





## Оглавление

ВВЕДЕНИЕ.....	3
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ .....	5
1.1    ОБЩАЯ ХАРАКТЕРИСТИКА ПРЕДПРИЯТИЯ МБОУ ДО СЮТ ...	5
1.2    ОБЩАЯ ХАРАКТЕРИСТИКА ПРЕДПРИЯТИЯ.....	7
1.3    ПРИМЕНЕНИЕ ЭКСПЕРТНЫХ СИСТЕМ .....	10
2. МЕТОДОЛОГИЯ РАЗРАБОТКИ ЭКСПЕРТНОЙ СИСТЕМЫ.....	13
2.1    СУЩЕСТВУЮЩИЕ ПОДХОДЫ ПО АВТОМАТИЗАЦИИ ПРОЦЕССА МЕДИЦИНСКОГО ОСМОТРА ПЕРСОНАЛА ОРГАНИЗАЦИИ .....	13
2.2    ВЫБОР СТРАТЕГИИ АВТОМАТИЗАЦИИ.....	13
2.3    ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ПРОЦЕССА ПРОЕКТИРОВАНИЯ .....	14
2.4    ПОСТРОЕНИЕ МОДЕЛИ «ТО-ВЕ» .....	16
2.5    ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ПРОЦЕССА МЕДОСМОТРА .....	25
2.6    ОПИСАНИЕ СРЕДЫ РАЗРАБОТКИ CLIPS .....	28
3. РЕАЛИЗАЦИЯ ЭКСПЕРТНОЙ СИСТЕМЫ.....	33
3.1    РАЗРАБОТКА ЭКСПЕРТНОЙ СИСТЕМЫ .....	33
3.2    ЭКОНОМИЧЕСКАЯ ЭФФЕКТИВНОСТЬ .....	47
ЗАКЛЮЧЕНИЕ.....	54
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ .....	55

## ВВЕДЕНИЕ

Когда человек или организация встречаются с задачей, которую они не могут решить самостоятельно — они обращаются к знающим людям, или к экспертам, к тем, кто обладает знаниями.

«Системы, основанные на знаниях», такой термин появился ещё в 1976 году одновременно с первыми системами, которые были способны аккумулировать опыт и знания. Это были экспертные системы MYCIN и DENDRAL для медицины и химии. Они ставили диагноз при обнаружении инфекционных заболеваний крови.

Экспертные системы были первым успешным продуктом такой научной области как искусственный интеллект, в то время переживавшей серьезный кризис. Требовался существенный прорыв в развитии практических приложений. Этот прорыв произошел, в то время, когда на смену поискам универсального алгоритма мышления и решения задач исследователям пришла идея моделировать конкретные знания специалистов-экспертов. Таким образом, появились первые коммерческие системы, основанные на знаниях, или экспертные системы (ЭС). Эти системы по праву стали первыми интеллектуальными системами, и до сих пор единственным критерием интеллектуальности является наличие механизмов работы со знаниями.

Таким образом, становится ясно, насколько может быть полезно применение экспертных систем. В частности, в случае, который рассматривается в данной работе, где крайне важна первая медицинская помощь, то есть в детском образовательном учреждении.

**Объектом исследования** - Муниципальное бюджетное образовательное учреждение дополнительного образования Станция юных техников г. Туапсе муниципального образования Туапсинский район.

**Предметом исследования** в настоящей работе выступает рабочее место медсестры.

**Целью** выпускной квалификационной работы является разработка прототипа экспертной системы, автоматизирующей процесс выявления заболеваний у работников организации.

**Задачи выпускной квалификационной работы:**

1. Спроектировать экспертную систему для объекта исследования;
2. Провести имитационное моделирование;
3. Реализовать прототип экспертной системы;
4. Провести анализ экономической эффективности.

Основная часть выпускной квалификационной работы содержит три главы:

1. Анализ предметной области. Эта глава содержит краткое описание объекта исследования, а также общую характеристику экспертных систем и их применение;

2. Методология разработки экспертной системы. В этой главе кратко рассматриваются известные методы решения поставленной задачи. Выбирается стратегия автоматизации, а также проектируется модель «ТО-ВЕ». Производится имитационное моделирование очередей, описывающих временные затраты до и после внедрения. Также кратко описывается среда CLIPS и её ключевые возможности;

3. Реализация экспертной системы. В этой главе описываются этапы разработки прототипа экспертной системы и обосновывается экономическая эффективность данной разработки.

# **1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

## **1.1 ОБЩАЯ ХАРАКТЕРИСТИКА ПРЕДПРИЯТИЯ МБОУ ДО СЮТ**

Преследуя цель развить внешкольную техническую деятельность в г. Туапсе Решением горисполкома с 1 сентября 1974 года была создана Станция юных техников. Являясь внешкольным учреждением технического профиля, СЮТ была призвана организовать досуг детей и подростков, способствовать обогащению их коммуникативной деятельности посредством организации первых кружков: авиамodelьного, радиоэлектроники и конструирования.

Наименование организации: Муниципальное бюджетное образовательное учреждение дополнительного образования Станция юных техников г. Туапсе муниципального образования Туапсинский район.

Ведомственная подчиненность: Учредителем и собственником имущества Организации является муниципальное образование Туапсинский район.

Форма собственности: Оперативное управление.

Организационно-правовая форма: учреждение.

Отрасль и вид деятельности: образовательная организация дополнительного образования.

Численность производственного персонала: 23 чел.

Размер: средний

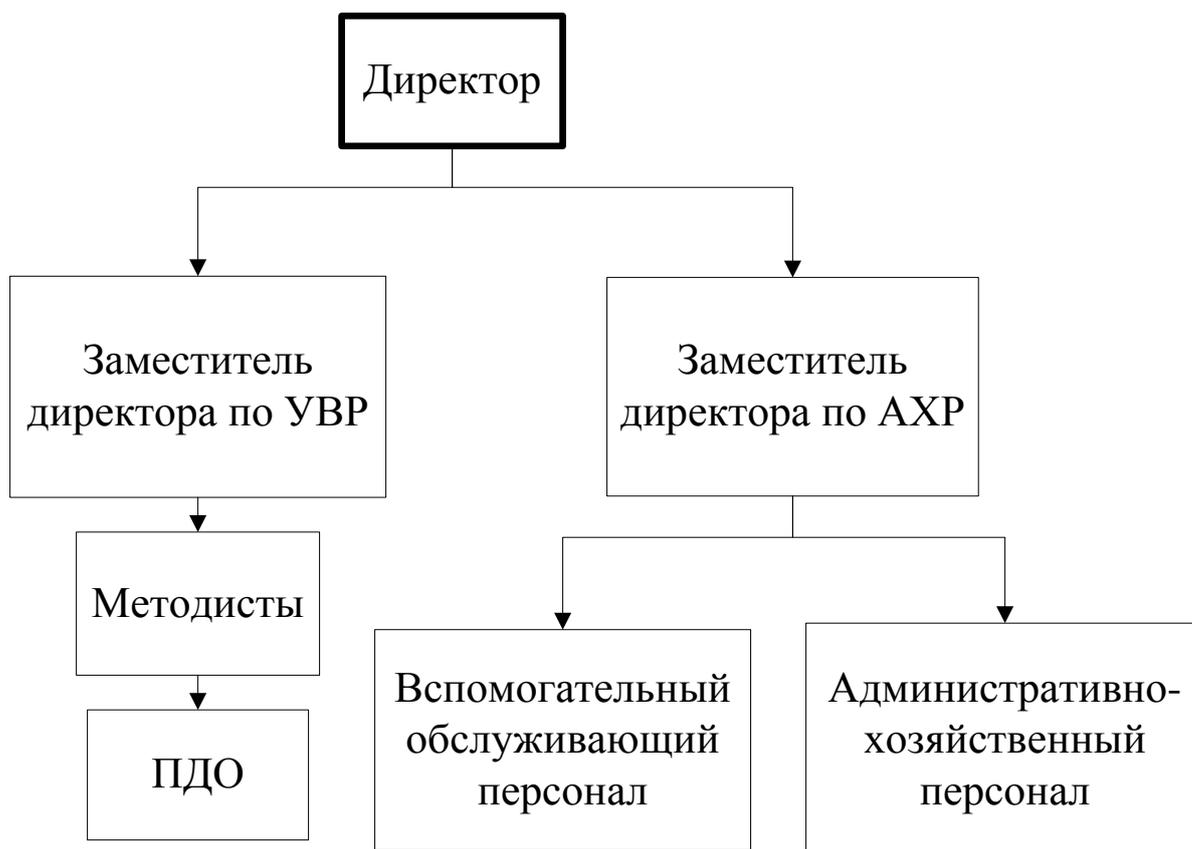
Должность руководителя: Директор.

Эффективность деятельности: Средняя.

Характеристика использования современных информационных технологий в решении задач управления:

Учет и контроль.

Производится учёт и контроль работ деятельности предприятия. Внутренняя система позволяет директору предприятия быстро узнавать и получать все отчеты, созданные внутри предприятия, так как используется сетевая передача данных.



**Рисунок 1 Организационно-управленческая структура предприятия**

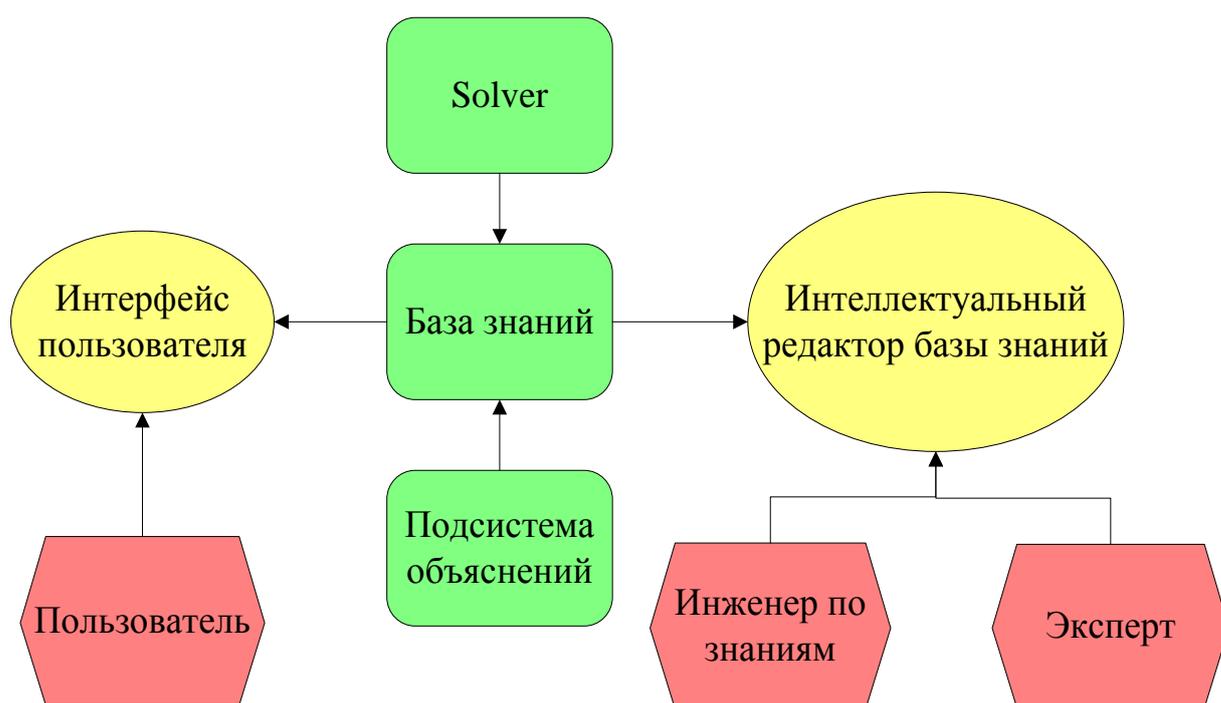
Организационная структура предприятия, представлена на рисунке 1, где УВР – учебно-воспитательная работа, АХР – административно-воспитательная работа, ПДО – педагог дополнительного образования.

На предприятии используется следующее программное обеспечение. Операционная система семейства Linux. Linux – это общее название Unix-подобных операционных систем, которые основаны на одноимённом ядре. LibreOffice – это офисный пакет, совместимый как с 32-битным, так и с 64-битными системами. Writer – это современный полнофункциональный текстовый редактор. Calc – является табличным процессором и визуальным редактором HTML. Impress – это программа для создания презентаций. Base – это специальная оболочка, предназначенная для работы с реляционными базами данных.

## 1.2 ОБЩАЯ ХАРАКТЕРИСТИКА ПРЕДПРИЯТИЯ

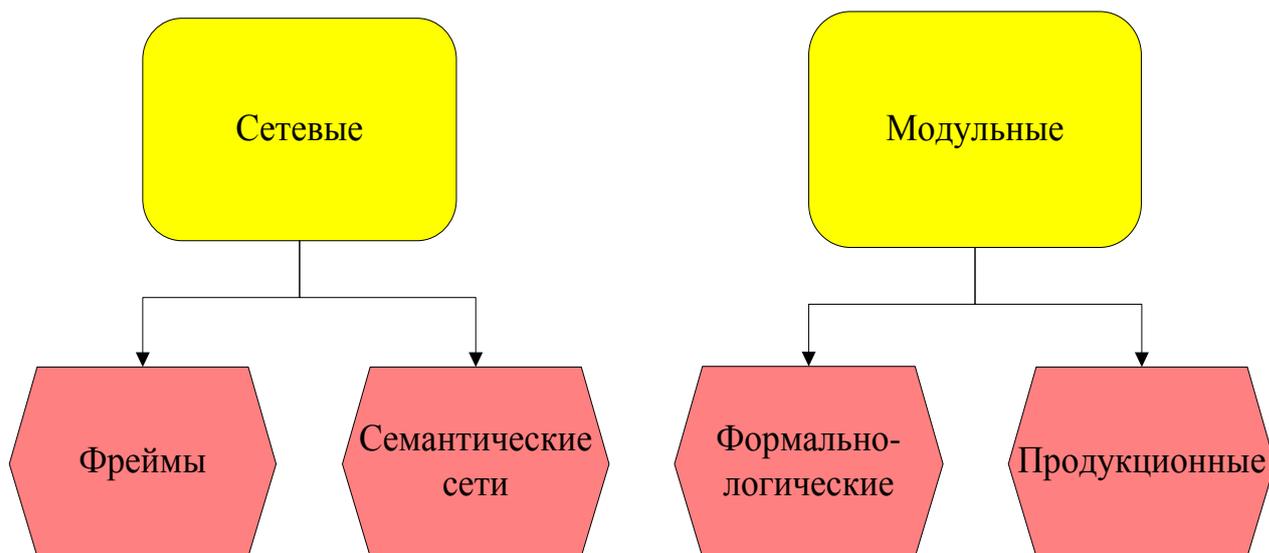
Экспертные системы - системы, основанные на знаниях. Ядро экспертной системы составляет база знаний, в которой описываются связи и закономерности предметной области, полученные в результате практической деятельности и профессионального опыта. Знаниями можно так же назвать хорошо структурированные данные, полученные из наблюдений и измерений.

Обобщённая структура экспертной системы показана на рисунке2, и состоит из интерфейса пользователя; solver, моделирующего ход рассуждений эксперта, на основании знаний имеющихся в базе знаний; базы знаний; интеллектуального редактора базы знаний, который позволяет инженеру по знаниям создавать базу знаний в диалоговом режиме.



**Рисунок 2 Обобщённая структура экспертных систем**

В базе знаний знания могут представляться по-разному. В основном эти представления можно разделить на два типа, которые показаны на рисунке 3:



**Рисунок 3 Модели представления знаний**

Фрейм – это абстрактный образ для представления стереотипа объекта, понятия или ситуации. Например, представлении объекта «комната», влечёт за собой образ: «жилое помещение с четырьмя стенами, полом, потолком, окнами, дверью, мебелью».

Семантическая сеть – это ориентированный граф, вершины которого – понятия, дуги – отношения между ними. Термин «семантическая» означает «смысловая», устанавливающая отношения между символами и объектами, которые они обозначают. Семантическая сеть является собой представлением долговременной памяти человека.

Формально-логические модели, основанные на классическом исчислении предикатов 1-го порядка, когда предметная область или задача описывается в виде набора аксиом. Реально исчисление предикатов 1-го порядка в промышленных экспертных системах практически не используется. Эта логическая модель применима в основном в исследовательских "игрушечных" системах, т. к. предъявляет очень высокие требования и ограничения к предметной области.

Продукционная модель – это модель представления, основанная на правиле «Если...<условие>, то...<действие>». Под <условием> (антецедентом) понимается некоторое предложение-образец, по которому

осуществляется поиск в базе знаний, а под <действием> (консеквентом) — действия, выполняемые при успешном исходе поиска.

Экспертные системы решают разнообразные задачи, основные из них это:

1. Интерпретация – процесс определения смысла данных;
2. Под диагностикой понимается соотнесение объекта с некоторым классом объектов и/или обнаружение неисправностей;
3. Мониторинг означает – непрерывная интерпретация данных в реальном времени и оповещение о выходе целевых параметров за допустимые пределы;
4. Проектирование — это процесс подготовки пакета документов – чертежей, пояснительных записок и т.д.;
5. Прогнозирование, позволяет предсказывать последствия некоторых событий или явлений на основании анализа имеющихся данных;
6. Планирование, составление планов действий или поиск уже имеющихся моделей, способные выполнять некоторые функции;
7. Обучение. Приобретение новых знаний у начинающих специалистов;
8. Управление – это функция организованной системы, поддерживать определённую деятельность;
9. Облегчение принятия решений, путём обеспечения человека необходимыми знаниями.

Примеров для каждого пункта может быть приведено достаточно много, например, для задач мониторинга – контроль за работой электростанций; для проектирования электронно-вычислительных машин (ЭВМ); предсказание погоды и прогнозы в экономике; планирование эксперимента; выбор стратегии выхода фирмы из кризисной ситуации. Как видно, экспертные системы могут решать широкий спектр задач. Но в данной работе будет разработан прототип экспертной системы способный определить распространённые острые респираторные вирусные инфекции (ОРВИ).

### 1.3 ПРИМЕНЕНИЕ ЭКСПЕРТНЫХ СИСТЕМ

Каждая организация нуждается в здоровом и работоспособном персонале, это особенно важно для организаций, работа которых связана с детьми и подростками. Таким образом, появляется необходимость контроля здоровья персонала. Для того, чтобы персонал был здоров, предусмотрены специальные процедуры медицинского осмотра, на которые выделяются деньги из государственного бюджета.

Процедура проходит таким образом, что каждый из работников организации в заданное время пребывает в назначенное место, после чего проходит всех медицинских специалистов. Со стороны организации весь процесс курируется медицинской сестрой работающей в самой организации в медицинском пункте. При прохождении этой процедуры между предприятием и клиникой используется преимущественно бумажный документооборот. Каждый работник, или группа работников посещает своего куратора в поликлинике, ответственного за сохранность документов каждого из представителей курируемого им работника организации. Далее каждый из работников, теперь уже пациентов клиники, проходит каждого врача-специалиста. Врачи делают отметки о состоянии здоровья, фиксируют отклонения, выдают рекомендации. После чего все документы возвращаются куратору, и выдаются по необходимости, например, в ситуации, когда работники не прошли всех специалистов. По итогам осмотра делается вывод о состоянии здоровья пациента.

Фактически времени не всегда достаточно для детального разбора жалоб каждого из пациентов, так как количество рабочего персонала в организации может переходить за несколько сотен, в рассматриваемой организации это несколько десятков. Эти временные пределы могла бы расширить экспертная система, не говоря уже о том, чтобы поспособствовать уточнению диагнозов, а также ускорению выздоровления, путём применения необходимых средств, по причине быстрой и точной постановки диагноза, что особенно важно в

сложных или осложняющихся ситуациях, или, например, исключить необходимость проходить тех или иных специалистов, так как экспертная система сможет определить отсутствие каких бы то ни было отклонений. Экономия времени и точность, это самые очевидные плюсы применения экспертных систем.

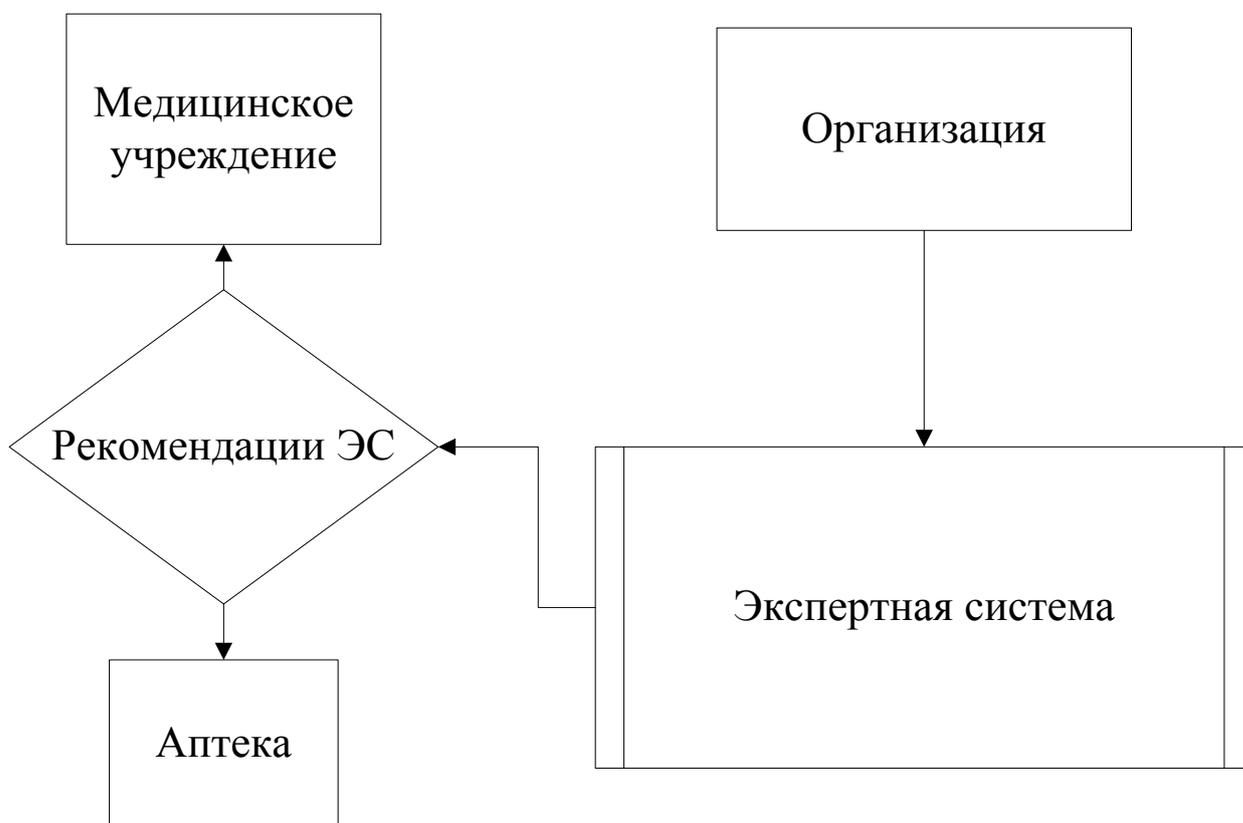
Экспертные системы на сегодняшний день играют очень важную роль в развитии здравоохранения в целом. Они служат для того, чтобы обеспечивать систематическую помощь медицинскому персоналу в случае возникновения спорных и проблемных ситуаций в вопросах лечения пациентов. Таким образом, главная задача экспертных систем — это выдача рекомендаций. Медицинские экспертные системы способны решать не только задачи диагностики, но и помогать в прогнозировании течения болезней, в выборе тактики их лечения. Тем не менее, работу любой экспертной системы не стоит осуществлять исключительно автоматической, другими словами, исключать человека из процесса принятия решения нельзя. Для медицинской экспертной системы, лицо принимающее решение (ЛПР), является в первую очередь терапевт. В рассматриваемой организации присутствует последний вариант, таким образом, формируется идея автоматизации рабочего места медсестры.

Для решения проблем экспертные системы, как правило, применяются:

1. Исходные данные, представленные экспертам, неоднозначны либо противоречивы.
2. Невозможно определить алгоритм однозначного решения задачи классическими методами.
3. Задача не может быть представлена в числовой форме.

То есть, если проблема характеризуется каким-либо (или несколькими) из вышеперечисленных факторов, то для её разрешения целесообразно применение экспертной системы. Медицинские проблемы, как правило, обладают именно такими свойствами, т.к. исходные данные представлены зачастую большим количеством запутанных и, возможно, противоречивых клинических данных. Использование экспертной системы должно помочь лицу,

принимающему решение, в данном случае это медсестра организации осуществить более точную диагностику. При проблемах со здоровьем, все, кто находится в организации могут обратиться к медицинской сестре, которая в свою очередь обращается к экспертной системе с целью получения рекомендаций. После чего, ЛПР, в данном случае это медицинская сестра, может принимать решение о том, стоит ли пациенту посетить медицинское учреждение или же достаточно посетить аптеку, для приобретения необходимых препаратов. На рисунке 4 условно изображена модель работы экспертной системы, и возможные варианты развития событий связанными с рекомендациями экспертной системы.



**Рисунок 4 Общая схема разрабатываемой системы**

## **2. МЕТОДОЛОГИЯ РАЗРАБОТКИ ЭКСПЕРТНОЙ СИСТЕМЫ**

### **2.1 СУЩЕСТВУЮЩИЕ ПОДХОДЫ ПО АВТОМАТИЗАЦИИ ПРОЦЕССА МЕДИЦИНСКОГО ОСМОТРА ПЕРСОНАЛА ОРГАНИЗАЦИИ**

Автоматизация процесса может проводиться в рамках онлайн-тестов. В такой ситуации проходимая пациентом процедура тестирования может не иметь реальной связи с клиникой, то есть результаты могут быть не использованы в дальнейшем.

Некоторые особо крупные компании разрабатывали экспертные системы для собственного использования. Это без сомнения принесло им большую выгоду. Например, повысило обучаемость начинающих специалистов, ускорило работу персонала, и качество предоставляемых услуг. Примером таких экспертных систем может служить MYCIN. Первая медицинская экспертная система, благодаря которой можно было диагностировать бактерии, которые вызывали тяжёлые инфекции, для рекомендации антибиотиков против них, в зависимости от массы тела пациента. MYCIN могла диагностировать заболевания связанные со свёртываемостью крови.

Существуют медицинские экспертные системы, такие как MYCIN, которая применяется в промышленности, также, как и на государственных объектах.

Недостатком существующей экспертной системы является высокая стоимость и специфичность выявляемых заболеваний.

### **2.2 ВЫБОР СТРАТЕГИИ АВТОМАТИЗАЦИИ**

При выборе стратегии главным является понимание поставленных целей. Так, главной целью является снижение временных и финансовых затрат на процесс прохождения медицинского осмотра работниками организации. Последовательность автоматизации стоит начать с обучения персонала, проводить это можно без отрыва от рабочего процесса, для чего стоит

установить одну копию внедряемой системы на рабочий компьютер. В первую очередь обучение персонала должно быть направлено на получение теоретических знаний о внедряемом решении и о том, как оно изменит целевой процесс в целом. После чего необходима направленная тренировка умения работать с экспертной системой.

На следующем этапе в реализации стратегии будет установка рабочей версии экспертной системы. Стоит применить такой подход, при котором старая (ручная) и новая системы работали параллельно, при этом выходные документы каждой из них сравнивались. Это позволит отрегулировать работу системы для дальнейшей эксплуатации в будущем.

Последним этапом будет сопровождение экспертной системы. Для этого необходимо что бы был назначен человек в самом предприятии, в таком случае необходимо дополнительное обучение для этого работника, в то же время можно обращаться за поддержкой к самим разработчикам экспертной системы. Хотя необходимость обновления базы знаний может отпасть на предыдущем этапе, из-за параллельного способа внедрения. Тем не менее стоит иметь ввиду что научный прогресс не стоит на месте, могут находиться новые заболевания, методы их диагностики, могут меняться методы диагностики уже известных заболеваний, что делает проблему актуальной.

### **2.3 ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ПРОЦЕССА ПРОЕКТИРОВАНИЯ**

Для проектирования экспертной системы были использованы такие программные продукты как IBM Rational Rose, ERWin, MS Project, AnyLogic.

RationalRose – это CASE-средство фирмы Rational Software Corporation, позволяющее автоматизировать процесс разработки и проектирования программного обеспечения. В среде RationalRose также существует возможность генерации программного кода, архитектура которого должна быть описана в самой среде, при помощи языка UML (Unified Modeling Language).

Этот язык отлично справляется с описанием объектно-ориентированного взгляда на программное обеспечение, анализа и проектирования, в свою очередь позволяющий общаться на общем языке проектировщикам и разработчикам программного обеспечения.

Благодаря RationalRose можно решать следующие задачи, которые возникают в команде разработчиков:

1. Проектирование систем, от простых до сложных;
2. Предоставлять развёрнутое представление о проекте;
3. Производить код, при помощи механизма кодогенерации;
4. Проводить обратное проектирование имеющихся систем, то есть фактически проводить так называемый reverse engineering;
5. Автоматически проверять на соответствие две модели.

ER Win Process Modeler, также как Rational Rose, является средством для проектирования программного обеспечения, т.е. это case-средство, помогающее создавать так называемые IDEF0, IDEF3 и DFD диаграммы.

IDEF0 представляет собой функции моделирования для описания производственных функций, который предлагает функциональный язык моделирования для анализа, разработки, реинжиниринга и интеграции информационных систем; деловые процессы; или программного обеспечения инженерного анализа.

IDEF3 от английского Integrated DEFinition for Process Description Capture Method. Методология моделирования и стандарт документирования процессов, происходящих в системе. Метод документирования технологических процессов представляет собой механизм документирования и сбора информации о процессах.

Microsoft Project – программа, разработанная для управления проектами. Microsoft Project создаёт расписания критического пути. Расписания могут быть составлены с учётом используемых ресурсов. Главным способом представления проекта является диаграмма Ганта. Благодаря чему можно легко разработать план проекта, представить его в понятной форме для других членов

команды, а также для демонстрации потенциальному заказчику.

AnyLogic – это программа, предназначенная для имитационного моделирования. Этот продукт поддерживает три методики моделирования:

1. Системная динамика;
2. Процессное моделирование;
3. Агентное моделирование.

В то время как для проектирования применяются такие продукты как RationalRose, ERWin, AnyLogic, для разработки экспертной системы будет использована среда CLIPS. Аббревиатура CLIPS расшифровывается как – C Language Integrated Production System; язык C, интегрированный с производственными системами. CLIPS это современный инструмент, состоит из оболочки со своим способом представления знаний, гибкого и мощного языка представления знаний. CLIPS это сочетание языка C и LISP.

## **2.4 ПОСТРОЕНИЕ МОДЕЛИ «ТО-ВЕ»**

Модель «ТО-ВЕ» призвана для того, чтобы демонстрировать то, как будет выглядеть разрабатываемая система в итоге. Целью является разработка экспертной системы, для этого стоит знать об основных этапах проектирования.

Выбор подходящей проблемы, определяет деятельность, предшествующую решению начать разрабатывать экспертную систему.

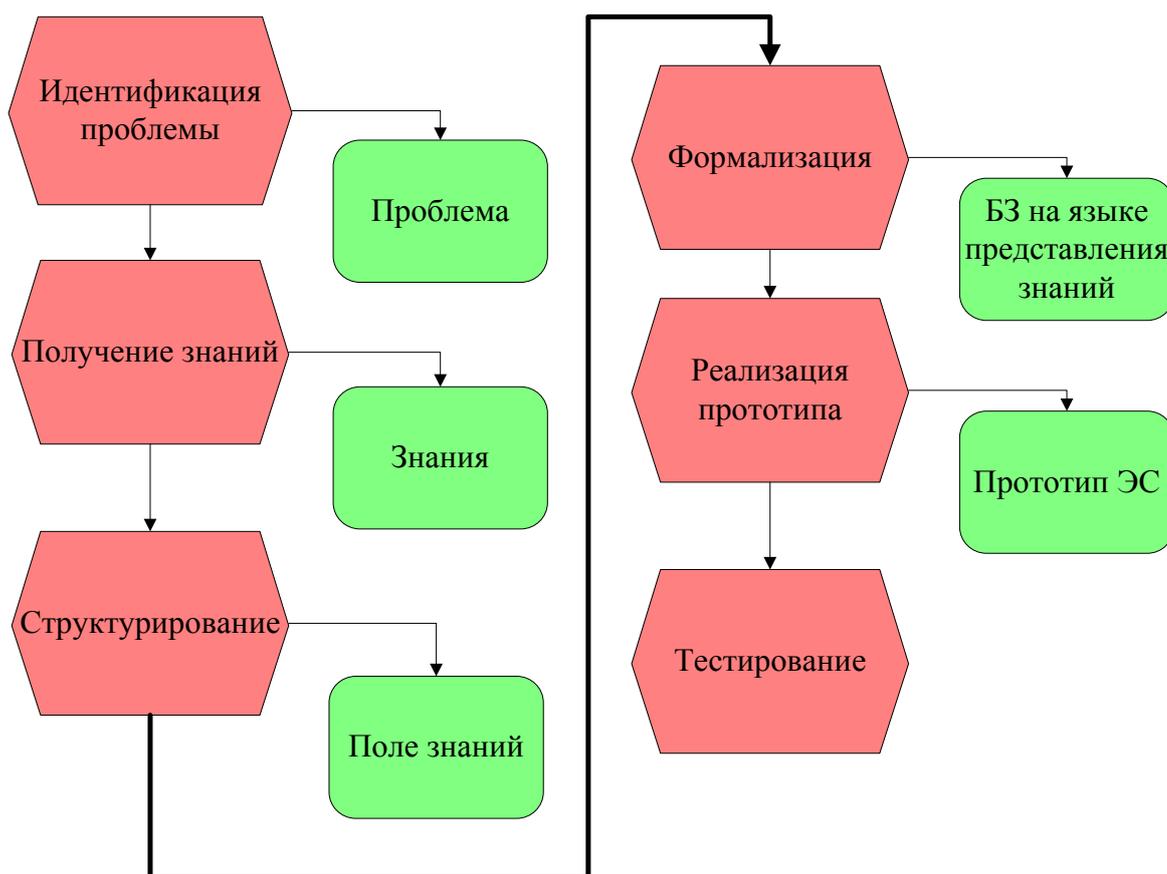
Этот этап можно так же разделить на такие пункты:

1. Определение проблемной области;
2. Поиск эксперта, желающего сотрудничать;
3. Анализ расходов и прибылей от разработки;
4. Подготовку подробного плана разработки.

Правильный выбор проблемы – это наиболее важная часть разработки. Если выбрать неподходящую проблему, можно остаться на этапе проектирования, по причине того, что никто не знает, как решать выбранную проблему. В рамках данной работы решается проблема проектирования

экспертной системы для диагностирования здоровья рабочих организации.

На рисунке 5 изображены все шесть стадий разработки прототипа экспертной системы, а также, в зелёных блоках, указаны результаты каждого из этапов. Результатом же последнего этапа, тестирования, является сама экспертная система. Рассмотрим каждый из этапов более подробно.



**Рисунок 5 Стадии разработки прототипа экспертной системы**

Идентификация проблемы – знакомство и обучение членов команды разработчиков, создание спецификации задачи, в том числе неформальной.

Извлечение знаний – это процесс переноса компетентности от эксперта к инженеру по знаниям с использованием различных методов, например, анализ тестов, диалоги, экспертные игры, лекции, дискуссии, интервью, наблюдение.

Это процесс, который состоит из большого количества серий систематических интервью, продолжающихся в течение нескольких месяцев.

Инженер должен работать вместе с экспертом и наблюдать, как он работает в конкретной ситуации. Очень эффективным оказывается подход, когда эксперту задаются вопросы о его методике решения задач напрямую. Обычно эксперты испытывают трудности при ответе на такие вопросы, поэтому этот аспект называется парадоксом экспертизы.

**Таблица 1 – Методы получения знаний**

<b>Метод</b>	<b>Описание</b>
Наблюдение на рабочем месте	Протоколирование хода решения задачи
Обсуждение задач	Выявить виды данных, знаний и процедур, необходимых для решения задачи
Описание задач	Просят эксперта написать прототипную задачу на каждую категорию ответов
Анализ задачи	Представить эксперту ряд реальных задач для решения вслух с целью выявления логики рассуждений
Доводка системы	Попросить эксперта представить несколько задач для решения с использованием правил, выявленных в интервью
Оценивание системы	Попросить эксперта проверить работу системы и подвергнуть критике правила и структуру управления ЭС
Проверка системы	Дать примеры, решенные экспертом и ЭС, другим экспертам для оценки

Самым узким местом при разработке экспертных систем, является приобретение знаний от эксперта, и дальнейшее структурирование этих знаний в удобную форму. Инженер получает все возможные знания от экспертов из интервью. Надо понимать, что интервью — это не просто беседа, это еще и

наблюдение за экспертом, как он решает практические задачи из рассматриваемой области. В таком случае стоит вести «протоколы рассуждения вслух». В таблице 1 приведены наиболее распространённые методы извлечения знаний от эксперта и их описание:

1. Наблюдение на рабочем месте. Здесь инженер по знаниям пассивно наблюдает, как эксперт решает задачу на своем рабочем месте. Это дает ему представление о сложности проблемы, подсказывает характер диалоговых возможностей. Этот метод может оказаться непригодным из-за необходимых затрат времени или соображений профессиональной этики.

2. Обсуждение задач. Инженер по знаниям выбирает некоторое множество задач, соответствующих рассматриваемой области, и в свободной неформальной манере обсуждает их с экспертом. Цель обсуждения – определить, как эксперт организует знания о каждой задаче, как представляет понятия и гипотезы, как обращается с неполными, неточными и противоречивыми данными. Обычно задаются следующие вопросы: *«Чем отличается данная задача от типичных задач рассматриваемой предметной области.»*. *«Какие способы решений соответствуют этой задаче?»*. *«Можно ли рассматриваемую экспертом задачу разбить на несвязанные друг с другом подзадачи?»*. *«Какие знания необходимы для решения задачи?»*. *«Что может выступить в качестве объяснения или обоснования решения задачи?»*.

3. Описание задач. Эксперт даёт инженеру по знаниям описание типичной задачи для каждой категории возможных ответов. Таким образом, появляется возможность определить примерные задачи для каждой категории ответов. То есть некоторую конструкцию, которую ЭС может использовать при выборе стратегии к решению задачи. Этот подход особенно интересен для задач диагностического поиска.

4. Анализ задач. Инженер по знаниям предлагает эксперту решить типичные для рассматриваемой области задачи. Инженер должен постичь стратегию решения, используемую экспертом.

5. Доводка системы. Эксперт дает инженеру задачи, сначала простые потом весьма трудные. До тех пор, пока ЭС не начала работать инженер решает задачи самостоятельно, при помощи полученных знаний. Это позволяет провести быструю проверку полноты и непротиворечивости знаний, извлеченных от экспертов. Но как только ЭС заработает, инженер должен решать задачи с ее помощью.

6. Оценивание системы. Эксперт анализирует и критически оценивает каждое правило ЭС, проверяет стратегии управления, используемые для выбора правил. Эксперт сравнивает стратегию управления в ЭС со своим методом решения.

7. Проверка системы. Инженер по знаниям предоставляет примеры, решенные экспертом и прототипной версией ЭС другим экспертам. Это позволяет сравнивать стратегии различных экспертов и находить важнейшие моменты, вызывающие разногласия.

Существуют так же условия работы эксперта и инженера по знаниям. Например, оба участника должны быть тщательно подготовлены. Сам эксперт должен быть не только отличным и компетентным специалистом, но и быть заинтересованным в построении экспертной системы. Эксперт также должен уметь объяснять свои знания, особенным преимуществом было бы, если бы эксперт имел преподавательский опыт.

Аналитику в свою очередь необходимо глубоко познакомиться со специальной литературой по предметной области, чтобы не задавать слишком «глупых» вопросов, а также уметь слушать и грамотно задавать вопросы. Инженер по знаниям должен настроиться на роль «ученика», а не «экзаменатора».

Структурирование знаний – процесс выявления структуры полученных знаний о предметной области: терминология; список основных понятий и их атрибутов; отношения между понятиями; структура входной и выходной информации; стратегия принятия решений.

На стадии формализации строится представление концепция предметной области на основе выбранного языка представления знаний. Для этого используются: логические методы, продукционные модели, семантические сети, фреймы, объектно-ориентированные языки.

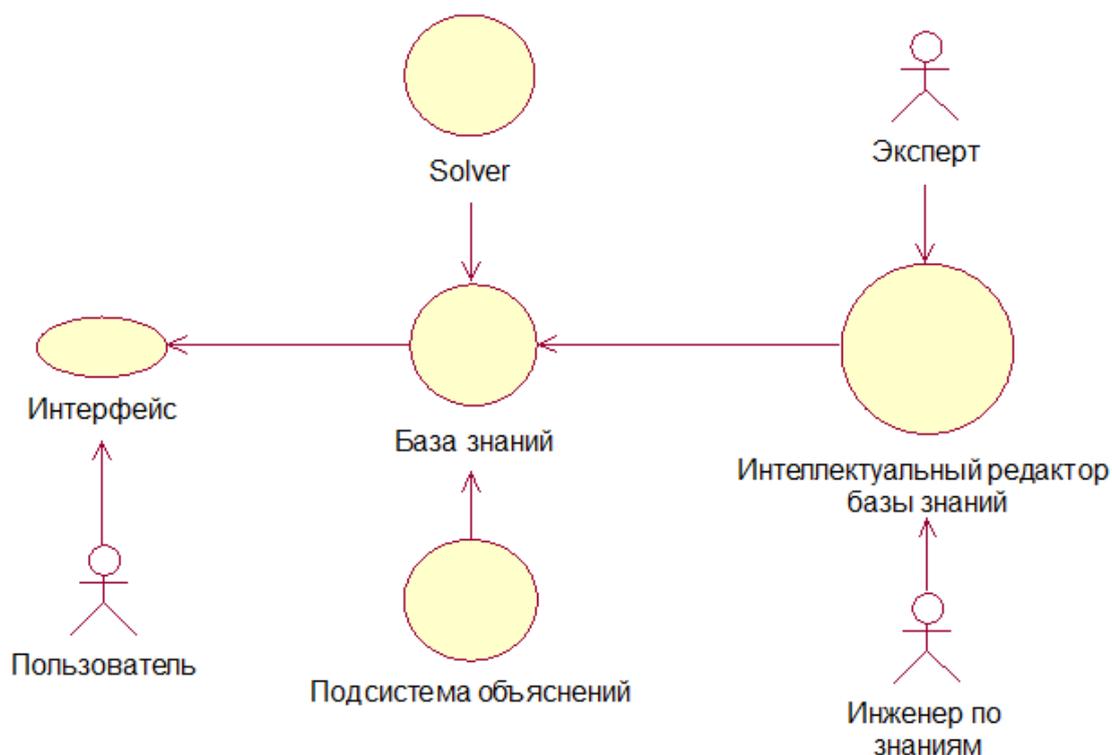
Реализацию прототипа, ещё можно назвать программной реализацией. Это разработка программного комплекса, демонстрирующего жизнеспособность подхода в целом.

Разработка прототипа, является важным моментом в разработке экспертной системы. Прототипную версию можно использовать не только для того чтобы убедиться в правильности кодирования фактов, но и привлечения экспертов данной области, и что самое важное заинтересовать потенциального заказчика.

Последний этап – тестирование. Тестирование – это процесс выявления ошибок в реализации прототипа и выработка рекомендаций по доводке системы до промышленного варианта.

Для демонстрации проектирования разрабатываемой системы будут использованы IBM RationalRose и ERWin Process Modeler. В первом случае используется диаграмма UML для многостороннего рассмотрения предлагаемой системы. Помимо этого, будет применено средство моделирования AnyLogic, целью которого будет демонстрация снижения нагрузок, связанных с обработкой одного клиента в больнице.

На рисунке 6 изображена диаграмма вариантов использования, которая описывает сценарий применения разрабатываемой системы. Три центральных элемента – solver, knowledgebase, explanation system – это решатель, база знаний, подсистема объяснений.



**Рисунок 6 Диаграмма вариантов использования**

Решатель (solver) предназначен для того, чтобы, используя исходные данные из рабочей памяти и знания из базы знаний, формировать такую последовательность правил, которые, применяясь к исходным данным, позволяют решать необходимую задачу.

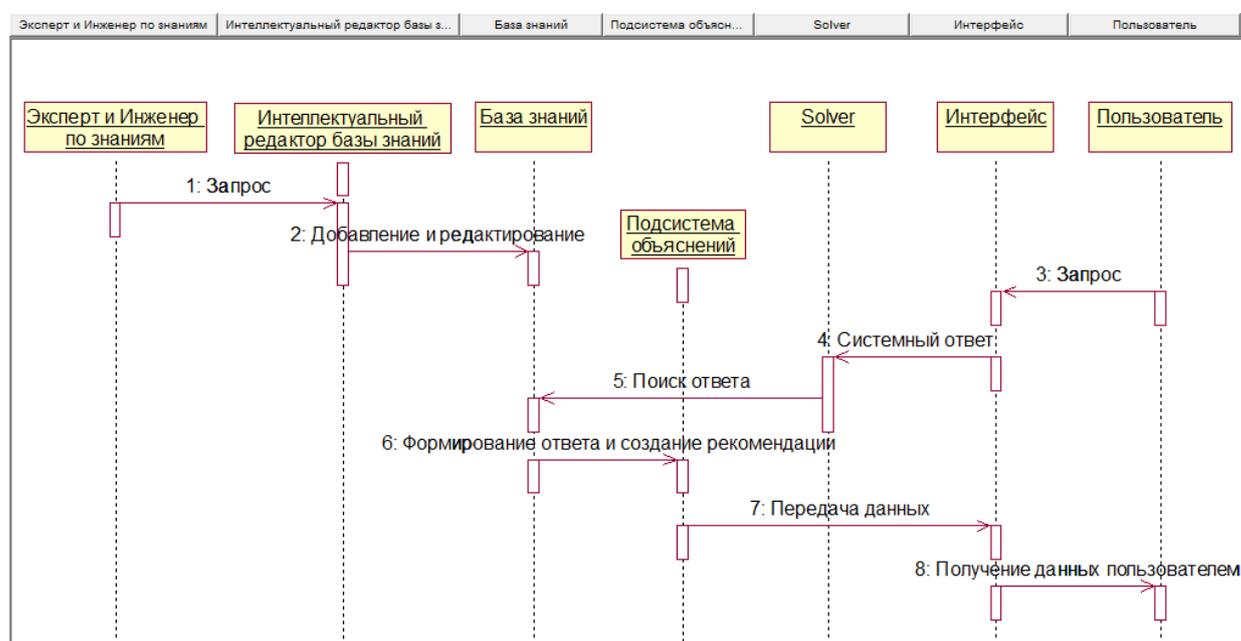
Подсистема объяснений (explanation system) интерпретирует в доступную для пользователя форму методы решения задачи или принятия системой определенного решения. Кроме того, он выполняет функции объяснения порядка использования данных, необходимых для принятия решения.

База знаний (knowledge base) необходима для хранения долгосрочных данных, описывающих рассматриваемую область, и правил, описывающих необходимые действия над данными этой области.

Interface – это интерфейс экспертной системы, это программное приложение, написанное на языке высоко уровня, позволяющее проще взаимодействовать с самой экспертной системой.

Intelligent Knowledge Base Editor – интеллектуальный редактор баз

знаний, позволяет дополнять базу знаний новыми знаниями, полученными от эксперта.



**Рисунок 7** Диаграмма последовательности действий

В целом процесс функционирования ЭС можно представить следующим образом: пользователь, желающий получить необходимую информацию, через пользовательский интерфейс посылает запрос к ЭС; решатель(solver), пользуясь базой знаний, генерирует и выдает пользователю подходящую рекомендацию, объясняя ход своих рассуждений при помощи подсистемы объяснений. Изображение этого сценария можно увидеть на рисунке 7.

Также для проектирования использовано такое средство как ERWin Process Modeler. С его помощью можно детально описать процесс создания целевой системы. В общем случае это будет выглядеть так, как показано на рисунке 8.

Для создания экспертной системы необходимы:

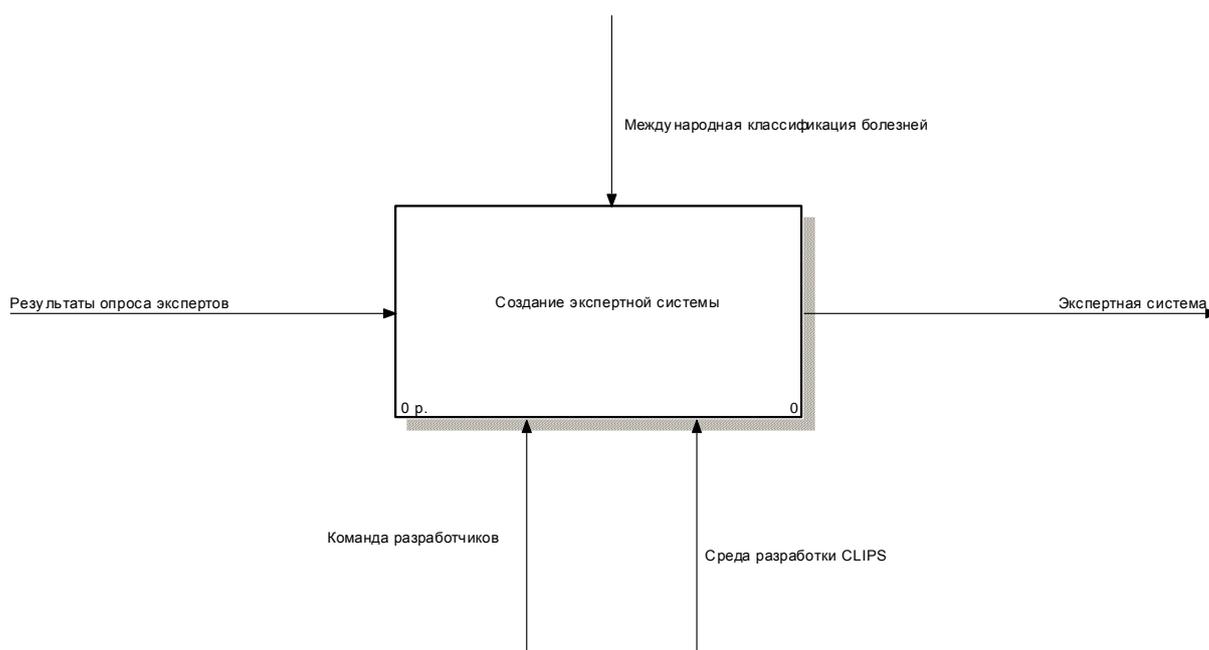
1. Результаты опросов экспертов, так как именно эти знания имеют наибольшую важность для создания базы знаний, и ценность при разработке и выдаче точных и правдивых результатов;
2. Международная классификация болезней – это особый документ,

который выступает как ведущая классификационная основа в здравоохранении;

3. Команда разработчиков, в которую обязательно входят инженер по знаниям, программист и тестировщик, а также к разработчикам можно отнести специалиста-эксперта;

4. Среда разработки CLIPS, в среде будет производиться существенная часть создания экспертной системы.

Руководствуясь планом разработки и используя полученные от экспертов знания, в CLIPS будет реализовано создание экспертной системы.

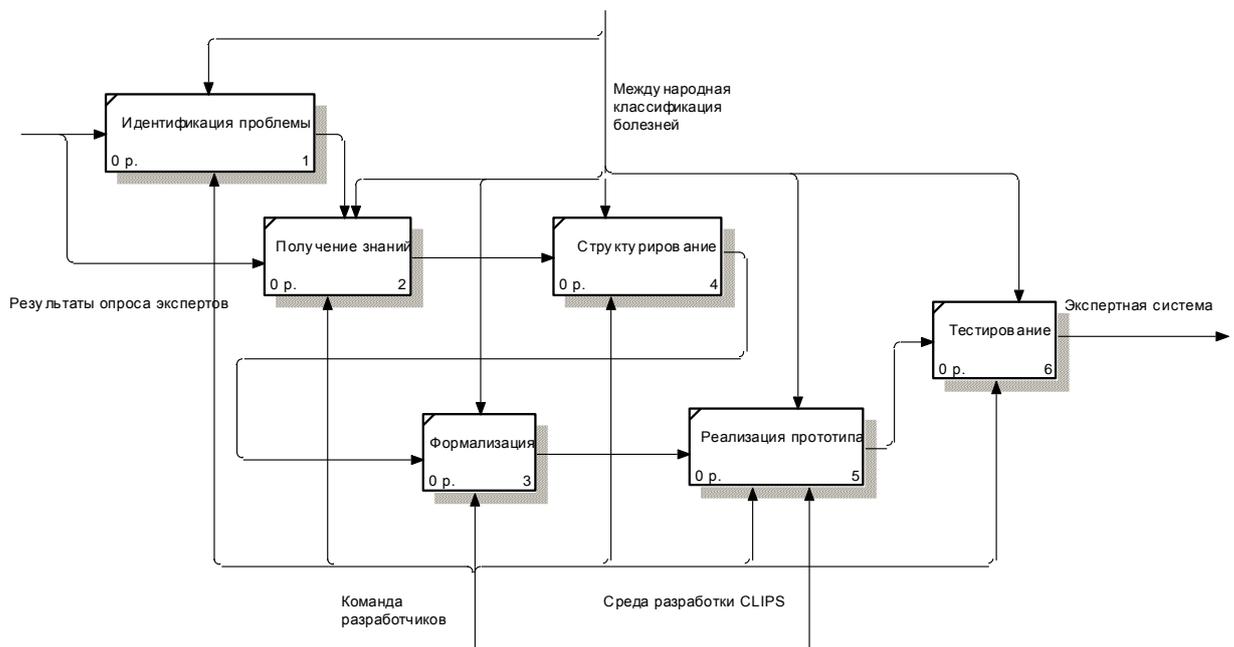


**Рисунок 8 Модель разработки экспертной системы**

На рисунке 9 изображена декомпозиция, благодаря которой можно понять роль каждого из участников команды разработчиков экспертной системы. Для идентификации проблемы нужно вмешательство такого управляющего элемента как инженер по знаниям, участие которого составляет больше половины всего процесса разработки. Для идентификации проблемы так же необходимо знать, какие вопросы существуют на данный момент в рассматриваемой предметной области. После определения проблемы, данные о которой будут выступать в качестве управления для всего дальнейшего

процесса, инженер по знаниям приступает к процессу получения знаний от экспертов, где в дальнейшем и происходит работа с этими полученными знаниями. Таким образом, каждый последующий процесс будет получать обработанные инженером по знаниям результаты, которые в итоге поступают к программисту и используются для создания прототипа.

Финальным этапом является тестирование и работа тестировщика, которая, в данном случае, не нуждается в специальном программном обеспечении, т.к. все тесты можно провести вручную.



**Рисунок 9** Декомпозиция модели разработки экспертной системы

## 2.5 ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ПРОЦЕССА МЕДОСМОТРА

Важным результатом внедрения системы является уменьшение временных затрат. Для демонстрации этого было применено моделирование системы при помощи AnyLogic. В обычных условиях работы предприятия, все работники проходят медосмотр имея записи в личных делах, которые хранятся в больнице. В этих делах есть история болезней, если такие наблюдались, но

реальная картинка может отличаться от того что происходит с человеком на данный момент.

Внедряемая экспертная система разрабатывается с целью предупредить возможные болезни или снять подозрения, чтобы при посещении врачей, работник имел рекомендации, которые он получил при работе с системой. Медицинский специалист использует полученные рекомендации от экспертной системы, это заменяет ему обычный опрос пациента, тем самым экономя время.

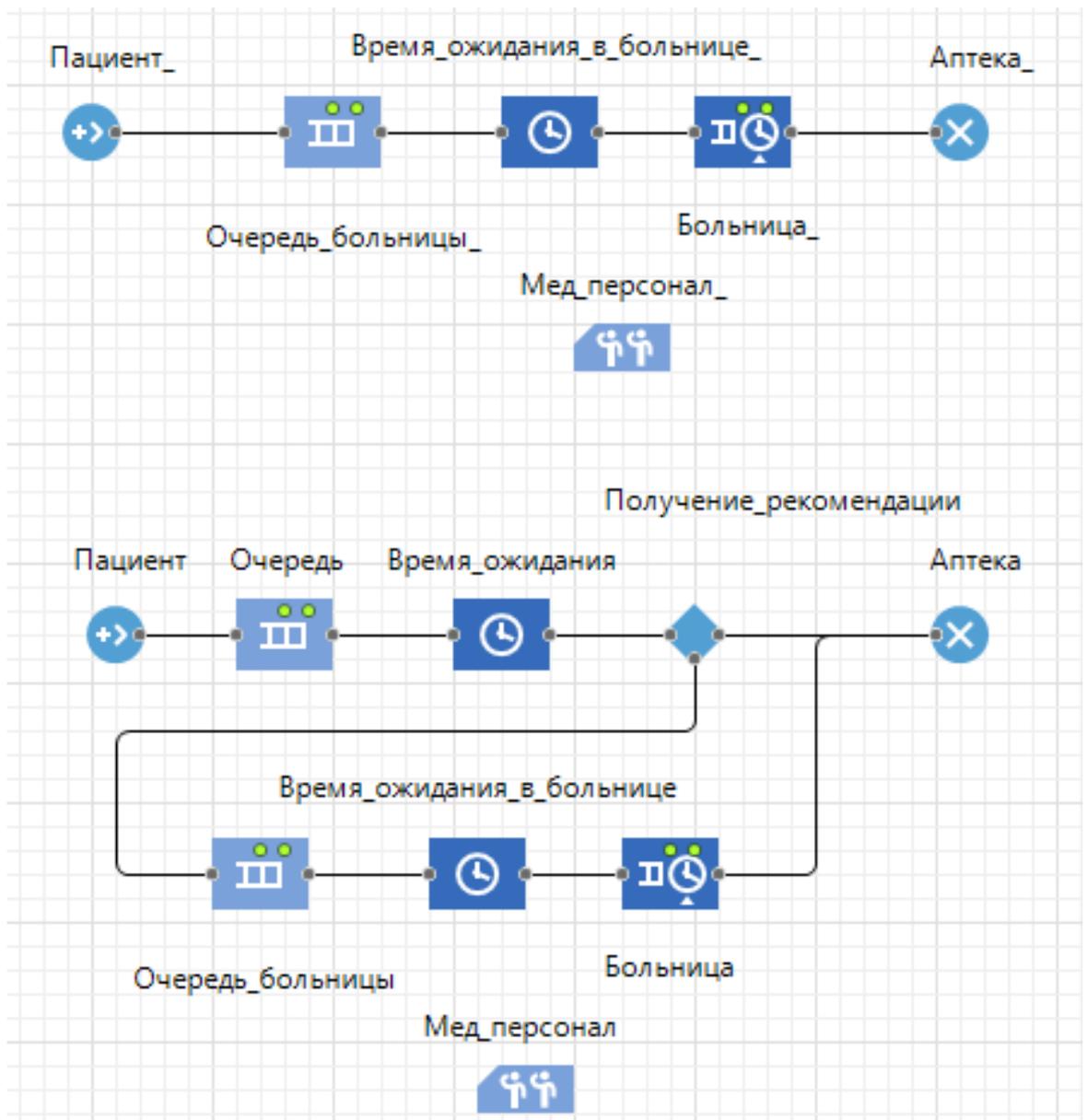


Рисунок 10 Моделирование очереди

На рисунке 10 изображены схемы моделирующие очереди работы больницы вверху. Та же модель, но с уже внедрённой в процесс медицинского осмотра, внизу. В схеме выше элемент Пациент\_ имеет частоту прибытия 0,5 в минуту, в то время как на нижней 0,3. Дело в том, что пациенты посещают больницу организованно, в соответствии с предписанным протоколом, обязующим всех членов муниципальных организаций и прочих образований проходить медицинский осмотр, в то время как медицинской экспертной системой могут воспользоваться для предупреждения проблем, связанных со здоровьем, которые могли бы быть спровоцированы деятельностью работника и/или несчастным случаем. Для больниц, как на верхней схеме, так и на схеме ниже, вместимость очереди равна 100, время же нахождения в очереди на схеме с экспертной системой меньше. Так, в больнице осмотр с пациентом занимает от 7 до 12 минут, а при использовании экспертной системы это время снижается до 5. Само тестирование с применением экспертной системы занимает от 2 до 5 минут. В обоих случаях работнику следует посетить аптеку.



**Рисунок 11 Запущенная модель очередей**

На рисунке 11 можно увидеть результаты работы модели до внедрения, и после внедрения экспертной системы. Соответственно, модель выше, это модель, описывающая работу больницы до внедрения экспертной системы, а

модель ниже - после. В то время как на выходе модели «AS-IS» наблюдается только 16 выпущенных пациентов, в модели «TO-BE» выпущенных пациентов уже 43. Очереди обеих моделей не перегружаются, но временные ограничения в модели «TO-BE» меньше, благодаря получаемым рекомендациям от экспертной системы. Так же очевидны задержки пациентов на разных этапах, самым частым из которых является опрос пациента врачом. При этом в модели «AS-IS» очереди больницы всегда полны, в то время как в модели «TO-BE» очередь больницы свободна, и врачи работают свободно.

## 2.6 ОПИСАНИЕ СРЕДЫ РАЗРАБОТКИ CLIPS

Для реализации экспертной системы была выбрана среда CLIPS, которая обладает большими возможностями. Так синтаксис языка можно разбить на три основные группы:

1. Примитивные типы данных;
2. Функции, использующиеся для обработки данных;
3. Конструкторы, необходимые для написания таких структур, как факты, правила, классы и т.д.

Основные элементы языка CLIPS можно разбить на три основных группы элементов, предназначенных для написания программ: примитивные типы данных; функции, использующиеся для обработки данных; конструкторы, предназначенные для создания таких структур языка, как факты, правила, классы и т.д.

CLIPS поддерживает восемь примитивных типов данных: float, integer, symbol, string, external-address, fact-address, instance-name, instance-address.

Для реализации прототипа необходимо небольшое количество знаний о среде, для начала стоит разобраться с понятием функции. Функцией в CLIPS называется часть кода, имеющая имя и возвращающая полезный результат или выполняющая полезные действия (например, отображение информации на

экране). Функции, не возвращающие результат и выполняющие некоторую полезную работу, как правило, называются командами.

CLIPS оперирует с несколькими типами функций — определенные пользователем внешние функции, системные (внутренние) функции, функции, определенные в среде CLIPS с помощью конструктора `deffunction`, родовые функции.

В CLIPS конструкторы выполняют действия отличные от тех, которые выполняют функции. В отличие от функций, конструкторы предназначены для добавления в базу знаний новых элементов. Конструкторы также не возвращают никаких значений.

В CLIPS определены следующие конструкторы: `defmodule`, `defrule`, `deffacts`, `deftemplate`, `defglobal`, `deffunction`, `defclass`, `definstances`, `defmessage-handler`, `defgenericidefmethod`. Вызовы всех конструкторов заключаются в круглые скобки. Конструкторы отличаются от встроенных функций по выполняемым ими действиям. Как правило, функции не меняют состояние базы знаний среды CLIPS (за некоторым исключением, например, функций, очищающих среду или загружающих на выполнение некоторый файл). Конструкторы, наоборот, предназначены для добавления в базу знаний новых элементов. Кроме того, в отличие от функций, конструкторы не возвращают никаких значений.

Для представления данных CLIPS использует три основных абстракции данных: факты, объекты и глобальные переменные.

Факты — одна из основных форм представления информации в CLIPS. Факты являются фундаментальным понятием теории экспертных систем и предназначены для использования в правилах системы. Каждый факт представляет фрагмент данных, помещенных в текущий список фактов системы (рабочую память).

CLIPS поддерживает как эвристическую, так и процедурную парадигму представления знаний.

Одним из основных методов представления знаний в CLIPS являются правила. Правила используются для представления эвристик или эмпирических

правил, определяющих действия, которые необходимо выполнить в случае возникновения некоторой ситуации. Разработчик экспертной системы создает набор правил, которые, работая вместе, решают поставленную задачу. Правила состоят из предпосылок и следствия. Предпосылки называются также ЕСЛИ-частью правила или LHSправила (left-hand side). Следствие называется ТО-частью правила или RHSправила (right-hand side).

Помимо эвристической, CLIPS поддерживает и процедурную парадигму представления знаний, используемую в большинстве языков программирования. Конструкторы `deffunction` и `defgeneric` позволяют пользователю определять новые выполняемые конструкции непосредственно в среде CLIPS, возвращающие некоторые значения или выполняющие какие-то полезные действия. Функции, родовые функции и обработчики сообщений представляют собой отрезки кода, заданного пользователем и выполняемого, в случае необходимости, интерпретатором CLIPS.

Для функционирования любой экспертной системы критически важным является наличие базы знаний. Об этом говорит даже тот факт, что в последнее время все чаще термин «система, основанная на знаниях» употребляется в качестве синонима термина «экспертная система». Как правило, в любой экспертной системе знания представляются фактами и правилами, заданными на некотором языке описания знаний. CLIPS не является исключением и предоставляет возможности для приобретения, хранения и обработки фактов и правил.

Для создания неупорядоченных фактов в CLIPS будет использоваться специальный конструктор `deftemplate`. Помимо конструктора `deftemplate`, CLIPS предоставляет конструктор `deffacts`, также предназначенный для работы с фактами. Данный конструктор позволяет определять список фактов, которые будут автоматически добавляться всякий раз после выполнения команды `reset`, очищающей текущий список фактов. Факты, добавленные с помощью конструктора `deffacts` могут использоваться и удаляться так же, как и любые другие

факты, добавленные в базу знаний пользователем или программой, с помощью команды `assert`.

Функция `assert` - одна из наиболее часто применимых команд в системе CLIPS. Без использования этой команды нельзя написать даже самую простую экспертную систему. Каждым вызовом этой функции можно добавить произвольное число фактов. Вывод фактов в данном случае производится командой (`facts`), которая выводит сумму двух чисел 3 и 4 тоже в списке фактов, как показано на рисунке 12.

```
CLIPS> (assert (color red))
<Fact-1>
CLIPS> (assert (color blue) (value (+ 3 4)))
<Fact-3>
CLIPS> (facts)
f-0      (initial-fact)
f-1      (color red)
f-2      (color blue)
f-3      (value 7)
For a total of 4 facts.
CLIPS> █
```

**Рисунок 12 Ввод фактов память и вывод на экран**

Функция `clear` полностью очищает систему, т. е. удаляет все правила, факты и прочие объекты базы знаний CLIPS, добавленные конструкторами, приводит систему в начальное состояние, необходимое для каждой новой программы. Функция `reset`, как уже упоминалось ранее, очищает список фактов и заносит в него факт (`initial-fact`), что очень важно для нормального функционирования программы. И, наконец, функция `run` запускает механизм логического вывода и приводит программу в движение.

Чтобы запустить программу на выполнение еще раз, достаточно вызвать функции (`reset`) и (`run`). Эти функции можно вводить с клавиатуры, кроме того, они доступны в меню `Execution` и имеют «горячие» клавиши `Ctrl+E` и `Ctrl+R` соответственно. Создавать правила конструктором `defrule` каждый раз, по мере необходимости используя для этого среду CLIPS, довольно неудобно.

Для облегчения участи пользователя CLIPS позволяет загружать конструкторы правил (как, впрочем, и все остальные конструкторы) из текстового файла. Для этого используется следующая команда:(load<имя-файла>).

Кроме вышеперечисленных конструкторов существует ещё конструктор `deffunction`, который позволяет пользователю создавать новые функции непосредственно в среде CLIPS. Способ вызова функций, определенных пользователем, эквивалентен способу вызова внутренних функций CLIPS. Вызов функции осуществляется по имени, заданному пользователю. За именем функции следует список необходимых аргументов, отделенный одним пробелом. Последовательность действий определенной с помощью конструктора `deffunction` функции исполняется интерпретатором CLIPS.

Синтаксис конструктора `deffunction` включает в себя 5 элементов:

1. имя функции;
2. необязательные комментарии;
3. список из нуля или более параметров;
4. необязательный символ групповых параметров для указания того, что функция может иметь переменное число аргументов;
5. последовательность действий или выражений, которые будут выполнены (вычислены) по порядку в момент вызова функции.

### **3. РЕАЛИЗАЦИЯ ЭКСПЕРТНОЙ СИСТЕМЫ**

#### **3.1 РАЗРАБОТКА ЭКСПЕРТНОЙ СИСТЕМЫ**

Разработка экспертной системы включает следующие этапы:

1. Определение проблемы, которую будет решать разрабатываемая экспертная система. Как уже стало ясно это экспертная система медицинского типа, а значит вопросы будут связаны со здоровьем пользователя. Проблемной областью были выбраны острые респираторные вирусные инфекции, или просто ОРВИ. ОРВИ – это группа подобных между собой острых воспалительных заболеваний органов дыхания, которые возбуждаются ввиду пневмотропных вирусов. Диагностика ОРВИ важна и должна быть зафиксирована, для чего стоит использовать все возможные средства.

2. Получение знаний. В данном вопросе об ОРВИ, можно выделить несколько простых заболеваний данного типа, это грипп, назофарингит (или простуда), ларингит, фарингит. В общем случае протекание болезни можно разделить на три периода: инкубационный период, лихорадочный период, завершающий период. При этом в инкубационном периоде выявить болезнь и определить тип, практически невозможно. В то время как в период лихорадки это становится реально, а значит именно в этот период нужно действовать наиболее быстро и точно.

3. Структурирование полученных знаний. Результатом структурирования будет поле знаний. Поле знаний — условное описание основных объектов предметной области, их атрибутов и закономерностей, их связывающих. Например, во время лихорадочного периода у человека могут быть следующие симптомы: насморк, кашель, головная боль, боль в горле, слабость. Эти симптомы характерны для всех ОРВИ, и благодаря этому можно с лёгкостью отличить эту группу болезней от других типов заболеваний.

Выделение набора эмпирических правил для каждой из четырёх болезней данного типа.

**Таблица 2 – Поле знаний**

<b>Болезнь</b>	<b>Симптомы</b>	<b>Назначение</b>
Грипп	Резкое повышение температуры, а затем рецидив резкого повышения температуры. Усиление кашля. Ухудшения самочувствия после начинающегося улучшения.	Профилактикой гриппа является вакцинация, а методом лечения – своевременное применение жаропонижающих и прочих препаратов.
Назофарингит	Першение в горле. Боль при глотании. Заложенность носа. Затруднение дыхания. Насморк. Боль в затылке.	В качестве разумных консервативных мер рассматриваются продолжительный отдых, усиленное потребление жидкости во избежание обезвоживания, полоскание горла тёплой соленой водой. Для смягчения проявлений заболевания используются анальгетики и антипиретики.
Фарингит	Основная причина фарингита— это вдыхание горячего, холодного или загрязненного воздуха, влияние химических раздражений. Першение и боль в горле. Сухой и порой мучительный кашель. При осмотре горла можно заметить ярко выраженное покраснение слизистой оболочки глотки.	Лечение фарингита производится при помощи антибиотиков.
Ларингит	При развитии остром ларингите слизистая оболочка выглядит резко покрасневшей. Припухание сильнее выражено в области складок преддверия.	Лечение ларингита состоит в устранении причин, вызвавших заболевание. Из лечебных процедур полезны тёплое питье, полоскание горла, тёплые ингаляции, тепло на шею.

Формализация, результатом которой является база знаний на языке представления знаний:

1. Для начала, ЭС должна знать в каком состоянии находится температура тела пациента. Если температура повышалась, то были ли рецидивы;

2. Бывает ли у пациента боль при глотании; першение в горле; сильный непрерывный кашель;

3. Вдыхал ли человек горячий, холодный, загрязнённый воздух; было ли воздействие химических раздражителей; употребление слишком горячей или слишком холодной пищи;

4. Присутствует ли покраснение слизистой, насколько она выражена. Подвержены ли покраснению трахея, гортань и надгортанник. Присутствуют ли красные точки на слизистой;

5. Бывают ли случаи изъязвление слизистой или выделением мокроты.

На данный момент имеющаяся информация позволяет описать некоторые факты будущей экспертной системы, которые были выведены на предыдущих этапах. Таким образом можно перейти к предпоследнему этапу – реализация прототипа.

Группа фактов, описывающих общее состояние здоровья:

1. Health-state normal – пациент здоров;
2. Health-state disease – пациент болен;
3. Health-state recurrent health deteriorating – рецидив ухудшения здоровья (грипп).

Группа фактов, описывающих состоянии температуры пациента:

1. Temperature normal – нормальная температура;
2. Temperature high – повышение температуры (грипп);
3. Temperature high-with-repeated – рецидив повышения температуры (грипп).

Группа фактов, описывающих присутствие кашля или болей в горле:

1. Throat pain-when-swallowing – боль при глотании (назофарингит);

2. Throat sore-throat – першение в горле (фарингит, назофарингит);
3. Throat gain-cough – усиление кашля (грипп);
4. Throat dry-cough – сухой кашель (фарингит).

Группа фактов, описывающих воздействия на человека вызванные состоянием воздухом:

1. Outside hot – горячий воздух (фарингит);
2. Outside cold – холодный воздух (фарингит);
3. Outside polluted – загрязнённый воздух (фарингит);
4. Outside chemical irritants – воздействие химических раздражителей (фарингит);
5. Outside hot-food – употребление слишком горячей пищи (фарингит);
6. Outside cold-food – употребление слишком холодной пищи (фарингит).

Группа фактов, описывающих присутствие покраснений слизистой, трахеи, гортани:

1. Redden mucosa – покраснения слизистой (фарингит, ларингит);
2. Redden points-on-the-mucous– красные точки на слизистой (ларингит);
3. Redden trachea – покраснения трахеи (ларингит);
4. Redden larynx – покраснения гортань (ларингит);
5. Redden epiglottis – покраснения надгортанник (ларингит).

Рекомендации к лечению.

Грипп:

1. Treatment “Vaccination.” – Лечение «Вакцинация»;
2. Treatment “Fever-reducing.” – Лечение «Жаропонижающие».

Назофарингит:

1. Treatment “Long rest.” – Лечение «Длинный отдых»;
2. Treatment “Increased fluid intake.” – Лечение «Повышенное потребление жидкости»;

3. Treatment “Gargling with warm salt water.” – Лечение «Полоскание тёплой солёной водой»;

4. Treatment “Analgesics and antipyretics.” – Лечение «Анальгетики и антипиретики».

Фарингит:

1. Treatment “Antibiotics.” – Лечение «Антибиотики»;

Ларингит:

1. Treatment “Warm drinking.” – Лечение «Тёплое питьё»;

2. Treatment “Gargling.” – Лечение «Полоскание»;

3. Treatment “Warm inhalation.” – Лечение «Тёплые ингаляции»;

4. Treatment “Hot compress.” – Лечение «Горячий компресс».

В разрабатываемой экспертной системе реализуются правила диагностики, которые в зависимости от ситуации будут задавать пользователю необходимые вопросы и получать ответ в строго заданной форме. Дальнейшая диагностика будет производиться с учетом предыдущих ответов на вопросы, заданные пользователю. Эти ответы будут формировать описание текущей ситуации с помощью фактов, приведенных выше. Для реализации подобной архитектуры будет необходимо реализовать функцию, задающую пользователю вопрос и получающую ответ из заданного набора корректных ответов. На рисунке 13 изображена логика ответов и вопросов, моделирующая мышление эксперта. Эта логика и загружается в solver.

Далее будет использован конструктор `deffunction`, для создания функции вопроса, который будет отображаться на экране. Функция принимает два аргумента: простую переменную `question`, которая содержит текст вопроса, и составную переменную `allowed-values` с набором допустимых ответов. Сразу после своего вызова функция выводит на экран соответствующий вопрос и читает ответ пользователя в переменную `answer`. После этого функция проверяет, является ли полученный ответ одним из заданных корректных ответов. Если нет, то процесс повторится до получения корректного ответа, иначе функция вернет ответ, введенный пользователем.

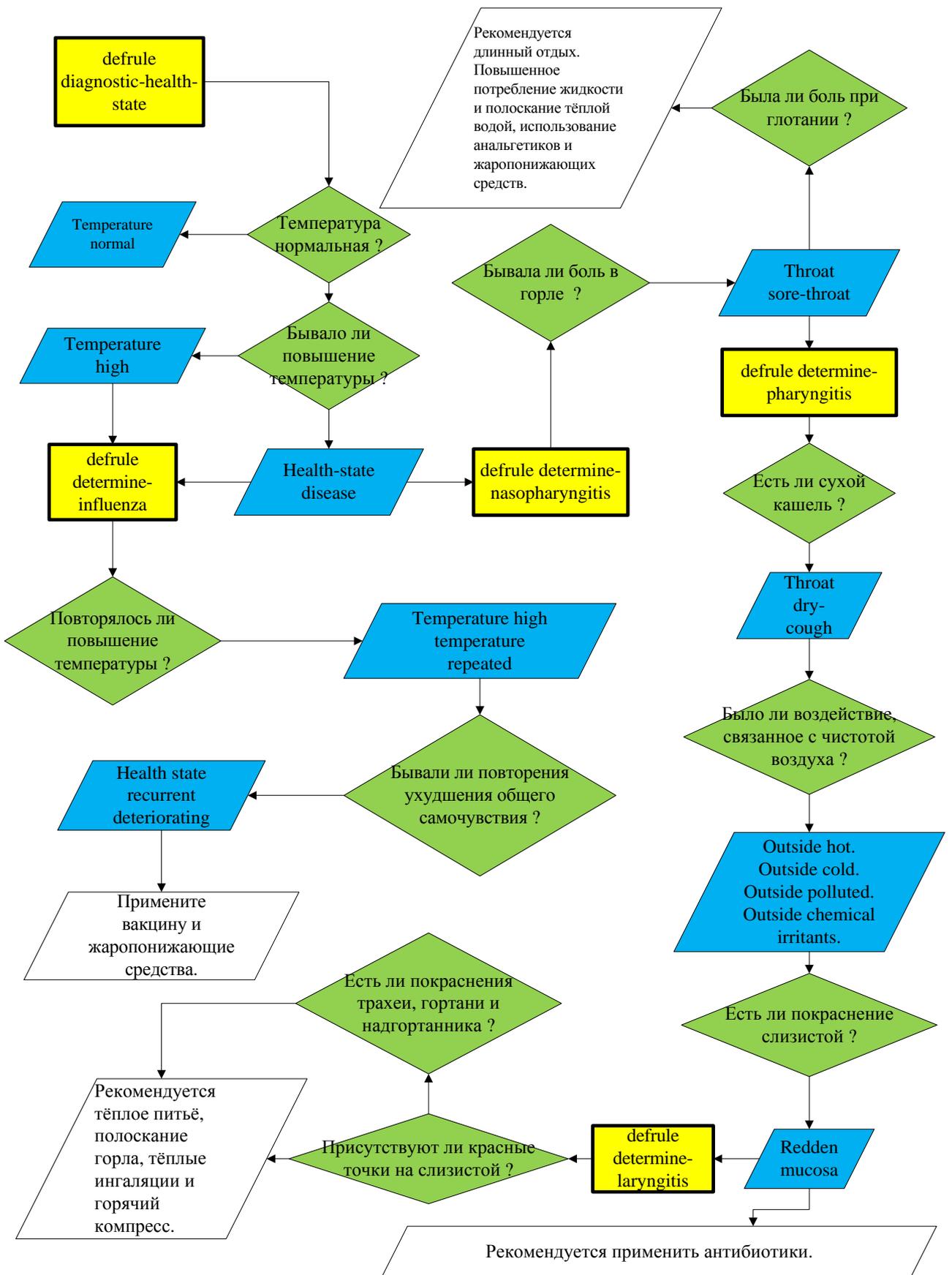


Рисунок 13 Логика Solver

```

(deffunction ask-question (?question $?allowed-values)
  (printout t ?question)
  (bind ?answer (read))
  (if (lexemep ?answer)
      then
      (bind ?answer (lowercase ?answer)))
      (while (not (member ?answer ?allowed-values)) do
        (printout t ?question)
        (bind ?answer (read))
        (if (lexemep ?answer)
            then
            (bind ?answer (lowercase ?answer))))
        ?answer)

```

Функция `yes-or-no-p` вызывает функцию `ask-question` с постоянным набором допустимых ответов: `yes`, `no`, `y` и `n`. В случае если пользователь ввел ответ `yes` или `y`, функция возвращает значение `TRUE`, иначе — `FALSE`. Обратите внимание, что поскольку функция `yes-or-no-p` использует функцию `ask-question`, то она должна быть определена после нее.

```

(deffunction yes-or-no-p (?question)
  (bind ?response (ask-question ?question yes no y n))
  (if (or (eq ?response yes) (eq ?response y))
      then TRUE
      else FALSE))

```

В некоторых случаях, когда экспертная системы заходит в тупик, не имея дальнейшего развития для анализа, будет выдано сообщение, чтобы пациент проконсультировался у врача.

Первым реализуем правило, определяющее общее состояние пациента.

```

(defrule diagnostic-health-state ""
(not (health-state normal ?))
(not (treatment ?))
=>
(if (yes-or-no-p "Температура нормальная(yes/no)? ")
    then
    (assert (temperature normal))
    (printout t crlf crlf)
    (printout t "Это хорошо !")
    (printout t crlf crlf))
    else
    (if (yes-or-no-p "Бывало ли повышение температуры(yes/no)? ")
        then
        (assert (temperature high))
        (assert (health-state disease))
        else
        (printout t crlf crlf)
        (printout t "Проконсультируйтесь у врача !")
        (printout t crlf crlf))))))

```

Условный элемент (not (health-state normal ?) ) гарантирует, что общее состояние пациента еще не определено. Если это так, то пользователю задаются соответствующие вопросы и в систему добавляется факт, описывающий текущее общее состояние пациента.

Следующим стоит проверить не было ли рецидивов повышения температуры, а также уточняет некоторые факты необходимые для выявления гриппа.

```

(defrule determine-influenza ""
(temperature high)

```

```

(health-state disease)
(not (temperature high-with-repeated ?))
(not (treatment ?))
=>
(if (yes-or-no-p "Повторялось ли повышение температуры(yes/no) ?")
    then
        (if (yes-or-no-p "Бывали ли повторения ухудшения общего самочувствия
(yes/no) ?")
            then
                (assert (temperature high-with-repeated))
                (assert (health-state recurrent health deteriorating))
                (assert (treatment "Vaccination."))
                (assert (treatment "Fever-reducing."))
                (printout t crlf crlf)
                (printout t "Примените вакцину и жаропонижающие средства.")
                (printout t crlf crlf)
            else
                (printout t crlf crlf)
                (printout t "Проконсультируйтесь у врача!")
                (printout t crlf crlf))))))

```

Следующим правилом стоит описать выявление назофарингита.

```

(defrule determine-nasopharyngitis ""
(health-state disease)
(not (treatment ?))
=>
(if (yes-or-no-p "Была ли боль в горле (yes/no) ? ")
    then
        (assert (throat sore-throat))

```

```

(if (yes-or-no-p "Была ли боль при глотании (yes/no) ?")
  then
    (assert (treatment "Long rest.))
    (assert (treatment "Increased fluid intake.))
    (assert (treatment "Gargling with warm salt water.))
    (assert (treatment "Analgesics and antipyretics.))
    (printout t crlf crlf)
    (printout t "Рекомендуется длинный отдых. Повышенное потребление
жидкости и полоскание тёплой водой. Также рекомендуется использование
анальгетиков и жаропонижающих средств.")
    (printout t crlf crlf)))

```

Следующим правилом стоит описать выявление фарингита.

```

(defrule determine-pharyngitis ""
(throat sore-throat)
(health-state disease)
(not (treatment ?))
=>
(if (yes-or-no-p "Есть ли покраснение слизистой (yes/no) ?")
  then
    (assert (redden mucosa))
    (if (yes-or-no-p "Есть ли сухой кашель (yes/no) ? ")
      then
        (if (yes-or-no-p "Было ли воздействие, связанное с чистотой
воздуха (yes/no) ?")
          then
            (assert (throat dry-cough))
            (assert (outside hot))
            (assert (outside cold))

```

```

(assert (outside polluted))
(assert (outside chemical irritants))
(assert (treatment "Antibiotics."))
(printout t crlf crlf)
(printout t "Рекомендуется применить антибиотики.")
(printout t crlf crlf)
else
(printout t crlf crlf)
(printout t "Проконсультируйтесь у врача !")
(printout t crlf crlf))))))

```

Следующим правилом будет описано выявление ларингита.

```

(defrule determine-laryngitis ""
(redden mucosa)
(health-state disease)
(not (treatment ?))
=>
(if (yes-or-no-p "Присутствуют ли красные точки на слизистой (yes/no) ?")
then
(assert (treatment "Warm drinking."))
(assert (treatment "Gargling."))
(assert (treatment "Warm inhalation."))
(assert (treatment "Hot compress."))
(printout t crlf crlf)
(printout t "Рекомендуется тёплое питьё, полоскание горла, тёплые
ингаляции и горячий компресс.")
(printout t crlf crlf))))
else
(printout t crlf crlf)

```

```

(printout t "Проконсультируйтесь у врача!")
(printout t crlf crlf)))
(if (yes-or-no-p "Есть ли покраснения трахеи, гортани и надгортанника (yes/no) ?")
    then
    (assert (treatment "Warm drinking.))
    (assert (treatment "Gargling.))
    (assert (treatment "Warm inhalation.))
    (assert (treatment "Hot compress.))
    (printout t crlf crlf)
    (printout t "Рекомендуется длинный отдых. Повышенное потребление
жидкости и полоскание тёплой водой, использование анальгетиков и
жаропонижающих средств.")
    (printout t crlf crlf)))
else
(printout t crlf crlf)
(printout t "Проконсультируйтесь у врача !")
(printout t crlf crlf)))

```

После, весь код, который был указан выше, можно загружать в CLIPS. Результатом будет отображение того, что правила были определены, о чём говорит запись «Defining defrule:». После чего используется команда(run), или комбинация Ctrl + R. Результат показан на рисунке 14.

```

Defining deffunction: ask-question
Defining deffunction: yes-or-no-p
Defining defrule: diagnostic-health-state +j+j+j

[CSTRCPSR1] Expected the beginning of a construct.
Defining defrule: determine-influenza +j+j+j+j+j
Defining defrule: determine-nasopharyngitis +j+j+j+j

[CSTRCPSR1] Expected the beginning of a construct.
Defining defrule: determine-pharyngitis +j+j+j+j

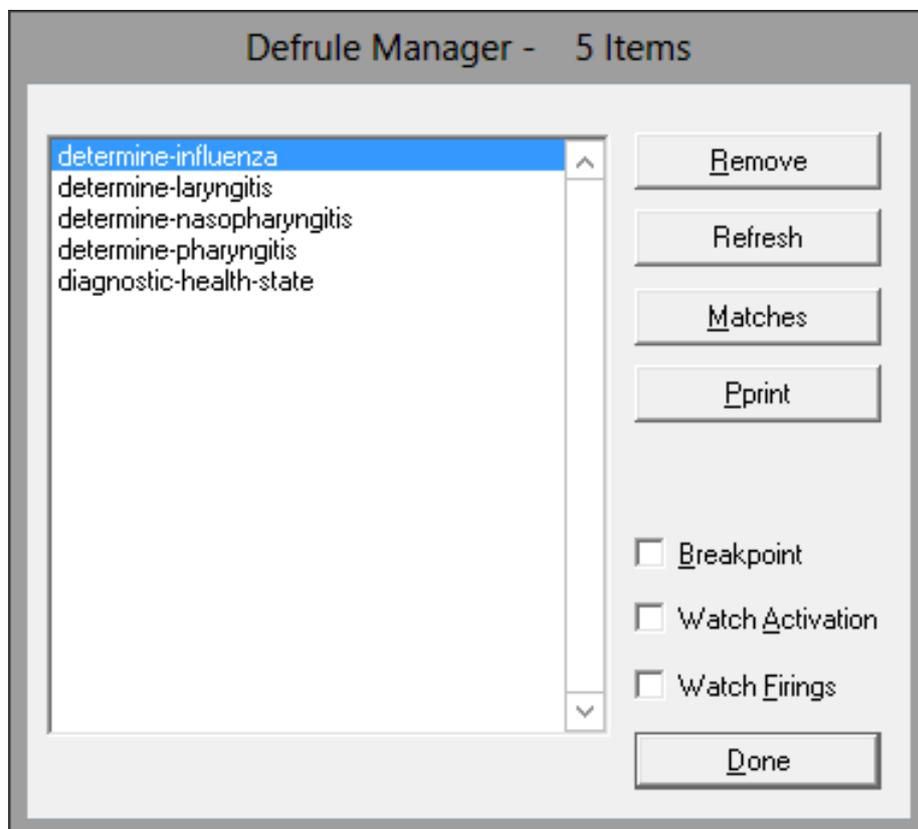
[CSTRCPSR1] Expected the beginning of a construct.
Defining defrule: determine-laryngitis +j+j+j+j

[CSTRCPSR1] Expected the beginning of a construct.
FALSE
CLIPS> (run)
Температура нормальная (yes/no)? █

```

**Рисунок 14** Запуск экспертной системы на исполнение

Можно убедиться, что загружены все правила, открыв «Defrule Manager», который изображен на рисунке 15. Там будут указаны пять правил, которые описаны выше.



**Рисунок 15** Загруженные правила

Температура нормальная (yes/no)? n  
Бывало ли повышение температуры (yes/no)? y  
Повторялось ли повышение температуры (yes/no) ? y  
Бывали ли повторения ухудшения общего самочувствия (yes/no) ? y

Примените вакцину и жаропонижающие средства.

### **Рисунок 16 Определение гриппа у пациента**

На рисунке 16, задаются вопросы, выявляющие то, что пациент болен гриппом. Ему выдаётся рекомендация о том, чтобы он использовал вакцину и жаропонижающие средства. После этого снова можно запустить работу системы, но прежде необходимо очистить память от уже загруженных фактов. Для этого можно использовать команду (reset) или нажать комбинацию клавиш Ctrl + E. На рисунке 17, изображённом ниже, указаны полученные рекомендации при условии что пациент болен назофарингитом.

Температура нормальная (yes/no)? n  
Бывало ли повышение температуры (yes/no)? y  
Повторялось ли повышение температуры (yes/no) ? n  
Была ли боль в горле (yes/no) ? y  
Была ли боль при глотании (yes/no) ? y

Рекомендуется глинный отдых. Повышенное потребление жидкости

### **Рисунок 17 Определение назофарингита у пациента**

Температура нормальная (yes/no)? n  
Бывало ли повышение температуры (yes/no)? y  
Повторялось ли повышение температуры (yes/no) ? n  
Была ли боль в горле (yes/no) ? y  
Была ли боль при глотании (yes/no) ? n  
Есть ли покраснение слизистой (yes/no) ? y  
Есть ли сухой кашель (yes/no) ? y  
Было ли воздействие, связанное с чистотой воздуха (yes/no) ? y

Рекомендуется применить антибиотики.

### **Рисунок 18 Определение фарингита у пациента**

Температура нормальная (yes/no)? n  
Бывало ли повышение температуры (yes/no)? y  
Повторялось ли повышение температуры (yes/no) ? n  
Была ли боль в горле (yes/no) ? y  
Была ли боль при глотании (yes/no) ? n  
Есть ли покраснение слизистой (yes/no) ? y  
Есть ли сухой кашель (yes/no) ? n  
Присутствуют ли красные точки на слизистой (yes/no) ? y

Рекомендуется тёплое питье, полоскание горла, тёплые ингаляции и горячий компресс.

### **Рисунок 19 Определение ларингита у пациента**

На рисунке 18 и 19 изображены череда вопросов, которые характерны для фарингита и ларингита, и соответствующие результаты.

Таким образом, была протестирована каждая из возможных сценариев использования экспертной системы, что в свою очередь означает, что этап тестирования завершен и имеется готовый прототип.

## **3.2 ЭКОНОМИЧЕСКАЯ ЭФФЕКТИВНОСТЬ**

Важным, и заключительным, пунктом данной работы, является рассмотрение экономической эффективности предлагаемой системы. Для прохождения медицинского осмотра, государство ежегодно выделяет до ста тысяч рублей, иногда больше, необходимых для обслуживания крупных организаций и поддержания персонала организации. Для определения различий между текущим экономическим состоянием и тем, которое возможно после внедрения экспертной системы, стоит внимательно рассмотреть этот вопрос.

Было выделено две целевые задачи:

1. Разработка экспертной системы;
2. Внедрение экспертной системы.

Каждая из этих задач была отдельно разбита на подзадачи. Так, разработка экспертной системы представляется такими подзадачами:

1. Идентификация проблемы;
2. Получение знаний;
3. Структурирование;

4. Формализация;
5. Реализация прототипа;
6. Тестирование.

Внедрение экспертной системы:

1. Обучение персонала;
2. Установка прототипной экспертной системы.

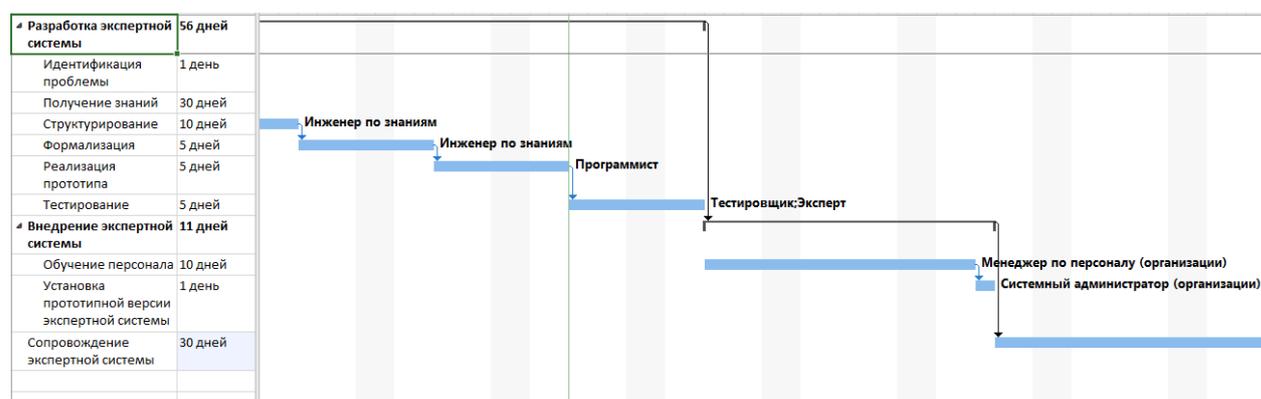
Заключительной частью проекта является сопровождение экспертной системы – для примера, период сопровождения будет составлять 30 рабочих дней.

При расчёте экономической эффективности необходимо знать сколько каждый из участников команды разработчиков зарабатывает. Команда разработчиков состоит минимум из 6 человек: инженер по знаниям, программист, тестировщик, менеджер по работе с персоналом, системный администратор, эксперт исследуемой области. В разработке так же участвует специалист исследуемой области, знания которого вкладываются в экспертную систему, участие же такого специалиста добровольно. Так же, для внедрения экспертной системы нужны системные администраторы и специалисты-менеджеры на предприятии, но фактически их работу оплачивает организация, в которой происходит внедрение экспертной системы. Таким образом наиболее важным являются программист, тестировщик и инженер по знаниям.

На текущий момент заработная плата программиста, привлекаемого для проектной работы составляет 30 тысяч рублей; тестировщика – 15 тысяч рублей; инженера по знаниям 20 тысяч рублей; менеджеру по работе с персоналом – 40 тысяч рублей; системный администратор 35 тысяч рублей. При условии, что каждый из членов команды работает двадцать дней в месяц и по десять часов в день, была получена таблица (см. таблицу 3). Для визуального изображения, сказанного выше, будет использоваться средство MS Project.

**Таблица 3 – Стандартная ставка ключевых профессий**

Название	Стандартная ставка
Инженер по знаниям	80,00 Р/ч
Программист	120,00 Р/ч
Тестировщик	60,00 Р/ч
Системный администратор (организации)	175,00 Р/ч
Менеджер по персоналу (организации)	200,00 Р/ч
Эксперт	0,00 Р/ч



**Рисунок 20 Диаграмма Ганта**

На рисунке 20 изображено общее представление разработки экспертной системы, включающее примерные временные затраты на прохождение каждого этапа. За каждой задачей закреплены соответствующие специалисты и члены команды разработки.

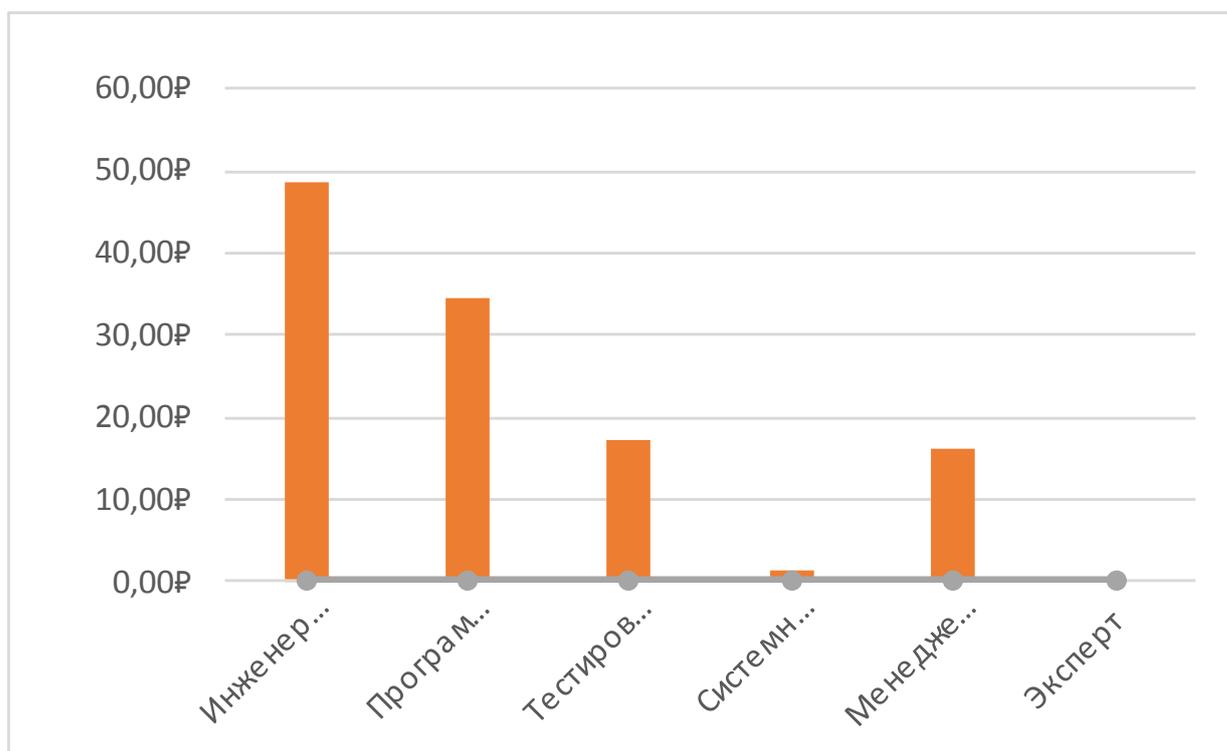
4 Разработка эксперт	512 ч 56 дней	Трудо:		8ч	8ч	16ч	16ч	16ч			16ч	16ч		
4 Идентификация п	32 ч 1 день	Трудо:												
Инженер по з	8 ч	Трудо:												
Программист	8 ч	Трудо:												
Тестировщик	8 ч	Трудо:												
Эксперт	8 ч	Трудо:												
4 Получение знани	240 ч 30 дней	Трудо:												
Инженер по з	240 ч	Трудо:												
4 Структурировани	80 ч 10 дней	Трудо:												
Инженер по з	80 ч	Трудо:												
4 Формализация	40 ч 5 дней	Трудо:												
Инженер по з	40 ч	Трудо:												
4 Реализация протс	40 ч 5 дней	Трудо:		8ч	8ч									
Программист	40 ч	Трудо:		8ч	8ч									
4 Тестирование	80 ч 5 дней	Трудо:				16ч	16ч	16ч		16ч	16ч			
Тестировщик	40 ч	Трудо:				8ч	8ч	8ч		8ч	8ч			
Эксперт	40 ч	Трудо:				8ч	8ч	8ч		8ч	8ч			
4 Внедрение эксперт	88 ч 11 дней	Трудо:										8ч	8ч	8ч
4 Обучение персон	80 ч 10 дней	Трудо:										8ч	8ч	8ч
Менеджер по	80 ч	Трудо:										8ч	8ч	8ч
4 Установка протот	8 ч 1 день	Трудо:												
Системный а	8 ч	Трудо:												
4 Сопровождение экс	3 200 ч 100 дней	Трудо:												
Инженер по зна	800 ч	Трудо:												
Программист	800 ч	Трудо:												
Тестировщик	800 ч	Трудо:												
Эксперт	800 ч	Трудо:												

**Рисунок 21 Использование ресурсов**

На рисунке 21 изображено использование ресурсов, при помощи которой можно определить кто из участников проекта в какое время занят и сколько времени затрачено на решение той, или иной задачи.

На рисунке 22 изображена итоговая сумма, которую каждый из участников проекта получает в результате разработки и внедрения экспертной системы. Таким образом, наибольшая сумма приходится на работу инженера по знаниям, так как он проходит почти через все стадии разработки принимая участие в создании большей части документов и требований необходимых для программирования экспертной системы. Программист, тестировщик и эксперт затрачивают примерно одинаковое время для разработки и сопровождения. Несмотря на то, что эксперт, в данном случае это медсестра организации, проводит так много времени над разработкой, и при этом не получает финансовой прибыли связана с тем, что в обязанности эксперта входит решать задачи своей предметной области. Меньше всего затрачивается времени на внедрение системы в техническом плане, чем занимается системный администратор, выступающий от лица организации. Фактически для этого необходимо всего лишь установить рабочую систему на действующий компьютер и провести тестирование. Немного больше на внедрение затрачивает менеджер по работе с персоналом, которому необходимо

интенсивно работать с сотрудниками организации над выработкой правильного понимания системы и тренировки навыка в её использовании.



**Рисунок 22 Трудозатраты**

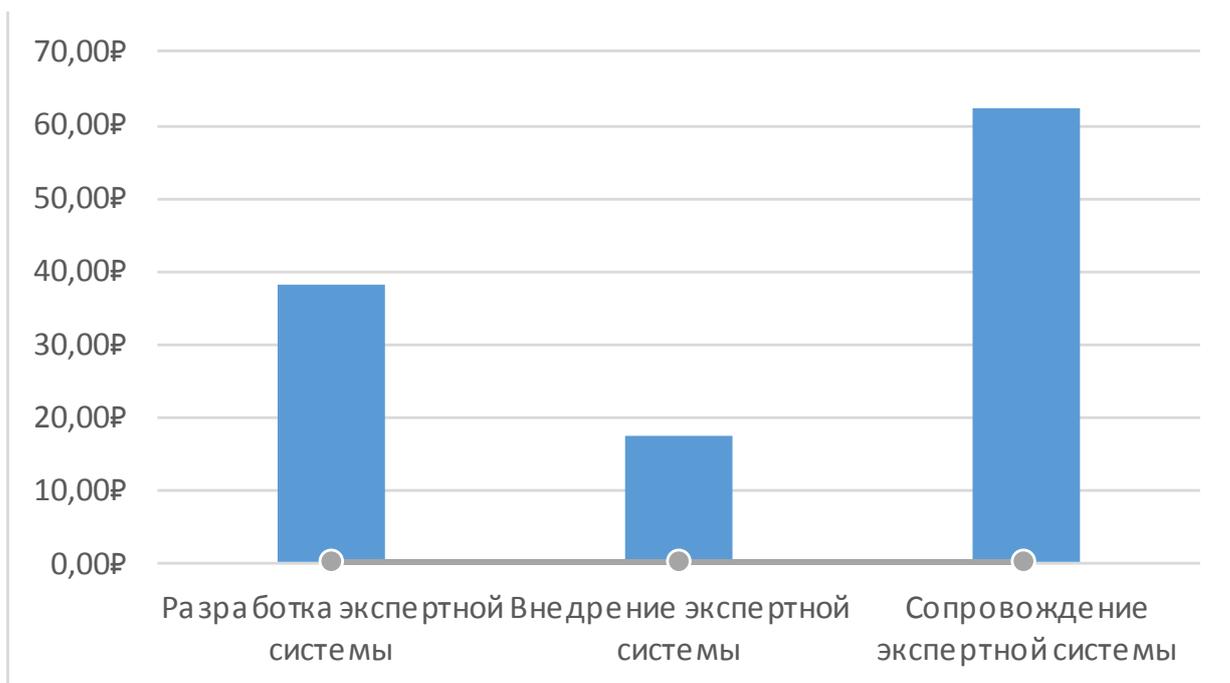
В таблице 4 приведены временные затраты каждого из участников проекта.

**Таблица 4 – Временные трудозатраты специалистов проекта**

Название	Временные трудозатраты
Программист	288 ч
Тестировщик	288 ч
Системный администратор организации	8 ч
Менеджер по персоналу организации	80 ч
Эксперт	288 ч
Инженер по знаниям	608 ч

Для наглядности при составлении отчёта период сопровождения системы было принято за 30 дней. В реальных условиях период сопровождения должен

составлять несколько лет. Причина состоит в том, что экспертным системам нужна поддержка в большом объёме, так как медицинское знание обновляется, могут меняться подходы к диагностированию заболеваний, например, открываться новые симптомы. Таким образом, каждое новое знание, должно быть своевременно добавлено в базу знаний.



**Рисунок 23 Затраты на разработку, внедрение и сопровождение**

Всего на разработку, внедрение и сопровождение экспертной системы требуется затратить 117 880 рублей. Из них, на разработку экспертной системы 38 080 рублей; 17 400 рублей – для внедрения; сопровождение – 62 400 рублей.

Очевидно, что наибольшую часть финансовых вложений составляет сопровождение, но это не превышает платы за медосмотр, который составляет 100 000 рублей ежегодно. Тем не менее экспертная система в первую очередь окупает себя в том смысле, что здоровый персонал не отправится для медицинского осмотра, они смогут продолжать работать, что принесёт выгоду для организации. Экспертная система предупреждает болезни человека, и поэтому работник может заниматься их лечением в домашних условиях, используя лишь рекомендации, полученные от системы. Помимо этого, врачи,

не будут переживать временных затрат, а государственный бюджет не будет выделять деньги на осмотр кадров, которые не испытывают проблем со здоровьем.

Но самое важное, при большем масштабе применения рассматриваемой экспертной системы, то есть применение не только в рамках одной организации, а применение в других организациях и образованиях, эта система могла бы принести существенную финансовую и временную пользу.

Экспертная система окупит себя быстрее только при условии масштабного внедрения на несколько организаций. В противном случае окупаемость будет отрицательна. Для того, чтобы окупаемость была положительной, необходимо внедрение на несколько организаций. Тогда общие затраты на внедрение вырастут, но цена на сопровождение останется прежней. Такая политика приведёт к тому, что сопровождение экспертной системы будет оплачиваться несколькими организациями, а окупаемость стремительно пойдёт вверх.

## ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы были сделаны следующие выводы:

1. Была спроектирована экспертная система для Муниципального бюджетного образовательного учреждения дополнительного образования Станция юных техников г. Туапсе муниципального образования Туапсинский район;
2. Был спроектирован и разработан прототип экспертной системы;
3. Была создана модель очередей, демонстрирующая эффективность связанную с временными издержками при помощи AnyLogic;
4. Были получены знания и навыки о методиках проектирования, внедрения и эксплуатации отдельных экспертных систем;
5. Был кратко изучен язык CLIPS, реализующий экспертные системы. Приобретены навыки исследования и проектирования экспертных систем;
6. Был проведен анализ экономической эффективности разработанной экспертной системы.

Разработанная экспертная система, представляет собой автоматизированное рабочее место медсестры, которое позволяет не только оптимизировать ее работу, но и значительно повысить качество выявления заболеваний у сотрудников, что является особо актуальным в связи с тем, что сотрудники предприятия работают с детьми. Таким образом, внедрение разработанной экспертной системы имеет социальный эффект.

Разработанная экспертная система является прототипом, а не завершенным программным продуктом. Поэтому рекомендуется в дальнейшем увеличить базу знаний экспертной системы и разработать более удобный пользовательский интерфейс, чем это позволяет среда CLIPS.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. А. П. Частиков, Т. А. Гаврилова, Д. Л. Белов: РАЗРАБОТКА ЭКСПЕРТНЫХ СИСТЕМ. СРЕДА CLIPS. - Санкт-Петербург, «БХВ-Петербург» 2003. – 393 с.
2. Системы искусственного интеллекта. [Электронный ресурс] – URL: <http://www.mari-el.ru/mmlab/home/AI/>.
3. Система помощи для работы с экспертной системы CLIPS. [Электронный ресурс] – URL: <http://www.iskhacov.narod.ru/lang/clips.html#commands>.
4. Программирование на языке CLIPS. [Электронный ресурс] – URL: <http://ryk-кyrc2.narod.ru/clips.htm>.
5. Применение средства языка CLIPS для организации и представления знаний в экспертных системах. [Электронный ресурс] – URL: <http://www.uran.donetsk.ua/~masters/2005/kita/kapustina/ind/clips.htm>.
6. Моделирование с использованием CLIPS. [Электронный ресурс] – URL: [http://alice.pnzgu.ru/~dvn/fb61499/festo/clips/FESTO\\_CLIPS.htm](http://alice.pnzgu.ru/~dvn/fb61499/festo/clips/FESTO_CLIPS.htm).
7. A Tool for building experts system. [Электронный ресурс] – URL: <http://clipsrules.sourceforge.net/>.
8. BPM Soft. Нотации серии IDEF. [Электронный ресурс] – URL: <http://bpmssoft.org/>.
9. Cit Forum. Объектно-ориентированные CASE-средства (Rational Rose). [Электронный ресурс] – URL: [http://citforum.ru/database/case/glava5\\_5.shtml](http://citforum.ru/database/case/glava5_5.shtml).
10. Cit Forum. Внедрение ERP-систем. [Электронный ресурс] – URL: [http://citforum.ru/seminars/cis99/epr\\_4.shtml](http://citforum.ru/seminars/cis99/epr_4.shtml).
11. Rational Rose для разработчиков. [Электронный ресурс] – URL: <http://citforum.ru/programming/application/rrose.shtml>.
12. Работа с Erwin Process Modeler. [Электронный ресурс] – URL: <http://www.interface.ru/home.asp?artId=22358>.

13. Портал искусственного интеллекта, роботы с искусственным интеллектом. [Электронный ресурс] – URL: <http://www.aiportal.ru/>.
14. Искусственный интеллект и экспертные системы. [Электронный ресурс] – URL: [http://expro.ksu.ru/materials/ii\\_i\\_es/book.html](http://expro.ksu.ru/materials/ii_i_es/book.html).
15. Интеллектуальные информационные системы. [Электронный ресурс] – URL: [http://www.habarov.spb.ru/new\\_es/index.htm](http://www.habarov.spb.ru/new_es/index.htm).
16. Проект медицинского справочного бюро «Бесплатная медицина». [Электронный ресурс] – URL: <http://xn--b1amqi.net/index.php>.
17. Портал искусственного интеллекта. [Электронный ресурс] – URL: <http://neuronus.com/>.
18. Клуб любителей и знатоков искусственного интеллекта. [Электронный ресурс] – URL: <http://ai.obrazec.ru/>.
19. Российская ассоциация искусственного интеллекта. [Электронный ресурс] – URL: <http://www.raai.org/>.
20. Постнаука. Искусственный интеллект. [Электронный ресурс] – URL: <https://postnauka.ru/themes/ai>.
21. Машинное обучение. [Электронный ресурс] – URL: <http://www.statsoft.ru/>.
22. ИНТУИТ. Национальный открытый университет. [Электронный ресурс] – URL: <http://www.intuit.ru/>.
23. Сеть разработчиков Microsoft. [Электронный ресурс] – URL: <https://msdn.microsoft.com>.
24. Научная электронная библиотека «Киберленинка». [Электронный ресурс] – URL: <http://cyberleninka.ru/>.
25. GeekBrains. Образовательная площадка для программистов. [Электронный ресурс] – URL: <https://geekbrains.ru/>.