**Оглавление**

[1. Цель практики 2](#_Toc507441389)

[2. Задачи практики 3](#_Toc507441390)

[3. Об организации 4](#_Toc507441391)

[4. Результаты прохождения практики в организации 5](#_Toc507441392)

[4.1. Основная задача 5](#_Toc507441393)

[4.2. Краткое описание решения 5](#_Toc507441394)

[4.2.1. goDeep.py 7](#_Toc507441395)

[4.2.2. mainMetaParser.py 8](#_Toc507441396)

[4.2.3. parselTongue.py 12](#_Toc507441397)

[4.3. Результат работы 14](#_Toc507441398)

[5. Результаты научно-исследовательской работы в рамках магистерской диссертации 15](#_Toc507441399)

[Заключение 16](#_Toc507441400)

[Список литературы 17](#_Toc507441401)

# Цель практики

Назначение производственной преддипломной практики заключается в формировании и развитии профессиональных знаний в контексте избранного направления обучения, закреплении теоретических знаний, полученных по дисциплинам направления и специальным дисциплинам магистерской программы, овладении необходимыми профессиональными компетенциями по выбранному направлению специализированной подготовки.

Производственная преддипломная практика состоит из прохождения практики на предприятии и самостоятельной научно-исследовательской работы в рамках утвержденной темы магистерской диссертации.

В соответствии с этим цель преддипломной практики подразделяется на две составляющие:

1. в рамках прохождения практики на предприятии:

* приобретение навыков практического решения информационных задач на конкретном рабочем месте в качестве исполнителя;
* применение полученных в ходе практики в организации навыков в написании магистерской диссертации.

1. в рамках самостоятельной научно-исследовательской работы по теме магистерской диссертации:

- подготовка теоретического материала и проведение вычислительного

эксперимента на разработанном программном продукте.

# Задачи практики

Основными задачами практики являются приобретение опыта в исследовании актуальной научной проблемы, закрепление и интеграция знаний, умений и навыков, полученных за время теоретического обучения, также подбор необходимых материалов для выполнения выпускной квалификационной работы - магистерской диссертации.

Основными задачами производственной преддипломной практики в рамках двух составляющих являются следующие положения:

1. в рамках прохождения практики на предприятии:

- самостоятельное выполнение комплекса разработки фрагмента проекта, реализуемого в организации.

1. в рамках самостоятельной научно-исследовательской работы по теме магистерской диссертации:

- написание теоретического материала в рамках магистерской диссертации;

- доработка программного продукта, а также проведение финального тестирования и исправления ошибок;

- проведение вычислительного эксперимента.

# Об организации

Общество с ограниченной ответственностью "Лаборатория информационных технологий Тверского госуниверситета".

Основано в 2012 году как малое инновационной предприятие при непосредственном участии ТвГУ.

Основной вид деятельности - разработка программного обеспечения, внедрение и сопровождение информационных систем и консультирование в этой области.

# Результаты прохождения практики в организации

## Основная задача

Внесение списка литературы из архивных выпусков журналов:

1. "Вестник Тверского государственного университета. Серия: Прикладная математика" (<http://pmk-vestnik.tversu.ru>)
2. "Нечеткие системы и мягкие вычисления" (<http://fuzzy.tversu.ru>)

в базу всероссийского математического портала MathNet: (<http://www.mathnet.ru/php/journal.phtml?jrnid=vtpmk&option_lang=rus> и

<http://www.mathnet.ru/php/journal.phtml?jrnid=fssc&option_lang=rus>).

Задача заключается в преобразовании ссылки из списка литературы в формате .tex в формат amsbib:

Строка типа:

«*\bibitem{consul:1989} { Гельфанд И.М., Шилов Г.Е.} Обобщенные функции и действия над ними. Т.1, { М.: ГИФМЛ, 1959}.*»

должна быть преобразована в следующий формат:

\RBibitem{}

\by Гельфанд~И.\,М., Шилов~Г.\,Е.

\book Обобщенные функции и действия над ними

\bookvol 1

\publaddr М.

\publ ГИФМЛ

\yr 1959

## Краткое описание решения

Для выполнения данной задачи была написана программа-скрипт на языке программирования Python для автоматического преобразования строк в формат amsbib.

Т.к. исходный формат содержит много мусора и списки литературы оформлялись не по общему стандарту, то программа позволяет выполнять редактирование результатов напрямую во время своей работы, открывая блокнот с результатами.

**На вход** программе подаются директории:

- с файлами статей;

- с таблицей метрик (для определения типов ссылок).

**На выходе** программа сохраняет для каждой рассмотренной статьи полученный список литературы в той же директории, где находится рассмотренная статья.

**Основные этапы работы программы:**

1. Программа обходит поочереди все файлы формата “tex”, которые может найти в указанной директории со статьями;
2. Вырезает из каждой статьи список литературы;
3. Далее по очереди обрабатывает каждую ссылку:

- очищает её от мусора;

- даёт пользователю произвести редактирование отфильтрованной ссылки, если нужно;

1. С помощью специальной таблицы метрик определяет тип ссылки;

5. Преобразовывает ссылку в формат amsbib;

- даёт пользователю произвести редактирование, если нужно, также предоставляет возможность изменить тип ссылки и выполнить заново преобразование строки.

- далее программа сохраняет новый файл со списком литературы в формате amsbib рядом с исходным файлом статьи.

6. Работа программы идёт пока пользователь не прервёт её выполнение,

либо пока не закончатся файлы статей в указанной директории.

**Основные проблемы при решении задачи:**

1. Ссылка из списка может быть написана на нескольких строчках и содержать мусор;
2. В статье может быть несколько списков литературы;
3. Список литературы некорректно оформлен;
4. Существует 7 типов ссылок, которые оформляются отлично друг от друга;
5. Некоторые типы ссылок трудно отличить друг от друга;
6. Изначально неизвестны всевозможные виды шаблонов для вычленения авторов, томов, страниц, номеров и т.д.

Вкратце рассмотрим модули программы и предоставим решения для представленного списка проблем.

Программа состоит из трёх модулей:

## goDeep.py

Это основной модуль программы, он запускает работу всех остальных модулей. Внутри него реализованы следующие функции:

**def countArticles(path)** – возвращает кол-во статей в указанной директории.

**def readAndSee(filename)** – вырезает списки литературы из tex файла, возвращает список из ссылок.

**def filterAndForget(data)** – возвращает очищенную от мусора ссылку в формате строки.

*Две предыдущих функции решают 1-ую проблему: они «собирают» ссылку из нескольких строк (если она определена на нескольких строках) в одну и очищают её от мусора.*

**def restructList(data, rawdata)** - убирает заголовок и объединяет два списка литературы в один, возвращает сокращенный список из ссылок.

*Эта функция решает 2-ую проблему – убирает дубликаты списков литературы, и если два списка написаны на русском и английских языках – объединяет их в один.*

**def findFriesAndFish(path, tablePath, ind\_end=-2) –** этот скрипт позволяет найти уже полученные списки литературы в формате amsbib для редактирования по указанному условию (на случай, если допущена ошибка при редактировании или работе скрипта).

**def goThrough(path, tablePath, ind\_start=-1)** – это главная функция, которая запускает работы программы. Она составляет список из директорий папок со статьями и идёт по очереди по найденным директориям:

- сначала находит файл со статьёй в формате “tex”, ищет в нём литературу, с помощью предыдущих трёх функций: **readAndSee, filterAndForget, restructList** получает список из очищенных ссылок. И далее начинает по очереди обрабатывать каждую ссылку. Для этого используется функция **readAndChooseThenParse** из следующего модуля.

## mainMetaParser.py

Данный модуль отвечает за редактирование полученной ссылки пользователем и выбор типа ссылки.

Для этого он использует файл таблицы метрик. В данном случае она сохранён в виде таблицы excel. Строки представляют собой наименования метрик, столбцы – наименования типов ссылок:

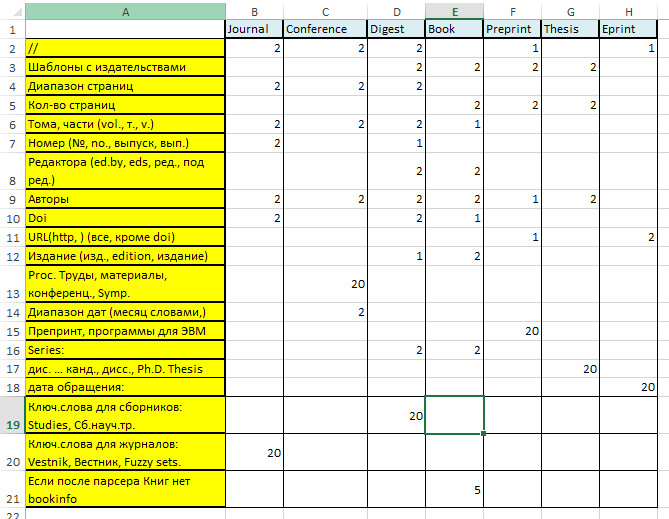


Рисунок 1 Таблица метрик

Данная таблица составлена по первоначальным сведениям о том, какие элементы могут включать отдельные типы ссылок. Данная таблица редактируется по мере выполнения работы по преобразованию ссылок в amsbib.

*Данная таблица решает 5-ую проблему – она позволяет различать такие сложные типы как конференция, главы из сборников и журналы.*

Большинство функций определенных в этом модуле соответствует наименованиям метрик из таблицы:

isThereDoubleSlash, isTherePattern, isTherePagesRange, isThereTotalPages,

isThereVolume, isThereIssue, isThereEditors, isThereAuthors,

isThereDoi, isThereReference, isThereEdition, areThereWordsLikeProc,

isThereProcInfo, areThereWordsLikePreprint, isThereSeries, areThereWordsLikeThesis,

areThereWordsLikeReference, areThereWordsLikeDigest, areThereWordsLikeJournal, isThereNoBookInfo.

Они возвращают True(1) – если ключевые слова или данные метрикой были определены в строке, иначе – False(0).

Далее идут следующие функции:

**def loadMetricsTable(filePath)** – загружает таблицу метрик из файла excel.

**def choiceFuncTable(table, line)** – прогоняет ссылку по всем метрикам таблицы. Далее получившийся вектор значений метрик (из нулей и единиц) перемножается на значения в ячейках таблицы. Из получившейся матрицы выбирается столбец, сумма значений ячеек которого наибольшая. Таким образом функция возвращает тип ссылки, указанный в заглавии выбранного столбца.

**def exchangePubls(s)** – функция для редактирования полученного формата amsbib. Меняет местами наименования «издательства» и «место издательства».

**def saveSourceStrToFile(filePath, source\_strings)** – записывает строчку с сырой/очищенной ссылкой в файл.

**def saveAmsbibToFile(filePath,amsbib)** – записывает ссылку в формате amsbib в файл.

**def cleanFile(filePath)** – стирает файл.

**def editThroughNotepad(filePath) –** открывает файл в программе notepad++ для редактирования пользователем.

**def checkSource(fileWithRez, ref) –** функция предоставляет пользователю возможность отредактировать отфильтрованную строку в notepad++, ведёт с ним интерактивный диалог через консоль.

**def readAndChooseThenParse(tablePath, filePath, refList, rawList) –**

это основная функция данного модуля, она принимает на вход сырой и очищенный от мусора списки ссылок. И начинает обработку. Сначала, предоставляет пользователю возможность исправить получившуюся ссылку, выводя «сырую» и очищенные ссылки на дисплей консоли:

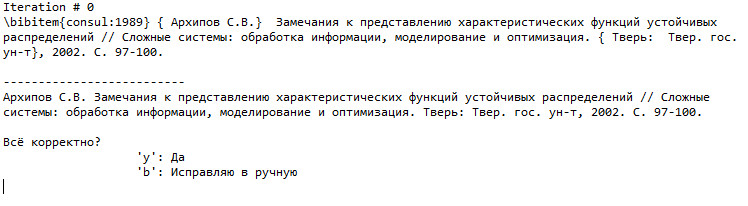


Рисунок 2 Вывод "сырой" и очищенной ссылок

Это выполняется с помощью функции **checkSource.** Далее, если всё корректно, используется функция **choiceFuncTable –** и программа определяет тип ссылки. Дальнейшие действия перекладываются на модуль parselTongue, где из класса Parser выбирается соответствующий парсер по типу ссылки. И возвращается ссылка в формате amsbib:

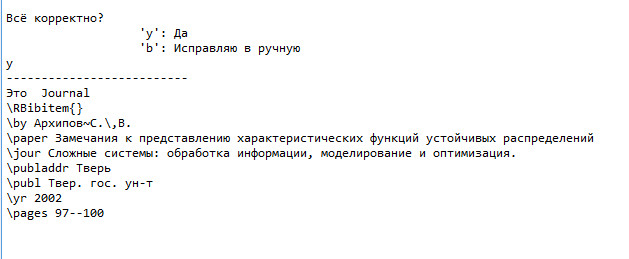


Рисунок 3 Ссылка в формате amsbib

Далее функция предоставляет ещё функционал для редактирования:

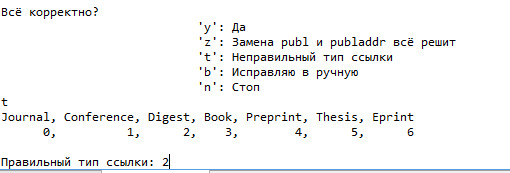


Рисунок 4 Выбор другого типа ссылки (парсера)

На этом этапе пользователь может поменять местами издательство и адрес издательства местами (это частая проблема при преобразовании, т.к. иногда трудно отличить название города от названия издательства).

Также можно поменять тип парсера на другой, если пользователь видит, что тип ссылки определен неправильно.

*Данный выбор также решает 5-ую проблему – он подстраховывает таблицу метрик на случай неправильного определения ссылки на время текущей итерации.*

Можно также продолжить работу и перейти на следующую ссылку, либо остановить процесс полностью.

Также можно прибегнуть к исправлению результатов вручную прямо во время исправления программы, не прерывая процесс обхода папок со статьями. Функция вызывает блокнот, в котором записан результат, который пользователь редактирует. Программа сохраняет исправленное.

*Данное ответвление решает сразу несколько проблем 3 и 6:*

*Если ссылка оформлена некорректно – её тут же можно исправить.*

*И можно исправить новый вид шаблона на старый вариант для вычленения корректного определения авторов, томов, страниц, номеров и т.д.*

## parselTongue.py

Данный модуль является фундаментом всех остальных.

Он состоит из двух классов со статичными функциями:

**Первый класс – HelpParse:**

Он помогает подчищать строки (убирать лишние точки и запятые по краям):

**def scratchStr(s) -** подчистить строку с двух краёв

**def scratchBegStr(s) –** подчистить начало строки

**def skipShortening(s, codeWord) –** пропустить сокращение

А также помогает извлекать основные элементы ссылок:

**def extract\_authors(s)** – авторы.

**def extract\_editors(s)** – редакторы.

**def getlangLetter(s\_check)** – язык (русский/английский).

**def findPages(s)** – диапазон страниц.

**def findDoiReference(s)** – ссылку на doi.

**def extractVolume(s)** – том, часть.

**def extractIssue(s)** – номер, выпуск.

**def extractShortName(s)** - краткая аббревиатура конференции.

**def extractSerial(s)** – серия.

**def extractInfo(s)** – информация о сборнике.

**def extractBookInfo(s)** – информация о книге.

**def extractTotalPages(s)** – количество страниц.

**def extractEdition(s)** – издание.

**def extractThesisInfo(s)** – информация о диссертации.

**def extractReference(s)** – интернет-ссылка.

**def extractArxiv(s)** – arxiv-ссылка.

**def \_RussianPatterns(string, publaddr, publ, year, procinfo='')** – шаблоны «издательство, место издания, год» на русском языке.

**def \_EnglishPatterns(string, publaddr, publ, year)** - шаблоны «издательство, место издания, год» на английском языке.

**def conferencesSpecialPatterns(string, procinfo)** – шаблоны с информацией о месте проведения конференции.

Функции в основном построены при помощи регулярных выражений, и организованы так, чтобы можно было добавлять новые шаблоны.

*Это решает 6-ую проблему, когда неизвестны некоторые виды шаблонов для вычленения авторов, томов, страниц, номеров и т.д.*

**Второй класс- Parser:**

Данный класс содержит 7 видов парсеров для конвертации строки в обычном формате в формат amsbib соотвественно для типа ссылки:

**def conf\_parser(string) –** конференция (Conference).

**def digest\_parser(string) –** главы, страницы из сборников, книг, монографий (Digest).

**def jour\_parser(string) –** журнал (Journal).

**def book\_parser(string) –** книга, монография (Book).

**def preprint\_parser(string) –** препринт, положение, закон, депонированная рукопись (Preprint).

**def thesis\_parser(string) –** диссертация.

**def eprint\_parser(string) –** электронный источник.

*Этот класс решает 4-ую проблему: предоставляет отдельному типу ссылки специфичное для него оформление.*

## Результат работы

Для журнала "Вестник Тверского государственного университета. Серия: Прикладная математика" требовалось извлечь список литературы для 391 статьи**.**

*К концу практики: извлечено и обработано 287 статей (проверено мной).*

Для журнала "Нечеткие системы и мягкие вычисления" требовалось извлечь список литературы для 72 статей**.**

*К концу практики: извлечено и обработано 72 статей (слепой прогон программы без участия пользователя).*

В итоге было обработано **359 статей из 463.**

# Результаты научно-исследовательской работы в рамках магистерской диссертации

В рамках самостоятельной научно-исследовательской работы были получены следующие результаты:

1. Написание теоретического материала в рамках магистерской диссертации, касающегося исследования детекторов ключевых точек изображений (углов, пятен), их достоинств и недостатков, исследования различных архитектур нейронной сети для применения в задаче сопоставления ключевых точек изображений;
2. Разработка ряда программ для продуцирования обучающей выборки из изображений, анализа корректности работы обученной нейронной сети (2 способами), а также проведение финального тестирования и исправления ошибок и неисправностей;
3. Проведение анализа обученных моделей сетей и их сравнение.

# Заключение

В результате прохождения производственной практики мной были закреплены навыки практического решения информационных задач на конкретном рабочем месте в качестве исполнителя. А также были применены полученные в ходе практики в организации навыки в написании магистерской диссертации.

С помощью написанной мной программы на языке программирования Python были обработаны 359 статей журналов. Для каждой из них получены списки литературы в формате amsbib. Таким образом, поставленная задача практики была выполнена.

# Список литературы

[1] Научная электронная библиотека [Электронный ресурс] <https://elibrary.ru/defaultx.asp> (дата обращения 26.02.2018)

[2] Общероссийский математический портал [Электронный ресурс] <http://www.mathnet.ru/> (дата обращения 26.02.2018)

[3] Tver State University Repository [Электронный ресурс] <http://eprints.tversu.ru/> (дата обращения 26.02.2018)

[4] Regular Expression HOWTO. (Использование регулярных выражений в Python) [Электронный ресурс] <https://docs.python.org/2/howto/regex.html> (дата обращения 26.02.2018)