Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Институт высоких технологий

наименование института

Кафедра вычислительной техники

наименование кафедры

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Допускаю к защите  Зав. кафедрой: | |
|  |  | \_\_\_\_\_\_\_\_\_\_\_  подпись | А.С. Дорофеев  инициалы, фамилия |
|  | | « » 2018 г. | |

|  |
| --- |
| Разработка рекомендательной системы для рынка Иркутской области |
| наименование темы |

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к выпускной квалификационной работе

(уровень магистратуры)

Программа магистратуры

Сети ЭВМ и телекоммуникации

наименование программы

по направлению подготовки

09.04.01 Информатика и вычислительная техника

код и наименование направления подготовки

0.007.00.00 – ПЗ

обозначение документа

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Разработал студент группы | ЭВМм-16-1 |  |  |  | Б. С. Шевченко |
|  |  | подпись |  | И.О. Фамилия |
| Руководитель | | |  |  | Е. А. Черкашин |
| подпись |  | И.О. Фамилия |
| Нормоконтроль | | |  |  | Е. А. Черкашин |
| подпись |  | И.О. Фамилия |

Иркутск 2018 г.

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Институт высоких технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ

Директор\_\_\_\_\_\_Е.А. Анциферов

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2018 г.

ЗАДАНИЕ

на выпускную квалификационную работу магистранту Шевченко Борису Сергеевичу группы ЭВМм-16-1

1. Тема работы: Проектирование и реализация рекомендательной информационной системы для рынка недвижимости Иркутской области

Утверждена приказом по университету от \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_№ \_\_\_\_\_

1. Срок представления студентом законченной работы в ГЭК 7 июня 2018 г.
2. Исходные данные: материалы практики, НИР
3. Содержание расчетно-пояснительной записки (перечень подлежащих разработке вопросов)

|  |
| --- |
| 4.1 Постановка задачи; |
| 4.2 Изучение предметной области рекомендательных систем; |
| * 1. Изучение методов выработки рекомендаций; |
| 4.4 Проектирование и разработка рекомендательной веб-системы; |
| 4.4.1 Проектирование базы данных систем; |
| 4.6 Реализация и тестирование полученного программного продукта. |

1. Перечень графического материала (с указанием обязательных чертежей)

|  |
| --- |
| 5.1 Диаграмма вариантов использования; |
| 5.2 Диаграмма классов (базы данных ORM); |
| 5.3 Методы и схемы алгоритмов рекомендательной системы; |
| 5.4 Тестирование системы. |

Календарный план

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Разделы | Месяцы и недели | | | | | | | | | | | | | | | | | | | |
| Март | | | | | Апрель | | | | | Май | | | | | Июнь | | | | |
| Постановка задачи | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Изучение предметной области рекомендательных систем; |  | x | x | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Изучение методов выработки рекомендаций |  |  |  |  | x | х |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Проектирование и разработка рекомендательной веб-системы; |  |  |  |  |  |  | х | х | x | x | x | x | x | x |  |  |  |  |  |  |
| Проектирование базы данных систем |  |  |  |  |  |  | x | x |  |  |  |  |  |  |  |  |  |  |  |  |
| Реализация и тестирование полученного программного продукта |  |  |  |  |  |  |  |  | x | x | х | х | х | х | х | х | х | х |  |  |
| Подготовка пояснительной записки |  |  |  |  |  |  |  |  |  |  |  | x | x | x | x | x | x | x |  |  |

Дата выдачи задания 12.03.2018 г.

Руководитель работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е. А. Черкашин

Заведующий кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. С. Дорофеев

Задание принял к исполнению магистрант \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Б. С. Шевченко

План выполнен \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(полностью / не полностью)

Руководитель работы «\_\_\_»\_\_\_\_\_\_\_\_\_\_2018 г. Е. А. Черкашин

Содержание

[Введение 5](#_Toc516426316)

[Глава 1. Описание предметной области 8](#_Toc516426317)

[1.1 Методы фильтрации содержания 11](#_Toc516426318)

[1.2 Методы коллаборативной фильтрации 11](#_Toc516426319)

[1.3 Подходы к оценке интереса 15](#_Toc516426320)

[1.4 Сопутствующие технические задачи 16](#_Toc516426321)

[1.5 Атрибуты описания объектов недвижимости на сайтах Avito 17](#_Toc516426322)

[Глава 2. Реализация рекомендательной системы 21](#_Toc516426323)

[2.1 Краткое описание основных понятий ООП 21](#_Toc516426324)

[2.2 Задачи, решаемые при разработке рекомендательных систем 24](#_Toc516426325)

[2.3 Представление объекта недвижимости в Интернет 26](#_Toc516426326)

[2.4 Эксклюзивные и уникальные предложения 29](#_Toc516426327)

[Глава 3. Методы моделирования бизнес-процессов 30](#_Toc516426328)

[3.1 Концептуальное (инфологическое) проектирование 36](#_Toc516426329)

[3.3 Принцип работы ORM 41](#_Toc516426330)

[3.4 Преимущества и недостатки использования ORM 41](#_Toc516426331)

[3.5 Прямое отображение сущностей 42](#_Toc516426332)

[3.6 Отображение в наборы данных 43](#_Toc516426333)

[3.7 Инструменты массовой загрузки (Bulk loading tools) 44](#_Toc516426334)

[3.8 Технология Entity Framework 44](#_Toc516426335)

[3.9 Выбор технологии ORM 45](#_Toc516426336)

[Глава 4. Регистрация пользователя и сбор информации о предпочтениях 47](#_Toc516426337)

[4.1 Первичная структуризация: кластерный анализ /средство первичной структуризации/ 48](#_Toc516426338)

[Заключение 50](#_Toc516426339)

[В данном списке литературы приведены примеры оформления различных источников. Там сейчас есть ошибки наверняка. 51](#_Toc516426340)

[Список использованной литературы 51](#_Toc516426341)

# Введение

/Рынок недвижимости, проблематика. Сложно найти покупателя./

/Риэлторские программы CRM, взаимоотношения риелторов друг с другом и анализ цен недвижимости/

/Задача – найти покупателя, т.е. направлять его на то, что ему нужно, предполагая, что интересы покупателя и риэлтерской конторы совпадут.. Одним из способов продвинуться в решении этой задачи – структурировать имеющуюся информацию относительно интересов покупателя. В этом может помочь разработка рекомендательной системы./

*Рекомендательные системы* (РС) [1] – Информационные систем поддержки принятия решений, предназначенные для оценки уровня интереса пользователя к определенному продукту или сервису объекту на основе имеющейся информации о пользователе и/или объекте. Отрасль разработки РС начала активно развиваться при появлении онлайн-сервисов продаж, и в настоящее время РС – одно из активных направлений развития систем поддержки принятия решений, ориентированное, прежде всего, на коммерческое использование, а также на решение задач повышения продуктивности поиска релевантной информации.

В коммерции РС позволяют решать задачи установления, что именно представляет ценность для потребителя в виде набора конкретных объектов (например, товаров или услуг), сужение вариантов выбора и предоставление схожих вариантов других объектов, тем самым упрощая выбор. РС позволяют также выявлять новые характеристики объектов, например, при помощи ведения классификаций объектов и анализа набора известных признаков. Использование РС позволяет отделам снабжения коммерческих фирм-поставщиков предоставлять уникальный сервис каждому потребителю, увеличивая его доверие и лояльность к поставщику, увеличивая продажи и конверсию, а также получая и накапливая больше знаний о потребителях.

Рекомендательные системы появились в интернете достаточно давно, около 20 лет назад. Однако настоящий подъем в этой области случился примерно 5-10 лет назад, когда произошло соревнование NetflixPrize. Компания Netflix тогда давала в прокат не цифровые копии, а рассылала VHS-кассеты и DVD. Для них было очень важно повысить качество рекомендаций. Чем лучше Netflix рекомендует своим пользователям фильмы, тем больше фильмов они берут в прокат. Соответственно, растет и прибыль компании. В 2006 году они запустили соревнование NetflixPrize. Они выложили в открытый доступ собранные данные: около 100 миллионов оценок по пятибалльной шкале с указанием ID проставивших их пользователей. Участники соревнования должны были как можно лучше предугадывать, какую оценку поставит определенному фильму тот или иной пользователь. Качество предсказания измерялось при помощи метрики RMSE (средне-квадратичное отклонение). У Netflix уже был алгоритм, который предсказывал оценки пользователей с качеством 0.9514 по метрике RMSE. Задача была улучшить предсказание хотя бы на 10% — до 0.8563. Победителю был обещан приз в $1000000. Соревнование длилось примерно три года. За первый год качество улучшили на 7%, дальше все немного замедлилось. Но в конце две команды с разницей в 20 минут прислали свои решения, каждое из которых проходило порог в 10%, качество у них было одинаковое с точностью до четвертого знака. В задаче, над которой множество команд билось три года, все решили каких-то двадцать минут. Опоздавшая команда (как и многие другие, участвовавшие в конкурсе) остались ни с чем, однако сам конкурс очень сильно подстегнул развитие в этой области [2].

# Глава 1. Описание предметной области

В области рекомендательных систем используется специальная терминология. Объектом обозначается то, что система рекомендует пользователям, например, продукты, услуги, товары, новости, книги, DVD и т.п. Профилем пользователя или объекта являются данные, характеризующие пользователя или объект. Именно эти данные используются в процессе оценивания релевантности объекта к желаниям пользователя. Этот процесс называется фильтрацией. В результате фильтрации объекты ранжируются в соответствии с полученной оценкой, а пользователю предоставляется некоторое конечное подмножество, элементы которого имеют максимальную релевантность, т.е. оцениваются как наиболее интересные пользователю. Далее под интересом будем понимать именно интерес пользователя к объекту. Т.к. РС — это прежде всего информационные системы, то все объекты и пользователи описываются при помощи атрибутов. Именно атрибуты являются входной информацией во все процедуры оценивания интереса. Качество рекомендации - оценка точности предсказания интереса, сделанного РС, например, в сравнении с имеющимися примерами, т.е. оценками конкретных объектов конкретными пользователями.

Рекомендательные системы полезны не только для информационных ресурсов и порталов электронной коммерции, но и могут также открыть новые возможности в области безопасности, автомобильной промышленности [3], рекламе [4] и др.

Существует ряд подходов к оценке интереса.

На основе фильтрации содержания (content-based information filtering), при этом в информационной системе создаются профили пользователей и объектов, включающие социальный статус пользователя, возраст, место проживания, род деятельности, а также характеристики, выражающие интерес пользователя к объекту; профили объектов включают позицию в системе классификации, его потребительские характеристики.

На основе коллаборативной фильтрации (collaborative filtering), где используется информация о поведении пользователей в прошлом, например, перечень покупок или оценок объектов, сделанных на сайте интернет-магазина в прошлом пользователями из той же группы интересов, при этом аналитическим блоком информационной системы автоматически формируются классификации объектов, производится ранжирование атрибутов по степени значимости в оценке интереса.

Интеллектные (knowledge-based), где оценка вычисляется на основе формализованных знаний.

Гибридные (hybrid prediction) методы, которые базируются на подходах пп. 1 и 2, включая элементы из 3, что призвано повышать эффективность 1 и/или 2.

Например, в Music Genome Project музыкальный аналитик оценивает каждую композицию по сотням различных музыкальных характеристик, при помощи которых выявляются музыкальные предпочтения пользователя. Перечень оценок формирует профиль музыкального произведения. Основная проблема первого типа РС (фильтрации содержания) — это работоспособность системы на начальном этапе ее эксплуатации, так называемый "холодный старт": для новых пользователей в системе нет необходимой информации в профиле для принятия решения о том, какие объекты следует предлагать. В связи с этим в современных рекомендательных системах реализуется механизм сбора и анализа данных о пользователях с применением явных и неявных методов.

Явные методы сбора данных выполняют следующие действия:

* запрос у пользователя оценки объекта по некоторой шкале;
* запрос у пользователя ранжировки группы объектов от наилучшего к наихудшему;
* предъявление пользователю двух объектов с вопросом о том, какой из них лучше;
* предложение создать список объектов, характеризующих предпочтения пользователя.

Примерами неявного сбора данных выступают:

* наблюдение за тем, что просматривает пользователь в интернет-магазине или базе данных;
* ведение записей о поведении пользователя онлайн.

Сбор информации из социальных сетей, например, как в [5-7].

Второй тип РС, основанные на коллаборативной фильтрации, сравнивают однотипные данные, полученные от разных людей и вычисляют список рекомендаций для конкретного пользователя. Для вычисления рекомендаций используется, например, граф интересов. Таким образом, РС представляют собой информационные системы, дополненные алгоритмами, позволяющими обнаружить в хранилище объекты, которые не имеют непосредственного отношения к запросу пользователя. Любопытно, что рекомендательные системы часто используют как поисковые машины для индексации необычных данных.

**Примеры использования РС**

В обзоре [8] рассмотрены РС в области предоставления пользователям текстовых документов, в частности, научных статей. Больше половины (55%, 34 из 62) систем построены на основе фильтрации содержания. Алгоритмы коллаборативной фильтрации использованы только в 18% (11 из 62) случаев. Представлены подходы, основывающиеся на стереотипировании и гибридных методах. Авторы исследования пришли к выводу, что в 81% случаев моделирование пользователя на основе автоматического сбора информации не приносит значимых результатов по сравнению с явным указанием набора ключевых слов.

В основе характеристик объектов, научных статей, в исследованных РС используют просто ключевые слова, содержащиеся в документах, реже N-граммы, а также нетекстовые элементы, такие как ссылки на другие статьи и фамилии авторов. Самая популярная модель для хранения представления статей - модель векторного пространства. Моделирование пользователя осуществляется при помощи графов и списков тем, назначенных пользователям в результате машинного обучения. Темы объединяются в иерархические справочники, например, на основе классификаторов АСМ. В рассмотренных подходах тексты извлекаются из заглавий, аннотаций, заголовков, введения, предисловия, предоставленных автором ключевых слов, библиографии, основного текста, социальных тегов и цитирований контекста.

В РС, где применялась коллаборативная фильтрация, и ни в одном из проектов не удалось успешно использовать явные рейтинги: пользователи были слишком ленивы, чтобы самостоятельно задавать рейтинг статьям. Неявные рейтинги получены из данных по количеству страниц, прочитанных пользователем, взаимодействию пользователей с документами (загрузка, редактирование, представление) и цитирования. Главная проблема коллаборативной фильтрации для научных работ - это дефицит информации, например, для РС научных статей Mendeley по сравнению с Netflix (онлайн фильмы) дефицит составляет три порядка.

Неявные рейтинги объектов получаются из анализа одновременной загрузки статьи (со-загрузка) разными пользователями одной группы, совместного просмотра (со-просмотр), совместное цитирование статьями (со-цитирование) одних и тех же источников. Оказалось, что со-цитирование, будучи эффективным в начале появления статьи на сервисе РС, через два года начинает уступать со-загрузке. Популярным подходом представления результата такого анализа являются графы. Вершины графа - это статьи, представленные наборами атрибутов, а дуги - соотношения между статьями.

Основными проблемами в области РС являются следующие:

* отсутствие общего базиса оценивания качества систем (по предметным областям), включая объективную информацию о реальных оценках реальных пользователей, нестабильность методов оценивания и высокая их зависимость от "шума";
* неиспользованный потенциал научных исследований: новые научные результаты не внедряются в практические приложения (большинство работающих РС базируются на простых методах), данные существующих практических реализаций РС научно не исследуются, нет тесного взаимодействия со снежными областями анализа данных, а так же друг с другом, низкий научный интерес к РС;
* в оценке удовлетворенности не учитывается факторы конфиденциальность, безопасности данных, разнообразие, разметка и презентация информации; в значимом количестве РС моделирование пользователя было крайне примитивно - набор ключевых слов, собственная статья или просто фрагмент текста, представляющих научные интересы пользователя.

Среди открытых проектов выделяются MyMediaLite, LensKit, Mahout, Duine, RecLabCore, Easyrec и Recommender.

# 1.1 Методы фильтрации содержания

В статье [5] решается задача анализа профиля пользователя в социальной сети ВКонтакте для решения проблемы холодного старта в решении задачи рекомендации жанров и произведений музыки и фильмов. Авторами разработана РС «EZSurf» автоматизирующая процесс веб-сёрфинга и фильтрации контента, используя профиль пользователя в социальной сети "ВКонтакте", а также API сервисов last.fm, TheMovieDB для получения сведений о схожих объектах (музыкальных произведений). Такой подход существенно упрощает хранилище данных РС, поскольку не требует создания собственной системы классификаций и базы объектов.

В статье [9] рассматривается задача выделения N объектов с наивысшими оценками интереса, задача top-N, при применении фильтрации контента. Предлагается математическая модель контентной рекомендательной системы, основанная на нечетких множествах, критерий оценки качества рекомендаций и алгоритм решения задачи. Математическая модель и алгоритм протестированы на данных сайта last.fm.

# 1.2 Методы коллаборативной фильтрации

Подходы, основанный на коллаборативной фильтрации, в настоящее время более популярны, чем подходы на основе фильтрации содержимого, вероятно из-за того, что представляет собой отражение практического опыта: большинство коммерческих РС вынуждены решать проблему недостатка информации ("холодный старт") а также адаптируемости существующих сообществ пользователей к новым объектам.

Математические обозначения элементов модели сравнения состоит из набора пользователей и набора объектов . В виде – множество элементов, оцененных пользователем – множество пользователей, которые оценили объект – оценка пользователя u для объекта - вектор всех оценок пользователя – вектор всех оценок объекта и – средние значения оценок пользователя и объекта соответственно. Сравнительная оценка обозначается . Для задания этой оценки сначала задается мера близости объекта к объекту . Рассмотрим несколько популярных вариантов оценки близости.

Коэффициент Пирсона [3]:

где \(U=U\_i\cup U\_j\) - множество пользователей, которые оценили объекты и .

Косинус угла между двумя векторами и:

Затем производится формирование конечного множества объектов наиболее близких к объекту Вычисление рейтинга объекта o делается по формуле:

Популярный подход к формированию множества рекомендаций - это упорядочивавшие всех объектов по критерию схожести и выборке некоторого фиксированного количества объектов с максимальным рейтингом [Нефедова]. В качестве меры similarity двух объектов выступает cos угла между -мерными векторами.

В [3,10] так же представлен обзор способов использования вышеупомянутых методов вычисления оценок, которые разделены на два класса – анамнестические, т.е. основывающиеся на одновременной обработке всех имеющихся данных, и модельные, где производится предварительная обработка данных, выполняемая, например, раз в сутки. Второй класс позволяет быстрее вычислять оценки интереса, однако не обеспечивает актуальности данных. В классе аналитических способов, как правило используются методы многомерного анализа данных на основе "ближайшего соседства" (Neighbourhood-based), в то время как в модельных методах используется методы анализа скрытых факторов (LatenetFactors). Существуют гибридные методы, объединяющие оба предыдущих класса.

Грубо суть методов коллаборативной фильтрации изображена рисунках 1-3.

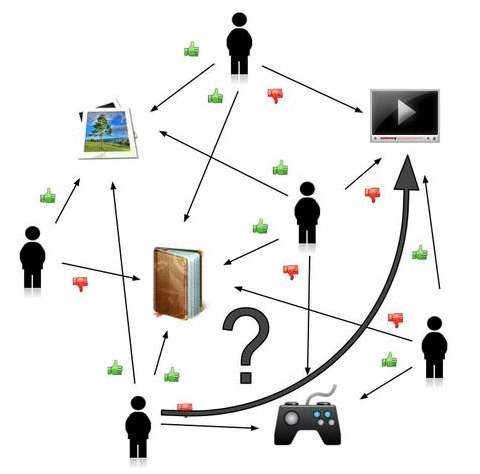


Рисунок 1 – Сбор информации о проявленном интересе пользователей

В процессе функционирования информационная система, представляющая интересующие объекты пользователям, запрашивает количественное выражение отношения к представленному объекту в виде ответа на вопрос: «Насколько представленный объект может вас заинтересовать?». Если человека интересует данный объект, то принято ставить положительную оценку, например, по шкале от 1 до 5 (от 0 до 4). Отрицательная оценка может выражаться как отрицательными числами, либо каким-либо поддиапазоном положительных. В самом «сложном» случае в качестве оценки выступает факт просмотра объекта (товара).

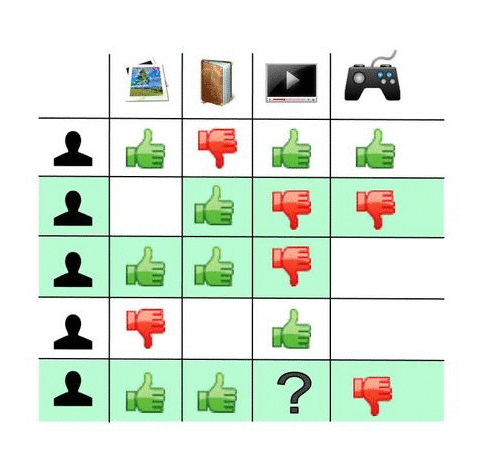


Рисунок 2 – Представление собранной информации

Полученный результат сбора периодически представляется в виде таблицы (рис. 2). Столбцами такой таблицы являются объекты (товары), а строками оценки пользователей этих объектов. Задачей рекомендательной системы состоит в том чтобы, по возможности, заполнить в этой таблице для заданного пользователя все ячейки. Например, на рис. 2 необходимо заполнить ячейку, помеченную знаком вопроса.

Один из умозрительных вариантов рассуждений следующий. Сравним 5-го (последнего) пользователя с остальными по критерию схожести его выбора с другими. Видно, что 5-й и 3-й схожи, т.к. оба положительно отнеслись к товарам 1 и 2. Пользователь 5 и 2 схожи тем, что положительно отнеслись к товару 2 и отрицательно к товару 4. Оба пользователя 3 и 2 отнеслись к интересующему нас товару 3 отрицательно. Следовательно, индуктивно можно предположить, что наш пользователь 5 отрицательно отнесется к товару 3 (рис. 3).

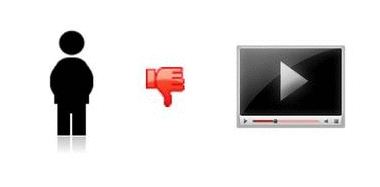


Рисунок 3 – Выработанная рекомендация

Полученная оценка означает, что в списке предлагаемых «на посмотреть» объектов данный товар будет ближе к концу, чем товары, которые «понравились бы».

**Гибридные методы**

В работе [11] рассмотрена задача разработки алгоритмов оценивания лекционного материала. Авторами предложен алгоритм вычисления близости лекций (объектов), где каждая лекция характеризуется подмножеством некоторого набора значений (например, подмножеством авторов лекций относительно множества всех авторов). Базовый алгоритм реализует подход фильтрации содержания. Для алгоритма подобранны коэффициенты, при помощи которых можно объединять оценки различных атрибутов в одну общую оценку лекции. Наиболее значимыми атрибутами оказались «категории», «авторы», «языки», «название» и «описание». Цель - синтезировать набор лекций фиксированной длины, рекомендованных для просмотра заданному пользователю, из фиксированного множества "новых", не использованных в построении профилей пользователя и объекта.

Далее алгоритм дополняется предсказателем последовательностей лекций: заданы примеры последовательностей из трех лекций, требуется для последовательностей из двух предложить третью, четвертую и т.д. Последовательности лекций приобретены системой неявно, т.е. фиксируя просмотренные пользователем лекции. Алгоритм занял первое место в соревновании, причем со значительным отрывом от второго места. Производится внедрение результатов исследований в области анализа сигналов и предсказания последовательностей событий.

**Приложения рекомендательных систем**

В [12] создан рекомендательный сервис новостей посетителям сайта, время пересчета рекомендаций в котором на каждую тысячу новых записей в журнале WEB-сервера составляет 1.5–2 с., что авторами заявлено как ресурс, функционирующий в режиме, близком к реальному времени. Для проекта Рамблер-новости подобный результат является удовлетворительным, так как 1000 новых запросов к сайту делается за чуть большее время. В исследовании использован адаптированный алгоритм MinHash для идентификации записей журнала и неточного их сравнения. Целью работы было показать целесообразность применения NoSQL-технологий для создания сервисов указанного качества. Важным свойством приведенной реализации является то, что задачи хранения и анализа данных удалось объединить с задачей предоставления доступа к результатам в единой системе, избежав накладных расходов на перемещение данных из одного источника в другой, что улучшило общую производительность сервиса. Кроме того, предложенный подход упрощает решение повседневных задач сбора статистики о взаимодействии пользователя с веб-приложением путем анализа структурированных логов мощным языком запросов СУБД MongoDB. В результате продемонстрировано, что применение NoSQL к решению подобного класса задач является весьма перспективным.

**Оценка эффективности работы рекомендательных систем**

В [13] поднята проблема сравнительной оценки различных подходов к построению РС. Классической оценкой точности предсказания интереса является оценк RMSE, среднеквадратическое отклонение, выполняется при помощи формулы:

где - множество пар пользователей и объектов – оценка интереса объекта o пользователем i, – оценка интереса, сделанная РС.

**Рекомендательные системы для рынка недвижимости**

В [18] предложен проект системы управления недвижимым имуществом, где варианты объектов предлагаются на основе вывода на прецедентах (case-based reasoning). Задача подсистемы вывода – найти прецедент в базе имеющихся прецедентов, похожий на запрос пользователя. Система помогает покупателям найти имущество, соответствующее их запросам. При этом система выводит суждения о свойствах объектов. Полученная информация затем используется в процессе фильтрации содержания и коллаборативной фильтрации. В дополнение к полученному списку выводится также наиболее популярные (most visited) варианты.

В теоретическом исследовании [15] в системе пользователи разделены на продавцов и покупателей. Продавцы "рекламируют" свое имущество, выставленное на продажу, выделяя те свойства недвижимости, которые сами считают важными. Предложена идея того, как получать данные для фильтрации содержания в задаче разработки РС управления имуществом: продавцы выступают в виде экспертов-оценщиков недвижимости, формируя информационную базу для фильтрации содержания. Предложенную идею можно дополнить, если ввести третий класс пользователей - экспертов-риелторов и позволить им дополнять базу данных прецедентов новыми суждениями.

В [16] опытным путем показано, что использование Интернета не влияет значительно на эффективность поиска недвижимости с целью ее покупки по критериям времени поиска, его гибкости и "удовлетворенность результатом". Согласно исследованию Национальной ассоциации риелторов, проведенном в 2011 году, показано, что 88% покупателей выбрали Интернет в качестве основного источника информации, но при этом среднее время на поиск жилья составило 12 недель, оно оказалось сравнимым с измерением, проведенным в 2009 году. Пользователь просматривает больше информации (как объектов, так и их свойств), но на анализ этой информации так же тратится много времени. Для повышения скорости выдачи результата авторами разработан алгоритм поиска на основе анализа поведения пользователей в процессе поиска объекта недвижимости, а также WEB-система, основанная на прецедентном выводе и онтологической концептуальной модели предметной области, ориентированная на пользователя.

В [23] авторами выделены три ключевые характеристики объекта недвижимости, которые в значительной мере является определяющими в процессе принятия решения – это "расположение" (location), "потребительская характеристика" (housing unit property) и "цена" (price). Большинство РС используют именно эти характеристики для фильтрации содержания, причем для критерия "потребительская характеристика" задаются формальные параметры объекта (площадь, номер этажа, количество балконов, комнат-спален и т.п.). На окончательное решение также влияет окружение объекта - расстояние до магазинов, школ, детских садов. Для того, чтобы учесть эти характеристики в [16] построена онтологическая модель, связывающая различные характеристики недвижимости в три древовидные структуры, описывающие варианты терминов "расположение", "потребительская характеристика" и "цена". Например, как вариант, под "расположением" понимается "расстояние" до места работы "пешком", выраженное в минутах. Так же "расположение" - это наличие в "окружении" (environment) объекта недвижимости "услуг" "фитнеса". При помощи онтологии получена возможность сравнивать не вполне "схожие" объекты, что повышает точность обработки информации.

Важным достижением авторов [16] является разработанный прототип РС, в котором пользователю предоставляется возможность указать на карте города область (окружность с заданным радиусом), в которой он хотел бы приобрести объект недвижимости, уточнить его потребительские характеристики и возможный диапазон цен.

Затем система выводит на карту варианты объектов недвижимости. Далее пользователь может уточнить другие характеристики, тем самым сужая количество предоставляемых вариантов. В сравнении с сервисами, подобным Avito.ru [17], пользователю предлагается меньше вариантов, т.е. система оценивает интерес пользователя более точно, и сильнее сужает количество альтернатив.

В [18] решается сложная логистическая задача организация процесса управления имуществом, в который вовлечены разнообразные группы людей в изменяющихся деловых и экономических условиях. В оценке учитываются не только экономические и бизнес-критерии, но и такие критерии, как "технологичность", "комфорт", "пространство", "административные" и "технические". Основная цель исследования - разработать модель, в которой различные группы людей будут максимально удовлетворены в "рациональной микро- и макро-среде".

Эффективность использования имущества предлагается оценивать по целой системе критериев, включающей цену объекта, цену владения этим объектом, цену ремонта, возможности его использования (capasity), количеству операций, которые необходимо выполнить по передаче собственности, надежность, комфорт, срок физической и технической эксплуатации, и др. Авторы разрабатывают математический аппарат для оценивания каждого объекта недвижимости – цены, эргономики, стоимости ремонта, назначение и т.п. Математическое обеспечение РС предложено развивать в направлении ухода от поиска "наиболее экономически выгодного управления недвижимостью" к мультикритериальному выбору и тем самым повысить эффективность вычислительных процессов

РС (анализа, оценки и т.п.).

Таким образом, на современном этапе развития РС важными вопросами, решаемыми в процессе проектирования РС объектов недвижимости являются:

• разработка концептуальной модели предметной области в виде онтологии;

• создание математических моделей предсказания значений атрибутов, описывающих объект недвижимости;

• реализация механизмов компьютерного обучения и логического вывода на основе прецедентов;

• обеспечение информационного наполнения РС для оценки качественных атрибутов (например, наличия школ и магазинов в шаговой доступности).

Решение данных задач в значительной мере влияет на качество и точность предсказания оценки значимости объекта недвижимости для пользователя.

**Сопутствующие технические задачи**

Одной из важных задач, решаемых при разработке РС, является создания пользовательского интерфейса, адекватно отображающего систему критериев, ко которым необходимо производить подбор объектов для пользователя. Например, в статье [14] представлен модуль естественно-языкового интерфейса к базе данных РС, который реализован на основе математических моделей семантических объектов. При помощи модели решаются задачи определения семантики языковой конструкции, заданной пользователем, включая синонимы, классы, отношения и ограничения. В статье приводятся сведения о программной реализации предложенного метода в среде PHP + SQL и результатах тестирования программы на задаче доступа к базе данных РС автомобильного салона.

В [15] решается проблема обеспечения ограничения доступа к личным данным пользователей в контексте построения РС встраивания рекламных сообщений в информационный поток. При этом необходимо контролируемо предоставлять в одностороннем порядке в РС информацию из БД пользователей. Предложено вместо традиционных средств VPN использовать режим функционирования сети с синхронным изменением IP-адреса сервера и переключение клиента на этот адрес.

# 1.5 Атрибуты описания объектов недвижимости на сайтах Avito

Перед тем, как разрабатывать информационную систему необходимо изучить предметную область, т.е. информационные потоки при решении задач поиска подходящей недвижимости. Первый этап естественный образом заключается в исследовании формы представления информации, т.е. выяснении какие атрибуты необходимо использовать для представления объектов недвижимости.

**Актуальность объявления**

• Объявления о продаже или аренде только существующих объектов недвижимости. Объявления с несуществующими объектами недвижимости нарушают правила Avito, к ним могут быть применены соответствующие меры.

• Обязательное снятие объявления с публикации, если объект недвижимости уже арендован, сдан, продан или куплен.

• Запрещено размещать объявления о платных услугах подбора недвижимости под видом аренды жилья.

**Уникальность объявления**

• Текст и фотографии объявления об объектах недвижимости должны быть уникальными. Неуникальные объявления нарушают правила Avito, к ним могут быть применены соответствующие меры. Поэтому составляется подробное описание и качественные фото для каждого объекта.

**Регион размещения**

• Объявление должно быть актуальным для того города, в котором находится недвижимость. Например, если объект находится в г. Долгопрудный Московской области, то местоположение в объявлении должно быть указано «Долгопрудный», а не «Москва».

• Объявления а продаже, покупке, сдаче или аренде объекта в другом государстве — размещаются в подкатегории «Недвижимость за рубежом».

**Местоположение**

• В поле «Адрес» указываются только точные данные о местоположении объекта недвижимости. Дополнительную информацию (ориентиры, район города) добавляются в описание объявления.

Правильно: «ул. Ленина, д.5»

Неправильно: «ул. Ленина, около гастронома», «ул. Ленина», «г. Москва, ул. Ленина».

**Заголовки объявлений**

• Для подкатегории «Коммерческая недвижимость» в заголовке указываются только вид объекта и основные параметры. Указание цены, слов, привлекающих внимание, контактной информации, адреса сайта или только слова «продам / куплю» является нарушением правил Avito.

Правильно: Торговое помещение, 208 м.

Неправильно: «Торговое помещение за 150000 руб.», «Торговое помещение срочно»,«Торговое помещение звоните ХХХХХХХХХХХ», «Торговое помещение site.ru».

**Параметры объявления**

• Выбирается подкатегория, соответствующая объекту.

• Профессиональный риэлторпредлагающий свои услуги (подбор, оценка, юридическое сопровождение сделки и т.д.), размещает объявление в категории «Услуги».

• В подкатегории «Коммерческая недвижимость» размещаются объявления об объектах недвижимости, предназначенных для коммерческих целей: «Офисное помещение», «Складское помещение» и т.д.

• В подкатегории «Коммерческая недвижимость» запрещено размещать продажу бизнеса. Для этого используется категория «Готовый бизнес».

• В подкатегории «Дома, дачи, коттеджи» размещаются объявления о готовых объектах или об объектах, находящихся в процессе строительства на момент размещения. Для размещения объявлений о продаже сборных объектов используется подкатегория «Предложение услуг-Строительство домов, коттеджей».

• Объявления о продаже, покупке, аренде комнаты или ее доли размещаются в подкатегории «Комнаты».

• Объявления об аренде койко-мест размещаются в подкатегории «Комнаты».

• В подкатегории «Гаражи и машиноместа» размещаются объекты с определенным местоположением. Объявления о продаже сборных гаражей размещаются в подкатегории «Ремонт и строительство».

• Новостройками в подкатегории «Квартиры» являются объекты из списка новостроек, размещенного на странице подачи объявления. Размещая объявление о продаже квартиры в новостройке до ввода дома в эксплуатацию, обязательно выбирается новостройку из списка или указывается самостоятельно в объявлении наименование застройщика с организационно-правовой формой, а также сведения о месте и способах получения проектной декларации многоквартирного дома для квартир по ДДУ или устава кооператива для квартир в кооперативах. Другие схемы для строящихся домов запрещены.

• Объявления о продаже или аренде апартаментов (помещений нежилого фонда, пригодных для кратковременного и длительного проживания) размещаются только в подкатегории «Коммерческая недвижимость» вид объекта «Помещения свободного назначения» или подкатегории «Квартиры».

**Цена**

• Указывается полная стоимость объектов, а не стоимость квадратного метра или сотки. Исключение: подкатегория «Коммерческая недвижимость» — здесь указывается стоимость или за квадратный метр, или за весь объект полностью, установив соответствующий параметр.

• В объявлениях типа «Сниму/Сдам» со сроком аренды «Посуточно» запрещено указывать цену за почасовую аренду. Указывается стоимость аренды за одни сутки. При этом о наличии бонусов за длительную аренду в виде сниженной цены сообщается в описании. Указывать акционнуюстоимость в поле «Цена» запрещено.

• Если цена объекта снижена по каким-либо причинам (плохое состояние объекта, цена указана без стоимости коммунальных платежей и т. д.), это указывается в описании объявлений.

**Фотографии**

• Используются только реальные фотографии предлагаемых объектов, иллюстрирующие текст объявления. Чужие фото (например, из интернета), снижают доверие потенциальных клиентов и могут быть отклонены модератором.

• Использование логотипов агентства недвижимости вместо фото объектов является нарушением правил сайта.

• Кроме фотографии дома (здания) следует добавить фотографии интерьера (комнат, прихожей, кухни, санузлов и т.д.), которые должны достоверно отражать актуальную обстановку внутри предлагаемого объекта. В случае, если фотографии интерьера отсутствуют, в обязательном порядке добавляются фотографии с актуальной планировкой.

• Объявления о поиске жилья могут сопровождаться реальными фотографиями потенциальных арендаторов только с их согласия.

**Личный кабинет компании**

• Агентствам недвижимости необходимо пользоваться одним личным кабинетом для всех сотрудников. Запрещается подавать одинаковые объявления, используя личный кабинет агентства недвижимости и личную учетную запись сотрудника агентства. Если у вашего агентства недвижимости уже есть личный кабинет на Avito, предоставьте доступ к нему своим коллегам.

**Право на размещение объекта третьим лицом**

• Объявления о продаже, аренде и покупке недвижимости могут быть размещены собственником либо третьим лицом (агентом, посредником и т.п.), имеющим соответствующее поручение от собственника. Третьи лица обязательно должны указать размер комиссии или иного вознаграждения по сделке. Третьим лицам необходимо выбирать значение «Посредник» для параметра «Право собственности». Объявления, в которых третьи лица указывают значение «Собственник», будут отклонены.

• Администрация сайта вправе потребовать правоустанавливающие документы (договор об оказании услуг, свидетельство о собственности и т.д.) как у собственника, так и у третьего лица.

**А.Б. Функциональное моделирование предметной области**

В результате анализа функций, которые должен реализовывать веб-сайт построена UML–диаграмма Use Case (рис. 4.).

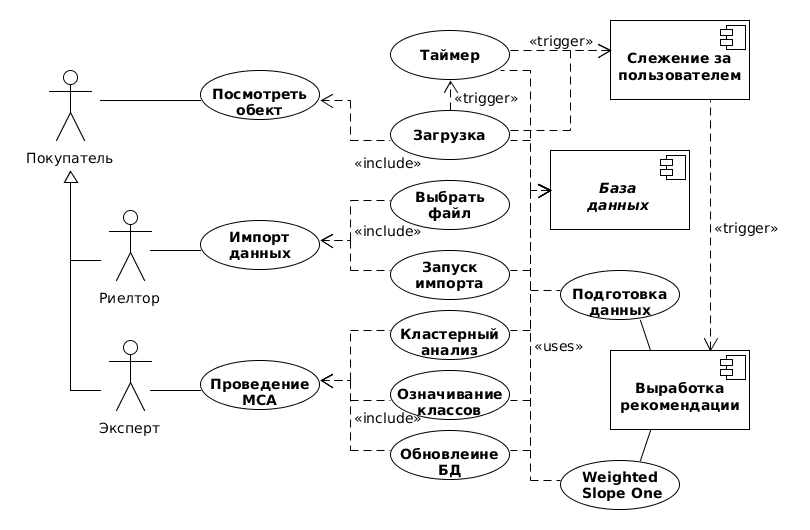


Рисунок 4 – Диаграмма вариантов использования веб-системы

Выделены три роли пользователей, взаимодействующих с веб-сайтом: покупатель, риелтор и эксперт. Эксперт и риелтор могут выполнять все функции покупателя, поэтому на диаграмме данные роли наследуют свойства покупателя.

Покупатель [недвижимости] выполняет одну только функцию - «просмотр объекта». Непосредственная покупка недвижимости осуществляется другими средствами. Задача РС – помочь покупателю быстро найти то, что ему нужно. Просмотр объекта недвижимости включает в себя вывод списка объектов на экран, для чего порождаются запросы к базе данных. В процессе просмотра система «наблюдает» за пользователем: если пользователь выбрал конкретный объект и просматривает детальную о нем информацию больше некоторого времени, то РС делает заключение, что данный объект представляет некоторый интерес. Затем, в список оценок добавляется «положительная» оценка, которая, в свою очередь, сохраняется в базе данных.

При просмотре конкретного объекта пользователем тратится некоторое время на восприятие информации. Тем временем подсистема «Выработка рекомендации» «готовит» ему перечень объектов (рекомендацию), которые потенциально должны его заинтересовать. Рекомендация строится на основе имеющейся информации. Если информации о предпочтениях достаточно, то используется метод Slope One, если нет, то пользователю представляется перечень объектов «схожих» с тем, что он просматривает.

Метод Slope One при построении оценки использует и весь набор положительных оценок пользователя и оценки других пользователей, т.е. его оценка является более информативно емкой, и, соответственно, более адекватной. Но в случае «холодного старта», когда информации для выполнения оценивания недостаточно, применимость метода не высока. Так как система должна вырабатывать хоть какие-то рекомендации, здравый смысл подсказывает, что необходимо применять какой-то другой способ. В качестве такого, запасного, способа в данной системе применяется оценка схожести объектов между собой. Подробно применение оценки схожести рассматривается в разделе «Кластерный анализ».

Вторая роль ­– это «риелтор». Его функция – обновлять базу данных с целью поддержания ее актуальности. Риелтором выполняется «импорт данных». Для того, чтобы выполнить эту функцию необходимо загрузить файл с данными, представленными в специальном стандартизованном обменном формате. Импортируемые данные загружаются риелтором с сайтов, содержащих информацию об объектах недвижимости. Для осуществления импорта необходимо выбрать файл и запустить процедуру.

Третья роль – «эксперт». Задача эксперта – структурировать имеющуюся информацию. Эксперт запускает процедуру кластерного анализа, в результате которой все множество объектов разбивается на подмножества (кластеры) схожих между собой объектов недвижимости. Данным процессом можно управлять, например, задавать количество объектов, подвергаемых структуризации (объекты выбираются случайным образом), объединять кластеры друг с другом, порождая новый кластер. Основная задача эксперта – это охарактеризовать каждый кластер, делается это путем присвоения кластерам некоторого имени («Означивание кластера»). Получаемые поименованные кластеры, затем, используются системой для выработки рекомендаций.

# Архитектура рекомендательной системы

Представленные на рис. 4. функции объединяются в функциональные блоки и сформировывают серверную часть архитектуры РС (рис. 5.).



Рисунок 5 – Архитектура рекомендательной системы

Пользователь и эксперт взаимодействуют с РС через веб-интерфейс, который реализуется при помощи Web-браузера (клиентская часть) и Web-сервера. Хранение информации в схеме оснащает блок «БД» (база данных). Функции выработки рекомендации реализуются в блоке «РС» (рекомендательная система), который взаимодействует с блоком «МСА» (многомерный статистический анализ [данных]). В МСА находятся функции оценки схожести объектов, кластерного анализа, вычисления оценок на основе метода Slope One.

Задача Web-сервера, таким образом, заключается в преобразовании наборов данных, получаемых из БД и РС в HTML5-страницы, представляемые пользователю Web-браузером в виде гипертекста. Импорт данных в системе реализован при помощи команды командной строки операционной системы (без графического пользовательского интерфейса), т.к. на этапе разработки программы необходимо было часто заново перестраивать и заполнять БД, что позволило обойтись без автоматизации процесса взаимодействия с интерфейсом загрузки (импорта) данных.

Таким образом, полученная архитектура представляет собой реализацию клиент-серверной схемы взаимодействия по протоколам Интернета.

Следующим этапом проектирования РС выступает информационное моделирование. Целью данного этапа являлось создание набора взаимодействующих классов, которые должны быть сохранены в долговременном хранилище. Достичь цели позволяют технологии ORM (Object-Relational Mapping), т.е. отображение объектов в реляционную базу данных. Под объектами мы понимаем структуры данных, представляемые в объектно-ориентированном программировании.

# 2.1 Краткое описание основных понятий ООП

Объекты – это абстрактные типы данных. Абстрагирование означает выделение значимой информации и исключение из рассмотрения незначимой. В ООП рассматривают лишь абстракцию данных (нередко называя её просто «абстракцией»), подразумевая набор значимых характеристик объекта, доступный остальной программе.

Инкапсуляция — свойство системы, позволяющее объединить данные и методы, работающие с ними, в классе. Одни языки (например, С++, Java или Ruby) отождествляют инкапсуляцию с сокрытием, но другие (Smalltalk, Eiffel, OCaml) различают эти понятия.

Наследование — свойство системы, позволяющее описать новый класс на основе уже существующего с частично или полностью заимствующейся функциональностью. Класс, от которого производится наследование, называется базовым, родительским или суперклассом. Новый класс — потомком, наследником, дочерним или производным классом.

Полиморфизм — свойство системы, позволяющее использовать объекты с одинаковым интерфейсом без информации о типе и внутренней структуре объекта. Другой вид полиморфизма — параметрический — в ООП называют обобщённым программированием.

Класс — универсальный, комплексный тип данных, состоящий из тематически единого набора «полей» (переменных более элементарных типов) и «методов» (функций для работы с этими полями), то есть он является моделью информационной сущности с внутренним и внешним интерфейсами для оперирования своим содержимым (значениями полей). В частности, в классах широко используются специальные блоки из одного или чаще двух спаренных методов, отвечающих за элементарные операции с определенным полем (интерфейс присваивания и считывания значения), которые имитируют непосредственный доступ к полю. Эти блоки называются «свойствами» и почти совпадают по конкретному имени со своим полем (например, имя поля может начинаться со строчной, а имя свойства — с заглавной буквы). Объект – сущность в адресном пространстве вычислительной системы, появляющаяся при создании экземпляра класса.

Другим проявлением интерфейсной природы класса является то, что при копировании соответствующей переменной через присваивание, копируется только интерфейс, но не сами данные, то есть класс — ссылочный тип данных.

Переменная-объект, относящаяся к заданному классом типу, называется экземпляром этого класса. При этом в некоторых исполняющих системах класс также может представляться некоторым объектом при выполнении программы посредством динамической идентификации типа данных. Обычно классы разрабатывают таким образом, чтобы обеспечить отвечающие природе объекта и решаемой задаче целостность данных объекта, а также удобный и простой интерфейс. В свою очередь, целостность предметной области объектов и их интерфейсов, а также удобство их проектирования, обеспечивается наследованием.

Механизм наследования позволяет производить дочерние классы на основе родительских. Различаются приватные (доступные из текущего класса) и защищенные (доступные из текущего класса и его потомков) компоненты классов. Обращение к методам родительского класса из класса наследника осуществляется с помощью ключевого слова base (super).

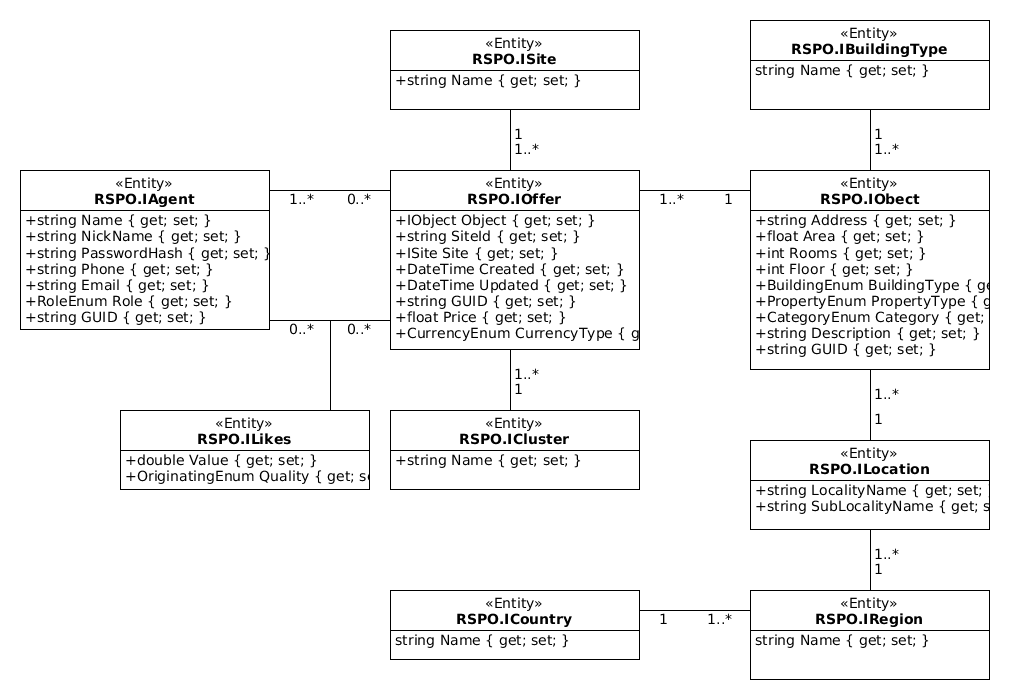


Рисунок 5. Диаграмма классов РС, на сохраняемых в БД

Недвижимость как информационный объект

# 2.3 Представление объекта недвижимости в Интернет

Данный раздел представляет собой результаты анализа способов представления объекта недвижимости на сайтах класса Avito, а также в базах данных и документах риэлтерских агентств.

Объявления о продаже или аренде только существующих объектов недвижимости. Объявления с несуществующими объектами недвижимости нарушают правила Avito, к ним могут быть применены соответствующие меры:

• Обязательное снятие объявления с публикации, если объект недвижимости уже арендован, сдан, продан или куплен.

• Запрещено размещать объявления о платных услугах подбора недвижимости под видом аренды жилья.

Уникальность объявления

Текст и фотографии объявления об объектах недвижимости должны быть уникальными. Неуникальные объявления нарушают правила Avito, к ним могут быть применены соответствующие меры. Поэтому составляется подробное описание и качественные фото для каждого объекта.

Регион размещения

Объявление должно быть актуальным для того города, в котором находится недвижимость. Например, если объект находится в г. Долгопрудный Московской области, то местоположение в объявлении должно быть указано «Долгопрудный», а не «Москва».

Объявления а продаже, покупке, сдаче или аренде объекта в другом государстве — размещаются в подкатегории «Недвижимость за рубежом».

Местоположение

В поле «Адрес» указываются только точные данные о местоположении объекта недвижимости. Дополнительную информацию (ориентиры, район города) добавляются в описание объявления.

Правильно: «ул. Ленина, д.5»

Неправильно: «ул. Ленина, около гастронома», «ул. Ленина», «г. Москва, ул. Ленина».

Заголовки объявлений

Для подкатегории «Коммерческая недвижимость» в заголовке указываются только вид объекта и основные параметры. Указание цены, слов, привлекающих внимание, контактной информации, адреса сайта или только слова «продам / куплю» является нарушением правил Avito.

Правильно: Торговое помещение, 208 кв.м.

Неправильно: «Торговое помещение за 150000 руб.», «Торговое помещение срочно»,«Торговое помещение звоните ХХХХХХХХХХХ», «Торговое помещение site.ru».

Параметры объявления

Выбирается подкатегория, соответствующая объекту.

Профессиональный риэлтор предлагающий свои услуги (подбор, оценка, юридическое сопровождение сделки и т.д.), размещает объявление в категории «Услуги».

В подкатегории «Коммерческая недвижимость» размещаются объявления об объектах недвижимости, предназначенных для коммерческих целей: «Офисное помещение», «Складское помещение» и т.д.

В подкатегории «Коммерческая недвижимость» запрещено размещать продажу бизнеса. Для этого используется категория «Готовый бизнес».

В подкатегории «Дома, дачи, коттеджи» размещаются объявления о готовых объектах или об объектах, находящихся в процессе строительства на момент размещения. Для размещения объявлений о продаже сборных объектов используется подкатегория «Предложение услуг-Строительство домов, коттеджей».

Объявления о продаже, покупке, аренде комнаты или ее доли размещаются в подкатегории «Комнаты».

Объявления об аренде койко-мест размещаются в подкатегории «Комнаты».

В подкатегории «Гаражи и машиноместа» размещаются объекты с определенным местоположением. Объявления о продаже сборных гаражей размещаются в подкатегории «Ремонт и строительство».

Новостройками в подкатегории «Квартиры» являются объекты из списка новостроек, размещенного на странице подачи объявления. Размещая объявление о продаже квартиры в новостройке до ввода дома в эксплуатацию, обязательно выбирается новостройку из списка или указывается самостоятельно в объявлении наименование застройщика с организационно-правовой формой, а также сведения о месте и способах получения проектной декларации многоквартирного дома для квартир по ДДУ или устава кооператива для квартир в кооперативах. Другие схемы для строящихся домов запрещены.

Объявления о продаже или аренде апартаментов (помещений нежилого фонда, пригодных для кратковременного и длительного проживания) размещаются только в подкатегории «Коммерческая недвижимость» вид объекта «Помещения свободного назначения» или подкатегории «Квартиры».

Цена

Указывается полная стоимость объектов, а не стоимость квадратного метра или сотки. Исключение: подкатегория «Коммерческая недвижимость» — здесь указывается стоимость или за квадратный метр, или за весь объект полностью, установив соответствующий параметр.

В объявлениях типа «Сниму/Сдам» со сроком аренды «Посуточно» запрещено указывать цену за почасовую аренду. Указывается стоимость аренды за одни сутки. При этом о наличии бонусов за длительную аренду в виде сниженной цены сообщается в описании. Указывать акционную стоимость в поле «Цена» запрещено.

Если цена объекта снижена по каким-либо причинам (плохое состояние объекта, цена указана без стоимости коммунальных платежей и т. д.), это указывается в описании объявлений.

Фотографии

Используются только реальные фотографии предлагаемых объектов, иллюстрирующие текст объявления. Чужие фото (например, из интернета), снижают доверие потенциальных клиентов и могут быть отклонены модератором.

Использование логотипов агентства недвижимости вместо фото объектов является нарушением правил сайта.

Кроме фотографии дома (здания) следует добавить фотографии интерьера (комнат, прихожей, кухни, санузлов и т.д.), которые должны достоверно отражать актуальную обстановку внутри предлагаемого объекта. В случае, если фотографии интерьера отсутствуют, в обязательном порядке добавляются фотографии с актуальной планировкой.

Объявления о поиске жилья могут сопровождаться реальными фотографиями потенциальных арендаторов только с их согласия.

Личный кабинет компании

Агентствам недвижимости необходимо пользоваться одним личным кабинетом для всех сотрудников. Запрещается подавать одинаковые объявления, используя личный кабинет агентства недвижимости и личную учетную запись сотрудника агентства. Если у вашего агентства недвижимости уже есть личный кабинет на Avito, предоставьте доступ к нему своим коллегам.

Право на размещение объекта третьим лицом

Объявления о продаже, аренде и покупке недвижимости могут быть размещены собственником либо третьим лицом (агентом, посредником и т.п.), имеющим соответствующее поручение от собственника. Третьи лица обязательно должны указать размер комиссии или иного вознаграждения по сделке. Третьим лицам необходимо выбирать значение «Посредник» для параметра «Право собственности». Объявления, в которых третьи лица указывают значение «Собственник», будут отклонены.

Администрация сайта вправе потребовать правоустанавливающие документы (договор об оказании услуг, свидетельство о собственности и т.д.) как у собственника, так и у третьего лица.

# 2.4 Эксклюзивные и уникальные предложения

Avito представляет собой торговую платформу, на которой представлен широкий спектр различных товаров и услуг без возможности специальной обработки специфичной параметрам объекта недвижимости. В этой связи покупка/продажа недвижимости, возможности поиска и отбора удовлетворяющего покупателя/продавца предложения по необходимым именно им параметрам ограничена. Поэтому, при учёте вышеизложенных корректировок и регулярном (постоянном) администрировании возможно создать необходимый и эффективный инструмент, основанный в первую очередь на эксклюзивности и уникальности предложений в сфере недвижимости, позиционируемых в информационной системе. Под эксклюзивными и уникальными предложениями подразумеваются такие предложения, которые являются как редкими, способными заинтересовать покупателя/продавца, так и, что не маловажно, являющиеся не повторяющимися, представленными в единственном экземпляре.

Основными недостатками Avito являются:

1. Повторение объявлений (т.е. один объект позиционируется несколькими людьми)

2. Присутствие в экспозиции объявлений «уток».

Все объекты, выставленные в экспозицию в информационной системе должны быть Эксклюзивными и содержать максимально подробную информацию и фотографии. Это должно быть либо контролируемо модератором или самой структурой сайта (при внесении данных, все пункты должны быть обязательными).

Для продавца (при добавлении своего объекта на сайт, с дополнением электронной копии выписки из ЕГРН):

1. Подробная информация (район города, адрес, площадь, количество комнат, санузлов, стоимость и т.д.)

2. Подробные фотографии (объект, внешний вид, вид из окна, окружающая инфраструктура)

3. Возможные формы расчёта

4. Сотрудничество с агентствами (возможно/нет)

5. Для риелторов/агентств (является эксклюзивом/нет)

Данная информация позволит воспользоваться инструментами «поиск» и «сортировка по параметрам», что в свою очередь существенно облегчит поиск предложений, удовлетворяющих покупателя/продавца по необходимым им параметрам.

# Глава 3. Методы моделирования бизнес-процессов

Деление методов на функциональные и объектные является достаточно условным, поскольку наиболее развитые методы используют элементы обоих подходов. К числу наиболее распространенных методов относятся:

• метод функционального моделирования SADT (IDEF0);

• метод моделирования процессов IDEF3;

• моделирование потоков данных DFD;

• метод ARIS;

• метод Ericsson\\_Penker;

•метод моделирования, используемый в технологии Rational Unified Process и др.

Метод SADT (Structured Analysis and Design Technique) - классический метод процессного подхода к управлению. Основной принцип процессного подхода заключается в структурировании деятельности организации в соответствии с ее бизнес-процессами, а не организационно-штатной структурой.

Модель, основанная на бизнес-процессах, содержит в себе и организационно-штатную структуру предприятия.

Метод SADT разработан Дугласом Россом (SoftTech, Inc.) в 1969 г. для моделирования искусственных систем средней сложности.

Данный метод успешно использовался в военных, промышленных и коммерческих организациях США для решения широкого круга задач, таких как долгосрочное и стратегическое планирование, автоматизированное производство и проектирование, разработка ПО для оборонных систем, управление финансами и материально-техническим снабжением и др.

Метод SADT поддерживается Министерством обороны США, которое было инициатором разработки семейства стандартов IDEF (Icam DEFinition), являющегося основной частью программы ICAM (интегрированная компьютеризация производства), проводимой по инициативе ВВС США.

Метод SADT реализован в одном из стандартов этого семейства – IDEF0, который был утвержден в качестве федерального стандарта США в 1993 г., его подробные спецификации можно найти на сайте http://www.idef.com.

Существует также российская версия данного стандарта [Методология функционального моделирования IDEF0. Руководящий документ РД IDEF0 – 2000. – М.: Госстандарт России, 2000]. Вместе со стандартом IDEF0 обычно используются стандарт моделирования процессов IDEF3 и стандарт моделирования данных IDEF1Х.

Метод SADT представляет собой совокупность правил и процедур, предназначенных для построения функциональной модели объекта какой-либо предметной области. Функциональная модель SADT отображает функциональную структуру объекта, т.е. производимые им действия и связи между этими действиями.

Модель SADT состоит из диаграмм, фрагментов текстов и глоссария, имеющих ссылки друг на друга.

Диаграммы – главные компоненты модели. Все функции организации на диаграммах представляются в виде блоков. Взаимодействие блоков друг с другом описывается посредством интерфейсных дуг, выражающих «ограничения», которые, в свою очередь, определяют когда и каким образом функции выполняются и управляются.

Место соединения дуги с блоком определяет тип интерфейса. Управляющая информация входит в блок сверху; входная информация, подвергающаяся обработке, показана с левой стороны блока, а результаты процесса (выход) показаны с правой стороны. Механизм (человек или автоматизированная система), который осуществляет операцию, представляется дугой, входящей в блок снизу (рис. 1).

Постепенно, по мере создания диаграмм, в модель SADT вводится все больше уровней детализации. Каждый компонент модели может быть декомпозирован на другой диаграмме. Каждая диаграмма иллюстрирует «внутреннее строение» блока на родительской диаграмме. На рис. 2 приведены четыре диаграммы и их взаимосвязи, т.е. показана структура SADT модели. Условного объекта

Последовательность разработки SADT-модели:

1. собрать информацию об объекте, определить его границы;

2. определить цели исследования и точки зрения модели;

3. построить, обобщить и декомпозировать диаграммы;

4. распространить модель среди заинтересованных лиц;

5. оценить, рецензировать и комментировать;

6. принять модель.

Принципиально бизнес модель построенная по методу SADT, должна выглядеть следующим образом:

1. Верхний уровень модели должен отражать только контекст системы – взаимодействие моделируемого предприятия единственным контекстным процессом с внешним миром. Это представление системы в виде простейшего компонента – одного блока и дуг, изображающих интерфейсы с функциями вне системы. Этот единственный блок отражает систему как единое целое, поэтому имя, указанное в блоке, является общим. Это верно и для интерфейсных дуг – они также соответствуют полному набору внешних интерфейсов системы в целом.

2. На втором уровне модели должны быть отражены основные виды деятельности предприятия и их взаимосвязи. Для этого блок, который представляет систему в качестве единого модуля, детализируется на другой диаграмме с помощью нескольких блоков, соединенных интерфейсными дугами. Для этого процессы должны быть тематически сгруппированы. В случае большого их количества некоторые из них можно вынести на третий уровень модели. Но в любом случае под виды деятельности необходимо отводить не более двух уровней модели.

Кроме того, модель не может опустить какие-либо элементы, т.е., родительский блок и его интерфейсы обеспечивают контекст. К нему нельзя ничего добавить, и из него не может быть ничего удалено.

Дальнейшая детализация бизнес-процессов осуществляется посредством бизнес-функций, сгруппированных по определенным признакам. Т.е. определяются блоки, представляющие собой основные подфункции исходного бизнес-процесса. Данная декомпозиция выявляет полный набор подфункций, каждая из которых показана как блок, границы которого определены интерфейсными дугами. Каждая из этих подфункций может быть декомпозирована подобным образом в целях большей детализации.

Во всех случаях каждая подфункция может содержать только те элементы, которые входят в исходную функцию.

4. Бизнес-функции детализируются с помощью элементарных бизнес-операций.

5. Бизнес-операции описываются посредством задания алгоритма их выполнения.

Если коротко, то бизнес модель SADT это: контекст – бизнес процессы – бизнес-функции – элементарные бизнес-операции – алгоритмы бизнес-операций.

Пример создания SADT модели (модель банка и оформления кредита).

Таким образом, модель SADT представляет собой серию диаграмм с сопроводительной документацией, разбивающих сложный объект на составные части, которые изображены в виде блоков. Детали каждого из основных блоков показаны в виде блоков на других диаграммах. Каждая детальная диаграмма является декомпозицией блока из диаграммы предыдущего уровня. На каждом шаге декомпозиции диаграмма предыдущего уровня называется родительской для более детальной диаграммы.

На SADT-диаграммах не указываются явно ни последовательность, ни время. Обратные связи, итерации, продолжающиеся процессы и перекрывающиеся (по времени) функции могут быть изображены с помощью дуг.

Обратные связи могут выступать в виде комментариев, замечаний, исправлений и т.д.

Стратегии декомпозиции при построении иерархии диаграмм SADT-модели.

1. Функциональная декомпозиция – декомпозиция в соответствии с функциями, которые выполняют люди или организация. Может оказаться полезной стратегией для создания системы описаний, фиксирующей взаимодействие между людьми в процессе их работы. Очень часто, однако, взаимосвязи между функциями весьма многочисленны и сложны, поэтому рекомендуется использовать эту стратегию только в начале работы над моделью системы.

2. Декомпозиция в соответствии с известными стабильными подсистемами – приводит к созданию набора моделей, по одной модели на каждую подсистему или важный компонент. Затем для описания всей системы должна быть построена составная модель, объединяющая все отдельные модели. Рекомендуется использовать разложение на подсистемы, только когда разделение на основные части системы стабильно. Нестабильность границ подсистем быстро обесценит как отдельные модели, так и их объединение.

3. Декомпозиция по физическому процессу – выделение функциональных стадий, этапов завершения или шагов выполнения. Хотя эта стратегия полезна при описании существующих процессов (таких, например, как работа промышленного предприятия), результатом ее часто может стать слишком последовательное описание системы, которое не будет в полной мере учитывать ограничения, диктуемые функциями друг другу. При этом может оказаться скрытой последовательность управления. Эта стратегия рекомендуется только если целью модели является описание физического процесса как такового или только в крайнем случае, когда неясно, как действовать.

Степень декомпозиции - одна из наиболее частых проблем, возникающих в процессе построения SADT-моделей, т.е. определение разумной степени полноты описания. Только длительная практика позволяет приобрести знания, необходимые для принятия правильного решения об окончании моделирования.

Несколько рекомендаций, которыми пользуются опытные аналитики для определения момента завершения моделирования.

В целом рекомендуется прекращать моделирование, когда уровень детализации модели удовлетворяет ее цель.

Для отдельной модели, которая создается независимо от какой-либо другой модели, декомпозиция одного из ее блоков должна прекращаться, если:

1. Блок содержит достаточно деталей. Проверить достаточность деталей обычно совсем легко, необходимо просто спросить себя, отвечает ли блок на все или на часть вопросов, составляющих цель модели. Если блок помогает ответить на один или более вопросов, то дальнейшая декомпозиция может не понадобиться.

2. Необходимо изменить уровень абстракции, чтобы достичь большей детализации блока. Иногда при декомпозиции блока выясняется, что диаграмма начинает описывать, как функционирует блок, вместо описания того, что блок делает. В этом случае происходит изменение уровня абстракции – изменение сути того, что должна представлять модель (т.е. изменение способа описания системы). В SADT изменение уровня абстракции часто означает выход за пределы цели модели и, следовательно, это указывает на прекращение декомпозиции.

3. Необходимо изменить точку зрения, чтобы детализировать блок. Изменение точки зрения происходит примерно так же, как изменение уровня абстракции. Это чаще всего характерно для ситуаций, когда точку зрения модели нельзя использовать для декомпозиции конкретного блока, т. е. этот блок можно декомпозировать, только если посмотреть на него с другой позиции. Об этом может свидетельствовать заметное изменение терминологии.

4. Блок очень похож на другой блок той же модели или на блок другой модели. Иногда встречается блок, чрезвычайно похожий на другой блок модели. Два блока похожи, если они выполняют примерно одну и ту же функцию и имеют почти одинаковые по типу и количеству входы, управления и выходы. Если второй блок уже декомпозирован, то разумно отложить декомпозицию и тщательно сравнить два блока. Если нужны ничтожные изменения для совпадения первого блока со вторым, то внесение этих изменений сократит усилия на декомпозицию и улучшит модульность модели (т.е. сходные функции уточняются согласованным образом).

5. Блок представляет тривиальную функцию. Тривиальная функция – это такая функция, понимание которой не требует никаких объяснений. В этом случае очевидна целесообразность отказа от декомпозиции, потому что роль SADT заключается в превращении сложного вопроса в понятный, а не в педантичной разработке очевидных деталей. В таких случаях декомпозиция определенных блоков может принести больше вреда, чем пользы. Тривиальные функции лучше всего описываются небольшим объемом текста. Следует заметить, что «тривиальный» не означает «бесполезный». Тривиальные функции выполняют очень важную роль, поясняя работу более сложных функций, а иногда и соединяя вместе основные подсистемы. Поэтому при анализе не следует пропускать тривиальные функции. Наоборот, их существование должно быть зафиксировано и они должны быть детализированы, как и любые другие функции. Однако следует предостеречь от больших затрат времени на анализ тривиальных функций системы. Усиленное внимание к мелочам может привести к созданию модели, которой будет недоставать абстракции, что сделает ее трудной для понимания и использования.

Общее число уровней в модели (включая контекстный) не должно превышать 5-6 (правило связано с ограничением мощности краткосрочной памяти человека). Практика показывает, что этого вполне достаточно для построения полной функциональной модели современного предприятия любой отрасли.

SADT, + Use case (человечки).

Метод SADT в наибольшей степени подходит для описания процессов верхнего уровня управления. Его основные преимущества заключаются в следующем:

1. полнота описания бизнес-процесса (управление, информационные и материальные потоки, обратные связи);

2. комплексность декомпозиции;

3. возможность агрегирования и детализации потоков данных и информации (разделение и слияние дуг);

4. наличие жестких требований, обеспечивающих получение моделей стандартного вида;

5. простота документирования процессов;

6. соответствие подхода к описанию процессов стандарту ISO 9000:2000.

В то же время метод SADT обладает рядом недостатков:

1. сложность восприятия (большое количество дуг на диаграммах);

2. большое количество уровней декомпозиции;

3. трудность увязки нескольких процессов, представленных в различных моделях одной и той же организации.

Метод SADT может использоваться для моделирования самых разнообразных процессов и систем. В существующих системах метод SADT может быть использован для анализа функций, выполняемых системой, и указания механизмов, посредством которых они осуществляются.

Концептуальное (инфологическое) проектирование — построение семантической модели предметной области, то есть информационной модели наиболее высокого уровня абстракции. Такая модель создаётся без ориентации на какую-либо конкретную СУБД и модель данных. Термины «семантическая модель», «концептуальная модель» и «инфологическая модель» являются синонимами. Кроме того, в этом контексте равноправно могут использоваться слова «модель базы данных» и «модель предметной области» (например, «концептуальная модель базы данных» и «концептуальная модель предметной области»), поскольку такая модель является как образом реальности, так и образом проектируемой базы данных для этой реальности.

Конкретный вид и содержание концептуальной модели базы данных определяется выбранным для этого формальным аппаратом. Обычно используются графические нотации, подобные ER-диаграммам.

Чаще всего концептуальная модель базы данных включает в себя:

• описание информационных объектов или понятий предметной области и связей между ними.

• описание ограничений целостности, то есть требований к допустимым значениям данных и к связям между ними.

Логическое (даталогическое) проектирование — создание схемы базы данных на основе конкретной модели данных, например, реляционной модели данных. Для реляционной модели данных даталогическая модель — набор схем отношений, обычно с указанием первичных ключей, а также «связей» между отношениями, представляющих собой внешние ключи.

Преобразование концептуальной модели в логическую модель, как правило, осуществляется по формальным правилам. Этот этап может быть в значительной степени автоматизирован.

На этапе логического проектирования учитывается специфика конкретной модели данных, но может не учитываться специфика конкретной СУБД.

Физическое проектирование — создание схемы базы данных для конкретной СУБД. Специфика конкретной СУБД может включать в себя ограничения на именование объектов базы данных, ограничения на поддерживаемые типы данных и т. п. Кроме того, специфика конкретной СУБД при физическом проектировании включает выбор решений, связанных с физической средой хранения данных (выбор методов управления дисковой памятью, разделение БД по файлам и устройствам, методов доступа к данным), создание индексов и т. д.

При проектировании реляционных баз данных обычно выполняется так называемая нормализация.

Модель «сущность-связь» (англ. “Entity-Relationship model”), или ER-модель, предложенная П. Ченом[1] в 1976 г., является наиболее известным представителем класса семантических (концептуальных, инфологических) моделей предметной области. ER-модель обычно представляется в графической форме, с использованием оригинальной нотации П. Чена, называемой ER-диаграмма, либо с использованием других графических нотаций (Crow’s Foot, Information Engineering и др.).

Основные преимущества ER-моделей:

• наглядность;

• модели позволяют проектировать базы данных с большим количеством объектов и атрибутов;

ER-модели реализованы во многих системах автоматизированного проектирования баз данных (например, ERWin). Основные элементы ER-моделей:

• объекты (сущности);

• атрибуты объектов;

• связи между объектами.

Сущность — объект предметной области, имеющий атрибуты.

Связь между сущностями характеризуется:

• типом связи (1:1, 1:N, N:М);

• классом принадлежности. Класс может быть обязательным и необязательным. Если каждый экземпляр сущности участвует в связи, то класс принадлежности — обязательный, иначе — необязательный.

Семантическая модель (концептуальная модель, инфологическая модель) — модель предметной области, предназначенная для представления семантики предметной области на самом высоком уровне абстракции. Это означает, что устранена или минимизирована необходимость использовать понятия «низкого уровня», связанные со спецификой физического представления и хранения данных.

Семантическое моделирование стало предметом интенсивных исследований с конца 1970-х годов. Основным побудительным мотивом подобных исследований (то есть проблемой, которую пытались разрешить исследователи) был следующий факт. Дело в том, что системы баз данных обычно обладают весьма ограниченными сведениями о смысле хранящихся в них данных. Чаще всего они позволяют лишь манипулировать данными определенных простых типов и определяют некоторые простейшие ограничения целостности, наложенные на эти данные. Любая более сложная интерпретация возлагается на пользователя. Однако было бы замечательно, если бы системы могли обладать немного более широким объемом сведений и несколько интеллектуальнее отвечать на запросы пользователя, а также поддерживать более сложные (то есть более высокоуровневые) интерфейсы пользователя.

Идеи семантического моделирования могут быть полезны как средство проектирования базы данных даже при отсутствии их непосредственной поддержки в СУБД. Наиболее известным представителем класса семантических моделей является модель «сущность-связь» (ER-модель).

Анализ свойств объекта недвижимости показал, что структуры данных для представления этого объекта не являются жестко заданными: невозможно представить все классы объектов недвижимости одной и той же структурой данных с эффективным использованием реляционной модели представления. В связи с этим необходимо использование систем управления базами данных, основанных на принципах NoSQL.

В классе NoSQL отдельный подклассом выступают системы, отображающие объекты приложения в реляционную модель представления. Данный подкласс называется – отображение объектов в реляционную модель, анг. – Object-Rational Mapping.

ХЪ

**Объектно-реляционное отображение** – это технология программирования, которая позволяет преобразовывать несовместимые типы моделей в ООП, в частности, между хранилищем данных и объектами программирования. ORM используется для упрощения процесса сохранения объектов в реляционную базу данных и их извлечения, при этом ORM сама заботится о преобразовании данных между двумя несовместимыми состояниями. Большинство ORM-инструментов в значительной мере полагаются на метаданные базы данных и объектов, так что объектам ничего не нужно знать о структуре базы данных, а базе данных — ничего о том, как данные организованы в приложении. ORM обеспечивает полное разделение задач в хорошо спроектированных приложениях, при котором и база данных, и приложение могут работать с данными каждый в своей исходной форме.

В среде программирования C\# основной технологией доступа к реляционным базам данных является интерфейс ADO.NET. Большинство популярных систем ORM адаптируют объекты именно к этому интерфейсу.

Говоря конкретнее, использование ORM решает проблему так называемой парадигмы «несоответствия», которая гласит о том, что объектные и реляционные модели не очень хорошо работают вместе. Реляционные базы представляют данные в табличном формате, в то время как объектно-ориентированные языки представляют их как связанный граф объектов. Основные проблемы и несоответствия возникают во время сохранения этого графа объектов в реляционную базу или его загрузки:

• реляционная модель может быть намного детальнее, чем объектная, т.е. для хранения одного объекта в реляционной базе данных используется несколько таблиц;

• реляционные СУБД не имеют ничего похожего на наследование — естественную парадигму объектно-ориентированных языков программирования;

• в СУБД определен только один параметр для сравнения записей — первичный ключ. В то время как ООП предоставляет как проверку идентичности объектов (a==b), так и их равенства (a.equals(b));

• для связи объектов СУБД использует понятие внешних ключей, в объектно-ориентированных языках связь между объектами может быть только однонаправленной. Если же нужно организовать двунаправленные отношения, то придется определить две однонаправленные ассоциации. Кроме того, нет возможности определить кратность отношения, глядя на модель предметной области;

• принцип доступа к данным в ООП кардинально отличается от доступа к данным в БД. Для доступа к данным в ООП используются последовательные переходы от родительского объекта к свойствам дочерних элементов и инициализации объектов по необходимости. Такой подход считается не эффективным способом извлечения данных из реляционных баз данных. Как правило, количество запросов к БД должно быть сведено к минимуму, необходимые сущности должны по возможности загружаться сразу с использованием JOIN-ов.

# 3.3 Принцип работы ORM

Ключевой особенностью ORM является отображение, которое используется для привязки объекта к его данным в БД. ORM как бы создает «виртуальную» схему базы данных в памяти и позволяет манипулировать данными уже на уровне объектов. Отображение показывает как объект и его свойства связанны с одной или несколькими таблицами и их полями в базе данных. ORM использует информацию этого отображения для управления процессом преобразования данных между базой и формами объектов, а также для создания SQL-запросов для вставки, обновления и удаления данных в ответ на изменения, которые приложение вносит в эти объекты.

# 3.4 Преимущества и недостатки использования ORM

Использование ORM в проекте избавляет разработчика от необходимости работы с SQL и написания большого количества кода, часто однообразного и подверженного ошибкам. Весь генерируемый ORM код предположительно хорошо проверен и оптимизирован, поэтому не нужно в целом задумывается о его тестировании. Это несомненно является плюсом, но в тоже время не стоит забывать и о минусах. Основной из них — это потеря производительности. Это происходит потому, что большинство ORM предназначены для обработки широкого спектра сценариев использования данных, гораздо большего, чем любое отдельное приложение когда-либо сможет использовать. Вопрос о целесообразности использования ORM по большому счету затрагивается только в больших проектах, которые сталкиваются с высокой нагрузкой, здесь приходится выбирать что более приоритетно — удобство или производительность? Конечно, работа с БД посредством грамотно написанного SQL-кода будет намного эффективнее, но не стоит забывать и о таком параметре, как время — то, что с легкостью пишется с использованием ORM за неделю, можно реализовывать ни один месяц собственными усилиями. Кроме того, большинство современных ORM позволяют программисту при необходимости самому задавать код SQL-запросов. Без сомнений, для небольших проектов использование ORM будет куда более оправдано, чем разработка собственных библиотек для работы с БД.

Одна из ошибок, которую делают разработчики (и я когда-то в их числе) — это утверждение о том, что вы должны использовать ровно одну ORM-стратегию для создаваемого приложения. В общем случае это неверно. Вы можете (и должны) привязывать выбор стратегии к конкретному сценарию, и быть уверенным в том, что выбираете правильные инструменты для конкретного случая.

В 99.9% случаев рекомендовано не использовать ADO.NET напрямую. Некоторые специалисты в области хранения данных не считают ORM технологией, заслуживающей внимания. Например, Мартин Фаулер написал: «Наибольшее разочарование от ORM заключается в завышенных ожиданиях». ORM предназначены решать задачу представления объектных данных и имеют множество разных подходов для этого.

# 3.5 Прямое отображение сущностей

В таком отображение почти всегда таблицы базы данных соотносятся 1:1 с сущностями в вашей системе. Когда вы добавляете свойство к объекту — добавляете и колонку к таблице. Использование такого способа строится вокруг загрузки сущности (или агрегата) по его идентификатору, управлению этим объектом и, возможно, связанными объектами, а затем сохранения этого объекта в базу данных посредством ORM.

ORM в этом случае предоставляет множество функционала, например:

• Слежение за изменениями

• Ленивая загрузка (lazy-loading)

• Предзагрузка (eager fetching)

• Каскадность

• Обеспечение уникальности объектов (Identity map)

• Работа с единицами работы (Unit of work)

Если работать с только одной сущностью или агрегатом одновременно, то такие ORM как NHibernate нам очень подходят. Они используют указанную конфигурацию для слежения за загруженными сущностями и автоматическим сохранением изменений во время коммита транзакции. И это приятно, потому что мы не должны таскать за собой свой слой работы с данными. NHibernate делает всю грязную работу за нас.

Пока мы загружаем объект по Id с единственной целью изменить его, всё это работает отлично. Это избавляет от большого количества кода, который бы мне потребовалось создать чтобы следить за добавлением объектов, их сохранением и т.д.

Обратная сторона такого подхода в том, что ORM не знает, собираетесь ли вы только читать объекты, или загружаете сущность, чтобы изменить её. Мы часто видим людей спотыкающихся, когда они не понимают, что трекинг изменений включен по умолчанию и как он работает.

Если вы хотите загрузить сущность чтобы её изменить и сохранить изменения (или создать новую сущность), этот подход обеспечивает большую гибкость от включения уровня доступа к данным в ваш инфраструктурный слой и позволяет вашим типам сущностей быть относительно независимыми от их метода сохранения. Эта независимость не означает, что моя модель C\# и схема данных могут расходиться. Напротив, это означает, что слой доступа к данным не проникнет в мою объектную модель, которую вместо этого я бы скорее предпочёл нагрузить бизнес-правилами.

# 3.6 Отображение в наборы данных

В большинстве приложений, требования к чтению данных существенно превосходят количество записей. Мы видели соотношение в 100:1 между SELECT и INSERT/UPDATE/DELETE в нашем недавнем приложении. Когда мы смотрим, в чём SQL действительно хорош, так это в работе с данными в сетах (наборах). Чтобы выбрать какой-то набор данных из SQL сервера, часто не имеет никакого смысла пытаться прямо отображать эти данные в сущности.

Но мы всё равно предпочитаем не работать напрямую с IDataReader или DataTable. Это плохо-типизированные объекты, тяжело переносимые в верхние слои приложения. Наоборот, мы часто строим объекты, приспособленные к данным. Эти объекты часто называется DTO (Data-Transfer Objects), или модели для чтения (Read Models). Такие DTO мы создаём для индивидуальных SQL выборок — и редко для того, чтобы повторно использовать их в других запросах.

Многие ORM имеют функционал, оптимизированный для таких сценариев. В NHibernate, вы можете использовать проекции чтобы выключить трекинг, и отобразить данные напрямую в DTO. Вы можете использовать SQL запросы чтобы сделать это и не нуждаетесь в конфигурации маппинга. Или вы можете использовать микро-ORM например PetaPoco.

Эти чтения также могут генерировать DTO объекты по мере их чтения. И NHibernate и несколько micro-ORMs позволяют получать индивидуальные DTO объекты последовательно один за одним во время чтения строк результатов запроса, тем самым минимизируя объем объектов содержащихся в памяти.

В наших приложения, мы до сих пор часто используем NHiberante для чтения, но не используем объекты сущностей, а вместо этого используем сырой SQL. Мы полагаемся на оптимизированные мапперы NHiberanate, чтобы просто подать тип DTO, а результат выборки отобразится автоматически.

Этот подход не очень хорошо работает, если нам надо применить бизнес правила и сохранить информацию обратно. Так как эти модели обычно отображаются в отдельные наборы данных, а не в таблицы базы данных.

Active Record – это другой пример сущностного отображения данных, в котором, функционал работы с данными включён в саму объектную модель.

4.6 Отображение DML-запросов (DML-based relational mapping)

Если вы знаете, какой SQL вам нужен для реализации CRUD, и предпочли бы создавать его вручную, то вы уже ищите что-то, чтобы эффективно абстрагировать DML команды на уровень выше, чем ADO.NET.

И это арена микро-ORM. Такие фреймворки как PetaPoco, Dapper, Massive и другие созданы, чтобы помочь решить пробемы работы ADO.NET. Они обычно всё равно позволяют нам работать с объектами ADO.NET, но наше взаимодействие сильно упрощается. Нам только нужно соединение, и эти фреймворки могут позволить работать со всеми CRUD операциями в виде, который предлагает намного более простой код, чем сам ADO.NET.

В случая, когда у вас нет сущностей и нужды отображать их в таблицы и обратно, микро-ORM дадут гораздо более лёгкий подход. А так микро-ORM не требуют предварительной конфигурации, то они полагаются на ленивое-исполнение и оптимизированные техники кеширования, чтобы налету маппить SQL параметры и результаты запросов. Многие приложение могут начать с маппигом основанном на DML, переходя на полноценную ORM, как только отношения или сущности потребуют этого.

# 3.7 Инструменты массовой загрузки (Bulk loading tools)

Это то, что занимает особое место — иногда вы не хотите вставлять/загружать данные объектным способом.

Вместо этого, вы бы предпочли работать с всеми наборами целиком. Такие инструменты, как SQL Bulk Copy, позволяют вам получать и выгружать данные в CSV или в табличных форматах,

Эти утилиты работают примерно как базука, вырывая все данные сразу туда и обратно, но не предоставляя ничего кроме этого. Вы не можете обновлять или удалять данные, но для того, чтобы получить большие объёмы данных из SQL, эти утилиты — то что вам нужно.

Во многих интеграционных сценариях, где вы предоставляете файлы с данными внешним партнёрам, или наоборот — эти загрузчики позволяют пользоваться файлами как таблицам и напрямую загружать их в базы данных.

Эти утилиты намного быстрее традиционных методов парсинга/загрузки данных. В некоторых из наших тестов, мы видели разницу в порядки по сравнению с построчной загрузкой. А в одном случае, мы видели разницу между несколькими часами и минутой. Обратная сторона всего этого, это то, что функционал ограничен только лишь INSERT и SELECT. Всё остальное требует других подходов.

# 3.8 Технология Entity Framework

ADO.NET Entity Framework (EF) — объектно-ориентированная технология доступа к данным, является ORM-решением для .NET Framework от Microsoft. Предоставляет возможность взаимодействия с объектами как посредством LINQ в виде LINQ to Entities, так и с использованием Entity SQL. Для облегчения построения web-решений используется как ADO.NET Data Services (Astoria), так и связка из Windows Communication Foundation и Windows Presentation Foundation, позволяющая строить многоуровневые приложения, реализуя один из шаблонов проектирования MVC, MVP или MVVM.

Версия 6.0 была выпущена 17 октября 2013 года[3] и сейчас это проект с открытым исходным кодом под лицензией Apache License v2. В версии 6.0 был сделан ряд улучшений в поддержке метода работы Code First.

Это альтернативный интерфейс LINQ API, используемый для обращения к базе данных. Он отделяет сущностную объектную модель данных от физической базы данных, вводя логическое отображение между ними. Так, например, схемы реляционных баз данных не всегда подходят для построения объектно-ориентированных приложений и в результате мы имеем объектную модель приложения, существенно отличающуюся от логической модели данных, в этом случае используется LINQ to Entities, который использует модель EDM (Entity Data Model). То есть, если вам нужно ослабить связь между вашей сущностной объектной моделью данных и физической моделью данных, например, если ваши сущностные объекты конструируются из нескольких таблиц или вам нужна большая гибкость в моделировании ваших сущностных объектов используйте LINQ to Entities

Изначально с самой первой версии Entity Framework поддерживал подход Database First, который позволял по готовой базе данных сгенерировать модель edmx. Затем эта модель использовалась для подключения к базе данных. Позже был добавлен подход Model First. Он позволял создать вручную с помощью визуального редактора модель edmx, и по ней создать базу данных. Начиная с 5.0 предпочтительным подходом становится Code First[5]. Его суть - сначала пишется код модели на C\#, а затем по нему генерируется база данных. При этом модель edmx уже не используется.

# 3.9 Выбор технологии ORM

Ключевым моментом к выбору ORM является то, что нет необходимости привязывать себя к определённому инструменту или подходу. Никакая ORM-стратегия не работает во всех сценариях, и не должна этого делать. NHibernate может работать со многими другими сценариями (кроме непосредственного маппинга сущностей), но не делает всего на свете. Сложность часто возникает из-за попыток использовать один и тот же подход всегда.

Каждое приложение, написано вне SQL сервера использует ORM. Или этого рукописный ADO.NET код, или NHibernate — вы должны преодолевать разрыв между .NET и SQL. Это преодоление — тяжёлая задача, и ничто не решает задачу полностью идеально.

Основной рекомендацией к выбору ORM является подбор подхода, который решает конкретную проблему. Не беспокойтесь, что у вас будет несколько ORM-стратегий в одном проекте. Это не означает, что бессистемные решения приемлемы. Но наоборот — применение выверенных решений, основанных на знании возможных вариантов — всегда хорошая идея.

ГЛАВА 2. Реализация рекомендательной системы

# Заключение

В выпускной квалификационной работе представлены результты проеутирования рекомендательной информационной системы (РС) для рынка недвижимости Иркутской области. Выделены основные проблемы, требующие решения как на этапе разработки РС, так и на этапе их эксплуатации:

1. Проведен анализ свойств объекта недвижимости как информационного объекта, выделены соответствующие классы свойств.

2. Изучены стандартные методики анализа бизнес-процессов в сфере услуг.

3. Проведен анализ современных подходов к построению баз данных для хранения информационных объектов гибкой структуры.

4.

5.

6.

7. Проведено тестирование программной системы.

К разрабатываемой системе выдвигались следующие требования:

1. Разрабатываемая информационная система должна быть реализована как система, функционирующая в Интернет.

2. Система должна поддерживать интеграцию данных с существующими системами класса Avito.

3. Импорт данных из баз данных и документов риэлтерского агентства, в том числе при помощи пользовательского интерфейса.

# Глава 4. Регистрация пользователя и сбор информации о предпочтениях

Пользователь регистрируется на сайте либо как покупатель, либо как риелтор. Программа сразу начинает следить за тем какие объекты просматривал покупатель. Потом при помощи кластерного анализа высчитывает какие объекты предлагать тому или иному пользователю.

По умолчанию, пользователь регистрируется автоматически при заходе на сайт использующий систему обмена. Минус такого ограничения для системы– каждое устройство пользователя для нас является отдельным новым пользователем. Пользователь же таким образом не может получать актуальные рекомендации и не может редактировать настройки. Таким образом, целесообразно, помимо скрытого механизма регистрации, предоставить пользователю, возможность, зарегистрироваться самостоятельно. В момент регистрации система обмена должна импортировать настройки текущего скрытого пользователя.

Импорт данных

Импорт данных производится с сайта Атлант-недвижимость и Яндекс макет. Таким же методом можно получать информацию об объектах с других сайтов.

Формат импортируемого файла

Объекты в базу данных поставляется из файлов в расширении Xml b Xml.Zip

# 4.1 Первичная структуризация: кластерный анализ /средство первичной структуризации/

При анализе и прогнозировании социально-экономических явлений исследователь довольно часто сталкивается с многомерностью их описания. Это происходит, например, при решении задач сегментирования рынка, построения типологии стран по достаточно большому числу показателей, прогнозирования конъюнктуры рынка отдельных товаров, изучения и прогнозирования экономической депрессии и многих других проблем.

Кластерный анализ — один из методов многомерной статистики — наиболее ярко отражает черты многомерности в процедуре классификации объектов. Название «кластерный анализ» происходит от английского слова «cluster» — гроздь, скопление. Впервые определил предмет кластерного анализа и дал его описание исследователь Трион (Тгуоп) в 1939 г. [3].

Главное назначение кластерного анализа — разбиение множества исследуемых объектов, характеризуемых совокупностью признаков, на однородные в соответствующем понимании группы (кластеры). Это означает, что решается задача классификации данных и выявления соответствующей структуры в ней. Иными словами, предполагается выделение компактных, удаленных друг от друга групп объектов или отыскание «естественного» разбиения совокупности на области скопления.

Кластерный анализ является одним из направлений статистического исследования социально-экономических процессов, которые связаны с изучением массовых явлений.

Выработка рекомендаций: метод Slope One /средство анализа предпочтений и выработки рекомендаций/

Подход Slope One

Slope One - это один из самых простых подходов к рекомендациям на основе коллаборативной фильтрации по схожести предметов, но в то же время точность рекомендаций алгоритма сравнима с более сложными и ресурсоемкими алгоритмами . Он был разработан Даниелем Лемайром и Анной Маклахман в 2004 году и опубликован в 2005 году в статье .

Разработка сайта

Тестирование

# Заключение

В выпускной квалификационной работе приведен процесс проектирования и тестирование рекомендательной системы (РС) для рынка недвижимости Иркутской области. Выделены основные проблемы, требующие решения как на этапе разработки РС, так и на этапе их эксплуатации, а также решены следующие задачи:

1.

Разработанная РС обладает следующими характеристиками:

1. функционирует в Интернет.

2. поддерживает интеграцию данных с существующими системами класса Avito;

3. поддерживает импорт данных из баз данных с сайтов продажи недвижимости и др.

# В данном списке литературы приведены примеры оформления различных источников. Там сейчас есть ошибки наверняка.

# Список использованной литературы

Понтрягин~Л.~С. Принцип максимума в оптимальном управлении. Изд.~2-е, стереотипное. М.\,: Едиториал УРСС, 2004.~–~64~с.

\bibitem{rzhddb} URL:\href{http://mapservis.ru/docs/tar\_ruc\_4.htm}{Тарифное руководство № 4. Книга 2. Часть 1. Алфавитный список железнодорожных станций.} [Электронный ресурс]\,{}: сайт. \url{http://mapservis.ru/docs/tar\_ruc\_4.htm} (дата обращения: 06.05.2015).

\bibitem{citycoords} \href{http://alextyurin.ru/?p=1037}{Географические координаты основных городов России}. [Электронный ресурс]\,{}: сайт. URL:\url{http://alextyurin.ru/?p=1037} (дата обращения: 06.05.2015).

\bibitem{Anderson} Андерсон~Р. \emphbib{Доказательство правильности программ}\,{}: пер. с англ.\,{}/ Р.~Андерсон. – М.\,:\,Мир, 1982. – 168~c.: ил.

\bibitem{Bratko} Братко~И. \emphbib{\href{http://royallib.ru/book/bratko\_ivan/programmirovanie\_na\_yazike\_prolog\_dlya\_iskusstvennogo\_intellekta.html}{Программирование на языке ПРОЛОГ для искусственного интеллекта}}\,{}: пер. с англ.\,/ И.~Братко. – М.\,:~Мир, 1990. – 560~c.: ил.

\bibitem{Vass:2000} Васильев~С.~Н. \emphbib{\href{http://bookfi.org/book/616050}{Интеллектное управление динамическими системами}}\,{}/ С.~Н.~Васильев, А.~К.~Жерлов, Е.~А.~Федосов, Б.~Е.~Федунов. – М.\,:~Физматлит, 2000. – 352~с: ил.

\bibitem {AIDictionary} \emphbib{\href{http://aihandbook.intsys.org.ru/index.php/intro/ai-handbook}{Искусственный интеллект\,{}: в 3~кн.}}\,{}/ под ред. Э.~В. Попова. – М.\,:~Радио и связь, 1990. – 464 c.:\,{}ил.

\bibitem{Lauriere} Лорьер.~Ж.-Л. \emphbib{\href{http://publ.lib.ru/ARCHIVES/L/LOR'ER\_Jan\_Lui/\_Lor'er\_J.L..html}{Системы искусственного интеллекта}\,{}: пер. с франц.}\,{}/ Ж.-Л. Лорьер. – М.\,:~Мир, 1991. – 568~с.: ил.

\bibitem{Malpas} Малпас~Дж. \emphbib{\href{http://padaread.com/?book=40731&pg=1}{Реляционный язык Пролог и его применение}}\,{}/ Дж.~Малпас. – М.\,:~Наука, 1990. – 464~с.

\bibitem{math\_slov:88} \emphbib{\href{https://app.box.com/shared/793ukgvblxmj0hh6btw4}{Математический энциклопедический словарь}}\,{}/ гл.~ред. Ю.~В.~Прохоров. – М.\,:~Сов.~энциклопедия, 1988. – 847~c.

\bibitem{DDW} Непейвода~Н.~Н. \emphbib{\href{http://www.logic-books.info/taxonomy/term/215}{Прикладная логика\,{}: учеб. пособие}}\,{}/ Н.~Н.~Непейвода. – 2-е изд. – Новосибирск\,{}:~Изд-во Новосиб. ун-та, 2000. – 521~c.: ил.

%\bibitem{DDWII} Непейвода~Н.~Н. \emphbib{\href{http://philosophy.ru/library/logic\_math/library/nepeivoda\_prog.pdf}{Основания программирования}}\,{}/ Н.~Н.~Непейвода, И.~Н.~Скопин. – Москва; Ижевск\,{}:~Институт компьютерных исследований, 2003 – 880~c.: ил.

\bibitem {Russell} Рассел~С. \href{http://www.aiportal.ru/downloads/books/ai-modern-approach-2-edition-by-rassel-norvig.html}{Искусственный интеллект: современный подход}\,{}: пер. с англ.\,{}/ С.~Рассел, П.~Новриг. 2-е изд. – М.\,:~Изд. дом <<Вильямс>>, 2006. – 1408~c.: ил.

\bibitem{WIKI-DCG} \emphbib{\href{https://en.wikipedia.org/wiki/Definite\_clause\_grammar}{DC-грамматика}} [Электронный ресурс]\,{}// Wikipedia, The Free Encyclopedia\,{}: сайт. – URL:\texttt{https://en.wikipedia.org/wiki/Definite\\_clause\linebreak\\_grammar}. (дата обращения: 28.11.2013).

\bibitem{GNUP} \emphbib{\href{http://www.gprolog.org/}{The GNU Prolog web site} [Электронный ресурс]\,{}: сайт}. URL:\url{http://www.gprolog.org/}. (дата обращения: 28.11.2013).

\bibitem{SWIP} \emphbib{\href{http://www.swi-prolog.org/}{SWI-Prolog's home} [Электронный ресурс]\,{}: сайт}. URL:\url{http://www.swi-prolog.org/}. (дата обращения: 28.11.2013).

\bibitem{pythondl} Welcome to Python.org. [Электронный ресурс]\,{}: сайт. URL:\url{https://www.python.org/} (дата обращения: 11.01.2015).

\bibitem{pythondoc} \href{http://younglinux.info/sites/default/files/python\_structured\_programming.pdf}{Основы программирования на Python.} [Электронный ресурс]\,{}: сайт. URL:\url{http://younglinux.info/sites/default/files/python\_structured\_programming.pdf}. (дата обращения: 11.01.2015).

\bibitem{pythonbook}М.~Лутц. \href{https://vk.com/doc10903696\_196246835?hash=4b80f3cf914c7d65dd&dl=f90686bee18e565271}{Изучаем Python, 4-е издание.}– Пер.~с англ.–СПб.:Символ-Плюс, 2011. 1280~с.,~ил.

\bibitem{nominatim} OpenStreetMap Nominatim: Search. [Электронный ресурс]\,{}: сайт. URL:\url{http://wiki.openstreetmap.org/wiki/Nominatim}. (дата обращения: 11.01.2015).

\bibitem{sethi}S.~P.~Sethi, G.~L.~Thomson. Optimal Control Theory: Applications to Management Science and Economics. 2nd Edition. 2005. 506~pp.

\end{thebibliography}

\chapter\*{Приложение}

\label{lastpage}