МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО

ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное

образовательное учреждение высшего образования

«**Вятский государственный университет**»

**(ФГБОУ ВО «ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Допущен к защите

Руководитель проекта

\_\_\_\_\_\_\_\_/Исупов К.С./

(подпись) (Ф.И.О.)

«\_\_\_» \_\_\_\_\_\_\_\_\_202\_г.

Разработка экспертной системы

Пояснительная записка курсового проекта по дисциплине

«Комплекс знаний бакалавра в области программного и аппаратного обеспечения вычислительно техники»

ТПЖА 090301.786 ПЗ

Разработал студент группы ИВТб-2301-04-00 \_\_\_\_\_\_\_\_\_\_\_\_/Шуткин Д.А./

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_/Исупов К.С./

Консультант \_\_\_\_\_\_\_\_\_\_/Коржавина А.С./

Работа защищена с оценкой «\_\_\_\_\_\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оценка) (дата)

Члены комиссии \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ \_\_\_\_\_\_\_\_\_\_\_\_/

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ \_\_\_\_\_\_\_\_\_\_\_\_/

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ \_\_\_\_\_\_\_\_\_\_\_\_/

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ \_\_\_\_\_\_\_\_\_\_\_\_/

Киров 2021

Изображение выглядит как текст, стол

Автоматически созданное описание

Реферат

Шуткин Д.А. Разработка экспертной системы. ТПЖА.090301.786 ПЗ: Курс. проект/ВятГУ, каф. ЭВМ; рук. Исупов К. С. – Киров, 2021. ПЗ 44 c., 6 рис., 2 прил.

Объект исследования и разработки курсового проекта – экспертная система.

Предмет курсового проекта – экспертная система, принимающая на вход базу знаний и реализующая ее в виде опроса.

Целью курсового проекта является изучение экспертных систем, их назначения и перспектив развития.

Результатом выполнения курсового проекта является демонстрационная экспертная система с прилегающей базой знаний, которая позволяет определить характер возможного недуга.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *№ строки* | *Формат* | *Обозначение* | | | | *Наименование* | | | | *Кол-во листов* | | *№ экз.* | *Примеч.* | |
| *1* |  |  | | | | *Документация общая* | | | |  | |  |  | |
| *2* |  |  | | | | *Вновь разработанная* | | | |  | |  |  | |
| *3* |  |  | | | |  | | | |  | |  |  | |
| *4* | *А4* | *ТПЖА 090301.786 ПЗ* | | | | *Пояснительная записка* | | | | *44* | |  |  | |
| *5* | *А2* | *ТПЖА 090301.786 ПЛ1* | | | | *Схемы алгоритмов* | | | | *1* | |  |  | |
| *6* | *А2* | *ТПЖА 090301.786 ПЛ2* | | | | *Модель экспертной системы* | | | | *1* | |  |  | |
| *7* | *А2* | *ТПЖА 090301.786 ПЛ3* | | | | *Экранные формы* | | | | *1* | |  |  | |
| *8* | *А2* | *ТПЖА 090301.786 ПЛ4* | | | | *Структура приложения* | | | | *1* | |  |  | |
| *9* |  |  | | | |  | | | |  | |  |  | |
| *10* |  |  | | | |  | | | |  | |  |  | |
| *11* |  |  | | | |  | | | |  | |  |  | |
| *12* |  |  | | | |  | | | |  | |  |  | |
| *13* |  |  | | | |  | | | |  | |  |  | |
| *14* |  |  | | | |  | | | |  | |  |  | |
| *15* |  |  | | | |  | | | |  | |  |  | |
| *16* |  |  | | | |  | | | |  | |  |  | |
| *17* |  |  | | | |  | | | |  | |  |  | |
| *18* |  |  | | | |  | | | |  | |  |  | |
| *19* |  |  | | | |  | | | |  | |  |  | |
| *20* |  |  | | | |  | | | |  | |  |  | |
| *21* |  |  | | | |  | | | |  | |  |  | |
| *22* |  |  | | | |  | | | |  | |  |  | |
| *23* |  |  | | | |  | | | |  | |  |  | |
| *24* |  |  | | | |  | | | |  | |  |  | |
| *25* |  |  | | | |  | | | |  | |  |  | |
| *26* |  |  | | | |  | | | |  | |  |  | |
|  |  |  |  |  | *ТПЖА 090301.786 ПЗ* | | | | | | | | | |
|  |  |  |  |  |
| *Изм* | *Лист* | *№ документа* | *Подпись* | *Дата* |
| *Разраб.* | | *Шуткин Д.А.* |  |  | *Разработка экспертной системы* | | *Литер* | | | | *Лист* | | | *Листов* |
| *Руковод.* | | *Исупов К.С.* |  |  |  |  |  | | *4* | | | *44* |
| *Т.контр* | |  |  |  | *Кафедра ЭВМ*  *ИВТб-2301-04-00* | | | | | | | |
| *Н.контр* | |  |  |  |
| *Утверд.* | |  |  |  |

Содержание

[Содержание 5](#_Toc91179943)

[Введение 6](#_Toc91179944)

[1. Анализ предметной области 7](#_Toc91179945)

[1.1. Основы экспертных систем 7](#_Toc91179946)

[1.2. Актуальность экспертных систем 14](#_Toc91179947)

[1.3. Проблемы и преимущества экспертных систем 16](#_Toc91179948)

[1.4. Методология разработки экспертных систем 17](#_Toc91179949)

[1.5. Анализ существующих аналогов 22](#_Toc91179950)

[1.6. Постановка задачи 23](#_Toc91179951)

[2. Разработка структуры приложения 24](#_Toc91179952)

[2.1. Разработка модели экспертной системы 25](#_Toc91179953)

[2.2. Разработка алгоритмов 26](#_Toc91179954)

[3. Реализация и отладка программы 27](#_Toc91179955)

[3.1. Разработка интерфейса пользователя 27](#_Toc91179956)

[3.2.Используемые при разработке средства 28](#_Toc91179957)

[3.3. Проверка работоспособности 30](#_Toc91179958)

[Заключение 34](#_Toc91179959)

[Приложение А 35](#_Toc91179960)

[Приложение Б 36](#_Toc91179961)

[Приложение В 43](#_Toc91179962)

## Введение

Область искусственного интеллекта имеет более чем сорокалетнюю историю развития. С самого начала в ней рассматривался ряд весьма сложных задач, которые, наряду с другими, до сих пор являются предметом исследований: автоматические доказательства теорем, машинный перевод (автоматический перевод с одного естественного языка на другой), распознавание изображений и анализ сцен, планирование действий роботов, алгоритмы и стратегии игр.

Экспертные системы возникли как значительный практический результат в развитии и применении методов искусственного интеллекта [1] - совокупности научных дисциплин, изучающих методы решения задач интеллектуального (творческого) характера с использованием ЭВМ.

Цель курсового проекта - изучение экспертных систем, их назначения и перспектив развития, получение практических навыков реализации экспертных систем.

Для достижения поставленной в данном проекте цели необходимо решить ряд задач, таких как:

- изучение теоретических основ экспертных систем, для чего определить понятие "экспертная система", а также описать процесс создания и эксплуатации экспертных систем;

- исследование проблем и преимуществ экспертных систем, для чего изучить применение экспертных систем, а также рассмотреть перспективы использования экспертных систем;

- создание демонстрационной экспертной системы, позволяющей выдавать прогнозы в какой-либо сфере на основе заранее заданной базы знаний, исходя из ответов пользователя.

## 1. Анализ предметной области

В данном разделе описаны основные моменты, касающиеся исследуемой области, анализ существующих аналогов и постановка задачи.

## 1.1. Основы экспертных систем

Экспертная система [2] - это набор программ или программное обеспечение, которое выполняет функции эксперта при решении какой-либо задачи в области его компетенции. Экспертная система, как и эксперт-человек, в процессе своей работы оперирует со знаниями. Знания о предметной области, необходимые для работы экспертной системы, определенным образом формализованы и представлены в памяти ЭВМ в виде базы знаний, которая может изменяться и дополняться в процессе развития системы.

Экспертные системы выдают советы, проводят анализ, выполняют классификацию, дают консультации и ставят диагноз. Они ориентированы на решение задач, обычно требующих проведения экспертизы человеком-специалистом [3]. В отличие от машинных программ, использующий процедурный анализ, экспертные системы решают задачи в узкой предметной области, то есть конкретной области экспертизы, на основе дедуктивных рассуждений. Такие системы часто оказываются способными найти решение задач, которые неструктурированны и плохо определены. Они справляются с отсутствием структурированности путем привлечения эвристик, то есть некоторых заранее определённых правил, что может быть полезным в тех системах, когда недостаток необходимых знаний или времени исключает возможность проведения полного анализа.

Области применения систем, основанных на знаниях, весьма разнообразны: бизнес, производство, военные приложения, медицина, социология, геология, космос, сельское хозяйство, управление, юриспруденция и другие.

Системы, основанные на знаниях (СОЗ) [4] — это системы программного обеспечения, основными структурными элементами которых являются база знаний и механизм логических выводов. Среди СОЗ можно выделить:

- интеллектуальные информационно-поисковые системы;

- экспертные системы.

Наиболее известным практическим примером СОЗ могут служить экспертные системы, способные диагностировать заболевания, оценивать потенциальные месторождения полезных ископаемых, осуществлять обработку естественного языка, распознавание речи и изображений и т.д. Экспертные системы являются первым шагом в практической реализации исследований в области искусственного интеллекта.

Базовая структура экспертной системы отображена на рисунке 1.

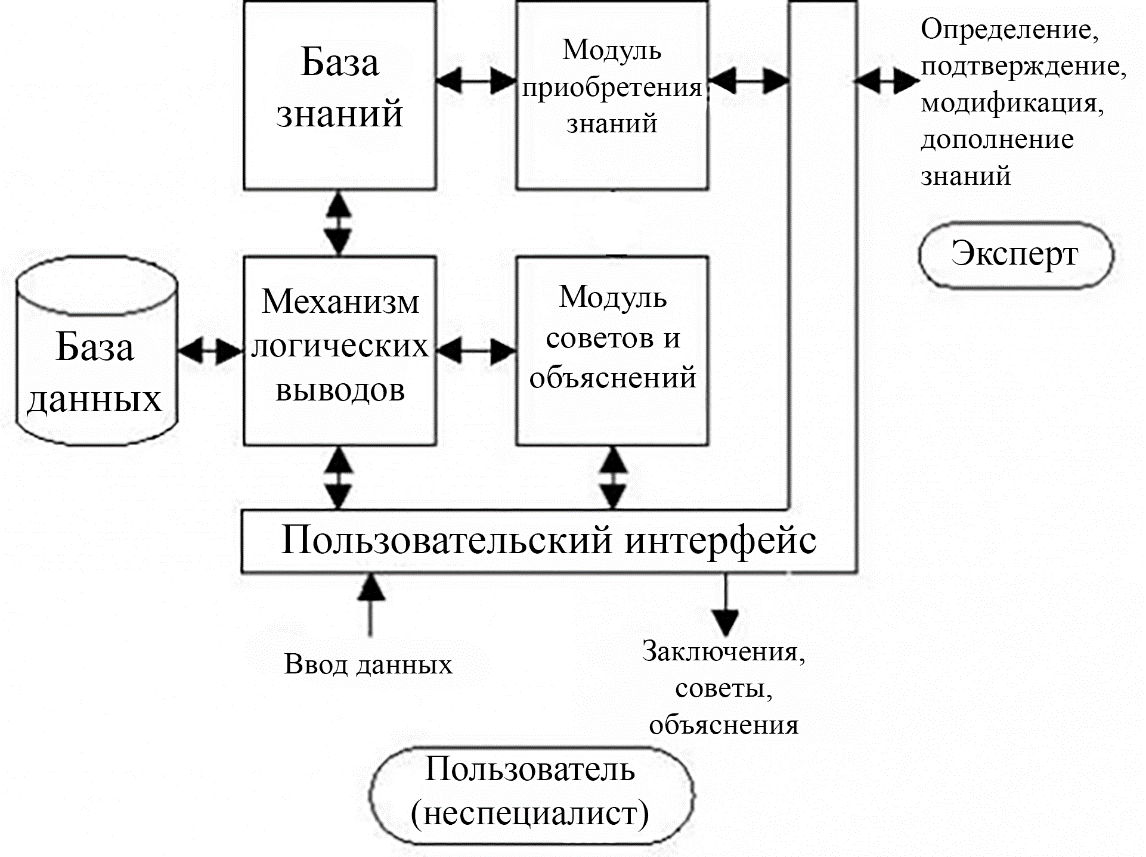


Рисунок 1 – Базовая структура экспертной системы

Структурные элементы, составляющие экспертную систему, выполняют следующие функции:

- база знаний реализует функции представления знаний в конкретной предметной области и управление ими;

- механизм логических выводов выполняет логические выводы на основании знаний, имеющихся в базе знаний;

- пользовательский интерфейс необходим для правильной передачи ответов пользователю, обеспечивая удобство для конечного пользователя;

- модуль приобретения знаний необходим для получения знаний от эксперта, поддержки базы знаний и дополнения ее при необходимости;

- модуль ответов и объяснений формирует заключение экспертной системы и представляет различные комментарии, прилагаемые к заключению, а также объясняет мотивы заключения.

Чтобы проводить экспертизу, компьютерная программа должна быть способной решать задачи посредством логического вывода и получать при этом достаточно надежные результаты [5]. Программа должна иметь доступ к системе фактов, называемой базой знаний.

Программа также должна во время консультации выводить заключения из информации, имеющейся в базе знаний. Некоторые экспертные системы могут также использовать новую информацию, добавляемую во время консультации. Экспертную систему, таким образом, можно представлять состоящей из трех частей [6]:

- база знаний (БЗ);

- механизм вывода (МВ);

- система пользовательского интерфейса (СПИ).

База знаний - центральная часть экспертной системы [7]. Она содержит правила, описывающие отношения или явления, методы и знания для решения задач из области применения системы. Можно представлять базу знаний состоящей из фактических знаний и знаний, которые используются для вывода других знаний. Утверждение "Великая Отечественная Война была с 1941 по 1945 год" - пример фактического знания. "Если у вас болит голова, то примите две таблетки цитрамона" - пример знания для вывода. Сама база знаний обычно располагается на постоянном запоминающем устройстве.

Механизм вывода содержит принципы и правила работы [7]. Механизм вывода "знает", как оперировать базой знаний так, чтобы можно было получать разумно согласующиеся заключения, то есть выводы, из информации, находящейся в ней.

Когда экспертной системе задается вопрос, механизм вывода выбирает способ применения правил базы знаний для решения задачи, поставленной в вопросе [8]. Фактически, механизм вывода запускает экспертную систему в работу, определяя какие правила нужно вызвать и организуя к ним доступ в базу знаний. Механизм вывода выполняет правила, определяет, когда найдено приемлемое решение и передает результаты программе интерфейса с пользователем.

Когда вопрос должен быть предварительно обработан, то доступ к базе знаний осуществляется через интерфейс с пользователем. Интерфейс - это часть экспертной системы, которая взаимодействует с пользователем [9].

Система интерфейса с пользователем принимает информацию от пользователя и передает ему информацию. Простыми словами, система интерфейса должна убедиться, что после того, как пользователь описал задачу, вся необходимая информация получена. Интерфейс, основываясь на виде и природе информации, введенной пользователем, передает необходимую информацию механизму вывода. Когда механизм вывода возвращает знания, выведенные из базы знаний, интерфейс передает их обратно пользователю в удобной форме. Интерфейс с пользователем и механизм вывода могут рассматриваться как "приложение" к базе знаний. Они вместе составляют оболочку экспертной системы. Для базы знаний, которая содержит обширную и разнообразную информацию, могут быть разработаны и реализованы несколько разных оболочек.

Хорошо разработанные оболочки экспертных систем обычно содержат механизм для добавления и обновления информации в базе знаний [10].

Как видим, экспертная система состоит из трех основных частей. Взаимосвязь между частями может быть сложной, зависящей от природы и организации знаний, а также от методов и целей вывода.

Представление знаний - это множество соглашений по синтаксису и семантике, согласно которым описываются объекты. Хорошее правило при проектировании представления знаний - это организация знаний в такой форме, которая позволяет легко осуществлять доступ с помощью естественных и простых механизмов [2].

Знания могут быть представлены несколькими способами, называемыми моделями представления знаний.

1. Логические модели представления знаний - реализуются средствами логики предикатов [5]. Предикатом называется функция, принимающая только два значения — истина и ложь — и предназначенная для выражения свойств объектов или связей между ними. Выражение, в котором утверждается или отрицается наличие каких-либо свойств у объекта, называется высказыванием. Наиболее простым языком логики является исчисление высказываний, в котором отсутствуют переменные. Любому высказыванию можно приписать значение истинно или ложно. Отдельные высказывания могут соединяться связками И, ИЛИ, НЕ, которые называются булевыми операторами.

2. Продукционные модели, или модели, основанные на правилах, позволяют представить знания в виде предложений типа: если (условие), то (действие) [6]. Под условием понимается некоторое предложение-образец, по которому осуществляется поиск в базе знаний, а под действием — действия, выполняемые при успешном исходе поиска (они могут быть промежуточными, выступающими далее, как условия, и терминальными или целевыми, завершающими работу системы).

3. Семантические сети, которые представляют из себя ориентированный граф, вершины которого — понятия, а дуги — отношения между ними[9]. Семантическая сеть описывает знания в виде сетевых структур. Например, «программист сел за компьютер и отладил программу». Объектами являются: программист (А1), компьютер (А2), программа (А3). Объекты связаны отношениями: сел за компьютер (р1), отладил (р2), загружена в компьютер программа (р3).

4. Фреймовые модели. Фреймовые модели представляют собой систематизированную в виде единой теории технологическую модель памяти человека и его сознания [10]. Под фреймом понимают минимальные структуры информации, необходимые для представления класса объектов, явлений или процессов. Фрейм можно представить в виде сети, состоящей из вершин и дуг (отношений), в которых нижние уровни фрейма заканчиваются слотами (переменными), которые заполняются конкретной информацией при вызове фрейма. Значением слота может быть любая информация: текст, числа, математические соотношения, программы, ссылки на другие фреймы. На заполнение слотов могут быть наложены ограничения, например, цена не может быть отрицательной.

Экспертные системы часто создаются проектировщиками экспертных систем, которые работают с человеком-экспертом, чтобы закодировать знания эксперта в базе знаний [4].

Первый способ - это классификация и помещение фактов и чисел (фрагментов фактического знания) в правила Турбо-Пролога [8].

Это представление подходит для использования в экспертных системах, базирующихся на правилах.

Другой способ - это организация фактов и числовой информации в утверждениях, которые образуют базу знаний на утверждениях.

Представление знаний в утверждениях подходит для использования в экспертных системах, базирующихся на логике.

Система пользовательского интерфейса обеспечивает взаимодействие между экспертной системой и пользователем. Это взаимодействие обычно включает несколько функций [1]:

- обработка данных, полученных с клавиатуры, и высвечивание вводимых и выводимых данных на экране;

- поддержка диалога между пользователем и системой;

- распознавание ситуации непонимания между пользователем и системой;

- обеспечение "дружественности" по отношению к пользователю.

Система интерфейса с пользователем должна эффективно обрабатывать ввод и вывод. Для этого необходимо обрабатывать вводимые и выводимые данные быстро, в ясной и выразительной форме. Необходимо также включить возможность работы с дополнительными средствами такими, как печатающие устройства, магнитные диски и дополнительные файлы данных.

Кроме того, система интерфейса должна поддерживать соответствующий диалог между пользователем и системой [3]. Диалог - это общая форма консультации с экспертной системой.

Консультация должна завершаться ясным утверждением, выдаваемым системой, и объяснением последовательности вывода, приведшей к этому утверждению.

Система пользовательского интерфейса должна также распознавать непонимание, между пользователем и системой, возникшее либо из-за ошибки, либо на принципиальной основе. Система должна реагировать соответствующим образом на эту ситуацию [7]. Например, не должно произойти сбоя системы, если пользователь вводит 1, когда ожидается "да" или "нет", или когда пользователь задает бессмысленный вопрос.

Способность экспертной системы моделировать человека эксперта может меняться от простых познавательных процессов до включения новых знаний или новых способов решения задачи. Система интерфейса должна информировать пользователя о методике работы системы и ее развитии, если такое развитие предусмотрено в системе.

Наконец, система пользовательского интерфейса должна быть "дружелюбной" к пользователю. Например, последовательность меню, показывающая задачи, которые пользователь может выбрать, является необходимой чертой экспертной системы.

Пользователь также должен иметь возможность взаимодействовать с экспертной системой естественным образом. В идеале пользователь должен иметь возможность использовать естественный язык.

## 1.2. Актуальность экспертных систем

История экспертных систем берет свое начало в 1965 году. Брюс Бучанан и Эдвард Фейгенбаум начали работу над созданием информационной системы для определения структуры химических соединений.

Результатом работы была система под названием Dendral. В основе системы формировалась последовательность правил подобных к «IF – THEN». Информационная система не перестала развиваться и получила множество наследников, таких как ONCOIN – информационная система для диагностики раковых заболеваний, MYCIN – информационная система для диагностики легочных инфекционных заболеваний [9].

Следующим этапом стали 70-е годы. Период не выделялся какими-либо особыми разработками. Было создано множество разных прототипов системы Dendral. Примером служит система PROSPECTOR, областью деятельности которой являлась геологические ископаемые и их разведка [2].

В 80-ых годах появляются профессия – инженер по знаниям [8]. Экспертные системы набирают популярность и выходят на новый этап эволюции интеллектуальных систем. Появились новые медицинские системы INTERNIS, CASNE.

С 90-ых годов развитие интеллектуальных систем приобретает новые и новые методы и особенности. Нововведением становится парадигма проектирования эффективных и перспективных систем. Гибкость, четкость решения поставленных задач дало новое название – мультиагентных систем. Агент – фоновый процесс, который действует в целях пользователя. Каждый агент имеет свою цель, «разум» и отвечает за свою область деятельности [10]. Все агенты в совокупности образуют некий интеллект. Агенты вступают в конкуренцию, настраивают отношения, кооперируются, что выглядит подобно деятельности людей.

В 21 веке интеллектуальные системы стали некой "обыденностью". Множество фирм внедряют экспертные системы в области своей деятельности, например:

- быстродействующая система OMEGAMON разрабатывается c 2004 года с IBM, цель которой отслеживание состояния корпоративной информационной сети. Служит для моментального принятия решений в критических или неблагоприятных ситуациях [6];

- G2 – экспертная система от фирмы Gensym, направленная на работу с динамическими объектами. Особенность этой системы состоит в том, что в нее внедрили распараллеливание процессов мышления, что делает ее быстрее и эффективней [4].

Класс экспертных систем сегодня объединяет несколько тысяч различных программных комплексов, которые можно классифицировать по различным критериям: решаемая задача, связь с реальным временем, тип ЭВМ, степень интеграции.

Как только инженеры смогут избавиться от таких недостатков экспертных систем, как отсутствие "здравого смысла", неспособность самообучаться и необходимость эксперта для поддержания работоспособности, экспертные системы вполне смогут занять рабочие места настоящих экспертов, за исключением тех, которые занимаются исследованиями и изобретениями в своей области, однако, замена таких экспертов также уже будет лишь вопросом времени [4].

## 1.3. Проблемы и преимущества экспертных систем

Главное достоинство экспертных систем - возможность накапливать знания, сохранять их длительное время, обновлять и тем самым обеспечивать относительную независимость конкретной организации от наличия в ней квалифицированных специалистов [3]. Накопление знаний позволяет повышать квалификацию специалистов, работающих на предприятии, используя наилучшие, проверенные решения.

Преимуществами экспертных систем можно назвать следующее.

1. Постоянство. Экспертные системы ничего не забывают в отличие от человека-эксперта, обеспечивая тем самым обхват всех доступных знаний.

2. Воспроизводимость. Можно сделать любое количество копий экспертной системы, в то время как обучение новых экспертов отнимает много времени и средств.

3. Эффективность. Работающая экспертная система может увеличить производительность компании и уменьшить затраты на персонал.

4. Постоянство выводов. С использованием экспертных системам похожие транзакции обрабатываются одним и тем же способом. Система будет делать сопоставимые рекомендации для похожих ситуаций.

5. Документация. Экспертная система может документировать процесс решения, тем самым предоставляя информацию для дальнейшей её настройки.

6. Законченность. Экспертная система может выполнять обзор всех транзакций, a человек-эксперт сможет сделать обзор только отдельной выборки его личного опыта.

7. Своевременность [1]. Погрешности в конструкциях и-или могут быть своевременно найдены.

8. Широта. В экспертной системе могут быть объединены знания многих экспертов, что дает системе больше широты, чем может достичь среднестатистический человек.

9. Снижение риска ведения дела благодаря последовательности принятия решения, документированности и компетентности.

Недостатками экспертных систем являются.

1. Здравый смысл [4]. В дополнение к широкому техническому знанию человек-эксперт имеет здравый смысл. Экспертная система руководствуется только теми правилами, которые в неё заложены, и чтобы гарантировать правильный результат в 100% случаев, нужно вложить немыслимое количество сил и средств, что перекроет любую итоговую ценность системы.

2. Творческий потенциал. Человек-эксперт может придумывать новые решения в нестандартных ситуациях, экспертные системы руководствуются только заранее известными данными и способами решения.

3. Обучение. Человек-эксперт автоматически адаптируются к изменению среды, в то время как экспертные системы нужно явно модифицировать.

4. Сенсорный опыт [7]. Человек-эксперт располагает широким диапазоном сенсорного опыта, однако экспертные системы в настоящее время основаны на вводе символов.

5. Экспертные системы не подходят, если решения не существует или когда проблема лежит вне области их компетенции.

Сегодняшним инженерам экспертных систем предстоит избавиться от текущих недостатков экспертных систем, чтобы в будущем они смогли полностью заменить специалистов в какой-либо области [4].

## 1.4. Методология разработки экспертных систем

Разработка интеллектуальных информационных систем отличается от создания обычного программного продукта. Опыт разработки ранних экспертных систем показал, что использование традиционной технологии программирования либо чрезмерно затягивает процесс разработки, либо вообще приводит к отрицательному результату. Это связано главным образом с необходимостью модифицировать принципы и способы построения по мере того, как увеличивается знание разработчиков о проблемной области [5].

Известно, что большая часть знаний в конкретной предметной области остается личной собственностью эксперта. Наибольшую проблему при разработке экспертной системы представляет процедура получения знаний у эксперта и занесения их в базу знаний, называемая извлечением знаний. Это происходит не потому, что он не хочет разглашать своих секретов, а потому, что не в состоянии сделать это — ведь эксперт знает гораздо больше, чем сам осознает. Для выявления знаний эксперта и их формализации на протяжении всего периода разработки с ним взаимодействует инженер по знаниям [7].

Чтобы избежать дорогостоящих и безуспешных попыток, был разработан набор рекомендаций для того, чтобы определить, является ли проблема подходящей для решения с помощью экспертной системы [1].

1. Потребность в решении должна соответствовать затратам на ее разработку. Суммы затрат и полученная выгода должны быть рациональными.

2. Невозможно использовать знания человека-эксперта там, где это необходимо. Если «экспертные» знания широко распространены, то маловероятно, что стоит разрабатывать экспертную систему. Однако в таких областях как разведка нефти и медицина могут быть редкие специализированные знания, которыми можно недорого снабдить экспертную систему, и не использовать очень высокооплачиваемого эксперта.

3. Проблема может быть решена с использованием символических методов рассуждения.

4. Проблема хорошо структурирована и не требует применения знаний, основанных на здравом смысле. Знания, основанные на здравом смысле, хорошо известны, поэтому их незачем фиксировать и представлять.

5. Проблема не может быть легко решена с использованием более традиционных вычислительных методов. Если имеется хорошее алгоритмическое решение проблемы, не следует использовать экспертную систему.

6. Существуют эксперты в данной проблемной области. Поскольку экспертная система проектируется для успешной работы, весьма существенно, чтобы эксперты желали помогать при ее проектировании, а не считали, что их работе угрожают. Кроме того, необходима поддержка администрации и потенциальных пользователей.

7. Проблема имеет подходящий размер и область применения. Как правило, проблема требует применения знаний высоко специализированных экспертов, но человек-эксперт должен тратить на ее решение короткое время, максимум час.

В настоящее время сложилась последовательность действий при разработке экспертных систем. Она включает следующие этапы: идентификация, получение знаний, концептуализация, формализация, выполнение, тестирование и опытная эксплуатация [4].

Идентификация

Этап идентификации связан, прежде всего, с осмыслением тех задач, которые предстоит решать будущей экспертной системе, и формированием требований к ней. На этом этапе планируется ход разработки прототипа системы, определяются источники знаний (книги, эксперты, методики), цели (распространение опыта, автоматизация рутинных операций), классы решаемых задач и т.д. Результатом идентификации является ответ на вопрос, что надо сделать и какие ресурсы необходимо задействовать.

Получение знаний

При решении проблемы получения знаний выделяют три стратегии.

1. Приобретение знаний. Под приобретением (acquisition) знаний понимается способ автоматизированного наполнения базы знаний посредством диалога эксперта и специальной программы.

2. Извлечение знаний. Извлечением (elicitation) знаний называют процедуру взаимодействия инженера по знаниям с источником знаний (экспертом, специальной литературой и др.) без использования вычислительной техники.

3. Обнаружение знаний. Термины «обнаружение знаний» (knowledge discovery), а также Data Mining связывают с созданием компьютерных систем, реализующих методы автоматического получения знаний [4].

Концептуализация

На этапе концептуализации проводится содержательный анализ проблемной области, выявляются используемые понятия и их взаимосвязи, определяются методы решения задач. Этот этап завершается созданием модели предметной области, включающей основные концепты и отношения. Модель представляется в виде графа, таблицы, диаграммы или текста.

Формализация

На этапе формализации все ключевые понятия и отношения выражаются на некотором формальном языке, который выбирается из числа существующих, либо создается заново. Другими словами, на данном этапе определяется состав средств и способов представления декларативных и процедурных знаний, осуществляется это представление и в итоге создается описание решения задачи экспертной системы на выбранном формальном языке.

Выполнение (реализация)

На этапе выполнения создается один или несколько реально работающих прототипов экспертной системы. Для ускорения этого процесса в настоящее время широко применяются различные инструментальные средства.

Тестирование

На данном этапе оценивается и проверяется работа программы прототипа с целью приведения ее в соответствие с реальными запросами пользователей. Прототип проверяется по следующим основным позициям:

- удобство и адекватность интерфейсов ввода-вывода (характер вопросов в диалоге, связность выводимого текста результата и др.);

- эффективность стратегии управления (порядок перебора, использование нечеткого вывода и т.д.);

- корректность базы знаний (полнота и непротиворечивость правил).

Задача стадии тестирования — выявление ошибок и выработка рекомендаций по доводке прототипа экспертной системы до промышленного образца.

Опытная эксплуатация

На этапе опытной эксплуатации проверяется пригодность экспертной системы для конечного пользователя. Пригодность определяется в основном удобством и полезностью разработки. Под полезностью понимается способность экспертной системы определять в ходе диалога потребности пользователя, выявлять и устранять причины неудач в работе, а также удовлетворять указанные потребности пользователя (решать поставленные задачи). Удобство работы подразумевает естественность взаимодействия с экспертной системой, гибкость (способность системы настраиваться на различных пользователей, а также учитывать изменения в квалификации одного и того же пользователя) и устойчивость системы к ошибкам (способность не выходить из строя при ошибочных действиях пользователя).

После успешного завершения этапа опытной эксплуатации экспертная система классифицируется как коммерческая система, пригодная не только для собственного использования, но и для продажи различным потребителям [4].

## 1.5. Анализ существующих аналогов

Отличный пример экспертной системы - небезызвестный Акинатор. Суть его в том, что пользователь должен загадать любого персонажа, а Джинн Акинатор должен его отгадать. В качестве персонажа могут выступать как реальные личности, так и выдуманные персонажи из любых произведений: фильмов, сказок, компьютерных игр и так далее.

Акинатор задаёт 40 вопросов. У него есть две дополнительные попытки (в каждой из которых несколько дополнительных вопросов) на тот случай, если он не смог отгадать загаданного пользователем персонажа за отведённые 40 вопросов. Или же, наоборот, он может задать меньше вопросов, если смог отгадать персонажа быстрее. Если он не знает вашего персонажа, вы можете добавить его.

На каждый вопрос предлагается выбрать один из пяти вариантов ответа по степени соответствия: «Да», «Возможно частично», «Я не знаю», «Скорее нет, не совсем», «Нет». После того, как пользователь ответил на вопрос, Акинатор задаёт следующий.

Примеры вопросов Акинатора: «Ваш персонаж существует на самом деле?», «Ваш персонаж женщина?», «Ваш персонаж носит усы?», «Ваш персонаж герой мультфильма?» и т.д.

Акинатор начинает с более общих вопросов, и каждый последующий вопрос носит уточняющий характер. Таким образом он фильтрует подходящих и неподходящих персонажей. Акинатор запоминает, как все пользователи ответили на тот или иной вопрос при загадывании того или иного персонажа, и таким образом на каждого персонажа создаётся некий реестр о том, как отвечали пользователи на вопросы о нём, и если данный пользователь ответит на вопросы так же, то Акинатор отгадает загаданного пользователем персонажа. Если Акинатор не смог отгадать персонажа, то он предлагает ввести имя, после чего запоминает его и все ответы, которые давал данный пользователь на вопросы об этом персонаже. И если другой пользователь загадает этого же персонажа, то Акинатор сможет уже его отгадать. Таким образом, количество персонажей, известных Акинатору, постоянно увеличивается. В случае, если Акинатор не отгадал персонажа, то он представляет вам возможных персонажей, которых он предполагал.

## 1.6. Постановка задачи

Главной задачей данного курсового проекта будет создание экспертной системы, позволяющей выдавать оценочные прогнозы на основе установленной пользователем базы знаний и полученных от пользователя ответов. Как уже было рассмотрено ранее, задачей экспертной системы является, в той или иной степени, замена какого-либо специалиста в своей области, либо оказание помощи работающему с ней специалисту. Однако такая концепция делает экспертную систему узконаправленной и, зачастую, не пригодной к повторному использованию в какой-либо другой сфере.

Чтобы решить данную проблему, целевая экспертная система должна быть разработана без привязки к какой-либо экспертной области, чтобы обеспечить её гибкость и сделать пригодной к повторному использованию. В связи с этим, экспертная система должна принимать на вход уже готовую базу знаний, в которой в некотором формате будут описаны выводы и правила, по которым система будет "склоняться" к тому или иному выводу (например, какие-либо вопросы). Исходя из заданных условий, вывод экспертной системы не будет однозначным, а будет в формате неких вероятностей, с которыми можно утверждать о каком-либо выводе. Как и в любой экспертной системе, корректность выводов по большей части будет зависеть от размера и корректности базы знаний, а также от корректности пользовательских ответов.

В данном разделе были проанализированы основные моменты, касающиеся исследуемой области, существующие аналоги, а также сформулирована и поставлена задача.

## 2. Разработка структуры приложения

На основании общепринятой модели экспертной системы, была разработана следующая структура приложения, изображенная на рисунке 2.

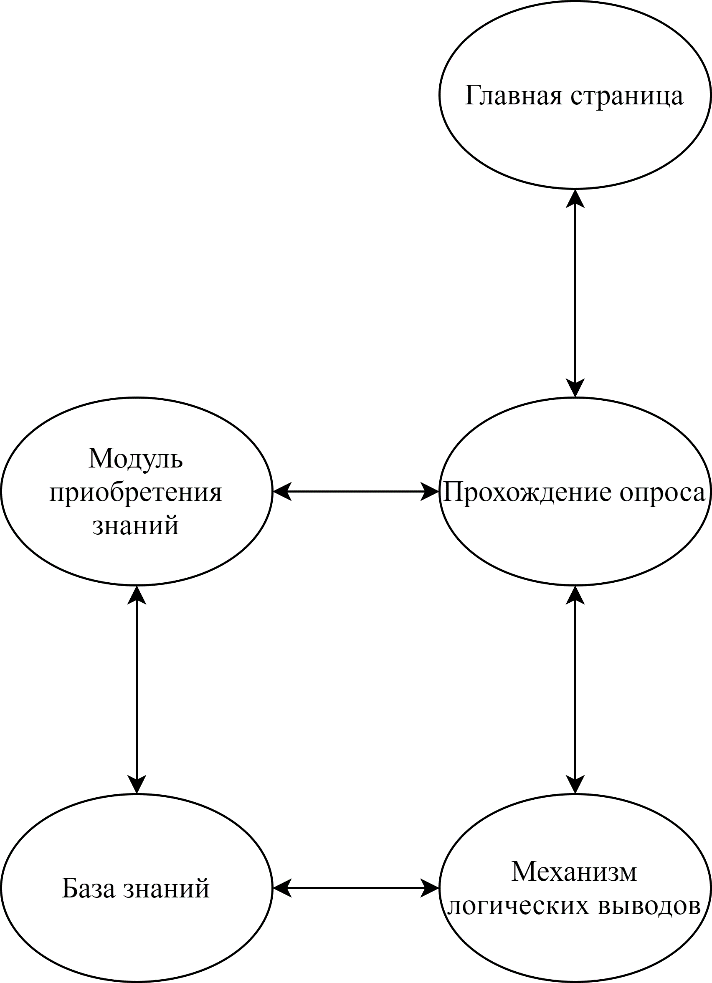


Рисунок 2 – Структура приложения

Главная страница представляет из себя часть пользовательского интерфейса, отвечающую за ввод и запуск опроса.

Прохождение опроса - часть пользовательского интерфейса, отвечающая за диалог между пользователем и экспертной системой.

Модуль приобретения знаний дополняет содержимое базы знаний.

База знаний реализует функции представления знаний в конкретной предметной области и управление ими.

Механизм логических выводов – модуль, оперирующий содержимым базы знаний и полученными фактами, а также выполняющий логические выводы на их основе.

## 2.1. Разработка модели экспертной системы

В данном курсовом проекте за основу была выбрана модель экспертной системы, основанная на продукционной модели представления знаний с прямым методом логического вывода.

В основе продукционной модели представления знаний находится конструктивная часть, продукция(правило): IF <условие>, THEN <действие>.  
Продукция состоит из двух частей: условие — антецендент, действие — консеквент. Правило срабатывает, если при сопоставлении фактов с антецедентом правила, которое подвергается проверке, имеет место совпадение.

Системы с прямым логическим выводом находятся под управлением фактов. Они начинают свою работу с известных начальных фактов и продолжают, используя правила для создания выводов или выполнения определенных действий. Результат работы правила заносится в базу данных, но в данном курсовом проекте сразу выводится пользователю.

Экспертная система базируется на вероятностях, где значение, равное 1, означает конкретный прогноз системы. С самого начала опроса у каждого возможного варианта ответа есть фиксированная вероятность, от которой отталкиваются дальнейшие вычисления. Значения max и min помогают определить, насколько сильно изменится вероятность возможного варианта ответа в зависимости от ответа пользователя на определенный вопрос.

Правило срабатывает, когда пользователь ответил на все вопросы либо их некоторую часть так, чтобы система точно могла сделать прогноз и вывести результат пользователю. Так как экспертная система базируется на вероятностях, то, как было указано ранее, в процессе опроса вывод экспертной системы не будет однозначным, а будет в формате вероятностей, с которыми можно утверждать о каком-либо выводе.

## 2.2. Разработка алгоритмов

Алгоритм работы приложения довольно прост. Сначала пользователю необходимо выбрать либо вбить опрос, после чего нажать кнопку начала опроса. Если опрос был введен, он будет считан, обработан и занесен в базу знаний. В итоге вне зависимости от способа запуска опроса он будет реализован и выведен пользователю.

Далее пользователю предстоит отвечать на вопросы, задаваемые системой, с помощью пяти кнопок-ответов. После нажатия на одну из кнопок-ответов система обрабатывает ответ пользователя и выводит результат своей работы. Повторяется до тех пор, пока пользователь не ответит на все вопросы, либо система не сделает конкретный прогноз.

По окончании прохождения опроса пользователь может снова пройти опрос либо вернуться на главную страницу и запустить интересующий его опрос.

В данном разделе были рассмотрены структура приложения, модель экспертной системы и алгоритмы работы программы.

## 3. Реализация и отладка программы

В данном разделе описаны основные моменты по разработке интерфейса пользователя, используемым при разработке средствам и проверке работоспособности конечного результата.

## 3.1. Разработка интерфейса пользователя

Система пользовательского интерфейса представляет из себя веб-страницу, состоящую из 2 страниц, главной и побочной, на которых пользователь может загрузить или выбрать интересующий его опрос, после чего пройти его и получить прогноз.

На главной странице располагаются кнопки «Начать» и «Формат данных», список доступных опросов и текстовую область, куда можно вбить опрос.

Кнопка «Начать» позволяет перейти к интересующему пользователя опросу, будь то выбор из списка доступных либо вбитый опрос. Если формат данных не соответствует указанному, появится окно, оповещающее об этом.

Кнопка «Формат данных» выводит в текстовой области формат данных, в котором необходимо предоставить опрос. Если до нажатия в области что-то находилось, оно будет замещено выведенным сообщением.

В списках доступных опросов находятся уже вбитые опросы, которые могут быть активированы нажатием на соответствующий опрос и кнопку «Начать».

На побочной странице располагаются возможные варианты ответа, напротив которых находятся значения вероятности, история вопросов, текущий вопрос и кнопки с вариантами ответов «Да», «Нет», «Не знаю», «Маловероятно», «Вероятнее всего», а также кнопки «Пройти снова» и «Вернуться на главную».

С помощью кнопок-ответов пользователь может вести диалог с экспертной системой в удобной форме. В зависимости от ответа на определенные вопросы, вероятности возможных вариантов ответа будут изменяться до тех пор, пока один из вариантов не достигнет значения, равного 1, либо не закончатся вопросы. По окончании опроса пользователь может либо заново пройти опрос, либо вернуться на главную.

Переход между двумя страницами осуществляется через сокрытие одних элементов и показ других в определенные моменты работы программы.

## 3.2. Используемые при разработке средства

Для разработки и реализации экспертной системы были использованы следующие средства:

- JavaScript;

- HTML/CSS;

- WebStorm.

JavaScript – прототипно-ориентированный сценарный язык программирования. Является диалектом языка ECMAScript.

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности web – страницам. Основные архитектурные черты: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса.

HTML (от англ. Hyper Text Markup Language – «язык гипертекстовой разметки») – стандартный язык разметки документов во Всемирной паутине. Большинство web-страниц содержат описание разметки на языке HTML (или XHTML). Язык HTML интерпретируется браузерами и отображается в виде документа в удобной для человека форме.

Язык HTML является приложением SGML (стандартного обобщённого языка разметки) и соответствует международному стандарту ISO8879.

Во всемирной паутине HTML-страницы, как правило, передаются браузерам от сервера по протоколам HTTP или HTTPS, в виде простого текста или с использованием сжатия.

Текстовые документы, содержащие разметку на языке HTML (такие документы традиционно имеют расширение.html или.htm), обрабатываются специальными приложениями, которые отображают документ в его форматированном виде. Такие приложения, называемые «браузерами» или «интернет – обозревателями», обычно предоставляют пользователю удобный интерфейс для запроса web – страниц, их просмотра (и вывода на иные внешние устройства) и, при необходимости, отправки введённых пользователем данных на сервер. Наиболее популярными на сегодняшний день браузерами являются Google Chrome, Mozilla Firefox, Opera, Internet Explorer и Safari .

CSS (англ. Cascading Style Sheets – каскадные таблицы стилей) формальный язык описания внешнего вида документа, написанного с использованием языка разметки.

Преимущественно используется как средство описания, оформления внешнего вида web-страниц, написанных с помощью языков разметки HTML и XHTML, но может также применяться к любым XML – документам, например, к SVG или XUL.

CSS используется создателями web – страниц для задания цветов, шрифтов, расположения отдельных блоков и других аспектов представления внешнего вида этих web-страниц. Основной целью разработки CSS являлось разделение описания логической структуры web-страницы от описания внешнего вида этой web-страницы. Такое разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом. Кроме того, CSS позволяет представлять один и тот же документ в различных стилях.

WebStorm – интегрированная среда разработки на JavaScript, CSS & HTML от компании JetBrains, разработанная на основе платформы [IntelliJ IDEA](https://ru.wikipedia.org/wiki/IntelliJ_IDEA).

WebStorm обеспечивает [автодополнение](https://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D0%B4%D0%BE%D0%BF%D0%BE%D0%BB%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5), анализ кода на лету, [навигацию по коду](https://ru.wikipedia.org/wiki/%D0%9D%D0%B0%D0%B2%D0%B8%D0%B3%D0%B0%D1%86%D0%B8%D1%8F_%D0%BF%D0%BE_%D0%BA%D0%BE%D0%B4%D1%83), [рефакторинг](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D1%84%D0%B0%D0%BA%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3), [отладку](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BB%D0%B0%D0%B4%D0%BA%D0%B0), и интеграцию с [системами управления версиями](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D1%8F%D0%BC%D0%B8). Важным преимуществом [интегрированной среды разработки](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8) WebStorm является работа с проектами (в том числе, рефакторинг кода JavaScript, находящегося в разных файлах и папках проекта, а также вложенного в HTML). Поддерживается множественная вложенность (когда в документ на HTML вложен скрипт на JavaScript, в который вложен другой код HTML, внутри которого вложен JavaScript) — то есть в таких конструкциях поддерживается корректный рефакторинг.

## 3.3. Проверка работоспособности

Система состоит из 3 файлов:

- index.html;

- expert.css;

- expert.js.

Первые два файла отвечают за веб-составляющую, а последний за логическую часть.

Для доступа к интерфейсу необходимо запустить файл index.html в любом доступном браузере.

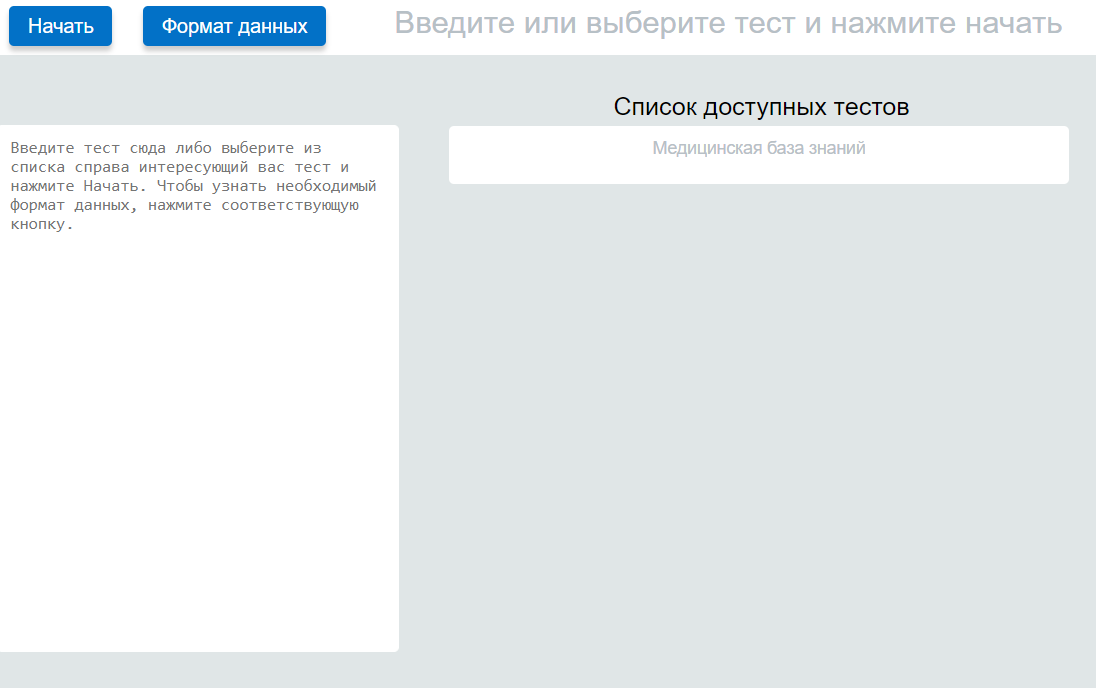


Рисунок 3 – Главная страница

В текстовую область следует вставить валидный опрос, состоящий из вопросов и выводов. Формат вводимых данных должен выглядеть так:

- на первой строке располагается название базы знаний;

- затем через знак перевода строки - список вопросов;

- после вопросов - следует список событий (вариантов) в формате:

[Событие]<Перевод строки>[Начальная вероятность события] [номер вопроса]) [вероятность max] [вероятность min]<перевод строки>

В данном примере будет продемонстрирована работа системы на заранее подготовленной медицинской базе знаний.

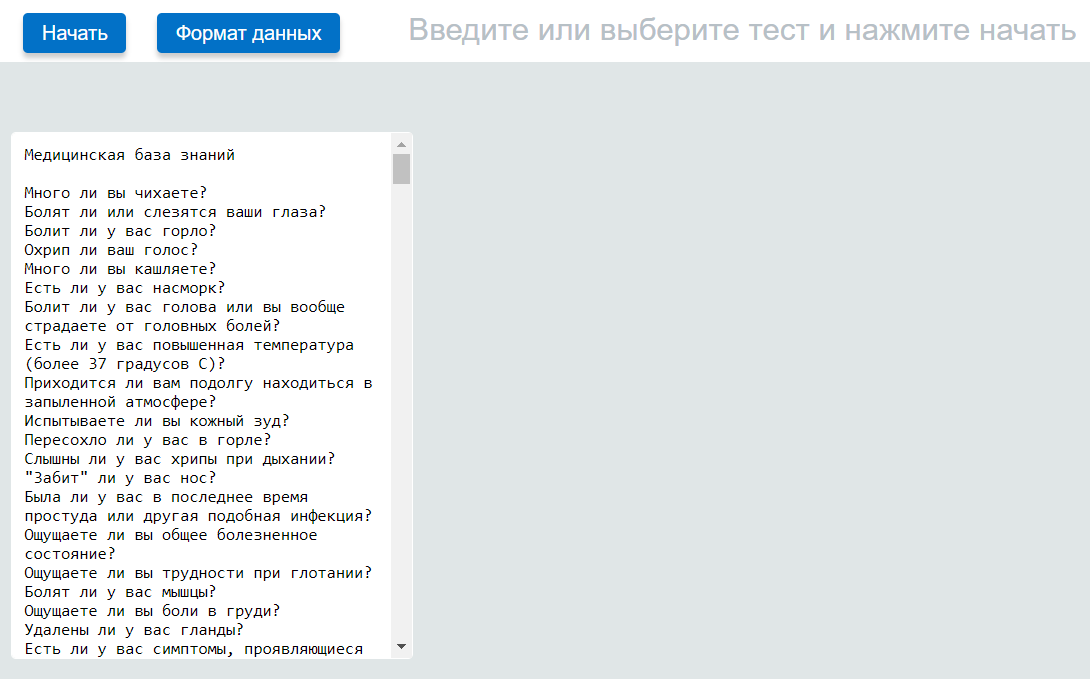


Рисунок 4 – База знаний

Затем система будет задавать вопросы, на которые необходимо ответить, нажав на одну из кнопок-ответов.

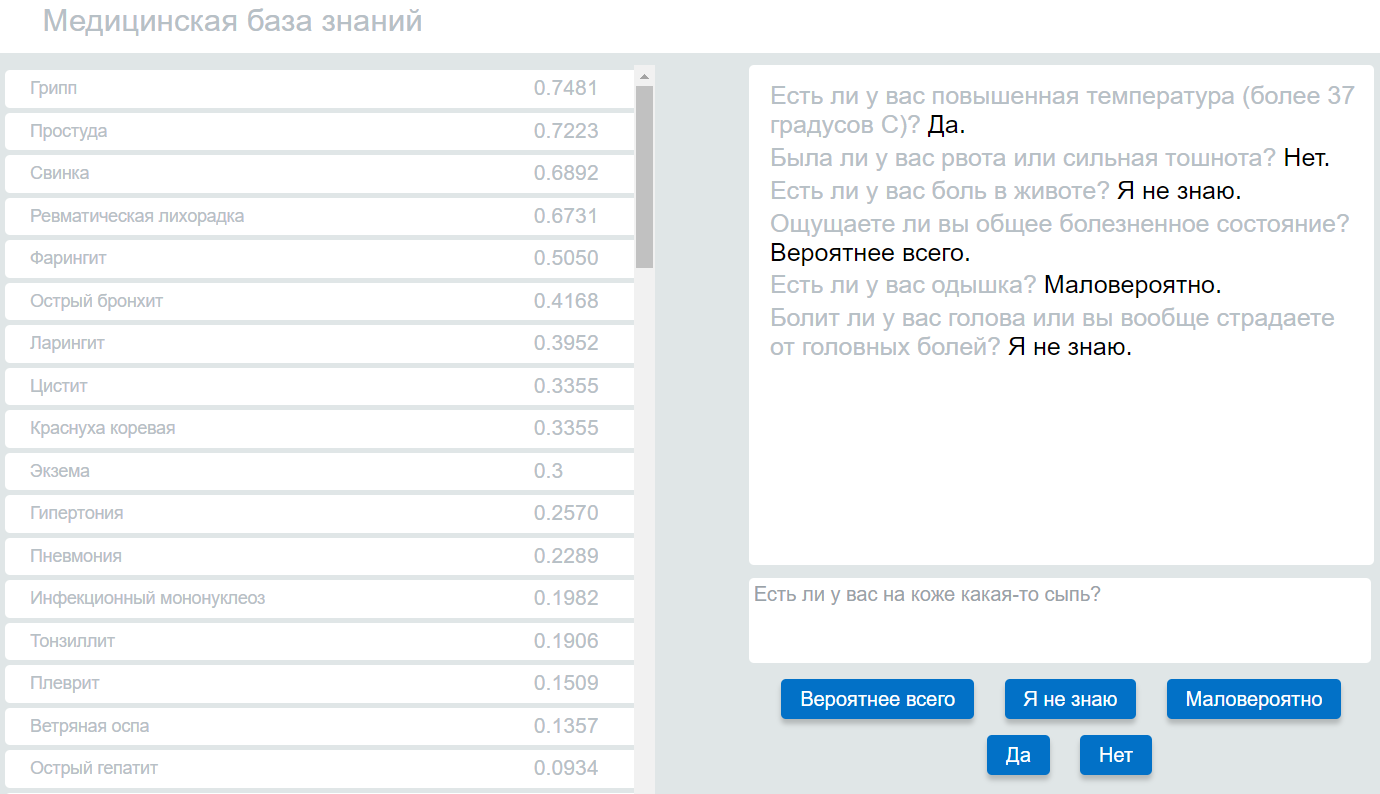


Рисунок 5 – Процесс прохождение опроса

По окончании опроса результат буден виден в списке слева.

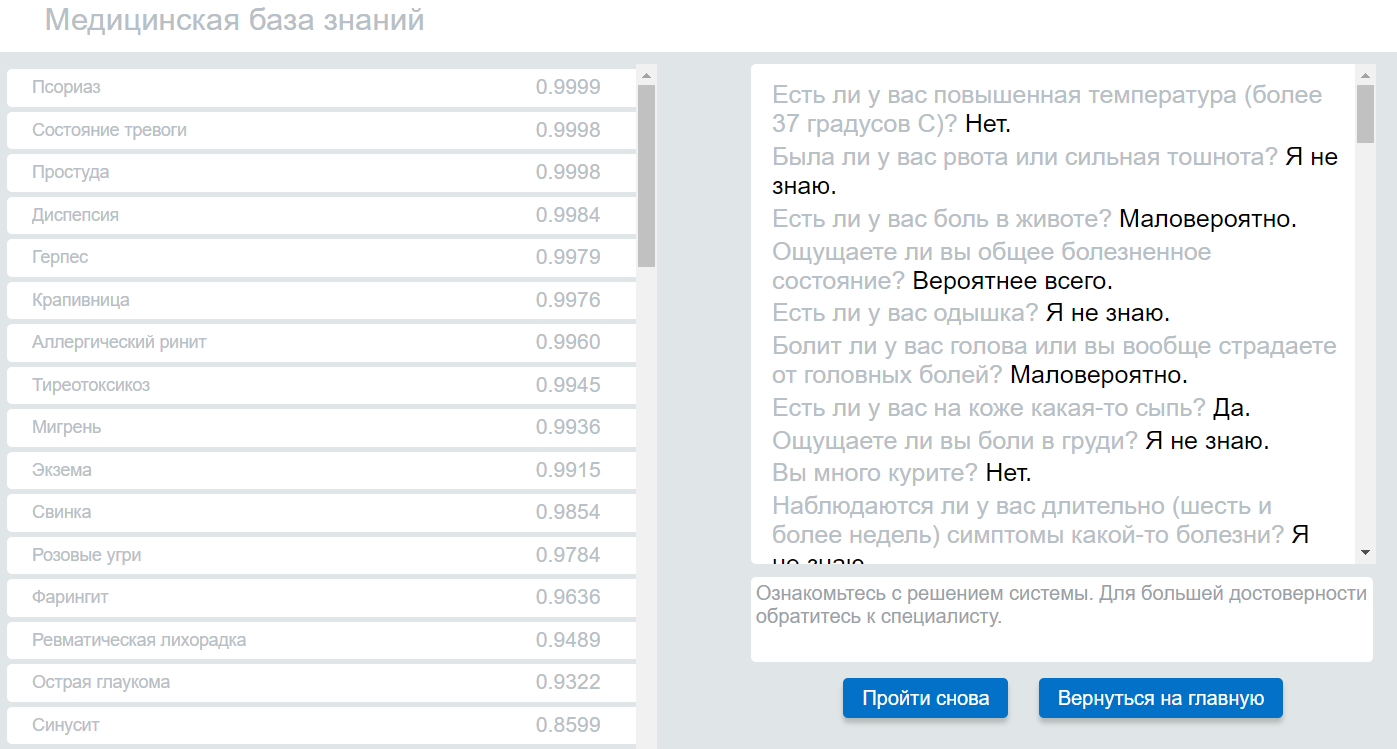


Рисунок 6 – Конец опроса

Как видно, программа отлажена и отлично работает, обрабатывая все необходимые действия пользователя и выводя прогноз.

В данном разделе были рассмотрены основные моменты по разработке интерфейса пользователя, используемым при разработке средствам и проверке работоспособности конечного результата.

## Заключение

В процессе выполнения курсового проекта были изучены теоретические основы экспертных систем, исследованы их проблемы и преимущества, перспективы развития и использования, а также рассмотрен процесс создания и эксплуатации экспертных систем. Помимо прочего, была сформулирована и поставлена задача, разработаны структура и модель экспертной системы, а также структура пользовательского интерфейса и алгоритмы работы.

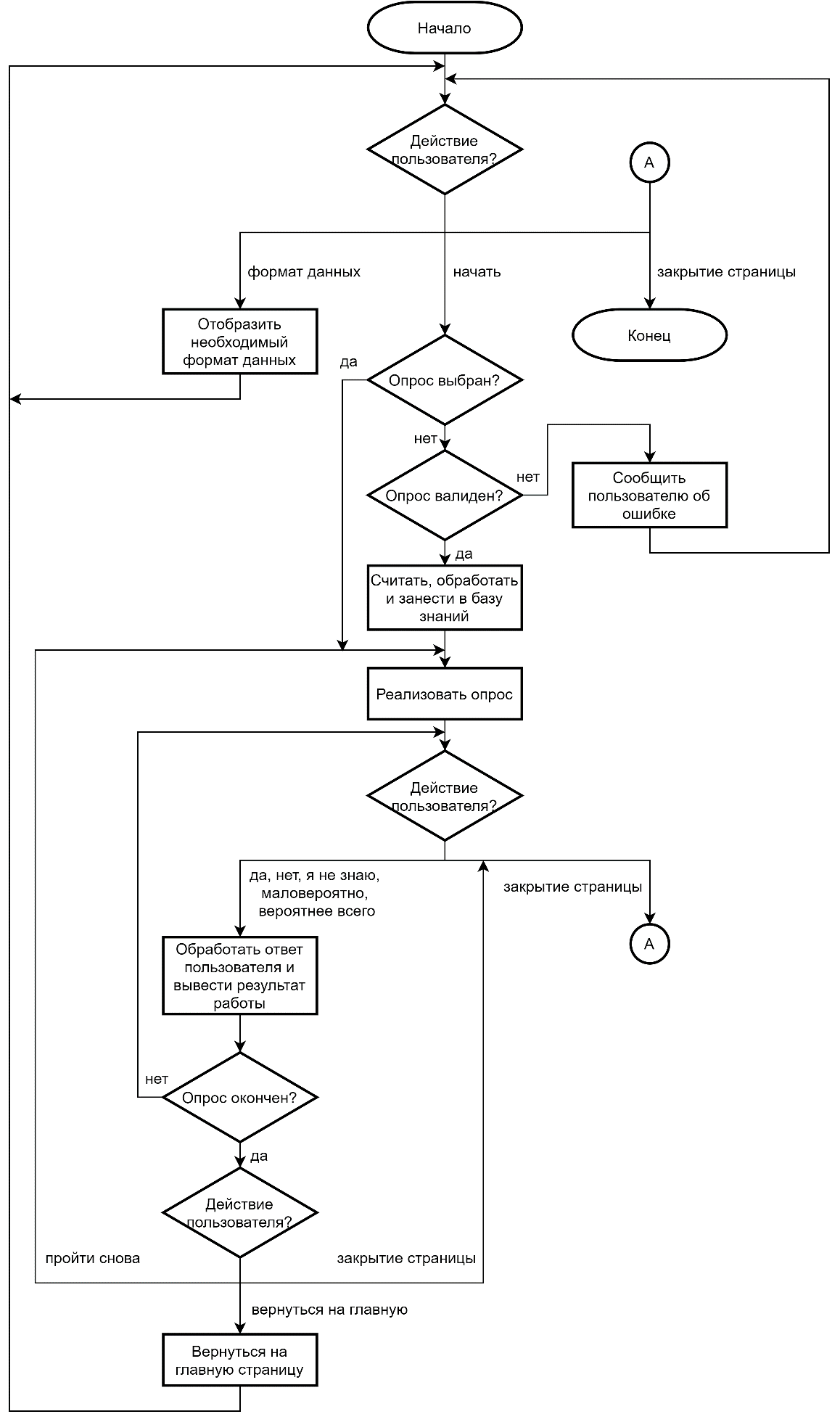
Анализ существующих аналогов помог понять, что из себя представляют реальные экспертные системы и как они должны работать.

В результате выполнения курсового проекта была создана и отлажена демонстрационная экспертная система, позволяющая выдавать прогнозы в какой-либо сфере на основе заранее заданной базы знаний, исходя из ответов пользователя.

## Приложение А

(обязательное)

Схема алгоритма



## Приложение Б

(обязательное)

Исходный код

**let testSystem;  
/\*\*  
 \* Проверка доступности localStorage  
 \* @returns {boolean} true - браузер поддерживает LocalStorage  
 \*/  
function checkLocalStorage()  
{  
 try {  
 return (localStorage !== null && localStorage !== undefined);  
 } catch (e) {  
 return false;  
 }  
}  
/\*\*  
 \* Загрузка опросов (пунктов меню), существующих в localStorage  
 \*/  
function loadItemsFromLocalStorage()  
{  
 if (!checkLocalStorage())  
 {  
 return;  
 }  
 let template = '<div class="b-page-test-switch-item b-page border-radius left-side" index="{1}">{0}</div>';  
 let target = $(".b-page-test-switch");  
 target.html('');  
 if (localStorage.length>0) {  
 $('.b-page-test-label').removeClass('hide');  
 for (let i = 0; localStorage["ExpertSys" + i]; i++) {  
 target.html(target.html() + template.replace("{1}", "ExpertSys" + i).replace("{0}", JSON.parse(localStorage["ExpertSys" + i]).title));  
 }  
 }  
 else{  
 $('.b-page-test-label').addClass('hide');  
 }  
}  
/\*\*  
 \* Загрузка опроса с localStorage  
 \* @param localStorageIndex - индекс опроса  
 \* @returns {Test} опрос (база знаний) для экспертной системы  
 \*/  
function loadFromLocalStorage(localStorageIndex)  
{  
 return new Test(JSON.parse(localStorage[localStorageIndex]));  
}  
/\*\*  
 \* Сохранение опроса (базы знаний) в localStorage (при условии поддержки оного браузером)  
 \* @param test - сохраняемый опрос (база знаний)  
 \*/  
function saveToLocalStorage(test)  
{  
 if (!checkLocalStorage())  
 {  
 return;  
 }  
 let i  
 for (i = 0; i<localStorage.length; i++)  
 {  
 if (localStorage["ExpertSys"+i].match(test.title))  
 {  
 localStorage.setItem("ExpertSys" + i,test.stringify());  
 return;  
 }  
 }  
 localStorage.setItem("ExpertSys" + i,test.stringify());  
}  
/\*\*  
 \* Реакция нажатия одной из кнопкок-ответов и переход к следующему вопросу  
 \*/  
function step(ans)  
{  
 testSystem.processAnswer(ans);  
 testSystem.nextStep();  
}  
/\*\*  
 \* Реакция нажатия кнопки "Пройти снова"  
 \*/  
function rerun()  
{  
 if ($(".b-page-test-switch-selected") && $(".b-page-test-switch-selected").length > 0)  
 {  
 testSystem = loadFromLocalStorage($(".b-page-test-switch-selected").attr("index"));  
 }  
 else  
 {  
 let i  
 for(i = 0;i<localStorage.length;i++)  
 {  
 if(JSON.parse(localStorage["ExpertSys" + i]).title === testSystem.title){  
 testSystem = loadFromLocalStorage("ExpertSys" + i);  
 break;  
 }  
 }  
 }  
 $('#yes-no-buttons').removeClass('hide');  
 $('#more-yes-answer').removeClass('hide');  
 $('#more-no-answer').removeClass('hide');  
 $('#idk-answer').removeClass('hide');  
 $('#rerun').addClass('hide');  
 $('#back').addClass('hide');  
 $('.b-page-questions-answers').html('');  
 testSystem.nextStep();  
}  
/\*\*  
 \* Реакция нажатия кнопки "Вернуться на главную"  
 \*/  
function back()  
{  
 $('.b-page-questions-answers').html('');  
 $('.b-page-test').addClass('hide');  
 $('#yes-no-buttons').removeClass('hide');  
 $('#more-yes-answer').removeClass('hide');  
 $('#more-no-answer').removeClass('hide');  
 $('#idk-answer').removeClass('hide');  
 $('#rerun').addClass('hide');  
 $('#back').addClass('hide');  
  
 $("#test-title").html('Введите или выберите тест и нажмите начать');  
 $('.b-page-test-switch-selected').removeClass('b-page-test-switch-selected');  
 $('.b-page-main').removeClass('hide');  
 $('#start-test').removeClass('hide');  
 $('#data-format').removeClass('hide');  
 $('#test').val('');  
 loadItemsFromLocalStorage();  
}  
/\*\*  
 \* Инициализация  
 \*/  
function init()  
{  
 loadItemsFromLocalStorage();  
  
 // Выбор опроса. Его загрузка с localStorage, либо парсинг с textarea  
 $("#start-test").bind("click keypress",function()  
 {  
 if ($(".b-page-test-switch-selected") && $(".b-page-test-switch-selected").length > 0)  
 {  
 testSystem = loadFromLocalStorage($(".b-page-test-switch-selected").attr("index"));  
 }  
 else  
 {  
 testSystem = new Test();  
 if (!testSystem.parseData($("#test").val()))  
 {  
 alert("Что-то пошло не так. Проверьте вводимые данные.")  
 return;  
 }  
 saveToLocalStorage(testSystem);  
 }  
 $(".b-page-test").removeClass("hide");  
 $(".b-page-main").add("#start-test,#data-format").addClass("hide");  
 $("#test-title").html(testSystem.title);  
 testSystem.nextStep();  
 });  
 // Присваивание функций кнопкам "Пройти снова" и "Вернуться на главную"  
 $('#rerun').bind('click keypress',rerun)  
 $('#back').bind('click keypress',back)  
  
 // Визуализация выбора теста  
 $(".b-page-test-switch-item").live("click keypress", function()  
 {  
 if ($(this).hasClass("b-page-test-switch-selected"))  
 {  
 $(this).removeClass("b-page-test-switch-selected");  
 return;  
 }  
 $(".b-page-test-switch-selected").removeClass("b-page-test-switch-selected");  
 $(this).addClass("b-page-test-switch-selected");  
 });  
 // Вывод информации о формате данных  
 $("#data-format").bind("click keypress",function()  
 {  
 $("#test").val(  
 "Формат вводимых данных:\n" +  
 "На первой строке располагается название базы знаний\n" +  
 "Затем через знак перевода строки - список вопросов\n" +  
 "После вопросов - следует список событий (вариантов) в формате:\n"+  
 "Событие<Перевод строки>Начальная\_вероятность\_события номер\_вопроса) вероятность\_max вероятность\_min<перевод строки>\n" +  
 "Пример ввода:\n\n" +  
 "Определение пола\n" +  
 "\n" +  
 "Вы мальчик?\n"+  
 "Вы девочка?\n"+  
 "\n" +  
 "Мальчик\n" +  
 "0.5 1) 1 0 2) 0 1\n" +  
 "Девочка\n"+  
 "0.5 2) 1 0 1) 0 1\n\n" +  
 "Номера вопросов у вариантов можно устанавливать не по порядку. Также вопросы, не влияющие на вероятность события, можно опускать.");  
 }) ;  
}  
window.onload = init;  
/\*\*  
 \* Сортировка вероятностей. Сравнение двух items  
 \* @returns {number} сравнение(1, 0, -1)  
 \*/  
function sortItems(a,b)  
{  
 if (a.points > b.points || (a.points === b.points && a.title > b.title)) return 1;  
 if (a.points === b.points && a.title === b.title) return 0;  
 return -1;  
}  
/\*\*  
 \* сравнение двух объектов - вопросов  
 \*/  
function sortQuestion(a,b)  
{  
 let aPoints = 0, bPoints = 0;  
 for (let i = 0; i < a.items.length; i++)  
 {  
 aPoints += a.items[i].questionPoints[a.index].min + a.items[i].questionPoints[a.index].max + a.items[i].points;  
 }  
 for (let i = 0; i < b.items.length; i++)  
 {  
 bPoints += b.items[i].questionPoints[b.index].min + b.items[i].questionPoints[b.index].max + b.items[i].points;  
 }  
  
 if (aPoints > bPoints) return -1;  
 if (aPoints === bPoints)  
 {  
 if (a.items.length > b.items.length) return -1;  
 if (a.items.length === b.items.length) return 0;  
 }  
 return 1;  
}  
/\*\*  
 \* Опрос (база знаний)  
 \* @param testObject - конструктор с уже введенным названием, вариантом и вопросом  
 \*/  
function Test(testObject)  
{  
 this.title = "";  
 this.items = [];  
 this.questions = [];  
 this.currentQuestion = -1;  
 this.complete = false;  
 if (testObject)  
 {  
 this.title = testObject.title;  
 this.items = testObject.items;  
 this.questions = testObject.questions;  
 }  
}  
/\*\*  
 \* Сериализация в строку, формата JSON  
 \* @returns {String} JSON  
 \*/  
Test.prototype.stringify = function()  
{  
 return JSON.stringify({  
 title : this.title,  
 items : this.items,  
 questions : this.questions  
 });  
}  
/\*\*  
 \* Вывод "вероятностей" ответов по заданному шаблону  
 \*/  
Test.prototype.printData = function()  
{  
 this.items.sort(sortItems);  
 let template = '<div title="{0}: {1}" class="b-page-test-items-item border-radius"><span class="b-page-test-items-item-title">{0}</span><span class="b-page-test-items\_\_item-percent">{1}</span></div>';  
 let t = $(".b-page-test-items");  
 t.html("");  
 for (let i = this.items.length-1; i >= 0; i--)  
 {  
 let res = this.items[i].points > 1? 1 : this.items[i].points;  
 t.html(t.html() + template.replace("{0}",this.items[i].title).replace("{0}",this.items[i].title).replace("{1}",res).replace("{1}",res));  
 if (res === 1)  
 {  
 this.complete = true;  
 }  
 }  
}  
/\*\*  
 \* Переход к следующему вопросу  
 \*/  
Test.prototype.nextStep = function()  
{  
 this.printData();  
 this.questions.sort(sortQuestion);  
 if (this.questions.length === 0 || this.complete)  
 {  
 $("#current-question").html("Ознакомьтесь с решением системы. Для большей достоверности обратитесь к специалисту.");  
 $('#more-no-answer').addClass('hide');  
 $('#idk-answer').addClass('hide');  
 $('#more-yes-answer').addClass('hide');  
 $('#yes-no-buttons').addClass('hide');  
 $('#rerun').removeClass('hide');  
 $('#back').removeClass('hide');  
 return;  
 }  
 $("#current-question").html(this.questions[0].q);  
};  
/\*\*  
 \* Обработка ответа. Измненение вероятности события по ответу  
 \* @param ans - ответ (вероятность, значение зависит от нажатой кнопки-ответа)  
 \*/  
Test.prototype.processAnswer = function(ans)  
{  
 for (let i = 0 ; i < this.items.length; i++)  
 {  
 let point = this.items[i].questionPoints[this.questions[0].index];  
 if (point)  
 {  
 let up = ((2\*point.max - 1)\*ans/100 + 1 - point.max) \* this.items[i].points;  
 let down = ((2\*point.max - 1)\*ans/100 + 1 - point.max) \* this.items[i].points + ((2\*point.min - 1)\*ans/100 + 1 - point.min)\*(1 - this.items[i].points);  
 this.items[i].points = down !== 0? up/down : this.items[i].points;  
 }  
 }  
 let template = '<div class="b-page-questions-answers-item">{0} <span class = "user-answers">{1}</span></div>'  
 switch (ans){  
 case 0:$("#answers").html($("#answers").html() + template.replace("{0}", this.questions[0].q).replace("{1}",$('#no-answer').html() + '.'));  
 break  
 case 25:$("#answers").html($("#answers").html() + template.replace("{0}", this.questions[0].q).replace("{1}",$('#more-no-answer').html() + '.'));  
 break  
 case 50:$("#answers").html($("#answers").html() + template.replace("{0}", this.questions[0].q).replace("{1}",$('#idk-answer').html() + '.'));  
 break  
 case 75:$("#answers").html($("#answers").html() + template.replace("{0}", this.questions[0].q).replace("{1}",$('#more-yes-answer').html() + '.'));  
 break  
 case 100:$("#answers").html($("#answers").html() + template.replace("{0}", this.questions[0].q).replace("{1}",$('#yes-answer').html() + '.'));  
 break  
 }  
 this.questions.shift();  
}  
/\*\*  
 \* Получение вопросов, событий и вероятностей с строки  
 \* @param data - строка, содержащая данные  
 \* @returns {boolean} успех обработки строки  
 \*/  
Test.prototype.parseData = function(data)  
{  
 try  
 {  
 //Пропуск лишних, пустых строк  
 let passEmptyStrings = function() {  
 while (position < items.length && items[position] == "" ) { position++; }  
 }  
 let items = data.split("\n");  
 let position = 0;  
 passEmptyStrings();  
 this.title = items[position++];  
 passEmptyStrings();  
 //Ввод вопросов  
 while (items.length > position && items[position] != "")  
 {  
 this.questions.push({  
 q :items[position++],  
 items: [],  
 index:this.questions.length  
 });  
 }  
 if (items.length <= position) throw "Invalid data format";  
 passEmptyStrings();  
 let index = 0;  
 //Ввод событий  
 while (items.length > position && items[position] != "")  
 {  
 let pointItems = items[position + 1].split(" ");  
 let newItem = {  
 title : items[position],  
 points : parseFloat(pointItems[0]),  
 index : index,  
 questionPoints : []  
 };  
 //Вероятности событий при 100 и 0% вероятностях ответа на вопрос  
 for (let i = 1; i < pointItems.length; i+= 3)  
 {  
 while (pointItems[i] == "") i++;  
 let questionIndex = parseFloat(pointItems[i]) - 1;  
 let questionPoint = {  
 max : parseFloat(pointItems[i+1]),  
 min : parseFloat(pointItems[i+2])  
 };  
 newItem.questionPoints[questionIndex] = questionPoint;  
 this.questions[questionIndex].items.push(newItem);  
 }  
 this.items.push(newItem);  
 index++;  
 position += 2;  
 }  
 return true;  
 }  
 catch(e)  
 {  
 console.log(e);  
 return false;  
 }  
}**

## Приложение В

(справочное)

Список используемой литературы

1. Автоматизированные информационные технологии в экономике: Учебник/ Под ред. Г.А.Титоренко. – М.: ЮНИТИ, 1998.

2. Долин Г. Что такое ЭС.- Компьютер Пресс, 2001/2.

3. Семенов М.И. и др. Автоматизированные информационные технологии в экономике // Финансы и статистика – 2000.

4. Миронов А. С. Экспертные системы // Молодой ученый. — 2016.

5. Малпасс Д. Р.. Реляционный язык Пролог и его применение.-М.:Мир,2000.

6. Д. Джарратано, Г. Райли «Экспертные системы: принципы разработки и программирование» : Пер. с англ. — М. : Издательский дом «Вильямс», 2006.

7. Марселлус Д. Н. Программирование экспертных систем на Турбо Прологе.- М.: Финансы и статистика, 2002.

8. Братко И. Программирование на языке Пролог для искусственного интеллекта.- М.: Мир, 2000.

9. Информационные технологии управления: Учебн. пособие для вузов/ Под ред. проф. Г.А.Титоренко. – М.: ЮНИТИ – ДАНА, 2003.

10. Нейлор К. Как построить свою экспертную систему.- М.: Энергоатомиздат, 2001.