Оглавление

|  |  |
| --- | --- |
| Введение…………………………………………………………………………. | 3 |
| Постановка задачи……………………………………………………………… | 5 |
| Глава 1. Экспертные системы………………………………………………….. | 6 |
| 1.1 Основные понятия ЭС…………………………………………………….. | 6 |
| 1.2 Классификация ЭС по решаемой задаче …………………………………. | 9 |
| 1.3 Достоинства и недостатки ЭС…………………………………………….. | 10 |
| Глава 2. Средства реализация ……………………………………………….. | 13 |
| Глава 3. Создание базы знаний………………………………………………. | 16 |
| 3.1. Обзор предметной области…………………………………………..….. | 16 |
| 3.2. Разработка семантической связи………………………………………… | 17 |
| 3.3. Заполнение базы знаний…………………………………………………. | 18 |
| Глава 4. Создание интеллектуального интерфейса………………………… | 21 |
| Глава 5. Тестирование интеллектуальной системы………………………… | 24 |
| 5.1 Апробация ………………………………………………………………… | 25 |
| Заключение…………………………………………………………………….. | 28 |
| Список литературы……………………………………………………………. | 29 |
| Приложение …………………………………………………………………….. | 30 |

# Введение

‏ㅤ Начиная ‏ㅤс середины ‏ㅤдвадцатого столетия ‏ㅤи по сей день, одним из самых значимых ‏ㅤнаправлений развития ‏ㅤнауки и техники ‏ㅤявляется искусственный ‏ㅤинтеллект. Задачей ‏ㅤданного направления ‏ㅤявляется обеспечение ‏ㅤразумных рассуждений ‏ㅤи действий ‏ㅤс помощью ‏ㅤвычислительных систем и других искусственных ‏ㅤустройств.

Интеллектуальная система — это техническая или программная система, способная решать задачи, традиционно считающиеся творческими, принадлежащие конкретной предметной области, знания о которой хранятся в памяти такой системы. Структура интеллектуальной системы включает три основных блока — базу знаний, механизм вывода решений и интеллектуальный интерфейс. Одним из наиболее развитых типов интеллектуальных систем являются экспертные системы, при изучении которых важным является представление структуры экспертных систем и их классификация, принципы построения и технология их разработки, представление знаний в экспертных системах и достижение целей[1].

Отдельная ‏ㅤобласть искусственного ‏ㅤинтеллекта - инженерия ‏ㅤзнаний – занимается ‏ㅤразработкой экспертных ‏ㅤсистем, развитием ‏ㅤсредств представления, структурирования ‏ㅤи обработки ‏ㅤзнаний на компьютере.

Экспертная ‏ㅤсистема - система ‏ㅤискусственного интеллекта, содержащая ‏ㅤзнания об определенной ‏ㅤслабо структурированной‏ㅤи трудно формализуемой ‏ㅤузкой предметной ‏ㅤобласти и способная ‏ㅤпредлагать пользователю ‏ㅤразумные решения ‏ㅤи их объяснения. Экспертные ‏ㅤсистемы могут принимать ‏ㅤрешения, схожие с решениями ‏ㅤэкспертов в заданной ‏ㅤпредметной области ‏ㅤи решают задачи диагностики, прогнозирования ‏ㅤили планирования.

Рассмотрим ‏ㅤпримеры наиболее ‏ㅤизвестных экспертных ‏ㅤсистем, с которых ‏ㅤначалось создание ‏ㅤи развитие ‏ㅤэтого типа программных ‏ㅤсредств:

MYCIN - это экспертная ‏ㅤсистема, разработанная ‏ㅤв начале 1970х годов в Стэндфордском ‏ㅤуниверситете. Система ‏ㅤпредназначена для диагностики ‏ㅤи лечения ‏ㅤмедицинских инфекций. Исходя из представленных ‏ㅤпациентом симптомов, система ‏ㅤставит диагноз ‏ㅤи рекомендует ‏ㅤкурс со- ответствующего ‏ㅤмедикаментозного лечения.[2]

PROSPECTOR - экспертная ‏ㅤсистема, применяемая ‏ㅤпри поиске место- рождений ‏ㅤполезных ископаемых. Система ‏ㅤделает выводы на основе геологических ‏ㅤанализов.[3]

DENDRAL - это старейшая ‏ㅤэкспертная система ‏ㅤв мире. Экспертная ‏ㅤсистема автоматизировала ‏ㅤпроцесс определения ‏ㅤхимической структуры ‏ㅤвещества и определяла ‏ㅤстроение органических ‏ㅤмолекул похимическим ‏ㅤформулам и спектрографическим ‏ㅤданным о химических ‏ㅤсвязях в молекулах. [4]

Мы видим, что экспертные ‏ㅤсистемы имеют дело с предметами ‏ㅤреального мира и выполняют ‏ㅤоперации, которые ‏ㅤобычно требуют ‏ㅤналичия значительного ‏ㅤопыта, накопленного ‏ㅤчеловеком. Обычно для создания ‏ㅤсложной экспертной ‏ㅤсистемы, требуются ‏ㅤзнания и практические ‏ㅤнавыки даже нескольких ‏ㅤспециалистов. Во-первых, для этого нужен эксперт ‏ㅤ- человек, умеющий ‏ㅤнаходить решения ‏ㅤпроблем в конкретно ‏ㅤзадано области. Во-вторых, нужен инженер ‏ㅤзнаний - человек, который ‏ㅤзнает каким образом ‏ㅤпостроить экспертную ‏ㅤсистему, как структурировать ‏ㅤи организовать ‏ㅤзнания эксперта. Ну и наконец, для построения ‏ㅤэкспертной системы ‏ㅤнеобходим программист, который ‏ㅤбудет разрабатывать ‏ㅤсаму программу, используя ‏ㅤспециальные инструментальные ‏ㅤсредства или создавая ‏ㅤновые.

Раньше на проектирование ‏ㅤи создание ‏ㅤодной экспертной ‏ㅤсистемы по- требовалось ‏ㅤбы около 20 человеко-лет. В настоящее ‏ㅤвремя имеется ‏ㅤряд инструментальных ‏ㅤсредств, значительно ‏ㅤускоряющих разработку.

Инструментальное ‏ㅤсредство разработки ‏ㅤэкспертных систем – это язык программирования, используемый ‏ㅤразработчиком для построения ‏ㅤэкспертной системы.

Постановка задачи

1. Изучить литературу, которая связанна с интеллектуальными

системами, базой знаний и выбранной предметной областью;

2. Разобрать понятие экспертная система

3. Разработать базу знаний

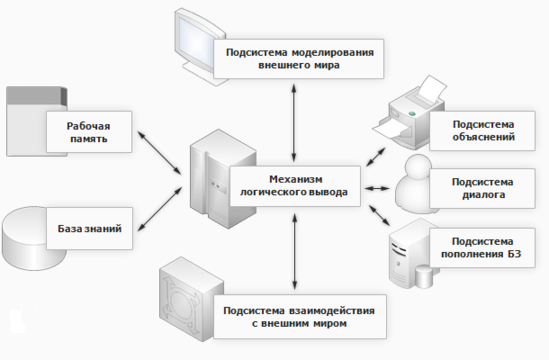
4. Реализовать экспертную систему выбора ресторана в г. Иркутске

# Глава 1. Экспертные ‏ㅤсистемы (ЭС)

## 1.1 Основные ‏ㅤпонятия экспертных систем

Экспертные ‏ㅤсистемы бывают двух типов: статические ‏ㅤи динамические. Статические ‏ㅤэкспертные системы ‏ㅤне меняются ‏ㅤс течением ‏ㅤвремени, а динамические ‏ㅤменяются, приспосабливаясь и ‏ㅤадаптируясь к изменениям ‏ㅤвнешнего мира. Например, динамические ‏ㅤэкспертные системы ‏ㅤмогут использовать ‏ㅤпоказания каких-либо датчиков ‏ㅤили заимствовать ‏ㅤдополнительную информацию ‏ㅤиз сетиInternet.

Поскольку ‏ㅤданная дипломная ‏ㅤработа посвящена ‏ㅤсозданию экспертной ‏ㅤсистемы статического ‏ㅤтипа, то на них мы и остановимся ‏ㅤподробнее. Рассмотрим ‏ㅤструктуру статической ‏ㅤэкспертной системы ‏ㅤ(рисунок. 1).

Рисунок ‏ㅤ1. Структура ‏ㅤэкспертной систем

Структура ‏ㅤэкспертной системы ‏ㅤсодержит следующие ‏ㅤобъекты:

1. базазнаний

2. рабочая‏ㅤпамять

3. механизм ‏ㅤлогическоговывода

4. подсистема‏ㅤдиалога

5. подсистема ‏ㅤприобретения и пополнения‏ㅤзнаний

6. подсистема‏ㅤобъяснения

Теперь остановимся ‏ㅤна каждом из этих понятий ‏ㅤподробнее.

База знаний - это одно из основополагающих ‏ㅤпонятий в области ‏ㅤэкспертных систем, означает ‏ㅤсовокупность знаний (фактов и правил) о предметной ‏ㅤобласти, обычно на некотором ‏ㅤязыке, близком ‏ㅤк естественному. Под правилами ‏ㅤв базе знаний понимаются ‏ㅤправила вывода, описывающие ‏ㅤотношения между фактами. Факты же имеют классическое ‏ㅤзначение и представляют ‏ㅤсобой знания в форме утверждений.

Помимо правил и фактов в базу знаний может входить ‏ㅤнекоторая процедурная ‏ㅤчасть, производящая ‏ㅤрасчетные, оптимизационные ‏ㅤили другие необходимые ‏ㅤалгоритмы.

Рабочая ‏ㅤпамять - предназначена ‏ㅤдля временного ‏ㅤхранения исходных ‏ㅤданных и промежуточных ‏ㅤфактов (полученных ‏ㅤот пользователя)‏ㅤ решаемой ‏ㅤв текущий ‏ㅤмомент задачи. Как правило, рабочая ‏ㅤпамять размещается ‏ㅤв оперативной ‏ㅤпамяти ЭВМ и отражает ‏ㅤтекущее состояние ‏ㅤпредметной области ‏ㅤв виде фактов с коэффициентами ‏ㅤуверенности в истинности ‏ㅤэтих фактов.

Механизм ‏ㅤлогического вывода предназначен ‏ㅤдля получения ‏ㅤновых фактов на основе сопоставления ‏ㅤисходных данных из рабочей ‏ㅤпамяти и знаний из базы знаний. Механизм ‏ㅤлогического вывода во всей структуре

экспертной системы ‏ㅤзанимает наиболее ‏ㅤважное место, т.к. именно он моделирует ‏ㅤход рассуждений ‏ㅤэксперта.

Подсистема ‏ㅤдиалога отвечает ‏ㅤза организацию ‏ㅤинтерфейса для общения ‏ㅤс пользователем ‏ㅤв процессе ‏ㅤрешения задачи и получения ‏ㅤрезультата. Именно этот элемент ‏ㅤэкспертной системы ‏ㅤв дальнейшем ‏ㅤбудет называться ‏ㅤпользовательским интерфейсом.

Подсистема ‏ㅤприобретения и пополнения‏ㅤзнаний автоматизирует ‏ㅤпроцесс наполнения ‏ㅤэкспертной системы ‏ㅤзнаниями, осуществляемый ‏ㅤпользователем-экспертом, и адаптации ‏ㅤбазы знаний системы ‏ㅤк условиям ‏ㅤее функционирования. Адаптация ‏ㅤэкспертной системы ‏ㅤк изменениям‏ㅤв предметной ‏ㅤобласти реализуется ‏ㅤпутем замены правил или фактов в базе знаний.

Подсистема ‏ㅤобъяснения позволяет ‏ㅤпользователю понять, каким образом ‏ㅤсистема пришла к итоговому ‏ㅤрешению задачи или почему решение ‏ㅤне было найдено. Это повышает ‏ㅤдоверие пользователя ‏ㅤк полученному ‏ㅤрезультату и упрощает ‏ㅤразработчику тестирование ‏ㅤпрограммы. Возможность ‏ㅤобъяснять свои действия ‏ㅤявляется одним из самых важных свойств ‏ㅤэкспертной системы, потому что еще больше приближает ‏ㅤее работу к работе эксперта ‏ㅤв данной области.

1.2. Классификация ‏ㅤэкспертных систем по решаемой ‏ㅤзадаче

Чтобы лучше понимать принципы работы экспертных ‏ㅤсистем, рассмотрим ‏ㅤклассы задач, которые ‏ㅤможно решать с их помощью

1. Интерпретация

‏ ㅤСогласование полученных ‏ㅤизвне данных и получение ‏ㅤисходя из них ответа на общий вопрос.

## 2. Диагностирование

Проведение различных ‏ㅤдиагностик (например, определение ‏ㅤизъянов в каком-либо механизме)‏ㅤ.

## 3. Мониторинг

ㅤНепрерывное отслеживание ‏ㅤданных и сообщение ‏ㅤо выходе тех или иных параметров ‏ㅤза допустимые ‏ㅤпределы. При решении ‏ㅤзадач мониторинга ‏ㅤс помощью ‏ㅤэкспертной системы, существует ‏ㅤриск того, что система ‏ㅤбудет подавать ‏ㅤложные сигналы, или наоборот, пропускать ‏ㅤплохие ситуации. Это связано ‏ㅤс тем, что границы ‏ㅤдопустимого обычно размыты.

## 4. Проектирование

ㅤЗадача проектирования ‏ㅤзаключается в том, чтобы подготовить ‏ㅤнеобходимый пакет документов ‏ㅤдля создания ‏ㅤобъекта, с заранее ‏ㅤзаданными свойствами.

## 5. Прогнозирование

Прогнозирование позволяет ‏ㅤпредсказать последствия ‏ㅤнекоторых событий ‏ㅤили явлений ‏ㅤна основании ‏ㅤанализа имеющихся ‏ㅤданных.

## 6. Обучение

ㅤСистема обучения ‏ㅤнаблюдает за процессом ‏ㅤизучения какой-либо дисциплины ‏ㅤпользователем-учеником ‏ㅤи, принимая ‏ㅤво внимание ‏ㅤего ошибки, подсказывает ‏ㅤправильные решения ‏ㅤи дает советы.

## 7. Планирование

ㅤ Экспертная система ‏ㅤсоставляет план действия ‏ㅤдля объекта, выполняющего ‏ㅤопределенные функции.

Таким образом, мы видим, что экспертные ‏ㅤсистемы позволяют ‏ㅤрешать довольно ‏ㅤширокий круг задач. Это тем более оправдывает ‏ㅤих использование ‏ㅤв различных ‏ㅤобластях.

## 1.3. Достоинства и недостатки ‏ㅤэкспертных систем

Конечно ‏ㅤне всегда машина, обладающая ‏ㅤискусственным интеллектом, может заменить ‏ㅤчеловека-эксперта ‏ㅤв своей области. Но в ряде случаев, применение ‏ㅤэкспертной системы ‏ㅤобладает заметными ‏ㅤпреимуществами перед использованием ‏ㅤзнаний эксперта. Итак, преимущества ‏ㅤэкспертных систем:

1. Постоянство

‏ㅤЧеловеку свойственно ‏ㅤзабывать. Поэтому ‏ㅤлюбой перерыв ‏ㅤв профессиональной ‏ㅤдеятельности эксперта ‏ㅤможет отразиться ‏ㅤна полноте ‏ㅤего знаний. В случае с экспертной ‏ㅤсистемой, это исключено, программа ‏ㅤможет годами хранить ‏ㅤпомещенные в нее знания в их первоначальном ‏ㅤвиде.

## 2. Решение ‏ㅤтрудоемкихзадач

Экспертная ‏ㅤсистема удерживает ‏ㅤв памяти многочисленные ‏ㅤфакты, правила ‏ㅤи условия ‏ㅤзадачи. Это позволяет ‏ㅤоперативно выполнять ‏ㅤдействия с данными ‏ㅤи быстро приходить ‏ㅤк какому-либо выводу.

## 3. Простота‏ㅤпередачи

Предоставление ‏ㅤзнаний от одного человека к‏ㅤдругому – длительный, трудоемкий ‏ㅤи часто дорогой ‏ㅤпроцесс. Гораздо легче передать ‏ㅤпрограмму, которая ‏ㅤбудет исполнять роль ‏ㅤэксперта своими действиями.

4. Стабильность‏ㅤрезультатов

Экспертные ‏ㅤсистемы устойчивы ‏ㅤк «помехам»‏ㅤ. Человек ‏ㅤже легко поддается влиянию ‏ㅤвнешних факторов, которые ‏ㅤнепосредственно не связаны ‏ㅤс решаемой ‏ㅤзадачей. Эксперт-человек ‏ㅤможет принимать ‏ㅤразнообразные решения ‏ㅤв одинаковых ‏ㅤситуациях из-за эмоциональных ‏ㅤфакторов.

## 5. Способность ‏ㅤк нечетким‏ㅤрассуждениям

Экспертные ‏ㅤсистемы могут ‏ㅤвоспринимать нечеткие ‏ㅤданные и проводить ‏ㅤпо ним рассуждения. К таким системам ‏ㅤотносятся системы ‏ㅤпланирования и прогнозирования, опирающиеся ‏ㅤна неполную ‏ㅤинформацию.

## 6. Стоимость

‏ㅤВысококвалифицированные специалисты ‏ㅤобходятся очень недешево. Экспертные ‏ㅤсистемы же хоть и отличаются ‏ㅤдороговизной разработки, но почти не требуют затрат при эксплуатации.

Вместе с тем разработка ‏ㅤэкспертной системы ‏ㅤне позволяет ‏ㅤполностью отказаться ‏ㅤот эксперта-человека. Хотя экспертная ‏ㅤсистема хорошо справляется ‏ㅤсо своей работой, тем не менее, в определенных ‏ㅤобластях человеческая ‏ㅤкомпетенция явно превосходит ‏ㅤискусственную. Однако и в этих случаях ‏ㅤэкспертная система ‏ㅤможет позволить ‏ㅤотказаться от услуг высококвалифицированного ‏ㅤэксперта, оставив ‏ㅤэксперта средней ‏ㅤквалификации, используя ‏ㅤпри этом экспертную ‏ㅤсистему для усиления ‏ㅤи расширения ‏ㅤего профессиональных ‏ㅤвозможностей.

Недостатки экспертных систем.

1. Здравый смысл. В дополнение к широкому техническому знанию, человек-эксперт имеет здравый смысл. Еще не известно, как заложить здравый смысл в экспертные системы.

2. Творческий потенциал. Человек-эксперт может творчески отнестись к необычности ситуаций в поставленной задаче, что экспертные системы не могут сделать.

3. Обучение. Человек-эксперт быстро адаптируется к изменением среды.

# Глава 2. Средства ‏ㅤреализации ЭС

Есть несколько ‏ㅤкатегорий инструментальных ‏ㅤсредств реализации ‏ㅤэкспертных систем. Все они имеют свои плюсы и минусы. Для того чтобы определить, какой метод лучше подходит ‏ㅤдля разработки ‏ㅤнами экспертной ‏ㅤсистемы, рассмотрим ‏ㅤкаждую категорию ‏ㅤподробнее.

СУБД - система рода совместимость управления языковых и базами данных программных – это средств, своего которые предназначены для создания, ведения и совместного использования баз данных многими пользователями. Она может быть классифицирована по способу доступа к базе данных, существуют файл-серверные СУБД, клиент-серверные и встраиваемые. Рассмотрим клиент-серверную СУБД, каждый из клиентских запросов на обработку данных обрабатывается клиент-серверной СУБД централизованно. Сама же СУБД находится на сервере вместе с базой данных и осуществляет доступ к базе данных напрямую, в монопольном режиме. Достоинства: удобство предоставления таких характеристик как высокая доступность, надежность, безопасность; допустима более низкая загрузка локальной сети; удобное централизованное управление. Недостаток – повышенные требования к серверу.

В таблице 1 представлено сравнение СУБД FireBird, MySQL и PostgreSQL.

Таблица 1.

|  |  |  |  |
| --- | --- | --- | --- |
| Характеристика | Firebird | MySQL | PostgreSQL |
| Разработчик | FirebirdFoundation | Oracle | PostgreSQL Global Development  Group |
| Язык реализации | С и С++ | С и С++ | С |
| Лицензия | Открытый исходный код | Открытый исходный код | Открытый исходный код |
| Серверные операционные системы | AIX FreeBSD HP-UX Linux OSX Solaris Unix Windows | FreeBSD Linux OS X  Solaris Windows | FreeBSD HP-UX Linux NetBSD OpenBSD OSX Solaris Unix Windows |
| Поддерживаемые языки программирования | C C++ Delphi Java JavaScript Lua Perl PHP Python Ruby | Ada C C# C++ D Delphi Eiffel Erlang Haskell Java JavaScript  Objective-C OCaml Perl PHP Python Ruby Scheme Tcl | .Net C C++ Delphi Java Perl PHP Python Tcl |

Для реализации экспертной системы в данной работе была выбрана СУБД Firebird. Это система, которая работает более чем на одной операционной системе и так же на разнообразных Unix платформах[5]

Для реализации интеллектуального интерфейса была выбрана система Borland C++ Builder. Эта система эксплуатируется программистами для разработки ПОна языке C++ и поддерживает работу с базами данных под управлением Firebird[12]. InterBase представляет собой многопользовательскую реляционную СУБД, которая взаимодействует с клиентом, принимая от него запросы на языке SQL, и возвращая запрошенные данные [6].

# Глава 3. Создание базы знаний

# 3.1. Обзор предметной области

# Предприятия общественного питания – это организации, которые оказывают следующие услуги: реализация и изготовление кулинарной продукции; организация ее потребления и обслуживания клиентов[7]. Существуют множество видов таких организаций, но разрабатываемая в данной работе система применима только в одной из них: [8]

# 1. Ресторан – это предприятие общественного питания, которое включает в себя различный ассортимент блюд, в том числе и сложные блюда, а также заказные и фирменные;

# 2. Кафе – отличие от ресторана в ограниченном ассортименте продукции;

# 3. Пиццерия – может объединить в себе формат ресторана и кафе. Главное отличие, это широкий ассортимент пиццы.

# Существует множество различных классификаций предприятий общественного питания – это классификации в зависимости от времени функционирования, в зависимости от ассортимента выпускаемой продукции, в зависимости от обслуживаемого контингента и другие. В данной работе следует отметить классификацию, которая согласно Государственному стандарту РФ, классифицирует рестораны в зависимости от ассортимента блюд, качества обслуживания, интерьера и т. д. На основании этого рестораны подразделяются на три класса [9]:

# 1. Ресторан люкс, включает в себя бар и банкетные залы. Интерьер этого ресторана должен иметь высокое архитектурно-художественное оформление. Стиль дизайна должен соответствовать названию ресторана. Персонал одет в фирменную одежду единого образца, в вечернее время 3организуется музыкальные шоу. В залах имеются специальные места для танцев. Меню в свою очередь должно иметь, не менее половины ассортимента, заказных и фирменных блюд;

# 2. Ресторан первого класса имеет оригинальный интерьер, разнообразный ассортимент блюд и предоставляемых услуг. В вечернее время организуется музыкальная программа;

# 3. Ресторан второго класса предполагает гармоничный интерьер, выбор услуг и разнообразный ассортимент блюд. Местоположение должно нравиться людям, которые будут его посещать.

# Для того чтобы человек мог выбрать какой из ресторанов является наиболее привлекательным для него, можно использовать интеллектуальную систему выбора, которая имеет свою базу знаний.

# 3.2. Разработка семантической связи.

# Семантическая сеть – это один из способов представления знаний. Она имеет вид ориентированного графа, вершины которого это объекты предметной области, а стрелки это святи между ними [10].

# В сематической сети, которая описывает данную предметную область, можно выделить такие типы понятий как: сущность (абстрактный объект предметной области), экземпляр (конкретный представитель сущности) и свойство (характеристика сущностей). В соответствии с этой классификацией и проведенным анализом предметной области, были выделены и разгруппированы следующие понятия:

# 1. Сущность: местоположение, средний чек, тип ресторана, качество блюд, кухня, музыка, танцпол, ремонт, доставка и стоянка.

# 2. Экземпляры: Охотников, Стрижи, OldCafe, Asador, Трапезников, Париж, Alex, Мамбу, Европа, Хутарок, Amstel, Delta, Banana.

# 3. Свойство: местоположение ресторана в центре или в спальном район; цена среднего чека на человека до 1000 рублей, от 1500 рублей до 2500 рублей, от 2500 рублей; тип ресторана люкс, первого класса или второго класса; среднее или высокое качество блюд; кухня ресторана русская, украинская, китайская и европейская; живая или фоновая музыка; вместимость танцпола от 50 человек или до 50 человек; ремонт от 5ти лет или до 5ти лет; доставка присутствует или отсутствует; стоянка присутствует или отсутствует.

# На рисунке 2 показано графическое представление представленной

# семантической сети. В котором обозначены объекты предметной области(сущности, экземпляры и свойства) и

# выделены связи между ними.

# 

# Рисунок 2. Графическое представление семантической сети.

# 3.3. Заполнение базы знаний.

# Данный раздел предполагает внесение в базу всех собранных знаний и сведений, в результате проведения анализа по данной предметной области. Главными требованиями к хранящейся в базе знаний интеллектуальной системы информации, считаются:

# 1. Правила релевантности информации, которые получаются с помощью вывода базы знаний, то есть обозначение субъективного представления чего-либо в данны момент времени;

# 2. Достоверность информации – показатель качества имеющейся информации в базе данных.

# С помощью системы управления базами данных — Firebird мы заполняем базу знаний. С помощью утилиты IBExpert была создана и зарегистрирована база данных, на рисунке 3 показана её регистрационная информация.

# 

# Рисунок 3. База данных

# Дальше была создана таблица Rest в которой в дальнейшем будет хранится и изменятся база знаний интеллектуальной системы.

# Код программы , использовавшийся при создание таблицы Rest:

# create table Rest

# (id\_restint primary key,

# namevarchar (60),

# addressvarchar (100),

# areavarchar (60),

# pricevarchar (60),

# cuisinevarchar (60),

# musicvarchar (60),

# type\_restvarchar (60),

# qualityvarchar (60),

# dancevarchar (60),

# deliveryvarchar (60),

# parkingvarchar (60),

# repairvarchar (60)

# )

# На рисунке 4 представлена наша заполненная база знаний.

# Рисунок 4. База знаний

# Глава 4.Создание интеллектуального интерфейса

# InterBaseExpress использовалась для создания клиентского приложения. На рисунке 5 представлена модульная структура разрабатываемого приложения.

# Рисунок 5. Модульная структура.

# Она включает в себя объекты для связи приложения с базой данных, а так же для вывода базы знаний в разрабатываемом приложении. Рисунок 5 – Модульная структура разрабатываемой системы На рисунках 6 и 7 представлен разработанный интерфейс интеллектуальной системы выбора ресторана в г. Иркутске. На рисунке 6 представлена вкладка «Выбор», на ней находятся такие компоненты как DBGrid, RadioGroup и Button. Компонент DBGrid выводит на экран базу знаний. Компонент RadioGroup позволяет выбрать необходимые критерии поиска ресторана. Button – это кнопка, которая выполняет заданные ей функции.

# Рисунок 6. Вкладка «Выбор».

# На рисунке 6 представлена вкладка «Дополнение», в ней содержатся следующие компоненты: DBGrid, Button, Edit, RadioGroup. Это схожие с предыдущей вкладкой компоненты, за исключением компонента Edit – это поле для ввода или вывода информации и компонента RadioGroup – компонент, предназначенный для объединения в группу нескольких кнопок или других разных компонентов.

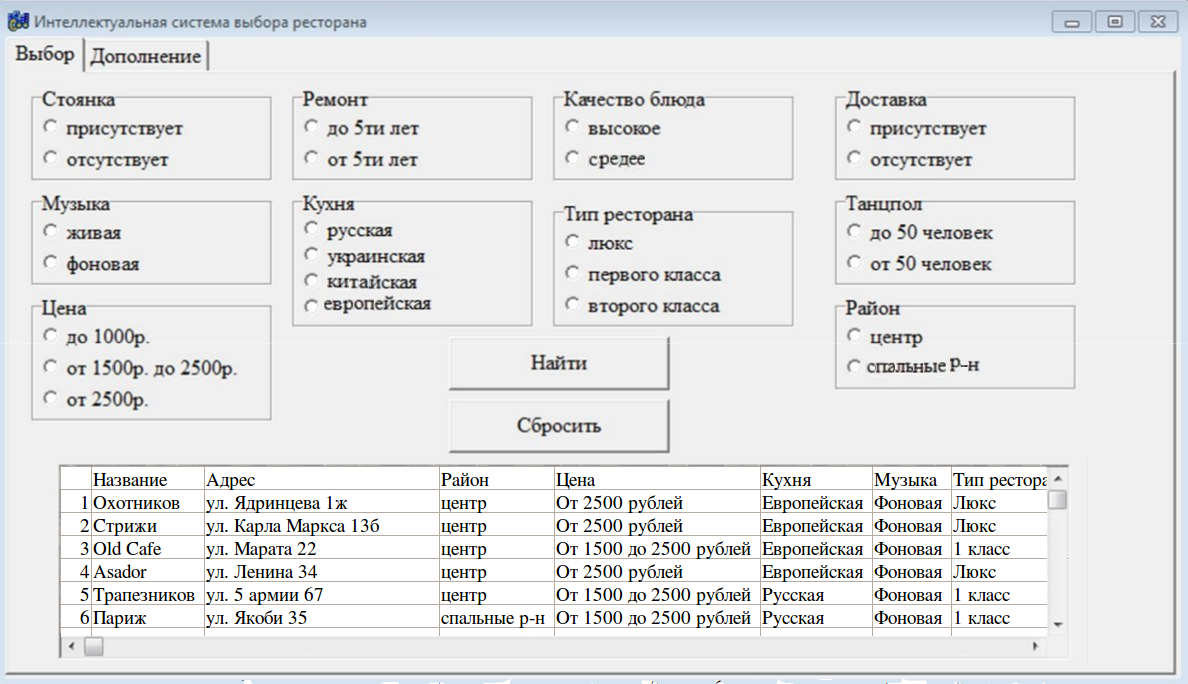
# Рисунок 7. Вкладка «Дополнение».

# Программный код разработки интеллектуального интерфейса представлен в приложении. В данной главе была рассмотрена разработка

# интеллектуального интерфейса выбора ресторана в г. Иркутске.

Глава 5. Тестирование интеллектуальной системы.

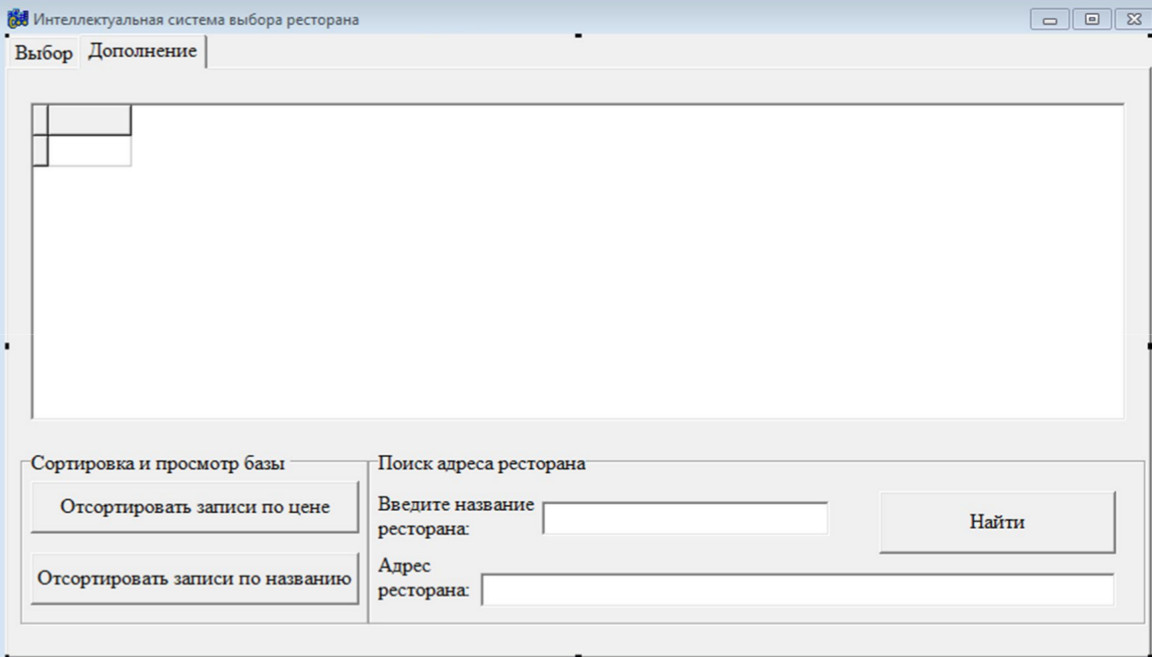
На вкладке «Выбор» мы видим базу знаний, а так же находящиеся компоненты выбора, для поиска подходящего ресторана по заданным критериям, стоит отметить необходимые критерии параметров в любом количестве, то есть можно выбрать все критерии, один или несколько. Затем нажать на кнопку «Найти» и в окне вывода появятся записи соответствующие запросу. Кнопка «Сбросить» предназначена для сброса выбранных критериев и сброса записей предшествующего вывода (см. рисунок 8).

Рисунок 8. Интеллектуальная система выбора ресторана, вкладка «Выбор»

Для перехода на вкладку «Дополнение» необходимо нажать на нее. На

этой вкладке размещается кнопки позволяющие просматривать базу знаний

и сортировать ее записи по необходимости. Нажав на кнопку «Отсортировать» записи по цене, записи базы знаний будут отсортированы в порядке возрастания цены. Кнопка «Отсортировать» записи по имени, позволяет отсортировать записи базы знаний по названию ресторана. На данной вкладке так же есть возможность найти адрес необходимого ресторана. В первом поле нужно ввести название ресторана с большой буквы, нажать на кнопку «Найти» и во втором поле появится адрес заданного ресторана (рисунок 9).

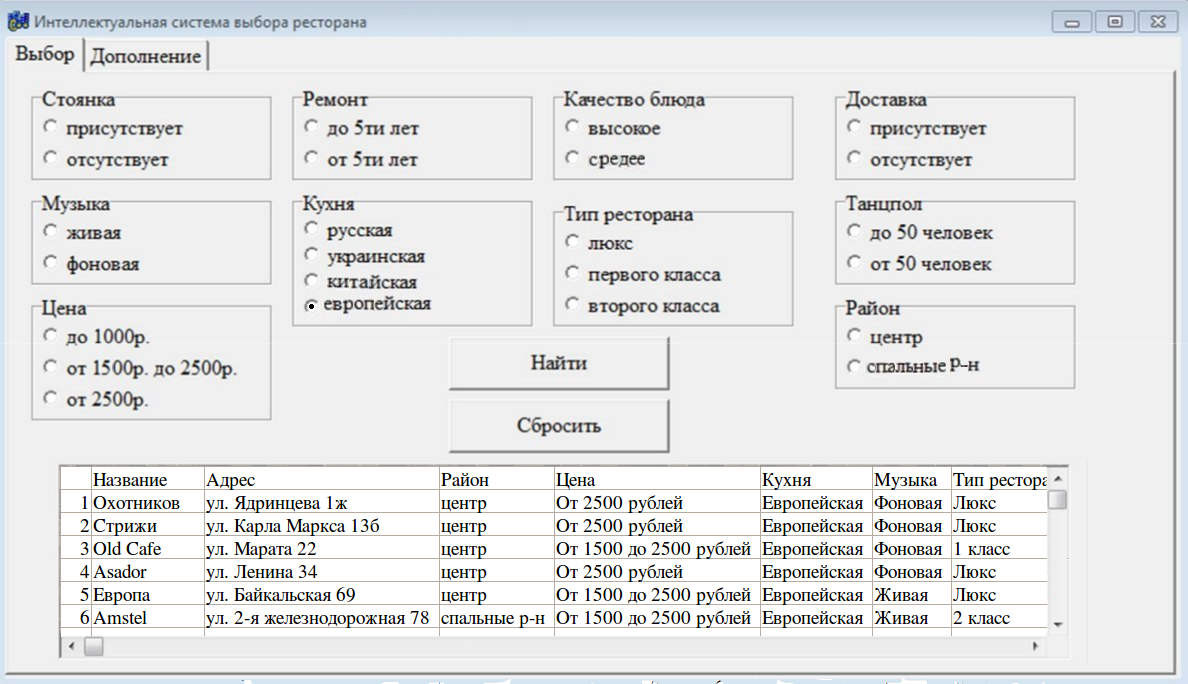
Рисунок 9. Интеллектуальная система выбора ресторана, вкладка «Дополнение»

5.1 Апробация.

Апробация–это проверка на практическом эксперименте работоспособности разработанной системы [11].

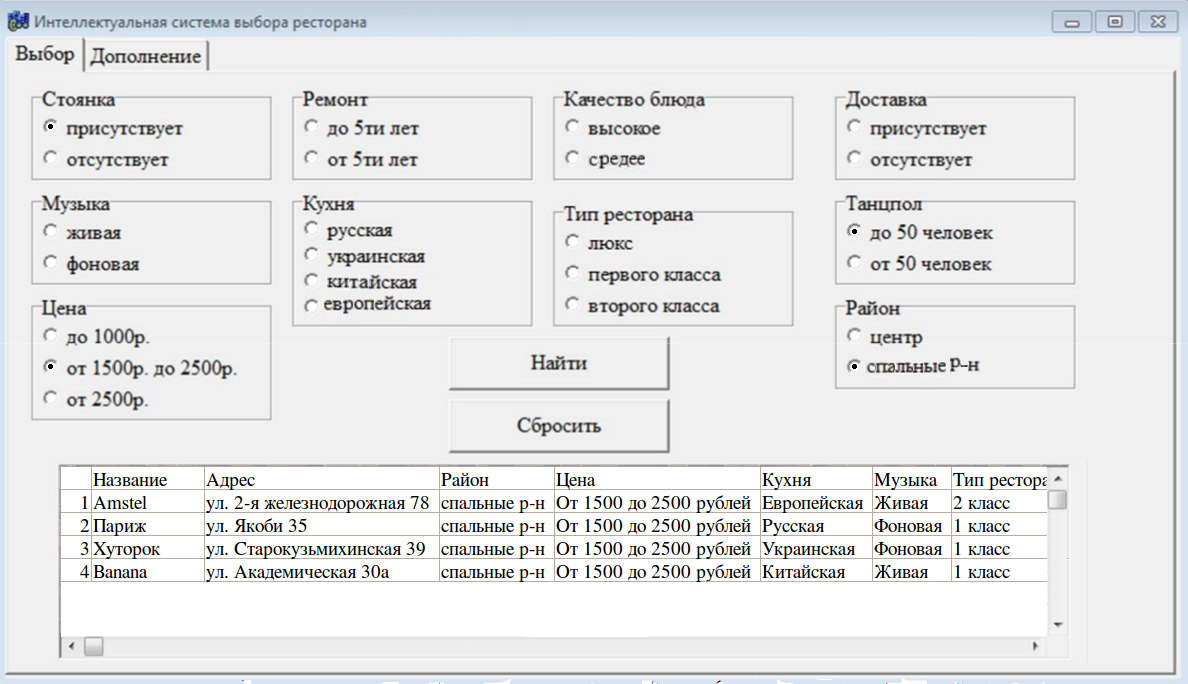
На рисунке 10, показан результат работы интеллектуальной системы

выбора ресторана при выборе одного наиболее значимого критерия: Европейская кухня.

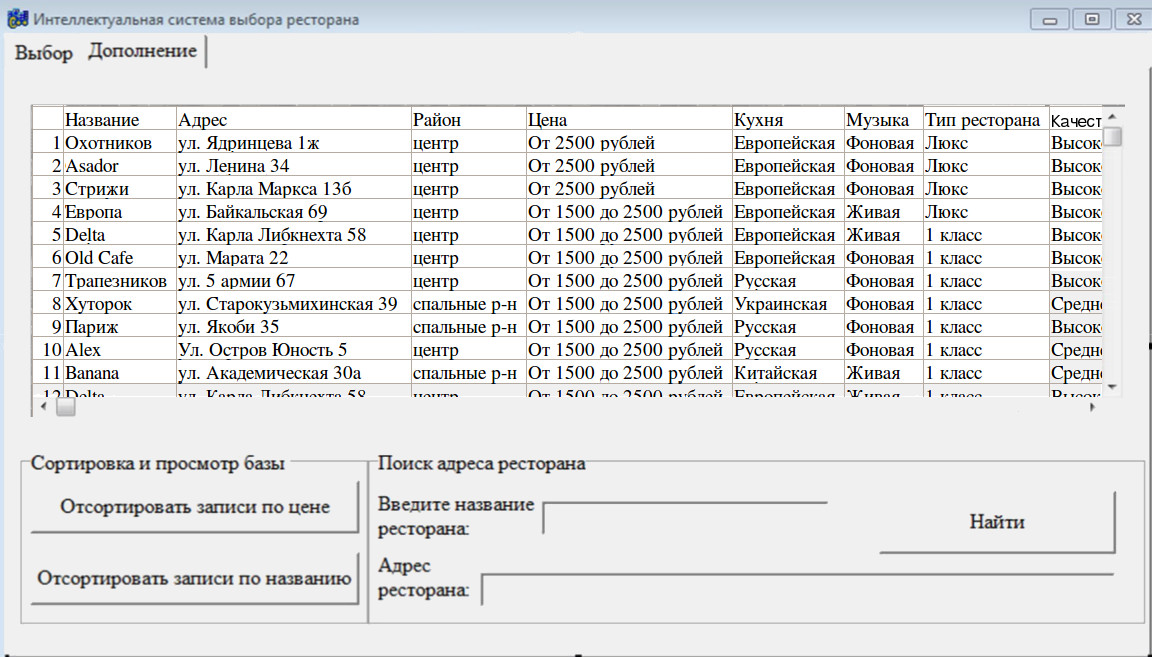
Рисунок 10. Результат по выбору критерия «европейская кухня»

На рисунке 11 показан результат работы интеллектуальной системы

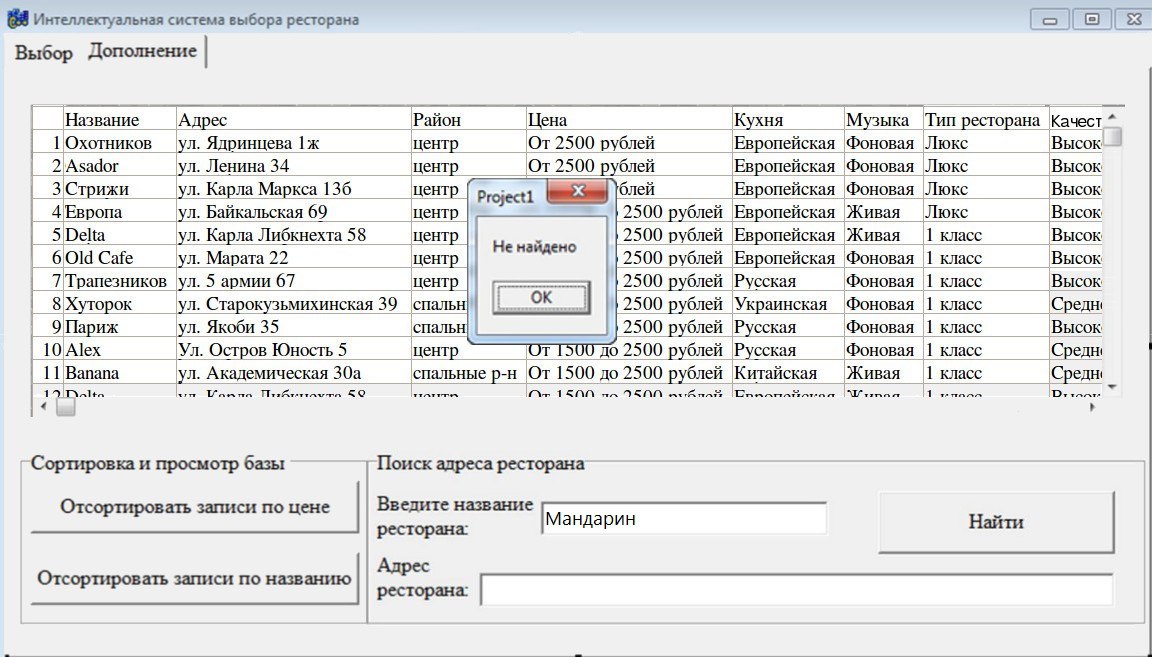
выбора ресторана при выборе нескольких критериев: стоянка у ресторана присутствует, танцпол до 50 человек, местоположение в спальных районах и средний чек от 1500 до 2500 рублей.

Рисунок 11. Выбор по нескольким критериям.

На рисунке 12 представлена сортировка записей по средней цене чека на одного человека в порядке убывания.

Рисунок 12. Сортировка по цене.

На рисунке 13 показан результат работы интеллектуальной системы при поиске ресторана «Мандарин». Так как этого ресторана нет в базе знаний, то системы выдает ошибку с сообщение «Не найдено».

Рисунок 13. Вывод ошибки.

Заключение

Входе написания выпускной квалификационной работы работы были получены практические навыки по проектированию базы знаний и разработки интеллектуальной системы выбора ресторана в г. Иркутске, она позволяет наиболее эффективно и быстро определять предпочитаемый ресторан, а так же найти и вывести адрес этого ресторана. Созданная интеллектуальная система удовлетворяет всем поставленным задачам: содержит в себе разработанную базу знаний на основе построенной семантической сети и позволяет производить выбор ресторана. Воспользовавшись этим приложением наименьшей затратой сил и времени пользователи могут с выбрать ресторан с наиболее предпочитаемой кухней, ценой, музыкой и другими критериями, а так же определить его адрес.

Список литературы

1. Гущин А.Н., Радченко И.А. Экспертные системы: учебное пособие, Балт. гос. техн. ун-т. — СПб., 2007

2.Экспертная система MYCIN  [Электронный ресурс]

http://ru.wikipedia.org/wiki/MYCIN

3.Экспертная система PROSPECTOR  [Электронный ресурс] http://penguin.photon.ru/doc/ai.shtml

4.Экспертная система Dendral  [Электронный ресурс] http://en.wikipedia.org/wiki/Dendral

5. Михелёв В.М. «Базы данных и СУБД», Белгород, 2007 г.

6. Пахомов Б.И. «C и Borland C++ Builder для студента» БХВ –

Петербург: Спб., 2012 г.

7. Предприятия общественного питания  [Электронный ресурс]

https://ru.wikipedia.org/wiki/Предприятие\_общественного\_питания

8. Радченко Л.А. «Организация производства на предприятиях

общественного питания»,Феник: Ростов, 2006 г.

9.Затуливетров А. «РЕСТОРАН ЗА 180 ДНЕЙ. ПРАКТИЧЕСКОЕ

ПОСОБИЕ», Ресторанные ведомости, 2012 г.,

10. Рассел С. «Искусственный интеллект. Современный подход», Вильямс:М., 2015г

11. Psilist.net  [Электронный ресурс]

<http://psylist.net/pedagogika/aprobac.htm>

12. Архангельский А.Я. «Программирование в C++ Builder 6», ООО

«Бином – Пресс»: М., 2013 г

Приложение

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include <iostream.h>

#include "Unit1.h"

#include "Unit2.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm1 \*Form1;

void \_\_fastcall TForm1::Button2Click(TObject \*Sender)

{

RadioGroup1->ItemIndex=-1;

RadioGroup2->ItemIndex=-1;

RadioGroup3->ItemIndex=-1;

RadioGroup4->ItemIndex=-1;

RadioGroup5->ItemIndex=-1;

RadioGroup6->ItemIndex=-1;

RadioGroup7->ItemIndex=-1;

RadioGroup8->ItemIndex=-1;

RadioGroup9->ItemIndex=-1;

RadioGroup10->ItemIndex=-1;

DataModule2->IBQuery1->Active=false;

DataModule2->IBQuery1->SQL->Clear();

DataModule2->IBQuery1->SQL->Add("select \* from rest");

DataModule2->IBQuery1->Active=true;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button1Click(TObject \*Sender)

{

AnsiString a="";

AnsiString b="";

AnsiString c="";

AnsiString d="";

AnsiString e="";

AnsiString f="";

AnsiString g="";

AnsiString h="";

AnsiString i="";

AnsiString j="";

if (RadioGroup1->ItemIndex!=-1) {

if (RadioGroup1->ItemIndex==0)

{a="присутствует";}

else if (RadioGroup1->ItemIndex==1) {

a="отсутствует"; }

}

if (RadioGroup2->ItemIndex!=-1) {

if (RadioGroup2->ItemIndex==0)

{b="до 5тилет";}

else if (RadioGroup2->ItemIndex==1) {

b="от 5 тилет"; }

}

if (RadioGroup3->ItemIndex!=-1) {

if (RadioGroup3->ItemIndex==0)

{c="высокое";}

else if (RadioGroup3->ItemIndex==1) {

c="среднее"; }

}

if (RadioGroup4->ItemIndex!=-1) {

if (RadioGroup4->ItemIndex==0){

d="присутствует";}

else if (RadioGroup4->ItemIndex==1) {

d="отсутствует"; }

}

if (RadioGroup5->ItemIndex!=-1) {

if (RadioGroup5->ItemIndex==0){

e="живая";}

else if (RadioGroup5->ItemIndex==1) {

e="фоновая"; }

}

if (RadioGroup6->ItemIndex!=-1) {

if (RadioGroup6->ItemIndex==0){

f="русская";}

else if (RadioGroup6->ItemIndex==1) {

f="украинская"; }

else if (RadioGroup6->ItemIndex==2) {

f="китайская"; }

else if (RadioGroup6->ItemIndex==3) {

f="европейская"; }

}

if (RadioGroup7->ItemIndex!=-1) {

if (RadioGroup7->ItemIndex==0){

g="люкс";}

else if (RadioGroup7->ItemIndex==1) {

g="первогокласса"; }

else if (RadioGroup7->ItemIndex==2) {

g="второгокласса"; }

}

if (RadioGroup8->ItemIndex!=-1) {

if (RadioGroup8->ItemIndex==0){

h="от 50 человек";}

else if (RadioGroup8->ItemIndex==1) {

h="до 50 человек"; }

}

if (RadioGroup9->ItemIndex!=-1) {

if (RadioGroup9->ItemIndex==0){

i="до 1000р.";}

else if (RadioGroup9->ItemIndex==1) {

i="от 1500р. до 2500р."; }

else if (RadioGroup9->ItemIndex==2) {

i="от 2500."; }

}

if (RadioGroup10->ItemIndex!=-1) {

if (RadioGroup10->ItemIndex==0){

j="центр";}

else if (RadioGroup10->ItemIndex==1) {

j="спальныеР-ны"; }

}

DataModule2->IBQuery1->Active=false;

DataModule2->IBQuery1->SQL->Clear();

DataModule2->IBQuery1->SQL->Add("SELECT \* from rest WHERE");

if (a!="") {

DataModule2→IBQuery1→SQL→Add("parking LIKE '"+a+"'");

}

if (b!=""){

if (a!=""){

DataModule2->IBQuery1->SQL->Add("and repeir LIKE '"+b+"'"); }

else {DataModule2->IBQuery1->SQL->Add(" repeir LIKE '"+b+"'");}

}

if (c!=""){

if ((a!="") || (b!="")){

DataModule2->IBQuery1->SQL->Add("and quality LIKE '"+c+"'"); }

else {DataModule2→IBQuery1→SQL→Add(" quality LIKE '"+c+"'");}

}

if (d!=""){

if ((a!="") || (b!="") || (c!="")){

DataModule2->IBQuery1->SQL->Add("and delivery LIKE '"+d+"'"); }

else {DataModule2->IBQuery1->SQL->Add("delivery LIKE '"+d+"'");}

}

if (e!=""){

if ((a!="") || (b!="") || (c!="") || (d!="")){

DataModule2->IBQuery1->SQL->Add("and music LIKE '"+e+"'"); }

else {DataModule2->IBQuery1->SQL->Add("music LIKE '"+e+"'");}

}

if (f!=""){

if ((a!="") || (b!="") || (c!="") || (d!="") || (e!="")){

DataModule2->IBQuery1->SQL->Add("and cuisine LIKE '"+f+"'"); }

else {DataModule2->IBQuery1->SQL->Add("cuisine LIKE '"+f+"'");}

}

if (g!=""){

if ((a!="") || (b!="") || (c!="") || (d!="") || (e!="") || (f!="")){

DataModule2->IBQuery1->SQL->Add("and type\_rest LIKE '"+g+"'"); }

else {DataModule2->IBQuery1->SQL->Add("type\_rest LIKE '"+g+"'");}

}

if (h!=""){

if ((a!="") || (b!="") || (c!="") || (d!="") || (e!="") || (f!="") || (g!="")){

DataModule2->IBQuery1->SQL->Add("and dance LIKE '"+h+"'"); }

else {DataModule2->IBQuery1->SQL->Add("dance LIKE '"+h+"'");}

}

if (i!=""){

if ((a!="") || (b!="") || (c!="") || (d!="") || (e!="") || (f!="") || (g!="") || (h!="")){

DataModule2->IBQuery1->SQL->Add("and price LIKE '"+i+"'"); }

else {DataModule2->IBQuery1->SQL->Add("price LIKE '"+i+"'");}

}

if (j!=""){

if ((a!="") || (b!="") || (c!="") || (d!="") || (e!="") || (f!="") || (g!="") || (h!="") || (i!="")){

DataModule2->IBQuery1->SQL->Add("and area LIKE '"+j+"'"); }

else {DataModule2->IBQuery1->SQL->Add("area LIKE '"+j+"'");}

}

DataModule2->IBQuery1->Active=true;

DBGrid1→Columns→Items[0]→Title→Caption=" ";

DBGrid1->Columns->Items[0]->Width=20;

DBGrid1->Columns->Items[1]->Title->Caption="Название";

DBGrid1->Columns->Items[1]->Width=85;

DBGrid1->Columns->Items[2]->Title->Caption="Адрес";

DBGrid1->Columns->Items[2]->Width=225;

DBGrid1->Columns->Items[3]->Title->Caption="Район";

DBGrid1->Columns->Items[3]->Width=60;

DBGrid1->Columns->Items[4]->Title->Caption="Цена";

DBGrid1->Columns->Items[4]->Width=142;

DBGrid1->Columns->Items[5]->Title->Caption="Кухня";

DBGrid1->Columns->Items[5]->Width=90;

DBGrid1->Columns->Items[6]->Title->Caption="Музыка";

DBGrid1->Columns->Items[6]->Width=60;

DBGrid1->Columns->Items[7]->Title->Caption="Типресторана";

DBGrid1->Columns->Items[7]->Width=105;

DBGrid1->Columns->Items[8]->Title->Caption="Качествоблюда";

DBGrid1->Columns->Items[8]->Width=86;

DBGrid1->Columns->Items[9]->Title->Caption="Танцпол";

DBGrid1->Columns->Items[9]->Width=96;

DBGrid1->Columns->Items[10]->Title->Caption="Доставка";

DBGrid1->Columns->Items[10]->Width=93;

DBGrid1->Columns->Items[11]->Title->Caption="Стоянка";

DBGrid1->Columns->Items[11]->Width=93;

DBGrid1->Columns->Items[12]->Title->Caption="Ремонт";

DBGrid1->Columns->Items[12]->Width=75;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button3Click(TObject \*Sender)

{

DataModule2->IBTable1->Active=false;

DataModule2->IBTable1->IndexName="ind\_price";

DataModule2->IBTable1->Active=true;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button4Click(TObject \*Sender)

{

Variant v;

v=DataModule2->IBTable1->Lookup("name", Edit1->Text, "address");

if(!(VarType(v)==varNull))

{Edit2->Text=v;}

else

ShowMessage("ненайдено");

}

//--------------------------------------------------------------------------

void \_\_fastcall TForm1::Button5Click(TObject \*Sender)

{

DataModule2->IBTable1->Active=false;

DataModule2->IBTable1->IndexName="ind\_name";

DataModule2->IBTable1->Active=true;

}

//---------------------------------------------------------------------------