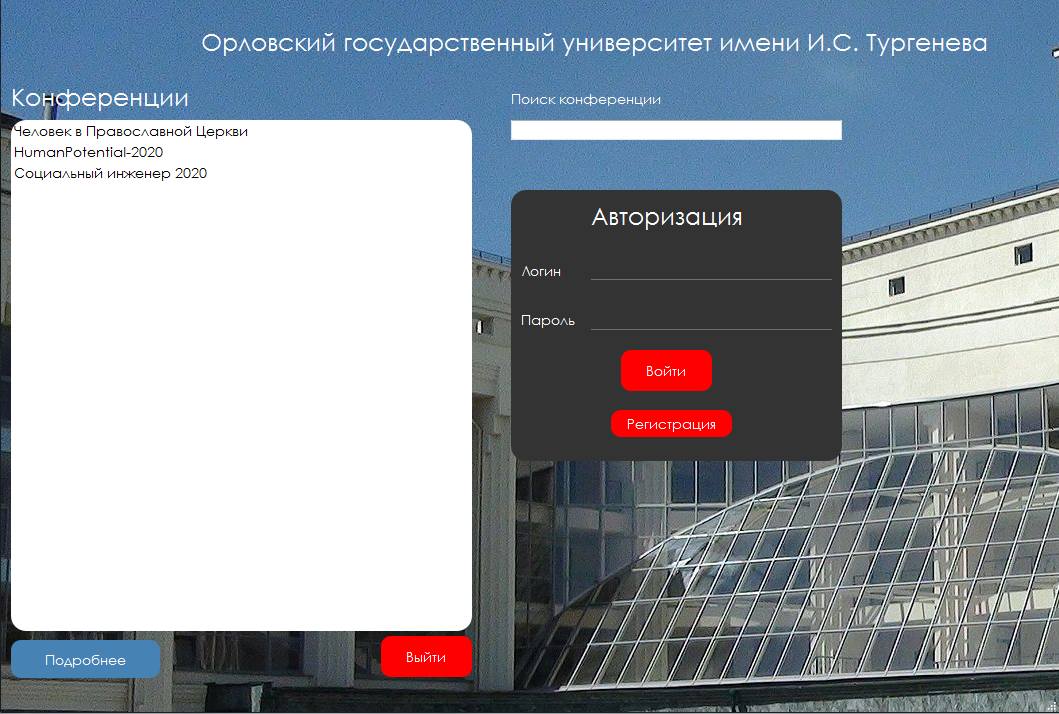
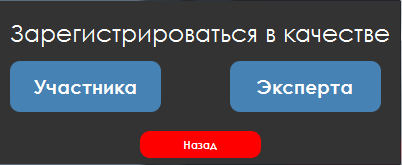
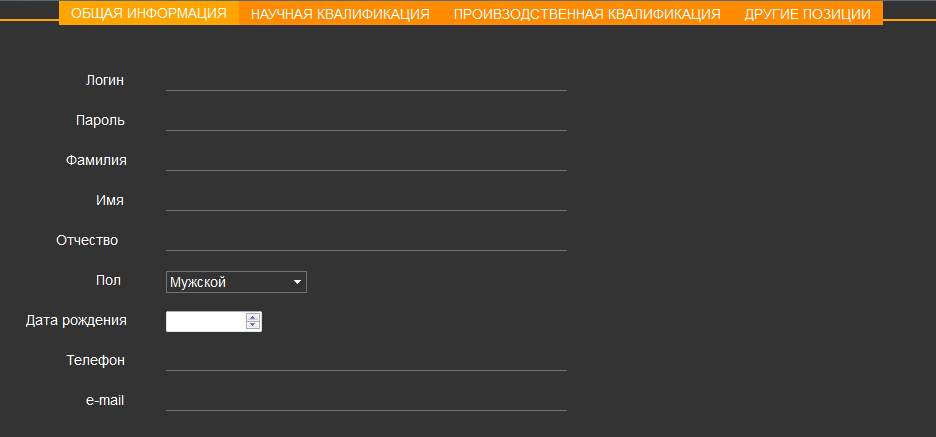


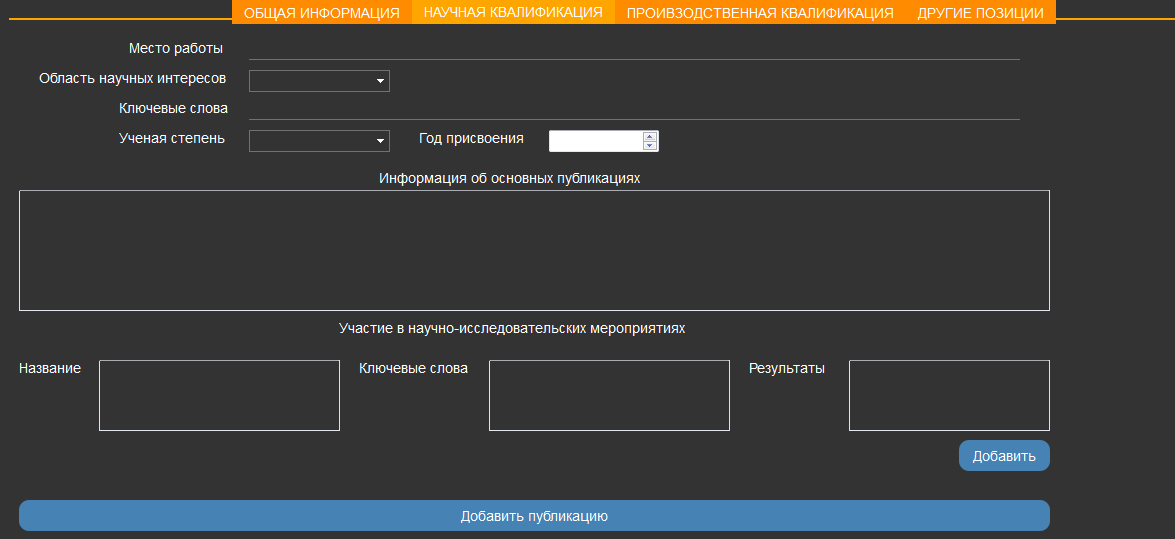
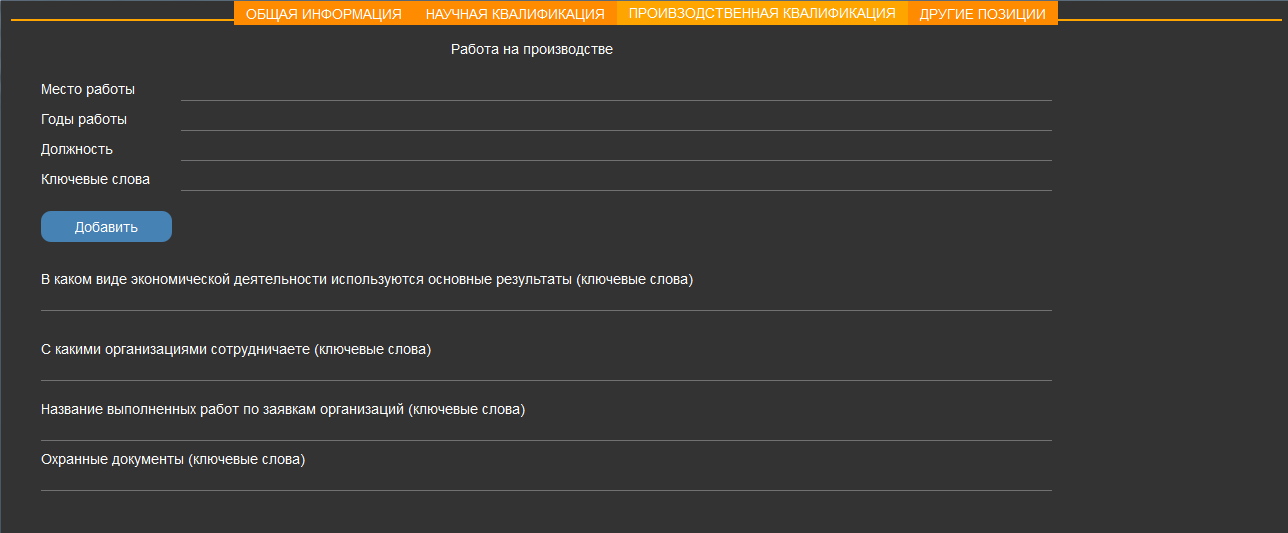
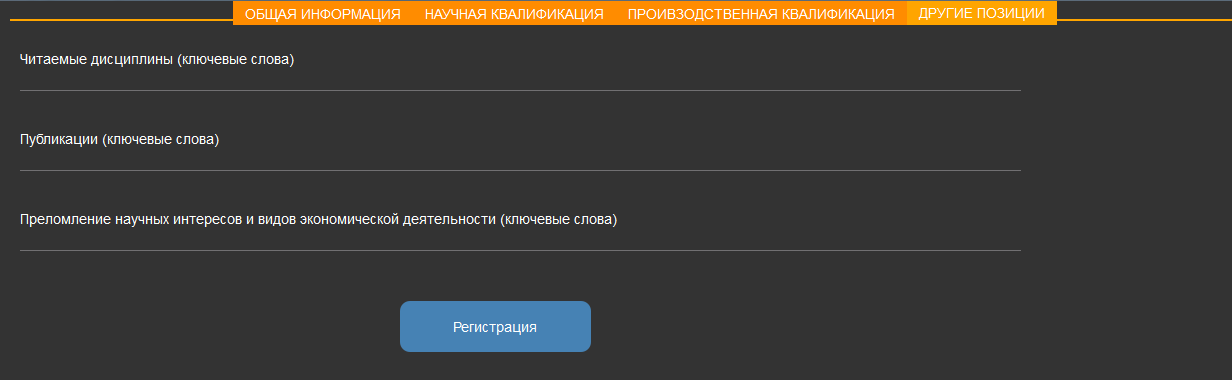
Рисунок 1 – Главное окно программы.

При нажатии кнопки «Регистрация» появится окно с выбором, предлагающее вам в качестве кого вы хотите зарегистрироваться. В нашем случаем нужно нажать на кнопку «Эксперта».

  
Рисунок 2 – Окно выбора.

После выбора эксперта, предлагается пройти процедуру регистрации, для этого нужно заполнить поля, находящиеся в четырех вкладках «Общая информация», «Научная квалификация», «Производственная квалификация», «Другие позиции». После заполнения всех полей на вкладках пользователь нажимает кнопку «Регистрация» (рисунок 6).

Рисунок 3 – Вкладка «Общая информация».

Рисунок 4 – Вкладка «Научная квалификация».Рисунок 5 – Вкладка «Производственная квалификация».Рисунок 6 – Вкладка «Другие позиции».

После регистрации все его данные переносятся в базу данных, где они и хранятся для последующего использования. В ней каждому эксперту присваивается свой id (Рисунок 7).

  
Рисунок 7 – База данных.

Далее эксперт попадает в свое рабочее окно (рисунок 8). Перед ним появляется список статей, которые ему нужно проверить. Для проверки статьи ему нужно выбрать статью в списке и нажать кнопку открыть. После этого он должен решить, что нужно сделать со статьей, одобрить ее, отправить на доработку или отказать в участии с этой статьей. После проверки статьи она отправляется в соответствующею папку, для следующего их распределения.

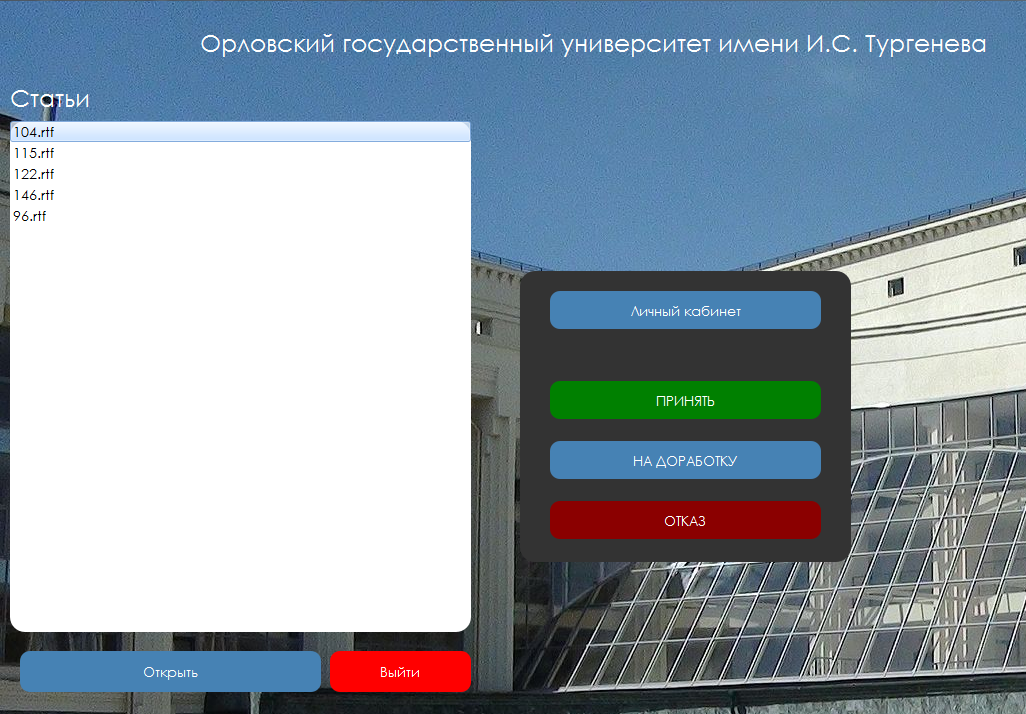


Рисунок 8 – Рабочее окно эксперта.

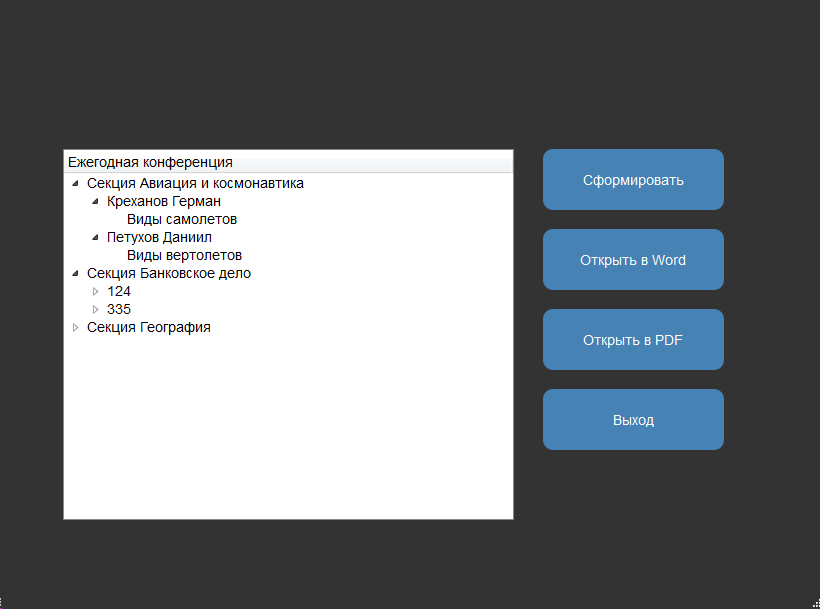
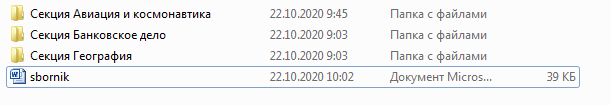
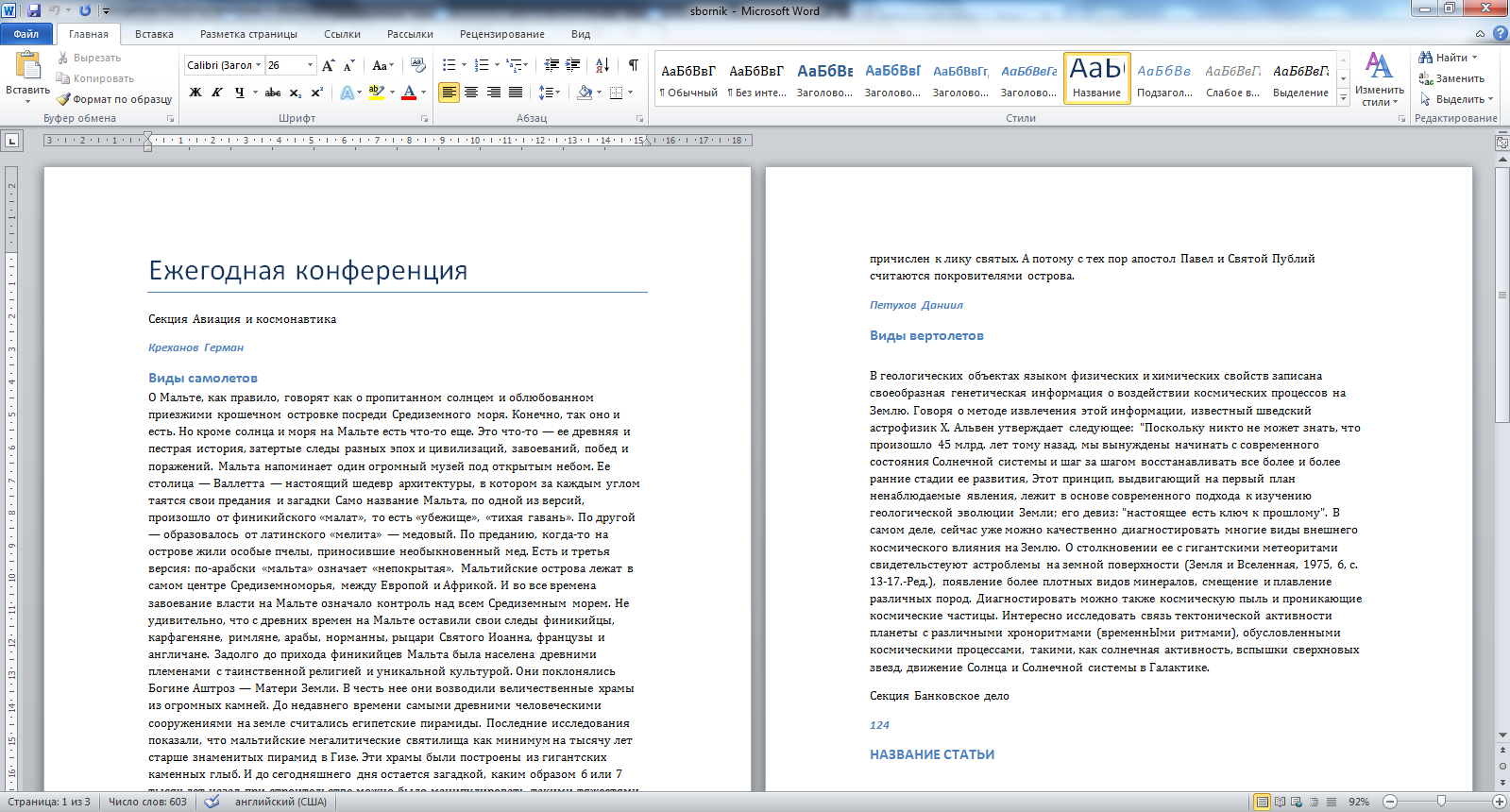
При вводе определенного логина и пароля, можно зайти в кабинет технического секретаря (рисунок 9). 

Рисунок 9 – Рабочее окно технического секретаря.

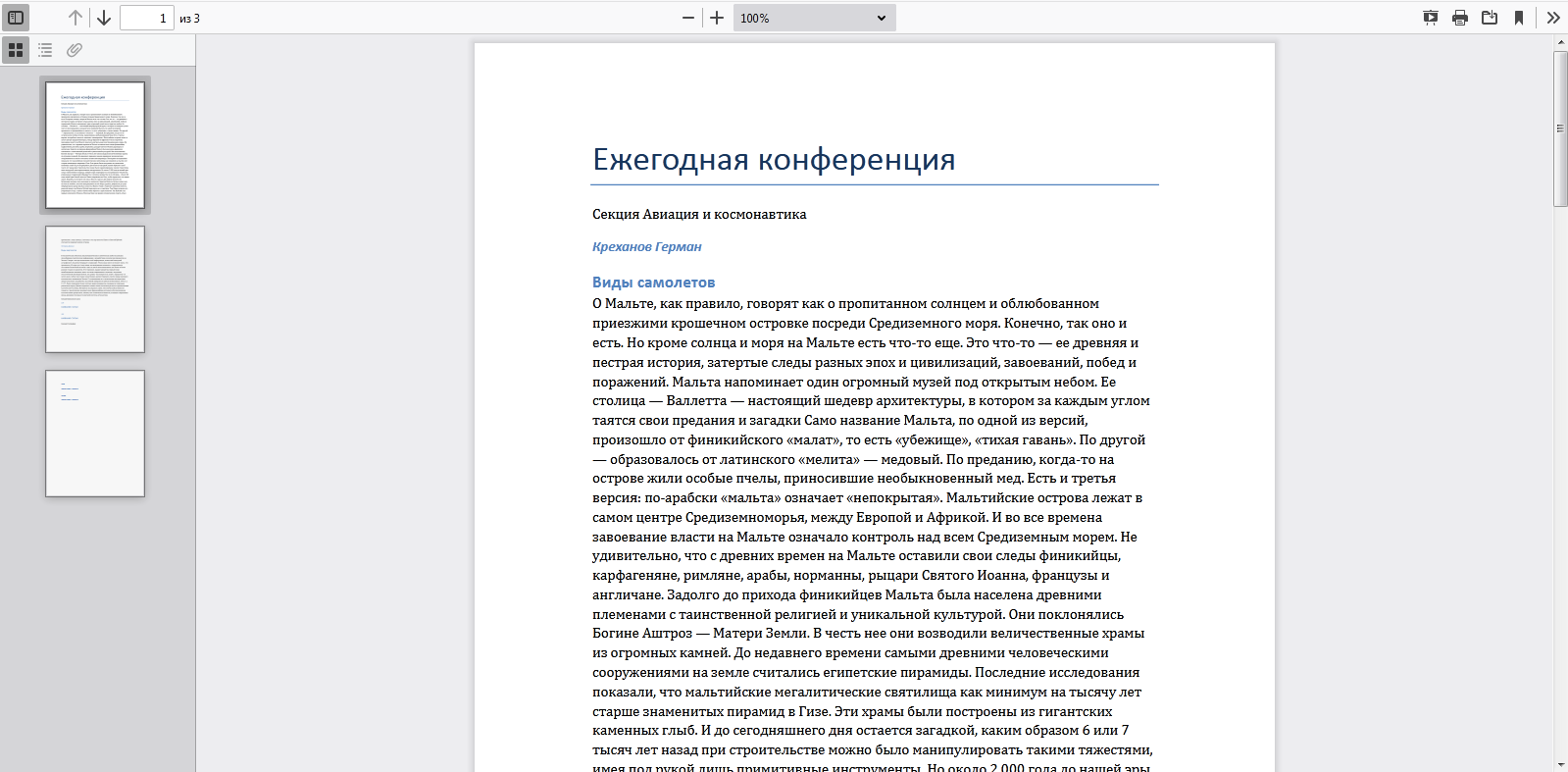
В списке слева схематично показываются секции, имена участников и темы их статей. В список попадают только те статьи, которые одобрили эксперты. При нажатии кнопки «Сформировать» формируется сборник статей (рисунок 10).

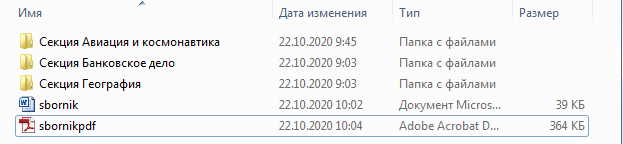
Рисунок 10 – Электронный сборник.

При нажатии кнопки «открыть в Word» можно просмотреть и отредактировать сформировавшийся сборник (рисунок 11).

Рисунок 10 – Электронный сборник.

При нажатии кнопки «открыть в PDF» файл форматируется в PDF формат и открывается той программой, которая выбрана в системе по умолчанию (рисунок 11 и 12).

  
Рисунок 11 – электронный сборник в PDF.

  
Рисунок 12 – Папка с электронным сборником.

# Заключение

В рамках практического задания была поставлена цель реализации приложения для проведения научной конференции с помощью языка программирования Python.

Для ее достижения были систематизированы теоретические сведения о проведении научной конференции. Рассмотрены алгоритмы и инструменты, предназначенные для решения отдельных задач по обработке полнотекстовых документов, а так же решения с базами данных. Разработано с помощью языка программирования Python проектное решение для программной реализации системы проведения научной конференции. Описаны программную реализацию регистрации экспертов, создание сборника статей. Подведены итоги по результатам проделанной работы.

Таким образом в результате выполнения практического задания было разработано приложение для проведения конференций.

# Список литературы

1) Бишоп Дж. С# в кратком изложении / Дж. Бишоп, Н. Хорспул. - М.: Бином. Лаборатория знаний, 2011. - 472 c.

2) Абрамян М. Visual C# на примерах / М. Абрамян. - М.: БХВ-Петербург, 2012. - 496 c.

3) Архангельский А.Я. Программирование в Delphi 6 / А.Я. Архангельский. - М.: Бином, 2018. - 258 c.

4) Сухарев М.В. Основы Delphi. Профессиональный подход / М.В. Сухарев. - М.: Наука и техника, 2018. - 600 c

5) Бриггс, Джейсон Python для детей. Самоучитель по программированию / Джейсон Бриггс. - Москва: Огни, 2013. - 177 c.

6) Гуриков, С.Р. Основы алгоритмизации и программирования на Python. Учебное пособие. Гриф МО РФ / С.Р. Гуриков. - М.: Инфра-М, Форум, 2018. - 707 c.

7) The Python Package Index [электронный ресурс] — URL: https://pypi.org/