Оглавление

[Введение 2](#_Toc477158192)

[Основная часть 5](#_Toc477158193)

[1. Изучение темы веб-скрапинга 5](#_Toc477158194)

[1.1 Определение 5](#_Toc477158195)

[1.2 Подходы к веб-скрапингу 7](#_Toc477158196)

[1.3 Инструменты веб-скрапинга 10](#_Toc477158197)

[1.4 Этапы веб-скрапинга 11](#_Toc477158198)

[1.5 Этические сложности веб-скрапинга 13](#_Toc477158199)

[1.6 Технические сложности веб-скрапинга 13](#_Toc477158200)

[2. Изучение рейтинга лучших университетов мира QS / QS World University Rankings. 15](#_Toc477158201)

[2.1 О рейтинге 15](#_Toc477158202)

[2.2 Методика расчета 16](#_Toc477158203)

[Заключение 18](#_Toc477158204)

[Список источников 19](#_Toc477158205)

# Введение

В современном цифровом мире данные приобретают все большую и

большую ценность. Если мы посмотрим вокруг, то увидим огромное количество различных сервисов, которые пытаются сделать нашу жизнь лучше. Стараются нам помочь, посоветовать, найти нужную информацию. От гигантов типа Google до стартапов и небольших экспериментальных проектов, все эти сервисы работают с данными. В основе любой задачи, которую предстоит сегодня решать машине или человеку, лежат данные.

Когда-то Интернет был маленькой американской сетью для нескольких сотен человек, где почти все друг друга знали. Теперь это гигантская информационная структура. Здесь практически невозможно контролировать потоки информации. То, что называется «контент, генерируемый пользователями» составляет колоссальный объем данных. Этот объем уже даже невозможно точно измерить. Точно так же как мы когда-то перестали измерять расстояние до звезд в километрах, применяемых на Земле, и стали использовать понятие «световой год», то есть характеристику скорости, также и об интернет-данных теперь пишут в терминах скорости прироста информации, а не ее объема. Так, за одну минуту в Интернете появляются, например, более 3 миллионов новых постов в сети Facebook[[1]](#footnote-1), более 350 тысяч сообщений в Twitter[[2]](#footnote-2), более 30 тысяч отзывов о книгах и покупках, не говоря уже об описаниях новых фильмов, товаров и т. д., и т. п.

Все это многообразие информации сейчас активно используется не только людьми, но и организациями для повышения эффективности своей деятельности. Жизнь людей сейчас настолько быстро меняется, что традиционных анкетных данных просто недостаточно для оценки поведения заемщика (если это банк), покупателя (если это розничная сеть). В стремительно меняющемся мире они банально устаревают. И тогда нужно обратиться к «внешним» данным. Произвести обогащение текстовыми данными. Банкам важно анализировать, что пользователи пишут в социальных сетях, на сайтах. Магазинам, производителям важно собирать отзывы покупателей об их товарах, анализировать их тональность, чтобы лучше спланировать рекламную компанию, использовать в ней текстовые формулировки, максимально близкие покупателю.

Тем не менее, для того чтобы эффективно работать с этой информацией, получать из нее пользу и реализовывать задачи, востребованные компаниями и людьми, данные нужно извлекать, обрабатывать, структурировать. То, что мы видим как веб-сайт с отзывом о фильме, для машины представляется сборищем разных «кусков» данных с непонятным назначением. Человек, взглянув на веб-страницу, сразу легко определяет нужный и значимый раздел, но для компьютера понимание того, какой именно текст следует обрабатывать, как отделить этот текст от рекламы, ненужных заголовков, ссылок является довольно сложной задачей.

По мере роста потока информации, возможностей по применению этой информации в прикладных задачах, развиваются технические подходы, объединяемые общим термином «веб-краулинг» или «веб-скрапинг». Они предназначены для сбора информации из сети Интернет и ее подготовки к автоматизированной обработке.[[3]](#footnote-3)

**Цель данной работы:** проанализировать методы и инструменты автоматизированного сбора с помощью веб-скрапинга.

**Гипотеза:** веб-скрапинг является эффективным решением для автоматизации сбора и изменения информации.

**Предмет изучения**: автоматизированный сбор информации с помощью веб-скрапинга.

**Актуальность данной работы** заключается в необходимости проведения анализа рейтинга Quacquarelli Symonds (QS) по направлениям подготовки на основе данных, полученных с помощью автоматизированного сбора.

# Основная часть

## 1. Изучение темы веб-скрапинга

### 1.1 Определение

**Что такое веб-скрапинг?**

Автоматизированный сбор данных из Интернета существует столько же, сколько сам Интернет. Несмотря на то, что веб-скрапинг (web scraping) не является новым термином, раньше это направление было больше известно под названием анализ экранных или интерфейсных данных (screen scraping), интеллектуальный анализ данных (data mining), сбор веб-данных (web harvesting). Похоже, что на сегодняшний день общее мнение склоняется в пользу термина веб-скрапинг (web scraping), который я и буду использовать в рамках данной работы.

В теории веб-cкрапинг – это сбор данных с помощью любых средств, кроме программ, использующих API. Чаще всего веб-скрапинг осуществляется с помощью программы, которая автоматически запрашивает веб-сервер, запрашивает данные (HTML и другие файлы, которые размещены на веб-страницах), а затем выполняет парсинг этих данных, чтобы извлечь необходимую информацию.

На практике веб-скрапинг охватывает широкий спектр методов и технологий программирования, таких как анализ данных и информационная безопасность.

**Зачем нужен веб-скрапинг?**

Если для вас единственным способом доступа к Интернету является браузер, вы теряете огромный спектр возможностей. Хотя браузеры удобны для выполнения JavaScript, вывода изображений и представления объектов в более удобночитаемом формате (помимо прочего), веб-скраперы удобны для сбора и обработки больших объемов данных (помимо прочего). Вместо однократного просмотра одной страницы на дисплее монитора вы можете просматривать базы данных, которые уже содержат тысячи или даже миллионы страниц.

Вы, вероятно, спросите: «Разве сбор данных – это не то, для чего используется API?» Ну, тогда API – это фантастическое средство, если вы с его помощью сразу найдете то, что искали. Они могут обеспечить передачу потока данных определенного формата с одного сервера на другой. Вы можете использовать API для получения различных видов интересующих вас данных, например, твитов или страниц Википедии. В общем случае лучше использовать API (если он есть), а не создавать программу-робот для сбора тех же самых данных. Тем не менее есть несколько причин, в силу которых использовать API не представляется возможным:

* вы собираете данные с сайтов, которые не имеют единого API;
* источник данных не имеет инфраструктуры или технических возможностей, позволяющих разработать API.

Даже в тех случаях, когда API уже есть, объем обрабатываемых запросов, ограничения скорости обработки запросов, тип или формат возвращаемых данных могут не удовлетворить ваши потребности.

Вот именно здесь и начинается сфера применения скрапинга веб-страниц. За некоторыми исключениями, вы можете просмотреть эти страницы в браузере или получить доступ к ним с помощью разработанного скрапера. Получив доступ к ним с помощью приложения, вы можете сохранить их в базе данных. Сохранив их в базе данных, вы можете выполнять любые действия с ними.

Очевидно, что существует очень много практических сфер, где требуется доступ к данным практически неограниченного объема. Рыночное прогнозирование рынка, машинный перевод и даже медицинская диагностика уже извлекли огромную пользу, воспользовавшись возможностью собрать и проанализировать данные новостных сайтов, переведенный контент и сообщения на медицинских форумах.

Независимо от вашей предметной области, почти всегда есть способ, благодаря которому веб-скрапинг может повысить эффективность бизнес-практик, улучшить производительность или даже открыть совершенно новое направление в бизнесе.[[4]](#footnote-4)

### 1.2 Подходы к веб-скрапингу

Существуют несколько подходов к извлечению данных из веб-ресурсов:

1. Ручной подход.
2. Использование регулярных выражений.
3. XML-парсинг
4. Анализ DOM-дерева, использование XPath.
5. Визуальный подход.

Рассмотрим все подходы более подробно.

1. Ручной подход.

В данном случае роль парсера выполняет человек. Он производит всю цепочку действий, необходимую для получения требуемой информации. Информация собирается обычным копипастом.

1. Использование регулярных выражений.

В парсинге регулярные выражения стоит использовать для решения небольших задач.

Его использование значительно затрудняется, когда нужно спарсить большой и сложный кусок HTML-кода, который, к тому же, не всегда соответствует какому-то определенному шаблону и вообще может содержать синтаксические ошибки.

Вообще регулярные выражения необходимо использоваться только для извлечения данных, которые имеют строгий формат — электронные адреса, телефоны и т.д., в редких случаях — адреса, шаблонные данные.

1. XML-парсинг

Еще одним неэффективным подходом является рассматривать HTML как XML данные. Причина в том, что HTML редко бывает валидным, то есть таким, что его можно рассматривать как XML данные. Библиотеки, реализовавшие такой подход, больше времени уделяли преобразованию HTML в XML и уже потом непосредственно парсингу данных. Поэтому лучше избегать этот подход.

1. Анализ DOM-дерева, использование XPath.

Этот подход основывается на анализе DOM дерева. Используя этот подход, данные можно получить напрямую по идентификатору, имени или других атрибутов элемента дерева (таким элементом может служить параграф, таблица, блок и т.д.). Кроме того, если элемент не обозначен каким-либо идентификатором, то к нему можно добраться по некоему уникальному пути, спускаясь вниз по DOM дереву, например:

body -> p[10] -> a[1] -> текст ссылки

или пройтись по коллекции однотипных элементов, например:

body -> links -> пятый элемент -> текст ссылки

Достоинства этого подхода:

* можно получить данные любого типа и любого уровня сложности
* зная расположение элемента, можно получить его значение, прописав путь к нему

Недостатки такого подхода:

* различные HTML / JavaScript движки по-разному генерируют DOM дерево, поэтому нужно привязываться к конкретному движку
* путь элемента может измениться, поэтому, как правило, такие парсеры рассчитаны на кратковременный период сбора данных
* DOM-путь может быть сложный и не всегда однозначный

Следующим эволюционным этапом анализа DOM дерева является использования XPath — т.е. путей, которые широко используются при парсинге XML данных. Суть данного подхода в том, чтобы с помощью некоторого простого синтаксиса описывать путь к элементу без необходимости постепенного движения вниз по DOM дереву.

1. Визуальный подход

Предпринимаются попытки с использованием машинного обучения и компьютерного зрения, которые пытаются идентифицировать и извлекать информацию с веб-страниц, интерпретируя страницы визуально как человек.

В данный момент визуальный подход находится на начальной стадии развития. Суть подхода в том, чтобы пользователь мог без использования программного языка или API «настроить» систему для получения нужных данных любой сложности и вложенности. Думаю, что парсеры будущего будут именно визуальными.[[5]](#footnote-5)

Проанализировав достоинства и недостатки выше перечисленных подходов, мы решили, что для нашей работы подойдет анализ DOM-дерева. Это достаточно мощный и в то же время не сложный для освоения подход к веб-скрапингу, который полностью отвечает всем заданным требованиям.

### 1.3 Инструменты веб-скрапинга

Для автоматизированного извлечения информации с веб-страниц существует 4 типа инструментов:

1. Библиотеки.

Этот подход требует понимания процесса формирования запросов и логики работы приложения, что влечет за собой дополнительные расходы на изучение низкоуровневой работы сайта. Возможно, это уместно и оправдано для единичного сайта, но в случае если требуется написать несколько парсеров для нескольких разнородных сайтов за ограниченное время — вряд ли.

К таким инструментам относятся многочисленные библиотеки для различных языков программирования: *jSoup* для Java, *SimpleHTMLDom* для PHP, *lxml.html* для Python и другие.

1. Headless-браузеры.

Данный подход позволяет обрабатывать страницу в браузере с поддержкой JavaScript, что позволяет писать свои сценарии для получения требуемой информации и даже использовать JavaScript библиотеки вроде jQuery для извлечения информации со страницы, что ускоряет разработку парсеров. Отсутствие графического интерфейса позволяет запускать данные браузеры даже на серверах, поддерживающих только консольный режим.

К таким инструментам можно отнести PhantomJS и SlimerJS.

1. SaaS решения.

Данные сервисы предоставляют графический интерфейс, с помощью которого можно указать адрес страницы, указать блоки, из которых нужно извлечь информацию, а также создать ряд правил по извлечению данных. Такие сервисы не обладают той гибкостью, которую предоставляют низкоуровневые решения. Некоторые из них стоят довольно дорого, зато ими просто пользоваться.

К таким сервисам можно отнести Mozenda, Octoparse (бесплатный), Import.io (бесплатный).

1. Настольные приложения.

Предоставляют графический интерфейс с возможностью задания правил для извлечения информации со страницы. Как и SaaS решения, они облегчают работу пользователя, избавляя его от необходимости написания кода, обладают определенной гибкостью, но не могут сравниться с решениями, разрабатываемыми под конкретный сайт.

Пример: IRobotSoft.

Основная проблема автоматических парсеров — они перестают работать при изменении логики работы сайта. Поэтому важно проектировать парсеры таким образом, чтобы они могли оповещать о невозможности извлечения необходимой информации, а в случае необходимости их можно было адаптировать под произошедшие изменения.[[6]](#footnote-6)

Поскольку мы собираемся разрабатывать собственное веб-приложение, и мы решили использовать подход анализа DOM-дерева, то в качестве инструмента для веб-скрапинга мы будем использовать библиотеки.

### 1.4 Этапы веб-скрапинга

Парсинг html-страницы представляет из себя процесс, который можно разбить на три этапа:

1. **Получение исходного кода веб-страницы.** В разных языках для этого предусмотрены различные способы, Например, в PHP чаще всего используют библиотеку *cURL* или же встроенную функцию *file\_get\_contents*.
2. **Извлечение из html-кода необходимых данных.** Получив страницу, необходимо обработать её – отделить обычный текст от гипертекстовой разметки, выстроить иерархическое дерево элементов документа, корректно среагировать на невалидный код, извлечь со страницы именно ту информацию, ради которой и затевается весь сыр-бор.
3. **Фиксация результата.** Благополучно обработав данные на странице, требуется их сохранить в необходимом виде для последующей обработки. Спарсенное обычно заносится в базу данных, однако есть и другие варианты. Иногда требуется записать в *CSV-файл* или строить иерархические *JSON-структуры*, иногда сконвертировать в *excel-таблицу*, а может даже сгенерировать динамический *rss-поток*.

Как правило, требуется спарсить не одну страницу сайта-донора, а множество, может даже и все. В этом случае после прохождении шагов 1-3 в алгоритм парсера должен быть заложен переход на следующую страницу сайта, дабы и с неё извлечь необходимый материал.

Обход всех нужных страниц сайта обеспечивается разными способами.

**Во-первых,** обрабатывая очередную страницу, парсер можно научить не только извлекать необходимые данные, но и заносить в свою базу данных все внутренние ссылки, встречающиеся по пути. Обращаясь к своему хранилищу ссылок, программа последовательно посещает страницы сайта, до тех пор пока не обойдёт их все.

**Во-вторых,** при первичном анализе сайта зачастую возможно проследить логику формирования url для страниц. И затем, генерировать адреса в соответствии с выявленными закономерностями.

**В-третьих,** некоторые парсеры рассчитаны, как ни странно, на «ручной» обход веб-ресурса. Пользователь, кликая по ссылкам, сам решает какие страницы посещать, какие нет. А программа в фоновом режиме запоминает необходимые данные.

Разумеется, совмещать разные методы ничто не мешает.[[7]](#footnote-7)

### 1.5 Этические сложности веб-скрапинга

Вопрос о том, является ли парсинг воровством контента, активно обсуждается во Всемирной сети. Большинство оппонентов считают, что заимствование части контента, не являющегося интеллектуальной собственностью, например, технических описаний, допустимо. Ссылка на первоисточник контента рассматривается как способ частичной легитимации. В то же время, наглое копирование, включая грамматические ошибки, осуждается интернет-сообществом, а поисковыми системами рассматривается как повод для блокировки ресурса.[[8]](#footnote-8)

### 1.6 Технические сложности веб-скрапинга

Кроме этических проблем парсер способен создать и технические. Он автомат, робот, но его вход на сайт фиксируется, а входящий и исходящий трафики учитываются. Количество подключений к веб-ресурсу в секунду устанавливает создатель программы. Делать этот параметр очень большим нельзя, поскольку сервер может не переварить потока входящего трафика. При частоте 200–250 подключений в секунду работа парсера рассматривается как аналогичная *DOS-атаке*. В связи с этим веб-мастера могут применять различные меры борьбы с автоматическими парсерами, а именно:

1. Динамическое изменение структуры кода страницы с целью усложнения извлечения информации из блоков.
2. Показ капчи при авторизации/регистрации/превышении числа запросов за единицу времени.
3. Блокировка клиентов по среднему объему трафика в единицу времени.
4. Анализ поведения клиентов и блокировка/требование пройти капчу для продолжения работы при подозрительном поведении.

## 2. Изучение рейтинга лучших университетов мира QS / QS World University Rankings.

### 2.1 О рейтинге

Рейтинг лучших университетов мира (QS World University Rankings) — глобальное исследование и сопровождающий его рейтинг лучших высших учебных заведений мирового значения по показателю их достижений в области образования и науки. Рассчитан по методике британской консалтинговой компании Quacquarelli Symonds (QS). Считается одним из наиболее влиятельных глобальных рейтингов университетов.

Рейтинг оценивает университеты по следующим показателям: активность и качество научно-исследовательской деятельности, мнение работодателей и карьерный потенциал, преподавание и интернационализация. Эти показатели охватывают ключевые стратегические миссии университетов мирового значения, за которые они отвечают перед участниками процесса: академическим сообществом, работодателями, учащимися и их родителями. Ежегодно в исследовании оцениваются свыше 2,5 тысяч высших учебных заведений по всему миру. По его итогам составляется рейтинг 500 лучших университетов мира, а также рейтинги университетов по отдельным дисциплинам.

Уровень достижений университетов оценивается на основании результатов комбинации статистического анализа деятельности учебных заведений, аудированных данных (включая информацию по индексу цитирования из базы данных Scopus, крупнейшей в мире библиометрической базы данных научных публикаций), а также данных глобального экспертного опроса представителей международного академического сообщества и работодателей, которые высказывают свои мнения об университетах.

### 2.2 Методика расчета

|  |  |
| --- | --- |
| Показатель | Вес |
| Индекс академической репутации ВУЗа. Показатель включает данные глобального экспертного опроса представителей международного академического сообщества, который отражает мнения руководства университетов, профессоров и преподавателей, ведущих научно-исследовательскую деятельность, о том, в каких учебных заведениях мира научные исследования по их сфере компетенций проводятся на самом высоком уровне. Учитываются данные за последние три года. | 40% |
| Индекс репутации ВУЗа среди работодателей. Показатель включает данные глобального экспертного опроса представителей организаций-работодателей, охватывающих все основные индустрии и имеющих размер от ста сотрудников и выше. В рамках опроса изучаются мнения руководства организаций-работодателей о профессиональной компетенции выпускников ВУЗов, принятых ими на работу. | 10% |
| Соотношение профессорско-преподавательского состава ВУЗа и численности обучающихся. Показатель включает данные о соотношении профессорско-преподавательского состава к числу студентов и содержит как сведения самих ВУЗов, так и сведения государственных организаций, курирующих сферу образования. | 20% |
| Индекс цитирования научных работ преподавательского состава ВУЗа по отношению к численности преподавательского состава. Показатель включает соотношение общей цитируемости научных публикаций и общей численности преподавателей и исследователей, работающих в ВУЗе на условиях полной занятости на протяжении как минимум одного семестра. Индекс цитирования научных работ оценивается на основе библиометрической базы данных Scopus. Учитываются данные за последние пять лет. | 20% |
| Доля иностранных преподавателей ВУЗа по отношению к общей численности преподавательского состава. Показатель включает долю иностранных преподавателей, работающих в ВУЗе на условиях полной занятости (по эквиваленту полной ставки) и проводящих в университете не менее одного семестра. | 5% |
| Доля иностранных студентов по отношению к общей численности, обучающихся в ВУЗе. Показатель включает долю иностранных студентов, обучающихся в ВУЗе по программам полного цикла обучения. Учитываются студенты, являющиеся гражданами стран, отличных от страны обучения, обучающиеся не менее одного семестра и не являющиеся студентами по обмену.[[9]](#footnote-9) | 5% |

Благодаря анализу данного рейтинга мы сможем рассмотреть динамику продвижения нашего Казанского федерального университета (КФУ) по рейтингу, увидим разницу с мировыми и российскими университетами, поймем в каких аспектах нужно работать и на что следует обратить особое внимание.

# Заключение

В работе было дано теоретическое обоснование автоматизированного сбора информации в виде веб-скрапинга, проанализированы подходы и инструменты для его проведения. На основе полученных теоретических знаний было получено понимание алгоритма работы данной технологии, и какие трудности могут возникнуть при реализации и работе веб-приложения, которое реализует автоматизированный сбор данных на основе веб-скрапинга. Также был рассмотрен мировый рейтинг QS, над которым будет производиться работа анализа данных, полученных на основе автоматизированного сбора с помощью веб-скрапинга.

# Список источников

1. How Much Data is Created on the Internet Each Day? - Gwava [Электронный ресурс]. – Режим доступа: https://www.gwava.com/blog/internet-data-created-daily. – (Дата обращения: 08.03.2017).
2. Twitter Usage Statistics – Internet live stats [Электронный ресурс]. – Режим доступа: http://www.internetlivestats.com/twitter-statistics/. – (Дата обращения: 08.03.2017).
3. Mitchell R. Web Scraping with Python: Collecting Data from the Modern. – O'Reilly Media, 2015. – 10-11 p.
4. Mitchell R. Web Scraping with Python: Collecting Data from the Modern. – O'Reilly Media, 2015. – 13-16 p.
5. Подходы к извлечению данных из веб-ресурсов – Хабрахабр [Электронный ресурс]. – Режим доступа: https://habrahabr.ru/post/99918/. – (Дата обращения: 08.03.2017).
6. Веб-парсинг данных – INOSTUDIO – студийные записи [Электронный ресурс]. – Режим доступа: https://inostudio.com/ru/article/web-parsing.html. – (Дата обращения: 08.03.2017).
7. Этапы парсинга – Парсинг: Что? Зачем? Как? [Электронный ресурс]. – Режим доступа: http://parsing.valemak.com/ru/what-why-how/stages-of-parsing/. – (Дата обращения: 09.03.2017).
8. Парсинг. Что это и где используется – iPipe [Электронный ресурс]. – Режим доступа: https://www.ipipe.ru/info/parsing.html/. – (Дата обращения: 09.03.2017).
9. РЕЙТИНГ ЛУЧШИХ УНИВЕРСИТЕТОВ МИРА ПО ВЕРСИИ QUACQUARELLI SYMONDS – Гуманитарная энциклопедия [Электронный ресурс]. – Режим доступа: http://gtmarket.ru/ratings/qs-world-university-rankings/info/. – (Дата обращения: 10.03.2017).

1. How Much Data is Created on the Internet Each Day? - Gwava [Электронный ресурс]. – Режим доступа: https://www.gwava.com/blog/internet-data-created-daily. – (Дата обращения: 08.03.2017). [↑](#footnote-ref-1)
2. Twitter Usage Statistics – Internet live stats [Электронный ресурс]. – Режим доступа: http://www.internetlivestats.com/twitter-statistics/. – (Дата обращения: 08.03.2017). [↑](#footnote-ref-2)
3. Mitchell R. Web Scraping with Python: Collecting Data from the Modern. – O'Reilly Media, 2015. – 10-11 p. [↑](#footnote-ref-3)
4. Mitchell R. Web Scraping with Python: Collecting Data from the Modern. – O'Reilly Media, 2015. – 13-16 p. [↑](#footnote-ref-4)
5. Подходы к извлечению данных из веб-ресурсов – Хабрахабр [Электронный ресурс]. – Режим доступа: https://habrahabr.ru/post/99918/. – (Дата обращения: 08.03.2017). [↑](#footnote-ref-5)
6. Веб-парсинг данных – INOSTUDIO – студийные записи [Электронный ресурс]. – Режим доступа: https://inostudio.com/ru/article/web-parsing.html. – (Дата обращения: 08.03.2017). [↑](#footnote-ref-6)
7. Этапы парсинга – Парсинг: Что? Зачем? Как? [Электронный ресурс]. – Режим доступа: http://parsing.valemak.com/ru/what-why-how/stages-of-parsing/. – (Дата обращения: 09.03.2017). [↑](#footnote-ref-7)
8. Парсинг. Что это и где используется – iPipe [Электронный ресурс]. – Режим доступа: https://www.ipipe.ru/info/parsing.html/. – (Дата обращения: 09.03.2017). [↑](#footnote-ref-8)
9. РЕЙТИНГ ЛУЧШИХ УНИВЕРСИТЕТОВ МИРА ПО ВЕРСИИ QUACQUARELLI SYMONDS – Гуманитарная энциклопедия [Электронный ресурс]. – Режим доступа: http://gtmarket.ru/ratings/qs-world-university-rankings/info/. – (Дата обращения: 10.03.2017). [↑](#footnote-ref-9)