**Содержание**

|  |  |
| --- | --- |
|  | Стр. |
| Введение………………………………………………………………. | 3 |
| 1 Основные принципы разработки Web приложений на Django……  2 Описание бизнес процесса по оформлению накладных в АО «ФНПЦ «ННИИРТ» и предложение по его оптимизации………….. | 4  9 |
| 3 Пример реализации Web приложения для оптимизации бизнес процесса по оформлению накладных на Django……………………. | 10 |
| Заключение……………………………………………………………. | 13 |
| Список литературы…………………………………………………… | 14 |

**Введение**

Деятельность компаний состоит из огромного количества повторяющихся бизнес-процессов, каждый из которых представляет собой последовательность действий и решений, направленных на достижение определенной цели. Прием заказа клиента, доставка товара клиенту, начисление зарплаты сотрудникам, оформление накладных и других документов. Эффективность деятельности компании в значительной степени определяется эффективностью реализации бизнес-процессов. В настоящее время даже в передовых, с точки зрения управленческих технологий, компаниях существуют возможности повышения эффективности отдельных подразделений и компании в целом путём оптимизации бизнес-процессов.

На крупном предприятии, имеющем большое количество контрагентов, в том числе по кооперации, таком как АО «ФНПЦ «ННИИРТ» каждый день оформляется большое количество накладных для передачи различных товарно-материальных ценностей.

В данной работе будет рассмотрен бизнес процесс по оформлению накладных в АО «ФНПЦ «ННИИРТ» и возможность оптимизации данного бизнес процесса с помощью web сервиса, разработанного на фреймворке Django.

1. **Основные принципы разработки Web приложений на Django.**

Django органично росла во время разработки реальных приложений, созданных командой разработки в Лоуренсе, штат Канзас, США. Она родилась в конце 2003 когда программисты газеты «Lawrence Journal-World», Эдриан Холовати (Adrian Holovaty) и Симон Виллисон (Simon Willison), начали использовать язык Python для разработки своих приложений. Команда World Online, ответственная за разработку и поддержку нескольких локальных новостных сайтов, процветала в среде разработчиков, диктуемой срочностью работы журналиста. Для сайтов, включая LJWorld.com, Lawrence.com и KUsports.com, журналисты и менеджеры требовали, чтобы новые возможности и целые приложения были разработаны максимально быстро, часто счёт шёл на дни или на часы. Таким образом, Симон и Эдриан создали среду разработки экономящую время, исходя из своих потребностей — это был единственный способ, с помощью которого они могли создавать управляемые приложения в жёстких рамках сроков.

Летом 2005, развив эту среду до такого состояния, что она поддерживала большинство сайтов World Online, команда разработчиков, включая Якоба Каплан-Мосса (Jacob Kaplan-Moss), решила выпустить среду в виде программного обеспечения с открытым исходным кодом. Они выпустили его в июле 2005 под именем Django в честь джазового гитариста Джанго Рэйнхарда (Django Reinhardt).

Сейчас Django является грамотным проектом с открытым исходным кодом с десятками тысяч пользователей, над которым работают разработчики по всей планете. Два автора из команды World Online («Великодушные пожизненные диктаторы», Эдриан и Якоб) всё ещё осуществляют общее руководство над развитием среды разработки, но теперь оно в большей степени зависит от совместных усилий команды разработчиков. World Online обеспечивает другие важные аспекты, такие как время разработчиков, маркетинговые материалы и хостинг/канал для сайта среды (http://www.djangoproject.com/).

Эта история уместна, потому что помогает объяснить два ключевых аспекта. Первый аспект — основное назначение Django. Так как Django родился в новостной среде, он предоставляет некоторые средства (такие как административный интерфейс), которые хорошо подойдут для контент-ориентированных сайтов, таких как Amazon.com, craigslist.org и washingtonpost.com, которые предоставляют динамическую информацию из базы данных. Существует разница между быть особенно эффективным в чём-то и быть неэффективным в другом.

Вторым аспектом является то, как происхождение Django сформировало культуру его сообщества открытого исходного кода. Так как Django был получен из реального кода, а не был академической разработкой или коммерческим продуктом, он полностью сфокусирован на решение проблем разработки, с которыми сталкивались и продолжают сталкиваться его авторы. В результате, Django постоянно совершенствуется. Основатели среды имеют свой интерес в том, чтобы Django экономил их время, создавал приложения лёгкие в обслуживании и хорошо работал под нагрузкой. При отсутствии других причин, разработчики мотивируются своими собственными эгоистичными желаниями сэкономить своё время и наслаждаться своей работой.

Django — свободный фреймворк для веб-приложений на языке Python, использующий шаблон проектирования MVC. Проект поддерживается организацией Django Software Foundation. MVC – схема разделения данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента: модель, представление и контроллер — таким образом, что модификация каждого компонента может осуществляться независимо.



Рисунок 1. Структура MVC шаблона.

Сайт на Django строится из одного или нескольких приложений, которые рекомендуется делать отчуждаемыми и подключаемыми. Это одно из существенных архитектурных отличий этого фреймворка от некоторых других (например, Ruby on Rails). Один из основных принципов фреймворка — DRY (англ. Don't repeat yourself). Этот принцип разработки программного обеспечения, нацеленный на снижение повторения информации различного рода, особенно в системах со множеством слоёв абстрагирования. Принцип DRY формулируется как: «Каждая часть знания должна иметь единственное, непротиворечивое и авторитетное представление в рамках системы». Он был сформулирован Энди Хантом и Дэйвом Томасом в их книге The Pragmatic Programmer. Они применяли этот принцип к «схемам баз данных, планам тестирования, сборкам программного обеспечения, даже к документации». Когда принцип DRY применяется успешно, изменение единственного элемента системы не требует внесения изменений в другие, логически не связанные элементы. Те элементы, которые логически связаны, изменяются предсказуемо и единообразно.

В отличие от других фреймворков, обработчики URL в Django конфигурируются явно при помощи регулярных выражений. Для работы с базой данных Django использует собственный ORM, в котором модель данных описывается классами Python, и по ней генерируется схема базы данных.

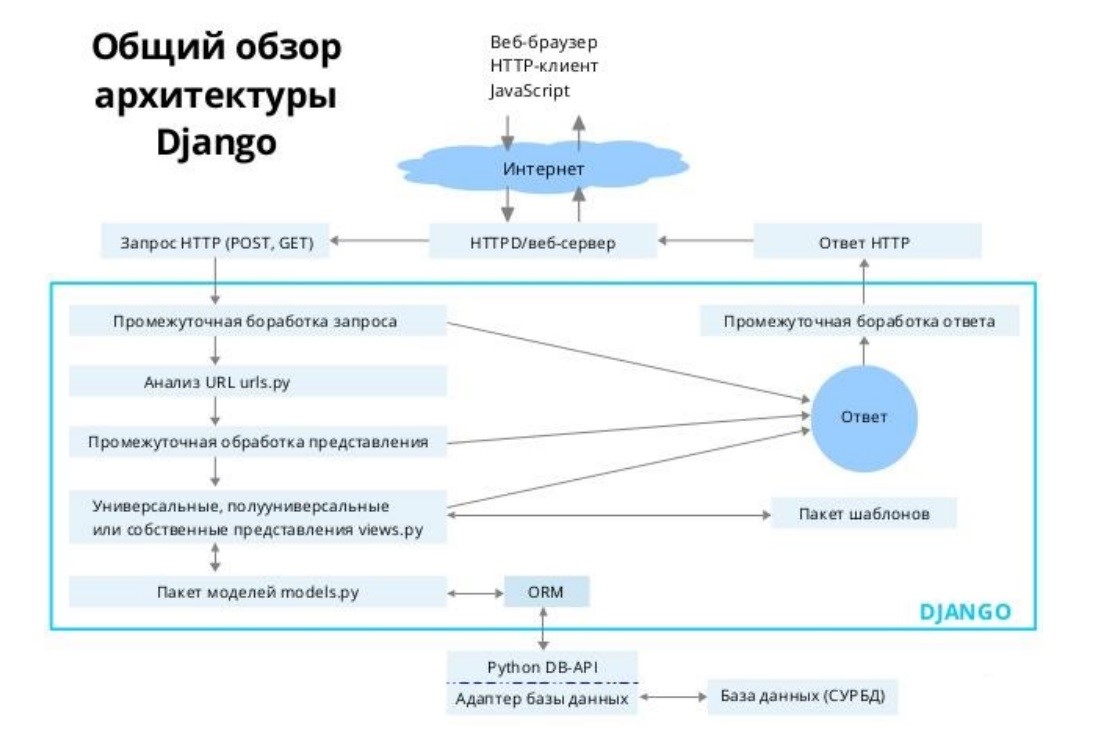


Рисунок 2. Общий обзор архитектуры Django

Рассмотрим более детально процесс создания Django-проектов.

Первоначальная разработка Django, как средства для работы новостных ресурсов, достаточно сильно отразилась на его архитектуре: он предоставляет ряд средств, которые помогают в быстрой разработке веб-сайтов информационного характера. Так, например, разработчику не требуется создавать контроллеры и страницы для административной части сайта, в Django есть встроенное приложение для управления содержимым, которое можно включить в любой сайт, сделанный на Django, и которое может управлять сразу несколькими сайтами на одном сервере. Административное приложение позволяет создавать, изменять и удалять любые объекты наполнения сайта, протоколируя все совершённые действия, и предоставляет интерфейс для управления пользователями и группами.

Django проект может включать в себя несколько приложений, каждое из которых состоит из четырех основных компонентов.

1. Модель данных (models): данные являются сердцевиной любого современного Web-приложения. Модель - важнейшая часть приложения, которое постоянно обращается к данным при любом запросе из любой сессии. Любая модель является стандартным Python классом. Объектно-ориентированный маппер (ORM) обеспечивает таким классам доступ непосредственно к базам данных. Если бы не было ORM, программисту пришлось бы писать запросы непосредственно на SQL. Модель обеспечивает облегченный механизм доступа к слою данных, инкапсулирует бизнес-логику. Модель не зависит от конкретного приложения. Данными можно манипулировать даже из командной строки, не используя при этом Web-сервер.

2. Представление (views): выполняют разнообразные функции, в том числе контролируют запросы пользователя, выдают контекст в зависимости от его роли. View - это обычная функция, которая вызывается в ответ на запрос какого-то адреса (URL) и возвращает контекст.

3. Шаблоны: являются формой представления данных. Шаблоны имеют свой собственный простой метаязык и являются одним из основных средств вывода на экран.

4. URL: механизм внешнего доступа к представлениям (views). Встроенные в URL регулярные выражения делают механизм достаточно гибким. При этом одно представление может быть сконфигурировано к нескольким урлам, предоставляя доступ различным приложениям. Здесь поддерживается философия закладок: URL становятся самодостаточными и начинают жить независимо от представления.

В дистрибутив Django также включены приложения для системы комментариев, синдикации RSS и Atom, «статических страниц» (которыми можно управлять без необходимости писать контроллеры и представления), перенаправления URL и другое.

1. **Описание бизнес процесса по оформлению накладных в АО «ФНПЦ «ННИИРТ» и предложение по его оптимизации.**

Несмотря на достаточно развитую инфраструктуру АО «ФНПЦ «ННИИРТ» и наличия корпоративной информационной сети бизнес процесс по оформлению накладных в АО «ФНПЦ «ННИИРТ» не оптимизировался уже более 10 лет. Ответственным за оформление накладных является экономист отдела сбыта. В отдел сбыта поступают служебные записки на оформление накладных от всех подразделений организации. Процесс оформления накладных достаточно прост, но занимает почти все рабочее время из-за большого количества заявок, за год оформляется порядка 1500 накладных.

В основном в организации используются две формы накладных: М-15 и Торг-12.

Накладная на отпуск материалов на сторону. Форма М-15. Форма утверждена постановлением Госкомстата РФ от 30.10.1997 N 71а.

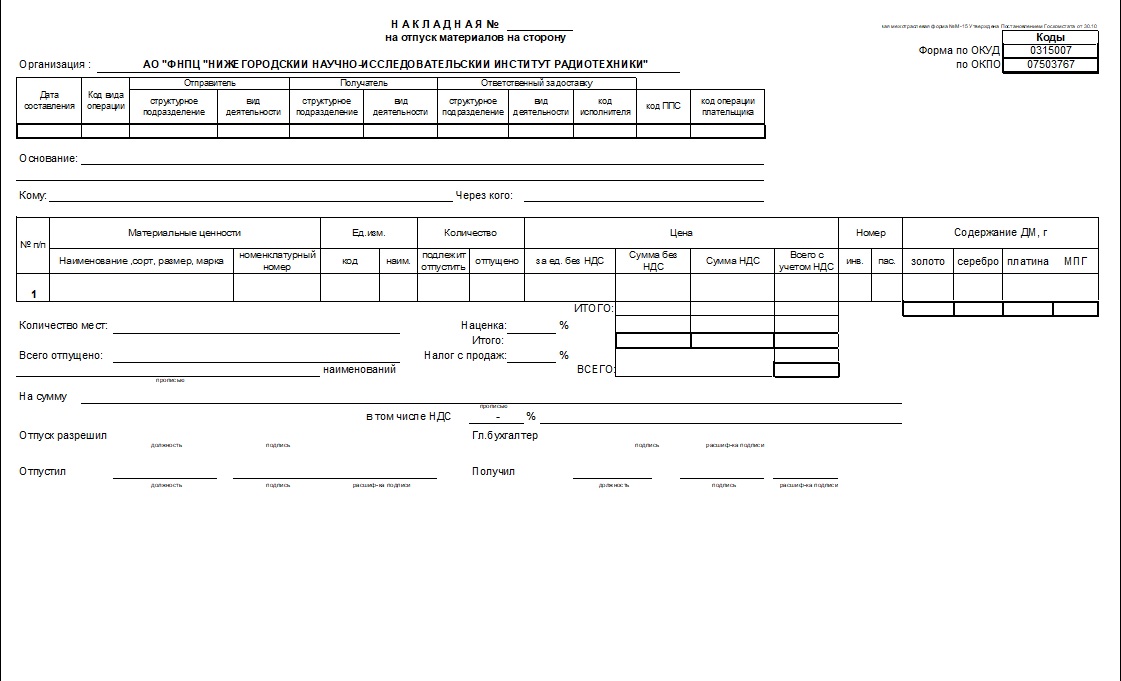


Рисунок 3. Образец накладной формы М-15.

При передаче материальных ценностей из одного предприятия в другое, либо представителям компании, которые находятся вне территории фирмы, используется накладная на отпуск материалов на сторону. Для этих целей применяется типовая форма М-15. Выписывается данная накладная в двух экземплярах для организации, отпускающей товар, и для получателя соответственно. Основанием для оформления накладной служит служебная записка на имя начальника отдела сбыта. Важным фактом является обязательное наличие доверенности у получателя ТМЦ, которую он предъявляет работнику подразделения, передающего имущество, для получения товарных ценностей.

Для оформления накладной используется форма файла, заполняемая в Microsoft Excel. Требуется указать дату выписки и номер документа, код операции (при необходимости) и наименование подразделения, поставляющего товар. Затем вносятся сведения о материальных ценностях (наименование и характеристика), наименование и код единицы измерения, номенклатурный номер (при наличии). В 7-8 строках заполняется количество материалов, необходимых для отпуска по сопроводительному документу, и количество отпущенных ТМЦ. Указывается стоимость единицы без учета налога. В 10 строке подсчитывается общая стоимость отпущенных материалов. Затем прописывается общая сумма НДС по всем материалам и общая сумма ТМЦ с учетом налога.

Ниже заполняется инвентарный номер товара, номер технического паспорта (только для драгоценных металлов и камней) и номер записи из карты учета. В конце формы прописью указывается число, сумма и НДС всех отпущенных материалов. Подписывается накладная сотрудником, разрешившим отпуск ТМЦ, отпустившим материалы, главным бухгалтером предприятия и представителем получателя.

Для передачи материальных ценностей в другие организации оформляются накладные, служащие подтверждением факта передачи имущества.

Форма № ТОРГ-12 «Товарная накладная» утв. Постановлением Госкомстата от 25.12.1998 № 132.

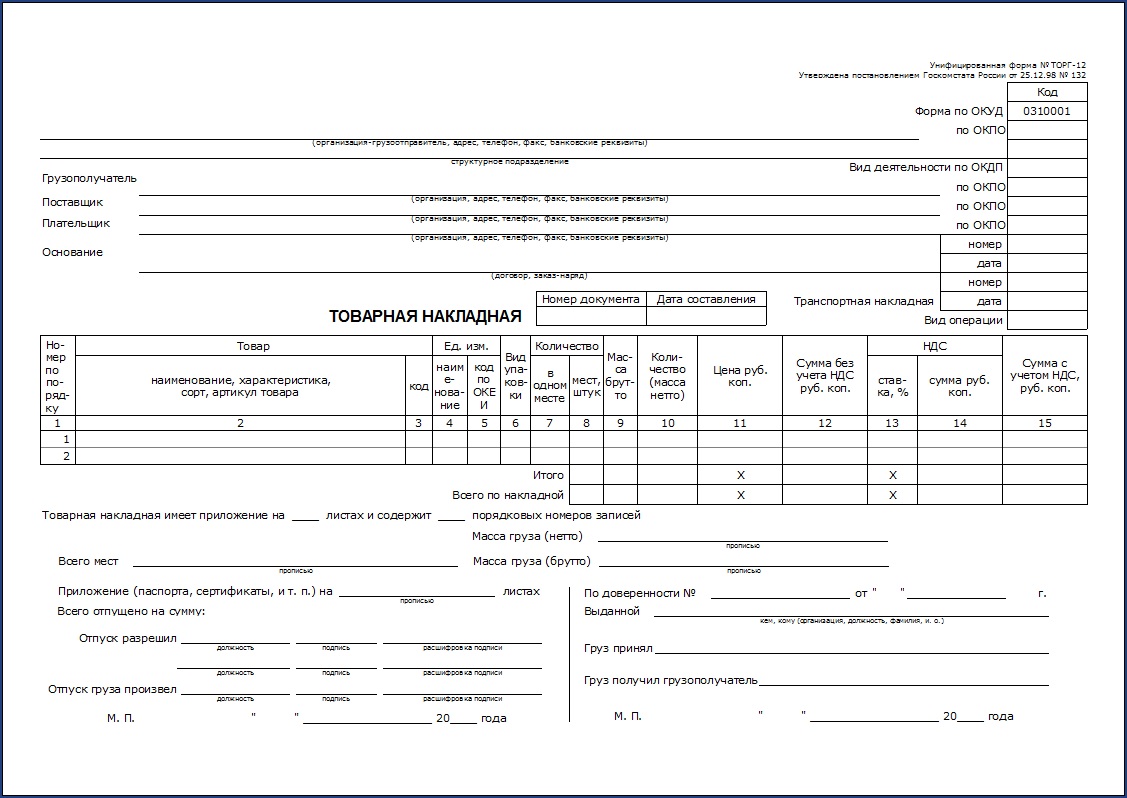


Рисунок 4. Образец формы Торг-12.

Накладная ТОРГ-12 применяется для оформления продажи (отпуска) товарно-материальных ценностей сторонней организации. Накладная составляется в 2 экземплярах. Первый экземпляр остается у организации, сдающей товарно-материальные ценности (т. е. у поставщика), и является основанием для их списания. Второй экземпляр передается сторонней организации (покупателю) и является основанием для оприходования этих ценностей.

Для того, чтобы накладная ТОРГ-12 могла считаться первичным учетным документом, необходимо контролировать наличие в ней, в частности, следующих обязательных реквизитов (ч. 2 ст. 9 Федерального закона от 06.12.2011 № 402-ФЗ):

* дата составления накладной;
* наименование составителя накладной (поставщика);
* наименование отпускаемых ТМЦ, их натуральное и денежное измерение;
* наименование должностей лиц, ответственных за отпуск ТМЦ и оформление накладной, а также фамилии, инициалы и подписи таких лиц.

Чтобы из накладной № ТОРГ-12 явно следовало, что это именно продажа ТМЦ, целесообразно заполнять реквизит «Основание». Здесь можно указать наименование договора, по которому отпускаются ТМЦ. Ведь ТОРГ-12 может быть оформлена и на возврат товаров – тогда в основании указывается, что это возврат.

Важно контролировать, чтобы со стороны получателя ТОРГ-12 была подписана уполномоченными лицами. К примеру, руководителем организации или лицом, действующим на основании доверенности. В последнем случае к накладной необходимо приложить такую доверенность.

Разберем подробно существующий в АО «ФНПЦ «ННИИРТ» бизнес процесс оформления накладных.

Подразделение, инициирующее передачу имущества направляет в отдел сбыта служебную записку, в которой в обязательном порядке указывает всю информацию, необходимую для заполнения формы накладной экономистом отдела сбыта:

1. Номер служебной записки
2. Структурное подразделение отправитель
3. Основание для отправки (письмо, рекламационный акт, протокол тех. Совещания и т.д.)
4. Сокращенное наименование предприятия получателя, его адрес
5. Перечень имущества, его количество, единицы измерения
6. Стоимость имущества (с НДС или без).
7. Количество мест
8. Дата вывоза.
9. Исполнитель и контактный телефон.

Далее экономист открывает форму накладной и заполняет её в соответствии со служебной запиской, после чего распечатывает и подписывает у руководства организации.

При данном построении бизнес процесса имеются следующие недостатки:

* Возможность появления ошибок. Экономисту отдела сбыта приходится еще раз вручную набирать тот же текст, который напечатан в служебной записке. При этом так же имеет место человеческий фактор – экономист может совершать опечатки при наборе данных.
* Потеря времени. Много времени уходит для перепечатывания текста из служебной записки в форму накладной и зависит от количества товаров (в некоторых случая может занимать более часа).
* Неполное указание данных. Служебная записка пишется в свободной форме, зачастую данные для оформления накладной указываются не полностью, при этом экономисту отдела сбыта приходится повторно обращаться в подразделение, отправляющее продукцию, за недостающими данными.

Для оптимизации бизнес процесса по оформлению накладных, достаточно создать веб сервис по подаче заявок в отдел сбыта на оформление накладных через единый информационный портал АО «ФНПЦ «ННИИРТ». Страница подачи заявок на оформление накладных будет включать в себя обязательные для заполнения поля и передавать их содержимое в базу данных. Далее экономист отдела сбыта через административную панель будет экспортировать данные из базы данных в формате .xls для последующего автоматического заполнения формы накладной. Данный способ сможет исключить ошибки при оформлении накладных экономистом, сэкономить время и избавить от необходимости приносить напечатанные служебные записки в отдел сбыта. Подробно реализацию данного проекта разберем следующей главе.

1. **Пример реализации Web приложения на Django, для оптимизации бизнес процесса по оформлению накладных.**

**Заключение**

В данной работе был разобран принцип построения экспертной системы на основе разработки модели на языке Python.

Были рассмотрены основные алгоритмы машинного обучения, а так же подробно рассмотрен алгоритм Decision Tree (дерево решений) и композиционный алгоритм Random Forest (случайный лес) базовыми элементами которого являются решающие деревья. В качестве базы знаний была взята база цен недвижимости пригорода N. Экспертная система была реализована с помощью библиотек pandas и scikit-learn, модель обучена и найдена её точность с использование кросс валидации. Была получена точность 87,5 %, которая является достаточно высокой для регрессионной модели.

**Список литературы**

1. Medium [Электронный ресурс]. – https://medium.com/@williamkoehrsen/random-forest-simple-explanation – (дата обращения 22.03.2019)
2. Scikit-learn [Электронный ресурс]. – http://scikit-learn.org/ – (дата обращения 22.03.2019)
3. Википедия – свободная энциклопедия [Электронный ресурс]. – http://wikipedia.org . – (дата обращения 22.03.2019)
4. Джоэл Грас, Data Science. Наука о данных с нуля / Джоэл Грас – BHV, 2017 – 336 c
5. Jake VanderPlas, Python DataScience Handbook / Jake VanderPlas – O’Reilly, 2017 – 529 c
6. Гасанов, Э.Э. Интеллектуальные системы. Теория хранения и поиска информации [Электронный ресурс] : учебник / Э. Э. Гасанов, В. Б. Кудрявцев. - 2-е изд. ; испр. и доп. - Электрон. дан. - М.: Издательство Юрайт, 2018. - 289 с.
7. Карпова, Т.С. Базы данных. Модели, разработка, реализация [Электронный ресурс] : учебное пособие / Т. С. Карпова. - Электрон. дан. – М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. - 403 c
8. Безопасность систем баз данных [Электронный ресурс]: учебное пособие/ А.В. Скрыпников [и др.].— Электрон. текстовые данные.— Воронеж: Воронежский государственный университет инженерных технологий, 2015.— 144 c.
9. Гордеев, С.И. Организация баз данных в 2 ч. Часть 1 [Электронный ресурс] : учебник / С. И. Гордеев, В. Н. Волошина. - 2-е изд. ; испр. и доп. - Электрон. дан. - М.: Издательство Юрайт, 2018. - 311 с
10. Гордеев, С.И. Организация баз данных в 2 ч. Часть 2 [Электронный ресурс] : учебник / С. И. Гордеев, В. Н. Волошина. - 2-е изд. ; испр. и доп. - Электрон. дан. - М.: Издательство Юрайт, 2018. - 501 с.
11. Ботуз, С.П. Интеллектуальные интерактивные системы и технологии управления удаленным доступом. Методы и модели управления процессами защиты и сопровождения интеллектуальной собственности в сети Internet/Intranet [Электронный ресурс]: учебное пособие/ С.П. Ботуз— Электрон. дан.— М.: СОЛОН-ПРЕСС, 2014.— 340c.
12. М.Ю. Карышев Социально-экономическая эффективность сферы информационно-коммуникационных технологий: методология международных статистических сопоставлений [Электронный ресурс] : монография. — Электрон. дан. — Режим доступа: http://e.lanbook.com/books/element.php?pl1\_id=53867 М.: Финансы и статистика, 2011. — 160 с.
13. Разработка баз данных [Электронный ресурс] : учебное пособие / А. С. Дорофеев [и др.]. - Электрон. дан. - Саратов : Ай Пи Эр Медиа, 2018. - 241 c.
14. Л. Г. Гагарина, Е. В. Кокорева, Б. Д. Виснадул Технология разработки программного обеспечения: учеб. пособие для вузов М.: ИД ФОРУМ-ИНФРА-М, 2008. - 400 с.