

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество  
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

кафедра Компьютерных технологий

«Допущен к защите»  
Заведующий кафедрой \_\_\_\_\_

(Ф.И.О., ученая степень, звание)

« \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ г.  
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка базы знаний для мобильного приложения консультации клиентов компьютерного магазина

Специальность 5В070400 ВТ/ПО

Выполнил (а) Торохова С.О. ВТ-12-2  
(Фамилия и инициалы) группа

Научный руководитель Ахметова М.А., к.т.н., доцент  
(Фамилия и инициалы, ученая степень, звание)

Консультанты:

по экономической части:

Секичева А.В., к.э.н., доцент  
(Фамилия и инициалы, ученая степень, звание)  
« 16 » 05 2016 г.  
(подпись)

по безопасности жизнедеятельности:

Тришодов Н.Г., Д.х.н., профессор  
(Фамилия и инициалы, ученая степень, звание)  
« 16 » 05 2016 г.  
(подпись)

по применению вычислительной техники:

Ахметова М.А., к.т.н., доцент  
(Фамилия и инициалы, ученая степень, звание)  
« 31 » 05 2016 г.  
(подпись)

Нормоконтролер: Ахметова М.А., к.т.н., доцент  
(Фамилия и инициалы, ученая степень, звание)  
« 31 » 05 2016 г.  
(подпись)

Рецензент: Батабаева А.Ш., к.т.н.  
(Фамилия и инициалы, ученая степень, звание)  
« \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ г.  
(подпись)

Алматы 2016 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество  
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Факультет Аэрокосмическая и информационная техника  
Специальность Высшеи техническая и программное обеспечение  
Кафедра Компьютерная техника

ЗАДАНИЕ

на выполнение дипломного проекта

Студент Борохова Светлана Олеговна  
(фамилия, имя, отчество)

Тема проекта Разработка базы знаний для мобильного приложения - конкурентный анализ компьютерного магазина

утверждена приказом ректора № 21 от «10» марта 2016 г.

Срок сдачи законченной работы «\_\_» \_\_\_\_\_ 20\_\_ г.

Исходные данные к проекту требуемые параметры результатов проектирования (исследования) и исходные данные объекта

Разработка базы знаний

Перечень подлежащих разработке дипломного проекта вопросов или краткое содержание дипломного проекта:

Анализ рынка компьютерной техники  
Обзор аналогов  
Выбор инструментария  
Разработка правил баз знаний  
Связь баз знаний с мобильным приложением



Перечень графического материала (с точным указанием обязательных чертежей)

Иерархия классов в виде графа  
 Список свойств объектов  
 Свойства типов данных  
 Структура датумов по разработке баз данных

Рекомендуемая основная литература

Гаврилова Т. А., Хорошевский В. Ф. Базы данных интеллектуальных систем.  
 Сивков В. П., Смирнов С. Т., Козлов А. Р. Сборник типовых расчетов по курсу "Охрана труда".  
 Лукашин Г. М., Жонкина от А до Я.

Консультанты по проекту с указанием относящихся к ним разделов

Раздел	Консультант	Сроки	Подпись
Эконом. часть	Вашева А. С.		AS
БЖД	Григорьев И. Г.	17.02 - 16.05.16	IG
Предметная область	Ахметова М. А.	25.02.16 - 16.05.16	MA
Техническая часть	Ахметова М. А.	18.03.16 - 16.04.16	MA
Проектирование	Ахметова М. А.	07.04.16 - 05.05.16	MA
Контроль	Ахметова М. А.	17.05.16 - 31.05.16	MA



## **Аннотация**

Тема дипломного проекта «Разработка базы знаний для мобильного приложения консультации клиентов компьютерного магазина».

Данная дипломная работа включает в себя разработку базы знаний для мобильного приложения, осуществляющего консультацию при выборе компьютерного устройства. Для разработки был использован редактор антологии, позволяющий сформировать правила.

Приложение будет востребовано на рынке продаж Казахстана.

## **Аңдатпа**

Дипломдық жоба тақырыбы: «Компьютерлік дүкен тұтынушыларына кеңес берудік мобильдік шешімі үшін білім базасын құру».

Бұл жобаға есептеуіш құрылғысының таңдауда кеңес өткізуді қамтамасыз ететін білім қорын өңдеу кіреді. Білім қорының ережелерін өңдеу үшін онтология редакторы қолданылды.

Қосымша Қазақстан нарығында сұранысқа ие болады.

## **Annotation**

The theme of the degree project "The development of a knowledge base for a mobile solution of clients consultation of a computer store."

This thesis includes the development of a knowledge base for mobile applications, carrying out consultation in the selection of the computing device. For the development of the editor of the anthology was used, allowing to form rules.

The application will be in demand in the Kazakh market sales.

## Содержание

Введение	12
1 Исследование предметной области	13
1.1 Введение в предметную область	13
1.2 Актуальность решения	13
1.3 Постановка задач	13
1.4 Исследование предметной области	14
1.5 Обзор аналогов мобильных приложений и экспертных систем для продажи компьютеров	15
1.6 Подбор инструментария для разработки базы знаний	19
2 Теоретическая часть	23
2.1 Структура экспертной системы	23
2.2 Модели базы знаний	25
2.3 Онтология, как способ представления знаний	27
2.3.1 Поддерживаемые редактором формализмы и форматы представления	28
2.3.2 Функциональность редактора онтологий	28
2.3.3 Сложные инструментальные средства	29
3 Проектирование базы знаний	30
3.1 Создание классов онтологии	30
3.2 Определение атрибутов класса и свойств экземпляров	33
3.3 Создание экземпляров класса	36
3.4 Создание DL-query запросов	37
3.5 Связка мобильного приложения, базы данных и базы знаний	39
4 Экономическая часть	40
4.1 Цель и задачи проекта	40
4.2 Трудовые ресурсы, используемые в работе	40
4.3 Оборудование, используемое в работе	41
4.4 Сроки реализации проекта	41
4.5 Затраты на разработку системы	42
4.5.1 Расчет фонда оплаты труда	43
4.5.2 Расчет затрат по социальному налогу	47
4.5.3 Расчет амортизационных отчислений	48
4.5.4 Расчет затрат на электроэнергию	48
4.5.5 Расчет накладных расходов	49
4.7 Цена реализации	50
4.8 Оценка научно-технической результативности и социальной эффективности	51
5 Безопасность жизнедеятельности	53
5.1 Анализ условий труда в помещении серверной	53
5.2 Расчет кондиционирования помещения серверной	54
5.3 Расчет мер защиты от поражения электрическим током	58
Заключение	62

Список использованной литературы	64
Приложение А	66

## Введение

Компьютерная техника стала неотъемлемой частью жизни современного человека. Практически в каждой сфере деятельности применяются всевозможные вычислительные устройства. Очевидно, каждый человек сталкивался с выбором компьютера. Ассортимент компьютерной техники очень многообразен, в связи с этим процесс выбора становится труднее и все чаще несет за собой не удачные покупки.

Для того чтобы помочь человеку с выбором и сэкономить его время, необходима система, которая могла бы учесть сразу несколько факторов одновременно и не иметь трудностей, с которыми сталкивается человек.

Существует множество рекомендаций по выбору компьютерной техники, но изучив их все равно нет гарантии, что человек выберет подходящий по всем характеристикам товар.

Посредством искусственного интеллекта стало возможным частично заменить знания человека в той или иной области знаний, для это в свою очередь необходимо наделить компьютер знаниями эксперта.

Искусственный интеллект начал свое развитие в девяностые годы, и уже нашел свое применение во многих областях. На данный момент существует множество экспертных систем.

Отдельная область искусственного интеллекта - инженерия знаний, занимающаяся разработкой экспертных систем, а также развитием средств представления и обработки знаний.

В свою очередь, экспертная система это - система искусственного интеллекта, которая использует экспертные знания для обеспечения высокоэффективного решения неформализованных задач в узкой предметной области. Экспертные системы играют роль эксперта и решают задачи диагностики, прогнозирования или планирования.

База знаний - центральная часть экспертной системы. 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.

На мой взгляд, экспертные системы не только позволяют заменить обычных экспертов, но, и в некоторых областях, превосходят их по точности консультации.



## **1 Исследование предметной области**

### **1.1 Введение в предметную область**

Современному человеку изо дня в день приходится выбирать товар. Компьютерная техника – это тот товар, который не один год будет служить своему владельцу. Поэтому подходить к выбору нужно очень обдуманно.

Если же покупатель не имеет никаких знаний в области компьютерной техники, то ему приходится уповать на помощь консультантов в магазинах компьютерной техники. Но не для кого не секрет, что их консультация, является навязыванием того или иного товара для выгоды магазина.

Основываясь на данные проблемы возникла идея, создания экспертной системы, позволяющей проводить опрос покупателя, а также подбирать нужную модель компьютера, исходя из рода деятельности и потребностей покупателя.

### **1.2 Актуальность решения**

Компьютеры являются фактически неотъемлемым элементом нынешней жизни. Рынок компьютерной техники в Казахстане в настоящее время считается более сформированным розничным направлением и конкурентоспособным.

В последние годы, высокоинтенсивный рост сегмента домашней электроники обусловлен потребительской активностью населения, связанной с улучшением общего благосостояния. С другой — непрерывное возникновение на рынке принципиально новых образцов техники, в следствии, бурного формирования инновационного, научно-технического прогресса способствует понижению жизненного цикла потребительских товаров, повышая частоту их обновления.

Актуальность проекта заключается в создании нового приложения не имеющего аналога в сфере продаж компьютерной техники, а также разработка данного проекта позволит уменьшить затраты времени по подбору компьютерной техники. Новизна программы сведет к минимуму риск приобрести клиентам залежавшийся, товар бывшего употребления, бракованную компьютерную технику.

### **1.3 Постановка задач**

Если обобщить те задачи, которые стояли в ходе разработки дипломной работы, то получиться следующие:

- Анализ рынка компьютерной техники
- Обзор аналогов
- Подбор инструментария
- Разработка правил Базы знаний
- Связка базы знаний с мобильным приложением

## 1.4 Исследование предметной области

Предметом исследования являются компьютеры с набором их характеристик. Базе знаний необходимо включать в себя большой спектр устройств, чтобы подобрать товар можно было из богатого ассортимента. Компьютеры различаются по множеству критериев, как по размерам, так и по присущим им возможностям.

Бытует мнение, что миром правят персональные компьютеры, наверняка так они и есть, просто их количество очень велико. Основываясь на это, в базе знаний будут описаны данные о персональных компьютерах. Персональный компьютер предназначен для работы с человеком на прямую, то есть компьютер даёт возможность получить понятную информацию для человека.

Персональные компьютеры могут быть представлены в виде стационарных персональных компьютеров, а также в виде портативных персональных компьютеров.

В базу знаний будут входить следующие виды персональных компьютеров:

- Настольные компьютеры(десктопы)
- Моноблоки
- Ноутбуки.

Настольные компьютеры являются самыми высокопроизводительными из персональных компьютеров. Настольный компьютер называется так потому, что местом размещения для него служит письменный. Основным компонентом в нем считается системный блок. К системному блоку присоединяется монитор, клавиатура, мышь, все эти составляющие ведут взаимодействие между собой. Системный блок является модульным устройством, что дает возможность замены каждого составляющего модуля, индивидуально, тем самым внести изменения в конфигурацию для себя.

Моноблок - это тот же стационарный компьютер, но без составляющего системного блока. Каркасом для такого вида персональных компьютеров как моноблок служит один общий корпус с монитором. Все комплектующие размещаются в тыловой части, за дисплеем, что создаёт определённое удобство для работы за компьютером.

Представителями портативных компьютеров в нашей базе знаний являются ноутбуки и ультрабуки. Прежде всего они обладают мобильностью, на них можно работать в любом месте, их можно переносить, а также есть возможность автономной работы. И в отличии от стационарных компьютеров их строение таково, что дисплей и клавиатура совмещены.

Ультрабуки, в свою очередь отличаются от ноутбуков еще меньшими габаритами, но при этом характеристики практически совпадают с ноутбуками.

Неотъемлемой комплектующей персонального компьютера являются всевозможные периферийные устройства, которые расширяют его

функциональные возможности. Поэтому в базе знаний необходимо наличие периферийного оборудования.

Рынок периферийной техники очень многообразен, начиная от принтеров и заканчивая наушниками.

Принтер – это периферийное устройство вывода текстовой и графической информации. Существует матричные, струйные и лазерные принтеры. В базе знаний будут представлены каждые из трех видов.

Мышь и клавиатура – те устройства которые облегчают ввод информации на персональном компьютере.

Из представленных выше устройств и будут составляться знания и факты в базе знаний.

### **1.5 Обзор аналогов мобильных приложений и экспертных систем для продажи компьютеров**

В настоящее время экспертные системы используются для решения различных типов задач и в самых разнообразных областях. Мобильные приложения, в свою очередь, являются не менее популярными и востребованными. Далее будут рассмотрены наиболее подобные разработки мобильных приложений и экспертных систем.

Индустрия компьютерных магазинов достаточно хорошо развита в Казахстане, что включает в себя информативность населения, посредством интернета. Многие магазины так же имеют мобильные приложения.

Мобильное приложение магазина «Алсер» (рисунок 1.1).

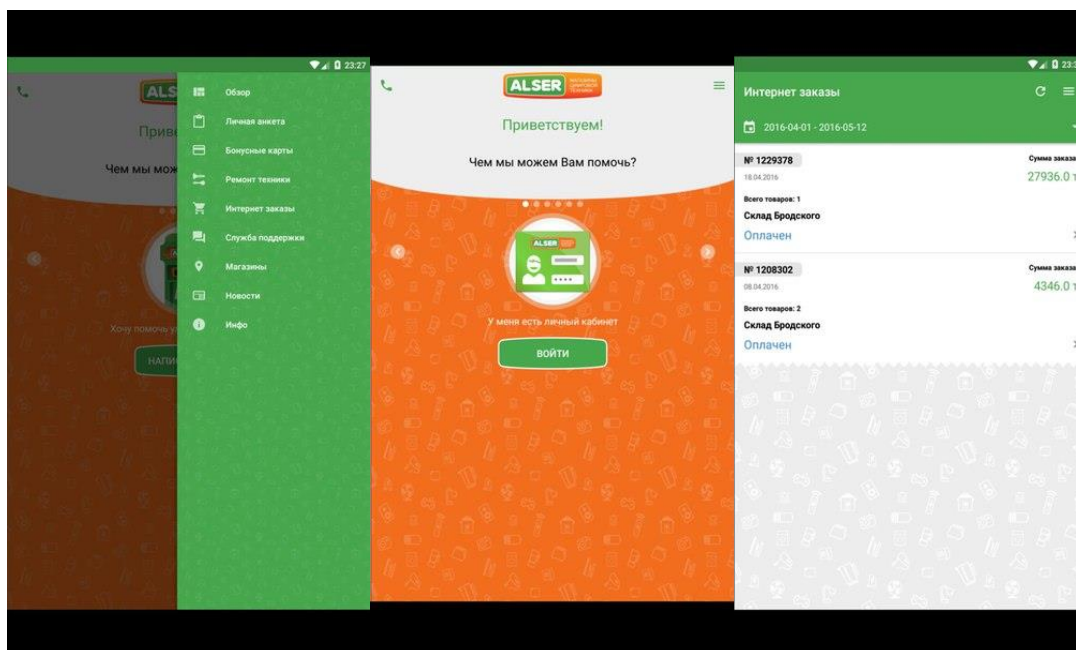


Рисунок 1.1 – Приложение магазина «Алсер»

Мобильные приложения казахстанских крупных компьютерных магазинов, в основном, разработаны по одному принципу.

Из плюсов данного приложения можно выделить:

- оно включает в себя регистрацию клиента по клубной карте, а также по этой карте можно следить за бонусами и за произведенными покупками;
- возможность интернет заказа;
- онлайн обращение в службу поддержки, в том числе и вопросы связанные с ремонтом техники
- информация о магазинах, их место расположения;
- классификация товара по видовой принадлежности;
- простота интерфейса.

Далее можно выделить следующие минусы:

- приложение не включает в себя экспертную систему, которая осуществляет диалог с пользователем, на основе чего осуществляет консультацию;
- нет возможности сравнения товара друг с другом;
- не удобный формат представления информации.

Я считаю, для разработки необходимо проанализировать не только казахстанские аналоги, но и зарубежные. Мною были выделены две индонезийские разработки.

Мобильное приложение Harga Laptop (рисунок 1.2)

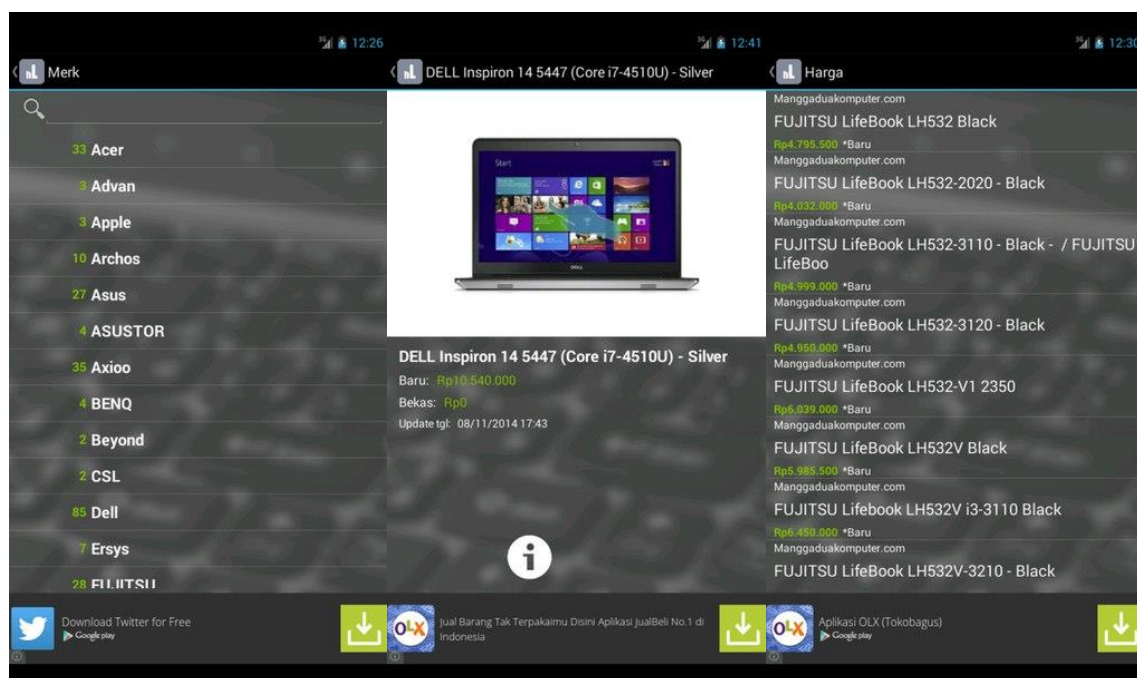


Рисунок 1.2 – Мобильное приложение Harga Laptop

Плюсы мобильного приложения:

- хороший выбор моделей компьютера;
- классификация товара по производителям;

- удобное представление информации;

Минусы мобильного приложения:

- нет возможности сравнения товара друг с другом;
- поиск товара возможен только по определенной модели компьютера;
- приложение не включает в себя экспертную систему;
- в приложении нет помощи в совершении покупки компьютера;
- нет привязки к магазину и ценовой категории.

Мобильное приложение Buying guide (рисунок 1.3)

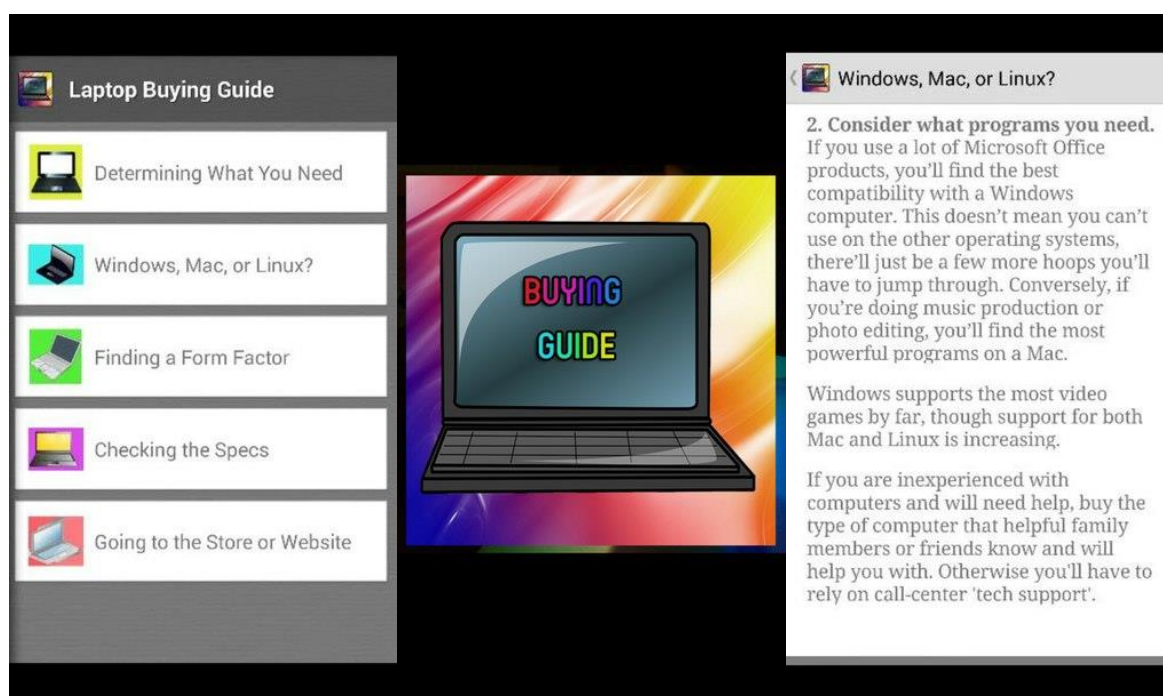


Рисунок 1.3. Мобильно приложение Buying guide.

Плюсы мобильного приложения:

- производиться консультация в покупке компьютера

Минусы мобильного приложения:

- нет представления техники по ее характеристики;
- не удобный формат представления информации;
- приложение не включает в себя экспертную систему;
- нет информации о магазинах.

На данный момент, не существует мобильных приложений, включающих в себя экспертные системы, поэтому были проанализированы аналоги экспертных систем. В то же время, не нашлось экспертных систем, разработанных для подбора компьютерной техники, поэтому была выбрана наиболее подходящая к нашей области.



## Экспертная система подбора видеокарты (рисунок 1.4)

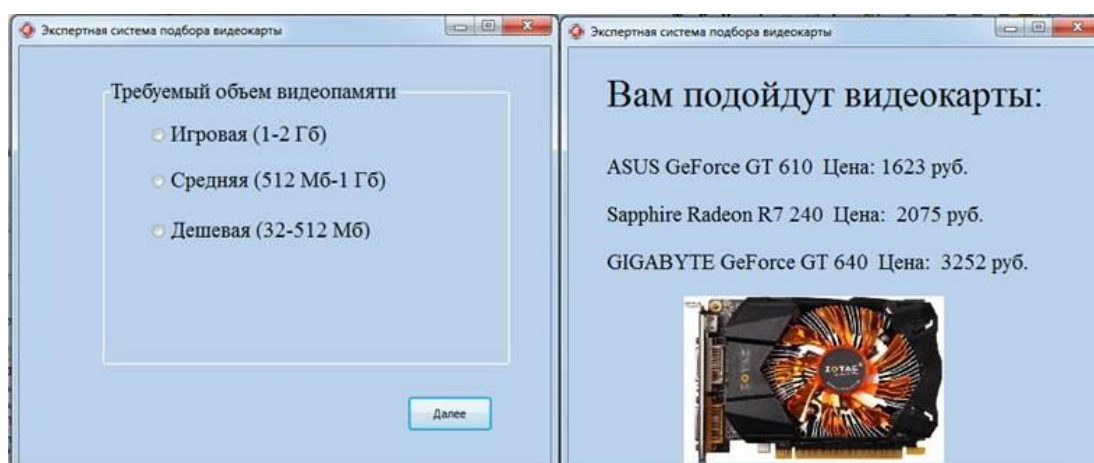


Рисунок 1.4 – Экспертная система подбора видеокарты.

Плюсы экспертной системы:

- экспертная система, разработанная с помощью онтологии Dое;
- диалог с пользователем;
- подбор видеокарт с разной ценовой категорией;

Минусы экспертной системы:

- не мобильное приложение;
- не удобный интерфейс;
- нет привязки к магазинам.

На основе обзора аналогов, стало очевидным использование экспертной системы, для более полной консультации потребителей. Ведь преимущество экспертной системы на лицо:

- постоянство
- решение трудоемких задач

Экспертная система способна удерживать в памяти многочисленные факты, правила и условия задачи, что позволяет оперативно выполнять действия с данными и быстро приходиться к какому-либо выводу.

- простота передачи

Передача знаний от одного человека другому – длительный, трудоемкий и часто недешевый процесс. Гораздо проще передать программу, которая будет заменять эксперта своими действиями.

- стабильность результатов

Экспертные системы устойчивы к «помехам». Человек же легко поддается влиянию внешних факторов, которые непосредственно не связаны с решаемой задачей.

- стоимость

Высококвалифицированные специалисты обходятся очень дорого. Экспертные системы же хоть и отличаются дороговизной разработки, но не требуют больших затрат при эксплуатации.

А также онтология на данный момент применяется в различных областях:

- искусственный интеллект;
- информационный поиск;
- вопросно-ответные системы;
- машинный перевод;
- системы понимания языка.

Становится очевидным, что благодаря разработки базы знаний с помощью онтологии, приложение получится уникальным, удобным и востребованным.

### **1.6 Подбор инструментария для разработки базы знаний**

Инструментальное средство разработки экспертных систем представляет собой язык программирования, который используется разработчиком для построения экспертной системы. На сегодняшний день, для создания экспертных систем существуют различные программные средства. С одной стороны, для создания полноценной экспертной системы требуются определение концепций предметной области, факты, а также правила, с помощью которых происходит процесс рассуждения. С другой стороны, необходимы процедурные знания, такие как алгоритмы.

Становится явным, что наиболее универсальным и удобным инструментальным средством для разработки экспертной системы является среда программирования, которая поддерживает несколько парадигм.

Нужным средством разработки является онтология. Онтология может составлять каркас базы знаний, создавать базис для описания основных понятий предметной области и служить основой для интеграции баз данных, содержащих фактически знания, необходимые для полноценного функционирования экспертной системы. Кроме всего прочего, в терминах онтологии могут описываться экспертные правила, что значительно повышает их уровень описания и понятность пользователями-экспертами.

Существует множество редакторов онтологий (Рисунок 1.5).

Наиболее известные редакторы онтологий на данный момент это Protégé и Doe. Выбор редактора напрямую зависит от функциональности экспертной системы.

Название	Краткое описание	Формализмы, языки, форматы
Ontolingua	Совместная разработка онтологий	OKBC , KIF
Protege	Создание, просмотр онтологий	JDBC , UML , XML , XOL , SHOE , RDF / RDFS , DAML+OIL , OWL
OntoSaurus	Web-браузер баз знаний на языке LOOM	LOOM
OntoEdit	Разработка и поддержка онтологий	F-Logic , RDFS , OIL , OXML
OilEd	Разработка онтологий, поддержка логического вывода	DAML+OIL
WebOnto	Многопользовательская разработка онтологий	OCML
WebODE	Создание онтологий с помощью методологии Methontology	F-Logic , LOOM , Ontolingua

Рисунок 1.5 – характеристика существующих онтологий.

Protege является локальной свободно распространяемой программой, разработанной в Стэнфордском университете. Инструмент имеет графический интерфейс, удобный для использования неопытными пользователями, снабжен справками и примерами. Protégé основан на фреймовой модели представления знания Open Knowledge Base Connectivity и снабжен рядом плагинов. В рамках Protégé выделяют Protégé Frames и Protégé OWL, каждый из них обладает специфическим интерфейсом.

В основе инструмента DOE лежит методология построения онтологий, разработанная Р. Тронси и А. Исааком. Данные исследователи отказались от использования существующих методов в связи с тем, что они не давали возможность строго и точно структурировать онтологии, в частности, подробно описывать родовидовые отношения.

Для онтологии важна простота ее переноса на другие платформы и внедрения в другие программные продукты и онтологии, переводимость на другие формальные языки:

DOE – предоставляет экспорт в RDFS; OWL; OIL; OIL XML; DAML+OIL; CGXML.

Protégé – экспорт в RDF; RDFS; XML; HTML; OWL; Clips; N3; TURTLE.

Специфика структуры редакторов не менее важна

В Doe присутствуют:

- класс;
- свойство;
- домен;
- разбиение на дифференциальную часть и референциальную.

В Protege присутствуют:

- классы;
- слоты (свойство);
- экземпляры;
- фасеты (ограничения слотов).

Для создания онтологии выполняются следующие этапы:

- Выделение понятий

- Формулирование отношений между понятиями
- Пополнение онтологии конкретными экземплярами
- Формулировка аксиом

Переходя от одного элемента онтологии к другому, продемонстрируем специфику сравниваемых редакторов.

Отражение классов в редакторе Doe удобно тем что, метод выявления оснований для деления помогает формализовать выделение классов, однако избыточны генерируемые определения, но зато встроены синонимы и лексические предпочтения. И полностью отсутствует поддержка русского языка для представления знаний.

В редакторе Protégé при создании классов есть возможность описания метаклассов, а также поддерживается множественное наследование и предоставляется выбор типа представления классов. Можно указать, как визуально выглядит класс и какие описания туда включить. Но к сожалению, нет встроенных синонимов и их приходится задавать отдельно при помощи слота.

Что касается отражения отношений в редакторе Doe, то оно позволяет строить иерархии отношений, но при этом нет привязки к классам или экземплярам. Это возможно лишь путем установки домена для отношения, однако неясно, как задавать отношения отличные от родовидовых.

В Protégé дела обстоят по-другому, есть возможность создавать и широко описывать новые отношения в дополнение к стандартным, список стандартных отношений также велик, кроме того, существует возможность связывать отдельные классы или экземпляры посредством различных отношений. Не стоит забывать, что присутствуют фасеты – ограничения на отношениях – мощность, домен, диапазон, также есть установка отношений между самими слотами.

При работе с конкретными экземплярами в редакторе Doe очень просто создавать экземпляры, но нельзя их напрямую описывать при помощи отношений и свойств. В Protégé же, много возможностей для приписывания конкретных свойств отдельному экземпляру, а также можно связывать отношением конкретный экземпляр и класс, и возможно задавать формы представления экземпляров.

Кроме всего прочего, у Protégé в отличии от Doe есть ряд функций. Например, в Protégé поддерживается создание аксиом как дополнительная функция. А также в этом редакторе возможно использование сложных запросов для поиска в рамках онтологии и осуществляется выборка информации и ее проверка и т.д.

Проведя сравнение, становится очевидным выбор редактора онтологии Protégé. Хоть он и сложнее в использовании, но предоставляет больше возможностей для адаптации под конкретную задачу и конкретного пользователя. А также дает возможность связать мобильное приложение с базой знаний.





## 2 Теоретическая часть

### 2.1 Структура экспертной системы

Экспертных системы являются компьютерными приложениями, разработанные для решения сложных проблем в той или иной области, на уровне человеческого интеллекта и опыта. Экспертная система может полностью взять на себя функции, выполнение которых обычно требует привлечения опыта человека-специалиста, или играть роль ассистента для человека, принимающего решение. Другими словами, система, требующая принятия решения, может получить его непосредственно от программы или через промежуточное звено — человека, который общается с программой [1].

Структура экспертной системы содержит следующие объекты (рисунок 2.1):

- база знаний
- рабочая память
- механизм логического вывода
- подсистема диалога
- подсистема приобретения и пополнения знаний
- подсистема объяснения



Рисунок 2.1 – Структура экспертной системы.

Теперь остановимся на каждом из этих понятий подробнее.

База знаний – ядро экспертной системы, совокупность знаний предметной области, записанная на машинный носитель в форме, понятной эксперту и пользователю [2]. Под правилами в базе знаний понимаются правила вывода, описывающие отношения между фактами. Факты же имеют классическое значение и представляют собой знания в форме утверждений.

Помимо правил и фактов в базу знаний может входить некоторая процедурная часть, выполняющая расчетные, оптимизационные или другие нужные алгоритмы.

Рабочая память предназначена для временного хранения исходных

данных и промежуточных фактов (полученных от пользователя) решаемой в текущий момент задачи. Как правило, рабочая память размещается в оперативной памяти ЭВМ.

Механизм логического вывода предназначен для получения новых фактов на основе сопоставления исходных данных из рабочей памяти и знаний из базы знаний. Механизм логического вывода во всей структуре экспертной системы занимает наиболее важное место, т.к. именно он моделирует ход рассуждений эксперта.

Подсистема диалога отвечает за организацию интерфейса для общения с пользователем в процессе решения задачи и получения результата. Именно этот элемент экспертной системы в дальнейшем будет называться пользовательским интерфейсом.

Подсистема приобретения и пополнения знаний автоматизирует процесс наполнения экспертной системы знаниями, осуществляемый пользователем-экспертом, и адаптации базы знаний системы к условиям ее функционирования. Адаптация экспертной системы к изменениям в предметной области реализуется путем замены правил или фактов в базе знаний.

Подсистема объяснения позволяет пользователю понять, каким образом система пришла к итоговому решению задачи или почему решение не было найдено. Это повышает доверие пользователя к полученному результату и упрощает разработчику тестирование программы. Возможность объяснять свои действия является одним из самых важных свойств экспертной системы, потому что еще больше приближает ее работу к работе эксперта в данной области.

Классификация экспертных систем по решаемой задаче. Чтобы лучше понимать принципы работы экспертных систем, рассмотрим классы задач, которые можно решать с их помощью:

Интерпретация. Согласование полученных извне данных и получение исходя из них ответа на общий вопрос.

Диагностирование. Проведение различных диагностик (например, определение неисправностей в каком-либо механизме).

Мониторинг. Непрерывное отслеживание данных и сообщение о выходе тех или иных параметров за допустимые пределы. При решении задач мониторинга с помощью экспертной системы, существует риск того, что система будет подавать ложные сигналы, или наоборот, пропускать плохие ситуации. Это связано с тем, что границы допустимого обычно размыты.

Проектирование. Задача проектирования заключается в том, чтобы подготовить необходимый пакет документов для создания объекта, с заранее заданными свойствами.

Прогнозирование. По полученному набору данных система должна предугадать дальнейшее развитие событий.

Обучение. Система обучения наблюдает за процессом изучения какой-либо дисциплины пользователем-учеником и, принимая во внимание его ошибки, подсказывает правильные решения и дает советы.

Планирование. Экспертная система составляет план действия для объекта, выполняющего определенные функции.

Таким образом, мы видим, что экспертные системы могут решать довольно широкий круг задач. Это тем более оправдывает их использование в различных областях.

## 2.2 Модели базы знаний

Представление знаний — еще одна функция экспертной системы.

На сегодняшний день разработано уже достаточное количество моделей. Каждая из них обладает своими плюсами и минусами, и поэтому для каждой конкретной задачи необходимо выбрать именно свою модель. От этого будет зависеть не столько эффективность выполнения поставленной задачи, сколько возможность ее решения вообще. Существуют два типа методов представления знаний (ПЗ):

1. Формальные модели ПЗ;
2. Неформальные (семантические, реляционные) модели ПЗ.

Очевидно, все методы представления знаний, которые рассмотрены выше, включая продукции (это система правил, на которых основана продукционная модель представления знаний), относятся к неформальным моделям. В отличие от формальных моделей, в основе которых лежит строгая математическая теория, неформальные модели такой теории не придерживаются. Каждая неформальная модель годится только для конкретной предметной области и поэтому не обладает универсальностью, которая присуща моделям формальным. Логический вывод - основная операция в СИИ - в формальных системах строг и корректен, поскольку подчинен жестким аксиоматическим правилам. Вывод в неформальных системах во многом определяется самим исследователем, который и отвечает за его корректность.

Каждому из методов ПЗ соответствует свой способ описания знаний.

1. Логические модели. В основе моделей такого типа лежит формальная система, задаваемая четверкой вида:  $M = \langle T, P, A, B \rangle$ . Множество  $T$  есть множество базовых элементов различной природы, например слов из некоторого ограниченного словаря, деталей детского конструктора, входящих в состав некоторого набора и т.п. Важно, что для множества  $T$  существует некоторый способ определения принадлежности или непринадлежности произвольного элемента к этому множеству. Процедура такой проверки может быть любой, но за конечное число шагов она должна давать положительный или отрицательный ответ на вопрос, является ли  $x$  элементом множества  $T$ . Обозначим эту процедуру  $\Pi(T)$ .

Множество  $P$  есть множество *синтаксических правил*. С их помощью из элементов  $T$  образуют *синтаксически правильные совокупности*. Например, из

слов ограниченного словаря строятся синтаксически правильные фразы, из деталей детского конструктора с помощью гаек и болтов собираются новые конструкции. Декларируется существование процедуры  $\Pi(P)$ , с помощью которой за конечное число шагов можно получить ответ на вопрос, является ли совокупность  $X$  синтаксически правильной.

В множестве синтаксически правильных совокупностей выделяется некоторое подмножество  $A$ . Элементы  $A$  называются *аксиомами*. Как и для других составляющих формальной системы, должна существовать процедура  $\Pi(A)$ , с помощью которой для любой синтаксически правильной совокупности можно получить ответ на вопрос о принадлежности ее к множеству  $A$ .

Множество  $B$  есть множество *правил вывода*. Применяя их к элементам  $A$ , можно получать новые синтаксически правильные совокупности, к которым снова можно применять правила из  $B$ . Так формируется *множество выводимых* в данной формальной системе *совокупностей*. Если имеется процедура  $\Pi(B)$ , с помощью которой можно определить для любой синтаксически правильной совокупности, является ли она выводимой, то соответствующая формальная система называется *разрешимой*. Это показывает, что именно правило вывода является наиболее сложной составляющей формальной системы.

Для знаний, входящих в базу знаний, можно считать, что множество  $A$  образуют все информационные единицы, которые введены в базу знаний извне, а с помощью правил вывода из них выводятся новые производные знания. Другими словами формальная система представляет собой генератор порождения новых знаний, образующих множество выводимых в данной системе знаний. Это свойство логических моделей делает их притягательными для использования в базах знаний. Оно позволяет хранить в базе лишь те знания, которые образуют множество  $A$ , а все остальные знания получать из них по правилам вывода.

2. Сетевые модели. В основе моделей этого типа лежит конструкция, названная ранее семантической сетью. Сетевые модели формально можно задать в виде  $H = \langle I, C_1, C_2, \dots, C_n, \Gamma \rangle$ . Здесь  $I$  есть множество информационных единиц;  $C_1, C_2, \dots, C_n$  - множество типов связей между информационными единицами. Отображение  $\Gamma$  задает между информационными единицами, входящими в  $I$ , связи из заданного набора типов связей.

В зависимости от типов связей, используемых в модели, различают *классифицирующие сети*, *функциональные сети* и *сценарии*. В классифицирующих сетях используются отношения структуризации. Такие сети позволяют в базах знаний вводить разные иерархические отношения между информационными единицами. Функциональные сети характеризуются наличием функциональных отношений. Их часто называют *вычислительными моделями*, т.к. они позволяют описывать процедуры "вычислений" одних информационных единиц через другие. В сценариях используются каузальные отношения, а также отношения типов "средство -

результат", "орудие - действие" и т.п. Если в сетевой модели допускаются связи различного типа, то ее обычно называют семантической сетью.

3. Продукционные модели. В моделях этого типа используются некоторые элементы логических и сетевых моделей. Из логических моделей заимствована идея правил вывода, которые здесь называются *продукциями*, а из сетевых моделей - описание знаний в виде семантической сети. В результате применения правил вывода к фрагментам сетевого описания происходит трансформация семантической сети за счет смены ее фрагментов, наращивания сети и исключения из нее ненужных фрагментов. Таким образом, в продукционных моделях процедурная информация явно выделена и описывается иными средствами, чем декларативная информация. Вместо логического вывода, характерного для логических моделей, в продукционных моделях появляется *вывод на знаниях*.

4. Фреймовые модели. В отличие от моделей других типов во фреймовых моделях фиксируется жесткая структура информационных единиц, которая называется протофреймом. В общем виде она выглядит следующим образом:

(Имя фрейма:

Имя слота 1(значение слота 1)

Имя слота 2(значение слота 2)

.....

Имя слота К (значение слота К)).

Значением слота может быть практически что угодно (числа или математические соотношения, тексты на естественном языке или программы, правила вывода или ссылки на другие слоты данного фрейма или других фреймов). В качестве значения слота может выступать набор слотов более низкого уровня, что позволяет во фреймовых представлениях реализовать "принцип матрешки".

При конкретизации фрейма ему и слотам присваиваются конкретные имена и происходит заполнение слотов. Таким образом, из протофреймов получают фреймы - экземпляры. Переход от исходного протофрейма к фрейму - экземпляру может быть многошаговым, за счет постепенного уточнения значений слотов [3].

### **2.3 Онтология, как способ представления знаний**

Онтология – это подробная формализация некоторой области знаний с помощью концептуальной схемы [4].

При создании онтологий, как и при проектировании программного обеспечения, рационально пользоваться подходящими инструментами. Инструментальные программные средства, сделанные намеренно для проектирования, редактирования и обзора онтологий, именуется редакторами онтологий.



Главная цель любого редактора онтологий состоит в поддержке процесса формализации знаний и понятии онтологии словно верного и обильного описания той либо другой области знаний.

В большинстве своем современные редакторы онтологий предоставляют средства кодирования формальной модели в том или ином виде. Некоторые дают дополнительные возможности по анализу онтологии, используют механизм логического вывода.

### **2.3.1 Поддерживаемые редактором формализмы и форматы представления.**

Под формализмом понимается абстрактный базис, лежащий в базе метода представления онтологических познаний. Примерами формализмов имеют все шансы работать логика предикатов (First Order Logic - FOL), дескриптивная логика, фреймовые модели (Frames), концептуальные графы и т.п. Формализм, применяемый редактором, имеет возможность не лишь только значимо воздействовать на внутренние структуры данных, но и предопределять формат представления или же в том числе и пользовательский интерфейс.

Формат представления онтологии задает картина сбережения и метод передачи онтологических описаний. Под форматами предполагаются языки представления онтологий: RDF, OWL, KIF, SCL. Этим образом, кое-какая формальная модель видется в формализме FOL и имеет возможность быть проявлена способами языка KIF.

Редакторы онтологий как правило поддерживают работу с несколькими формализмами и форматами представления, но нередко лишь только раз формализм считается родным для предоставленного редактора.

### **2.3.2 Функциональность редактора онтологий**

Важной характеристикой является функциональность редактора, т.е. множество сценариев его использования.

Базовый набор функций обеспечивает:

- работу с одним или более проектами;
- сохранение проекта в нужном формализме и формате (экспорт);
- открытие проекта;
- импорт из внешнего формата;
- редактирование метаданных проекта (в широком смысле: от настройки форм редактирования и представления данных до поддержки версий проекта);
- редактирование онтологии.

Набор возможных действий обычно включает создание, редактирование, удаление понятий, отношений, аксиом и прочих структурных элементов онтологии, редактирование таксономии.

К дополнительным возможностям редакторов относят поддержку языка запросов (для поиска нетривиальных утверждений), анализ целостности,

использование механизма логического вывода, поддержку многопользовательского режима, поддержку удаленного доступа через Интернет.

### **2.3.3 Сложные инструментальные средства**

Данные средства необходимы чтобы не только лишь вводить и редактировать онтологическую информацию, да и рассматривать ее, исполняя обычные операции над онтологиями, к примеру:

- выравнивание (alignment) онтологий – установка разного вида соответствий меж 2-мя онтологиями чтобы им предоставлялась возможность применять информацию друг дружку;
- отображение (mapping) одной онтологии на иную - нахождение семантических взаимосвязей меж схожими деталями различных онтологий;
- соединение (merging) онтологий [5].

### 3 Проектирование базы знаний

Целью исследования онтологии – первоначальный шаг для анализа настоящей сферы, что в последующем достаточно применяться в свойстве классов и подклассов.

#### 3.1 Создание классов онтологии

Классы трактуются как множества, и состоит из экземпляров. Они описываются, с использованием формальной конструкции, который определяет требование для членства в классе. Класс может быть в иерархии отношений вида “подкласс – суперкласс. Каждый подкласс является подмножеством своего суперкласса.

Если учесть то, что выбрана предметная область «Компьютеры», для нее были декларированы новые классы и их подклассы. На месте основного класса представляется класс «компьютер», имеющий подклассы в виде сферы деятельности для которой он будет использоваться. А именно, следующие подклассы как «для it сферы», «игровой», «мультимедиа» и «офисный».

В Protege 5.0 создание и декларирование классов выполняется в закладке “Classes” (рисунок 3.1). Исконно бессодержательная онтология содержит один класс по умолчанию – `THING`. `THING` – это тот класс, который представляет набор, содержащий в себе все объекты предметной области.

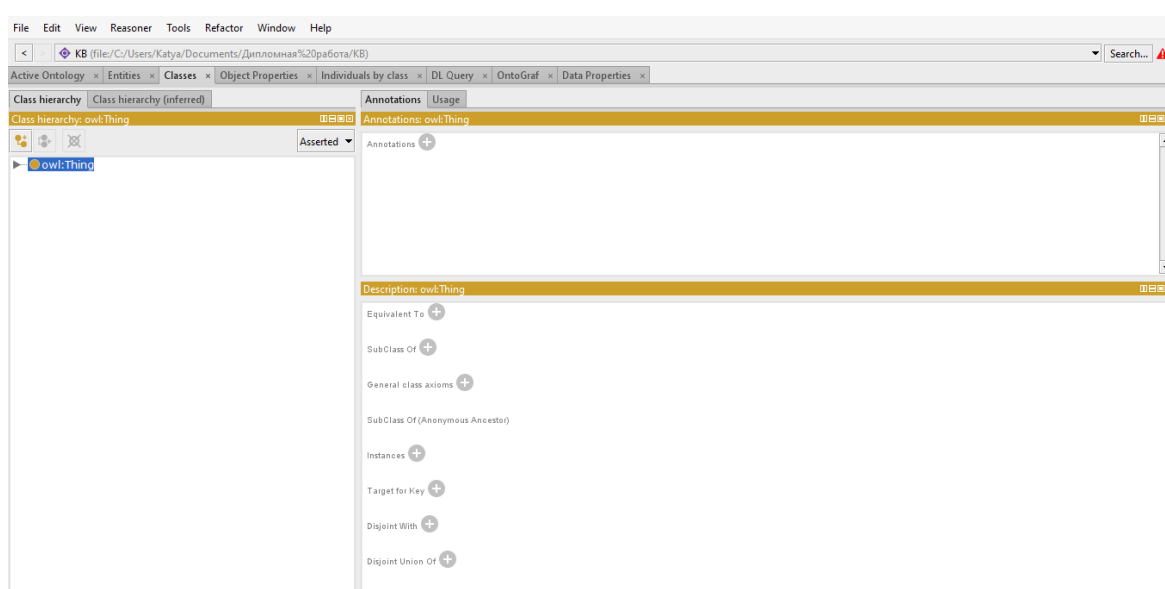


Рисунок 3.1 – Окно редакции классов

Фактически всегда вкладки с целью формирования компонентов онтологии вводят в себе 3 сферы редактирования:

- сфера формирования и редакции (Рисунок 3.2);

- сфера отображения(Рисунок 3.3);
- сфера статей(Рисунок 3.4). Что является очень удобным расположением.

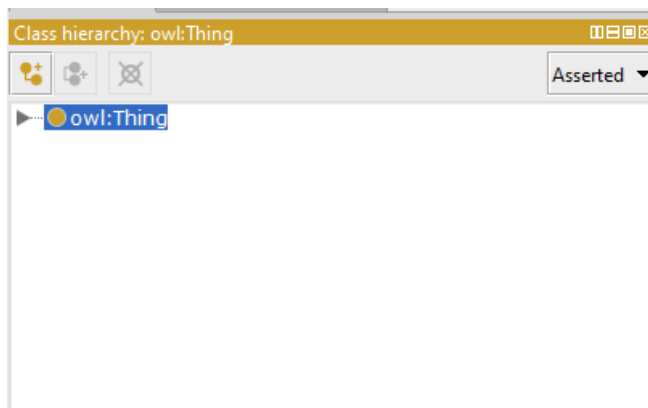


Рисунок 3.2 – Область создания и редактирования классов

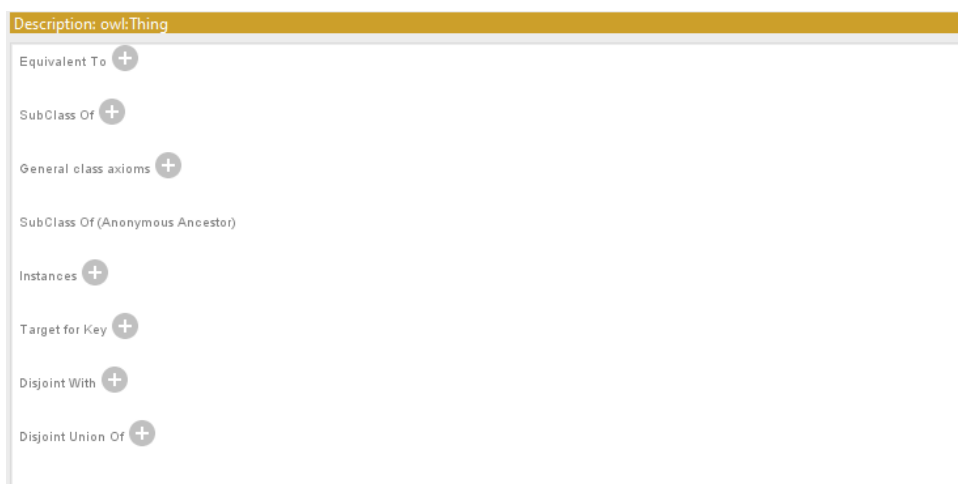


Рисунок 3.3 – Область декларирования классов

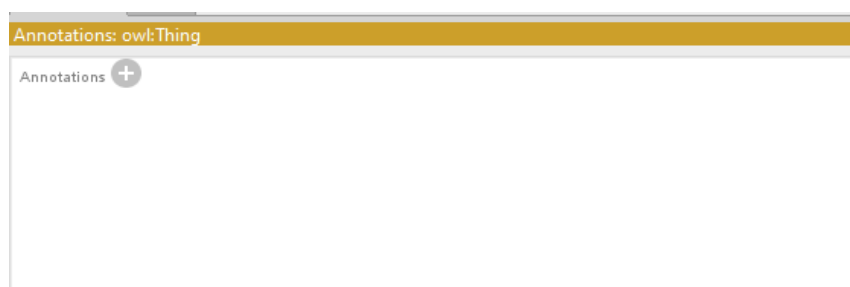


Рисунок 3.4 – Область аннотации для класса

Для того, чтобы создать класс “Компьютер” надо проделать следующее: надо выбрать закладку “Classes”, потом нажать “Добавить подкласс” (Рисунок 3.5). При нажатии этой кнопки создается новый класс как подкласс уже выбранного класса (в данном контексте как подкласс класса THING).

Покажется новое диалоговое окно для ввода названия нового класса, после ввода необходимо нажать Enter.

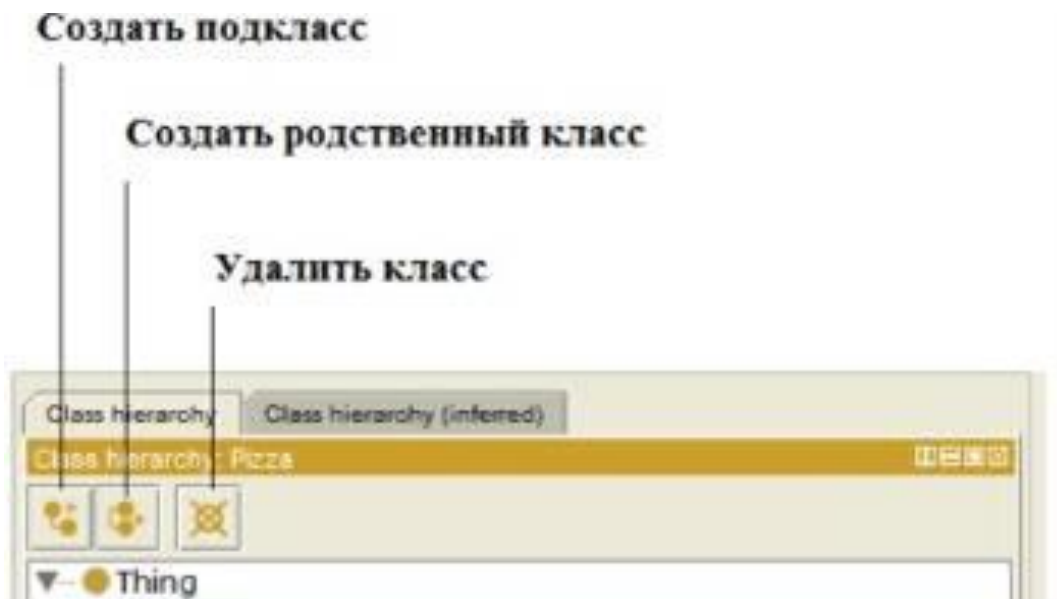


Рисунок 3.5 – Создание классов

На выводе получена данная иерархия классов (рисунок 3.6).

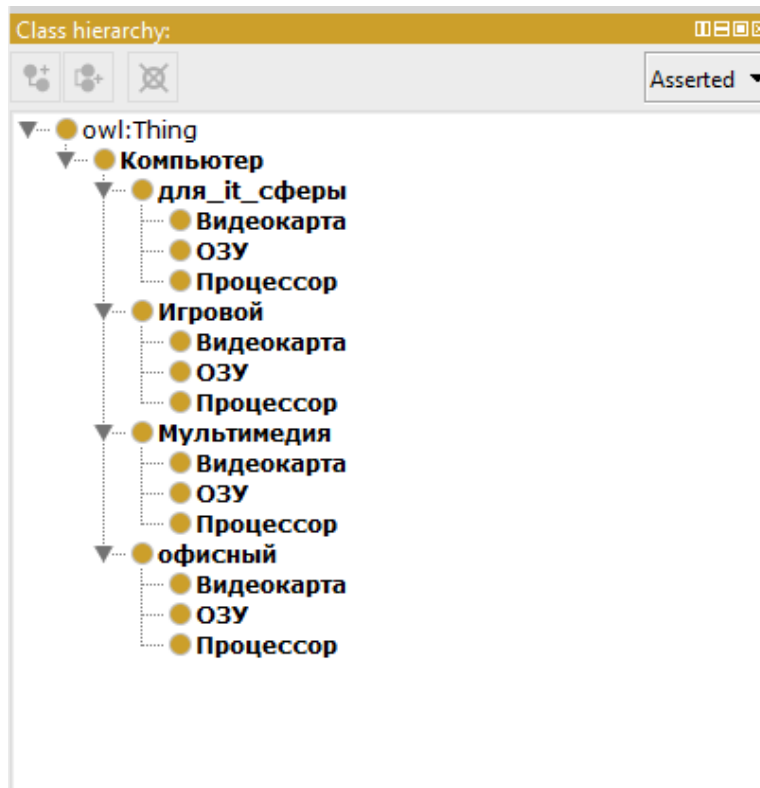


Рисунок 3.6 – полученная иерархия классов.

Также, сгенерированный вывод можно выстроить как граф в закладке “OntoGraf” (рисунок 3.7).

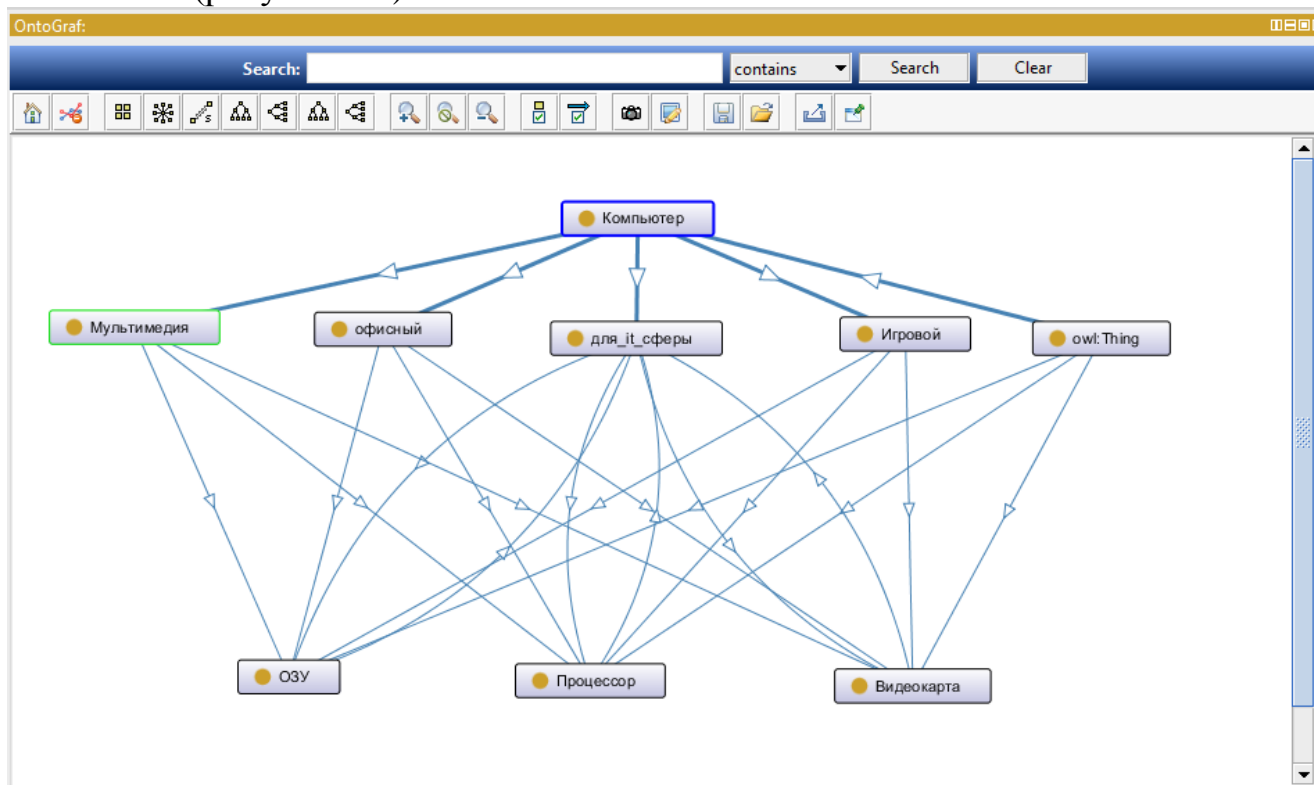


Рисунок 3.7 – Иерархия классов в виде графа.

### 3.2 Определение атрибутов класса и свойств экземпляров

Protege 5.0 имеет несколько основных типов объектов: свойства объекта (свойства объекта) и свойства типов данных (Properties). Свойства являются объектом отношений между этими двумя случаями.

В Protege 5.0 свойства автоматически по умолчанию объекты верхнего уровня, установленные (TopObjectProperties) по аналогии с классом. Для создания свойств объекта, перейдите в раздел "Object Properties".

Каждое свойство имеет ряд особенностей.

Функциональные свойства (Functional) – свойство которое обеспечивает функционал того что, например, через данное свойство первым лицом не может быть связан не более чем один человек.

Обратные функциональные свойства (Reverse Functional) - это означает, что свойство обеспечивает обратный функционал.

Переходная Свойство (Переходная) - Если свойство транзитивно свойство говорит, что, а и b индивида и личности, связанной с человеком, то можно сделать вывод, что человек не должен зависеть от человека через эту собственность. Симметричные свойства (Simmetric) – если свойство p симметричное, и свойство связывает индивида а с индивидом b, то индивид b связан также с индивидом а через свойство p.

Асимметричные свойства (Asymmetric) – если свойство  $p$  асимметричное, и свойство связывает индивида  $a$  с индивидом  $b$ , то индивид  $b$  не может быть связан с индивидом  $a$  через свойство  $p$ .

Отражённые характеристики (Reflexive) – качество  $p$  именуется отражённым, если человек  $a$  обязан являться сопряжен с собою.

Иррефлексивные характеристики (Irreflexive) – в случае если качество  $p$  иррефлексивное, в таком случае оно имеет возможность быть охарактеризовано равно как качество, что объединяет индивида  $a$  с индивидом  $b$ , в каком месте человек  $a$  и человек  $b$  непременно различные.

Помимо данного, любое качество предмета имеет возможность обладать надлежащее противоположное качество. В случае если определенное качество объединяет индивида  $a$  с определенным индивидом  $b$ , в таком случае его противоположное качество объединяет человек  $b$  с индивидом  $a$ .

В предоставленной труде в свойстве образца с целью классов онтологии станут сформированы последующие характеристики предметов: содержит такого рода ведь группа равно как (Всегда подклассы класса “Компьютер”), согласно области использования (Всегда подклассы класса “Компьютер”), обладают свойства в (Всегда подклассы класса “Компьютер”). В скобках показаны игра, чьи экземпляры станут владеть упомянутыми качествами. Окончательный список свойств объектов представлен на Рисунке 3.8.

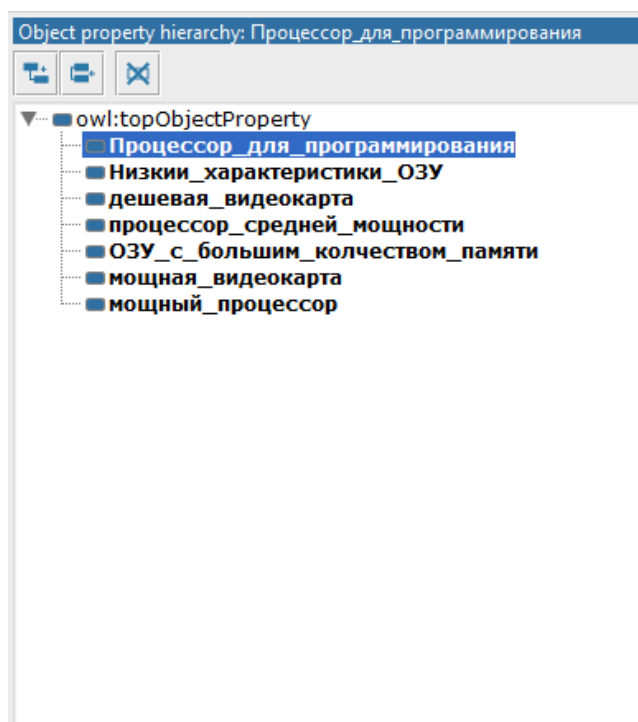


Рисунок 3.8 – Список свойств объектов.

Свойства типов данных (атрибуты класса) - описывают связи между индивидом и значения данных.



Для создания типовых параметров в Protégé 5.0 нужно перейти в закладку “Data Properties”. (Рисунок 3.9).

Процесс создания типовых параметров подобен созданию классов и параметров объекта.

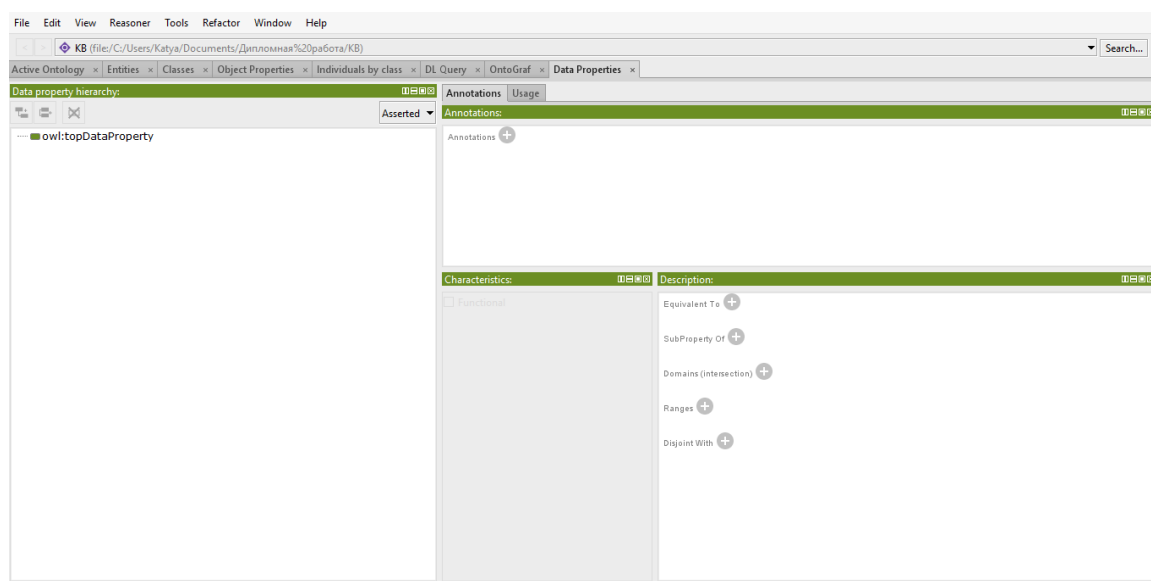


Рисунок 3.9 – Создание свойств для типов данных.

В виде образца для классов онтологии станут сделаны последующие характеристики типов этих: микропроцессор, видеоплата, ОЗУ, тактовая частота, размер памяти, разрядность микропроцессора.

Конечный перечень параметров типов этих представлен на Рисунке 3.10.

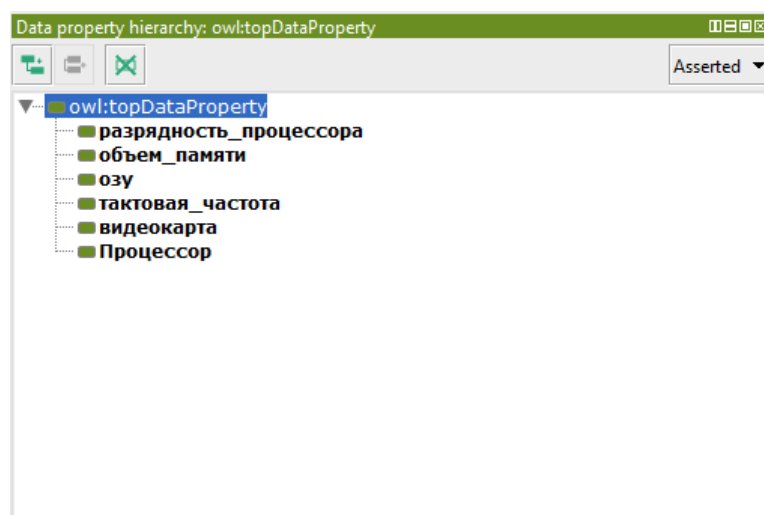


Рисунок 3.10 – Свойства для типов данных.

### 3.3 Создание экземпляров класса

Завершающим шагом в создании онтологии считается наполнения классов экземплярами.

Для создания экземпляров в Protégé 5.0 нужно перейти в закладку “Individuals by class”, надавить на иконку add individuals и установить имя экземпляра.

В масштабах этой работы было сотворено 35 экземпляров (рисунок 3.11), которые распределены по классам так: микропроцессор (5), видеоплата(7), ОЗУ (3), тактовая частота (8), размер памяти (10), разрядность микропроцессора (2).

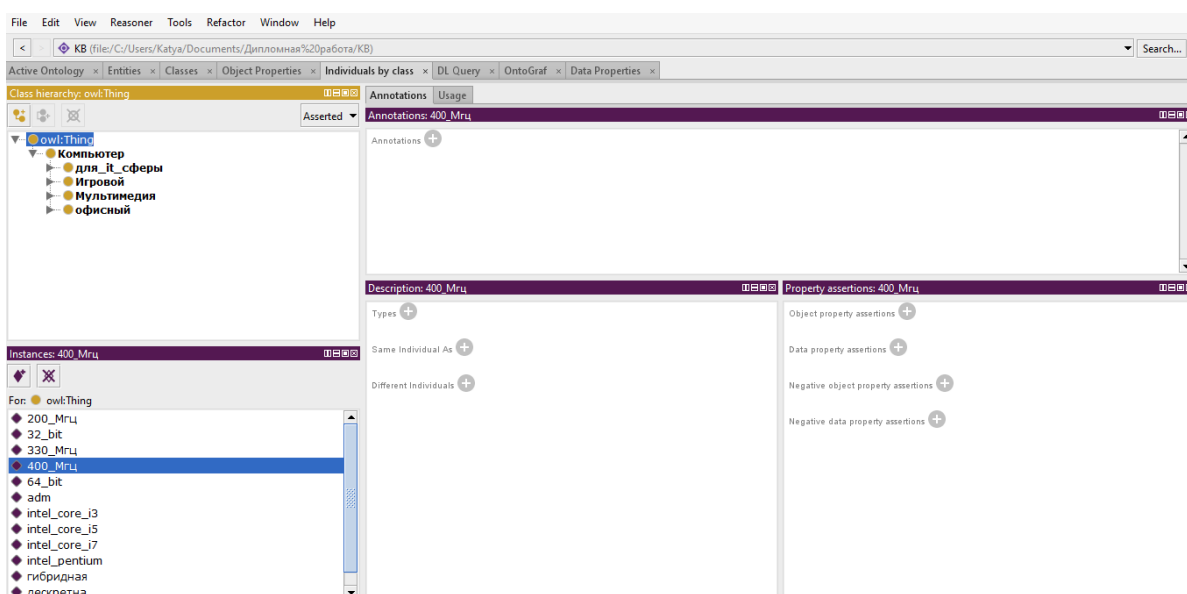


Рисунок 3.11 – Созданные экземпляры классов.

После создания экземпляров, необходимо задать свойства объектов между экземплярами классов. (Рисунок 3.12).

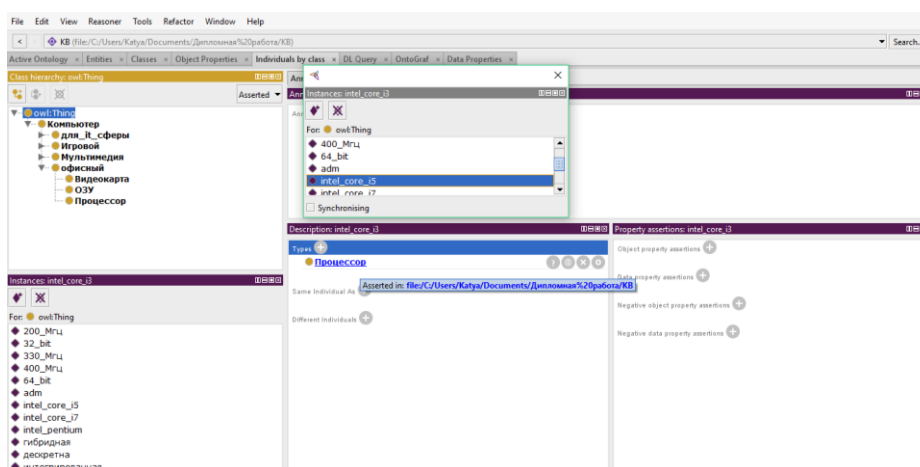


Рисунок 3.12 - Заполнение свойств объектов между экземплярами класса.

При заполнении атрибутов класса необходимо указывать тип данных, для данной онтологии для каждого атрибута используется тип данных “string”, так как это универсальный тип данных, позволяющий использовать как цифры, так и символы.

### 3.4 Создание DL-запросов

Последним шагом является создание конструкции основных DL-запросов. Результат запроса, и заполненная онтология может помочь в поиске информации, представляющей интерес.

Перейдите на вкладку запроса DL-запросов, а также вкладку "Reasoner", "Start Reasoner". (Рисунок 3.13).

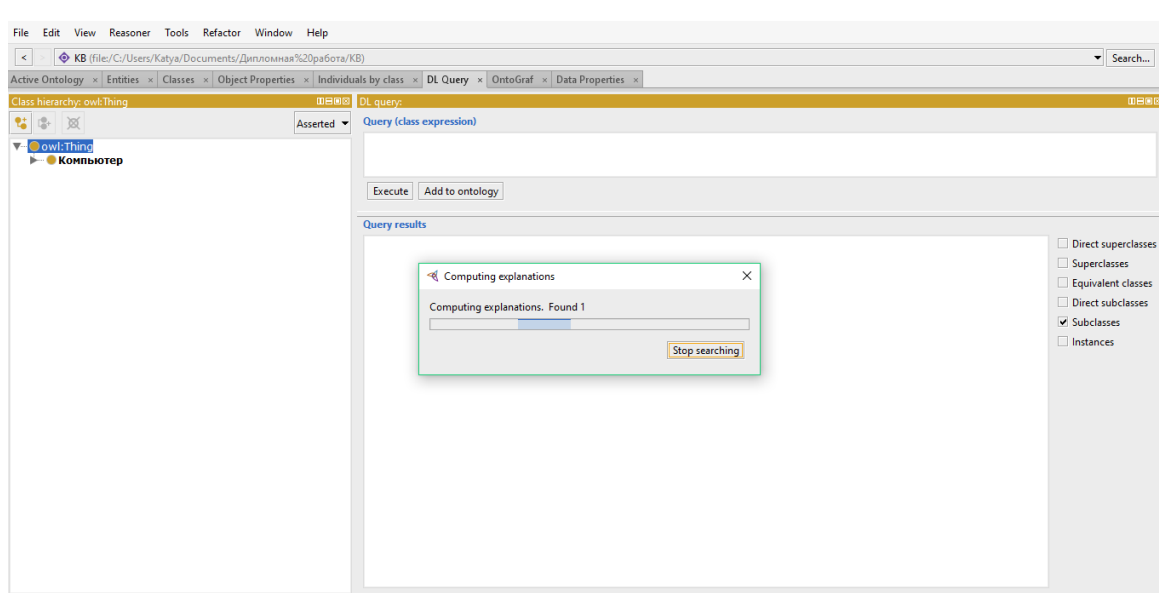


Рисунок 3.13 – Создание запросов

Например, мы желаем вывести все подклассы класса “офисный”. Для этого в запросном окне нужно написать наименования класса и выбрать на панели объектов “Descendant classes”. (Рисунок 3.14).

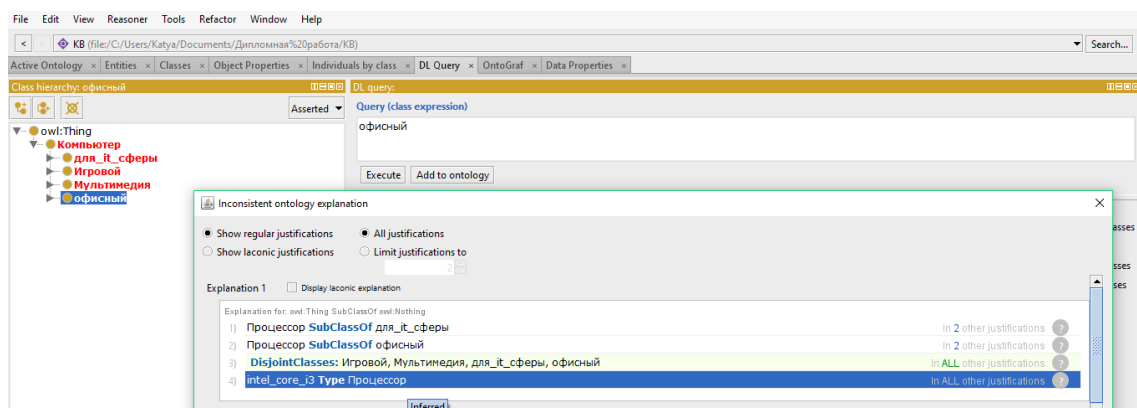


Рисунок 3.14 – Поиск подклассов класса Офисный

Примеры сформированных правил содержащихся в онтологии, выглядят следующим образом:

ЕСЛИ «компьютер для работы в офисе» и «для работы с изображениями», ТО «компьютер с хорошей видеокартой и не дорогой»

ЕСЛИ «компьютер игровой», ТО «компьютер с видеокартой игрового класса, процессор с частотой не менее 2,7 ГГц, оперативная память не менее 8ГБ со скоростью шины не менее 1600 МГц»

ЕСЛИ «компьютер для работы со звуком и для студии» ТО «компьютер с хорошей звуковой картой и процессором»

ЕСЛИ «компьютер для работы с видео», ТО «компьютер с хорошим процессором, видеокартой и объем оперативной памяти не менее 16ГБ, жесткий диск SSD».

Работа запросов онтологии в общем случае представляет собой повторяющийся процесс: решение задачи, запоминание этого решения в качестве прецедента, решение новейшей задачи и т.д.. Цикл может быть описан представленными ниже процессами (Рисунок 3.15).

Адаптация. Осуществляется решение новейшей задачи на базе имеющегося в системе прецедента. Во время этого процесса совершается проверка приобретенного решения на правильность и обработку ошибок. Вероятна некоторая корректировка решения.

Сохранение прецедента



Рисунок 3.15 – цикл работы онтологии.

### 3.5 Связка мобильного приложения, базы данных и базы знаний

Мобильное приложение является связующим звеном между базой знаний и базой данных. На основе информации хранящейся в базе данных формируются правила в базе знаний. В базе знаний же содержатся основные правила, по которым определяется выбор компьютера. Мобильное приложение выводит информацию с базы данных основываясь на правилах из базы знаний, посредством генерации html – документа, листинг приведен в Приложение А.



Рисунок 3.10 – Схема взаимосвязи частей приложения

## **4 Экономическая часть**

### **4.1 Цель и задачи проекта**

В любой профессиональной деятельности человека постоянно возникает большое количество наиболее сложных проблем, требующих оптимального решения.

Цель данной работы – разработать мобильное приложение, позволяющее помочь подобрать подходящий и выгодный вариант при покупке компьютера и тем самым облегчить выбор человека.

Проект включает следующие элементы разработки:

- мобильное приложение;
- экспертная система для выбора компьютера;
- разработка базы знаний;
- разработка базы данных.

В этой части работы осуществляются расчеты экономической составляющей реализации данного проекта, показывающие временные, трудовые и финансовые затраты на проект. Также в данном разделе представлено технико-экономическое обоснование разработки базы знаний для мобильного приложения по консультации клиентов компьютерного магазина. Экономическая часть разработки проекта содержит:

- анализ трудовых ресурсов, используемых в работе;
- анализ оборудования, используемого в работе;
- анализ программного обеспечения;
- расчет затрат на разработку системы.

### **4.2 Трудовые ресурсы, используемые в работе**

В данной дипломной работе задействованы:

1. Разработчики;
2. Руководитель проекта.

Количество сотрудников, задействованных в разработке мобильного приложения, включающего в себя базу знаний и базу данных, представлено в таблице 4.1.

Таблица 4.1 – Данные о работниках, задействованные в проекте, и их заработная плата

Наименование	Количество	Зарботная плата, тенге
Разработчик	3	100000
Руководитель проекта	1	90000
Итого	4	390000

### 4.3 Оборудование, используемое в работе

Оборудование, используемое при разработке ПП, представлено в таблице 4.2.  
Таблица 4.2 – Перечень оборудования, необходимое для разработки ПП

Наименование	Характеристики	Количество единиц	Цена за единицу, тенге
Системный блок	Intel Core i7/4,4GHz/4Gb/1000Gb/Radeon HD 7770/1Gb	1	180000
Монитор	Samsung 23EP75GGWP/23"/1920x1080/HDMIx2	1	75000
Ноутбук	Asus N550 (Intel Core i7 4200H(2.8 ГГц), 8 Gb DDR3L, GeForce GTX 850M, 1000 Гб, 15.6, Wi-Fi, Win 7)	1	175000
Всего			430000

Цены на оборудование приведены без учета НДС.

### 4.4 Сроки реализации проекта

Проектирование и разработка программного продукта состоит из определенных этапов и включает следующие виды работ, представленные в таблице 4.4:

1-й этап – постановка задач, сбор необходимой информации, разработка структуры дипломного проекта;

2-й этап – разработка 1 части проекта (анализ работы базы знаний экспертной системы);

3-й этап – разработка 2 части проекта (анализ, создание онтологии базы знаний);

4-й этап - разработка мобильного приложения;

5-й этап - тестирование системы;

6-й этап – оформление отчетов.



Таблица 4.4 – Этапы и сроки реализации проекта

Наименование этапа		Недели от начала работ									
		1	2	3	4	5	6	7	8	9	10
1 этап	Постановка задачи	■									
	Подбор и изучение литературы	■	■								
	Разработка структуры проекта		■								
2 этап	Анализ работы базы знаний экспертной системы			■	■						
3 этап	анализ, создание онтологии базы знаний					■					
4 этап	разработка мобильного приложения						■				
5 этап	Техническое описание теоретических аспектов базы знаний							■			
	Занесение знаний в экспертную систему, связка с базой данных								■		
	Тестирование системы									■	
6 этап	Проверка и сдача отчета										■

#### 4.5 Затраты на разработку системы

Затраты на разработку системы (производственные затраты) представляют собой единовременные расходы на всех этапах инновационного процесса: исследование, разработка, опытная проверка. Определение затрат на разработку проекта производится путем составления калькуляции плановой себестоимости. В плановую себестоимость включают все затраты, связанные с ее выполнением, независимо от источника их финансирования.

Вся стоимость разработки проекта определяется по формуле 4.1:

$$Cб = ФОТ + O_{сн} + A + Э + H, \quad (4.1)$$

где  $Cб$  – себестоимость;

$ФОТ$  – фонд оплаты труда;

$O_{сн}$  – отчисления на социальные нужды;

$A$  – амортизационные отчисления;

$Э$  – затраты на электроэнергию;

$H$  – накладные расходы.

#### 4.5.1 Расчет фонда оплаты труда

Фонд оплаты труда (ФОТ) – это суммарные издержки предприятия на оплату труда всех работников за определенный период, который формируется из основной и дополнительной заработной платы и определяется по формуле 4.2:

$$ФОТ = З_{осн} + З_{доп}, \quad (4.2)$$

где  $З_{осн}$  – основная заработная плата;

$З_{доп}$  – дополнительная заработная плата.

На этапах разработки ПП, участники разработки задействованы неравноценно, для этого необходимо рассчитать средний дневной заработок, а затем общий размер заработной платы.

Средний дневной заработок каждого работника рассчитывается по формуле 4.3:

$$D = \frac{ЗПм}{Др}, \quad (4.3)$$

где  $ЗПм$  – ежемесячный размер заработной платы, тенге;

$Др$  – количество рабочих дней в месяце (22 дня, при пятидневной рабочей недели).

1. Разработчик:  $D = \frac{100000}{22} = 4545$  тенге/день;

2. Руководитель:  $D = \frac{90000}{22} = 4090$  тенге/день;

Заработная плата за один час рассчитывается по формуле:

$$H = \frac{ЗПм}{Др \cdot Чр}, \quad (4.4)$$

где  $ЗПм$  – ежемесячный размер заработной платы, тенге;

$Др$  – количество рабочих дней в месяце (22 дня, при пятидневной рабочей недели);

$Чр$  – продолжительность рабочего дня, час (при 8-часовом рабочем дне).

1. Разработчик:  $D = \frac{100000}{22 \cdot 8} = 568$  тенге/день;

2. Руководитель:  $D = \frac{90000}{22 \cdot 8} = 511$  тенге/день;

Длительность цикла в днях по каждому виду работ определяется по формуле:

$$t_n = \frac{T}{q_n \cdot z \cdot K}, \quad (4.5)$$

где  $T$  – трудоемкость этапа, норма-час;

$q_n$  – количество исполнителей по этапу;

$z$  – продолжительность рабочего дня ( $z = 8$  часов);

$K$  – коэффициент выполнения норм времени,  $K = 1,1$ .

Полученную величину  $t_n$  округляем в большую сторону, до целых дней:

$t_1 = \frac{14}{1 \cdot 8 \cdot 1.1} \approx 2$  дня – руководитель (постановка задач);

$t_2 = \frac{21}{1 \cdot 8 \cdot 1.1} \approx 2$  дня – разработчик (подбор и изучение литературы);

$t_3 = \frac{21}{1 \cdot 8 \cdot 1.1} \approx 2$  дня – руководитель (подбор и изучение литературы);

$t_4 = \frac{28}{1 \cdot 8 \cdot 1.1} \approx 3$  дня – разработчик (разработка структуры проекта);

$t_5 = \frac{70}{1 \cdot 8 \cdot 1.1} \approx 8$  дней – разработчик (анализ принципов построения и средств мобильного приложения);

$t_6 = \frac{28}{1 \cdot 8 \cdot 1.1} \approx 3$  дня – разработчик (анализ, классификация и краткая характеристика баз знаний);

$t_7 = \frac{21}{1 \cdot 8 \cdot 1.1} \approx 2$  дня – разработчик (разработка онтологии базы знаний);

$t_8 = \frac{7}{1 \cdot 8 \cdot 1.1} \approx 1$  день – разработчик (техническое описание теоретических аспектов);

$t_9 = \frac{70}{1 \cdot 8 \cdot 1.1} \approx 8$  дней – разработчик (связка Базы знаний с интерфейсом);

$t_{10} = \frac{21}{1 \cdot 8 \cdot 1.1} \approx 2$  дня – разработчик (тестирование системы);

$$t_{11} = \frac{14}{1 \cdot 8 \cdot 1.1} \approx 2 \text{ дня} - \text{разработчик (проверка и сдача отчета);}$$

$$t_{12} = \frac{14}{1 \cdot 8 \cdot 1.1} \approx 2 \text{ дня} - \text{руководитель (проверка и сдача отчета).}$$

Результаты расчета затрат на основную заработную плату работников, задействованных в разработке базы знаний для мобильного приложения представлены в таблице 4.5.

Таблица 4.5 - Результаты расчета затрат на основную заработную плату

Наименование работ	Исполнитель	Трудоемкость		Длительность цикла, днях	Зарботная плата за час работы, тенге	Сумма заработной платы, тенге
		Нормо-час	% от общей трудоемкости			
1. Постановка задач	Руководитель	14	2,71	2	511	7154
2. Подбор и изучение литературы	Руководитель	21	4,07	2	511	10731
	Разработчик	21	6,79	2	852	17892
3. Разработка структуры проекта	Разработчик	28	9,05	3	852	23856
4. Анализ принципов построения и средств мобильного приложения	Разработчик	70	22,63	8	852	59640
5. Анализ, классификация и краткая характеристика баз знаний	Разработчик	28	9,05	3	852	23856
6. Разработка онтологии базы знаний	Разработчик	21	6,79	2	852	17892

7. Техническое описание теоретических аспектов	Разработчик	7	2,26	1	852	5964
--	-------------	---	------	---	-----	------

Продолжение таблицы 4.5

8. Связка Базы знаний с интерфейсом мобильного приложения	Разработчик	70	22,63	8	852	59640
9. Тестирование системы	Разработчик	21	6,79	2	852	17892
10. Проверка и сдача отчета	Разработчик	14	4,53	2	852	11928
	Руководитель	14	2,71	2	511	7154
Итого		329	100	52		263599

Дополнительная заработная плата составляет 10% от основной заработной платы и рассчитывается по формуле 4.6:

$$Z_{дон} = Z_{осн} \cdot 0,1, \quad (4.6)$$

$$Z_{дон} = 263599 \cdot 0,1 = 26359,9 \text{ тенге.}$$

Таким образом, суммарный фонд оплаты труда составит:

$$\text{ФОТ} = 263599 + 26359,9 = 289958,9 \text{ тенге.}$$

#### 4.5.2 Расчет затрат по социальному налогу

Социальный налог составляет 11% (ст.358 п.1 НК РК) от дохода работника, и рассчитывается по формуле 4.7:

$$C_{\text{н}} = (\text{ФОТ} - \text{ПО}) \cdot 0,11, \quad (4.7)$$

где ПО – пенсионные отчисления, которые составляют 10% от ФОТ и социальным налогом не облагаются.

$$\text{ПО} = \text{ФОТ} \cdot 0,1, \quad (4.8)$$

$$\text{ПО} = 289958,9 \cdot 0,1 = 28995,9 \text{ тенге.}$$

Размер отчислений на социальные нужды составит:

$$C_{\text{н}} = (289958,9 - 28995,9) \cdot 0,11 = 28705,93 \text{ тенге.}$$

### 4.5.3 Расчет амортизационных отчислений

Амортизационные отчисления рассчитываются по формуле 4.9:

$$A_i = \frac{H_A \cdot C_{\text{пер}} \cdot N}{100 \cdot 12 \cdot n} \quad (4.9)$$

где  $H_A$  - норма амортизации;

$C_{\text{пер}}$  - первоначальная стоимость оборудования;

$N$  – количество дней на выполнение работ;

$n$  – количество дней в рабочем месяце.

Норма амортизации  $H_A$  на компьютерную технику и на программное обеспечение составляет 40% от всей стоимости.

Амортизационные отчисления по используемому оборудованию и программному обеспечению составят:

$$A_1 = \frac{40 \cdot 180000 \cdot 52}{100 \cdot 12 \cdot 22} = 14181,8 \text{ тенге};$$

$$A_2 = \frac{40 \cdot 75000 \cdot 52}{100 \cdot 12 \cdot 22} = 5909,1 \text{ тенге};$$

$$A_3 = \frac{40 \cdot 175000 \cdot 52}{100 \cdot 12 \cdot 22} = 13787,8 \text{ тенге};$$

$$A = 14181,8 + 5909,1 + 13787,8 = 33878,7 \text{ тенге}.$$

### 4.5.4 Расчет затрат на электроэнергию

Так как в процессе производства используется электрооборудование, то необходимо рассчитать затраты на электроэнергию. Затраты на электроэнергию для производственных нужд включает в себя расходы электроэнергии на оборудование и дополнительные нужды.

$$\mathcal{E} = \mathcal{Z}_{\text{эл.оборуд}} + \mathcal{Z}_{\text{доп.нужды}} \quad (4.10)$$

где  $\mathcal{Z}_{\text{эл.оборуд}}$  – затраты на электроэнергию оборудования;

$\mathcal{Z}_{\text{доп.нужды}}$  - затраты электроэнергии на дополнительные нужды.

Расходы электроэнергии на оборудование рассчитывается по формуле 4.11:



$$Z_{\text{эл.оборуд}} = W \cdot T \cdot S \cdot K_{\text{исп}}, \quad (4.11)$$

где  $W$  - потребляемая мощность, Вт;

$T$  – количество часов работы оборудования;

$S$  - стоимость киловатт-часа электроэнергии (1кВтч = 16,02 тенге);

$K_{\text{исп}}$  - коэффициент использования ( $K_{\text{исп}} = 0,9$ ).

$$W = 90 \text{ Вт} = 0,09 \text{ кВт} \text{ (мощность ноутбука);}$$

$$T = 52 \cdot 5 = 260 \text{ часов;}$$

$$S = 16,02 \text{ тенге.}$$

Сумма затрат на электроэнергию основного оборудования составляет:

$$Z_{\text{эл.оборуд}} = 0,09 \cdot 260 \cdot 16,02 \cdot 0,9 = 337,4 \text{ тенге.}$$

Затраты на дополнительные нужды берутся по показателю в размере 5% от затрат на оборудование:

$$Z_{\text{доп.нужды}} = 0,05 \cdot Z_{\text{эл.оборуд}}, \quad (4.12)$$

$$Z_{\text{доп.нужды}} = 0,05 \cdot 337,4 = 16,8 \text{ тенге.}$$

Суммарные затраты на электроэнергию составят:

$$\mathcal{E} = 337,4 + 16,8 = 354,3 \text{ тенге.}$$

#### 4.5.5 Расчет накладных расходов

Накладные расходы составляют 50% от всех затрат и рассчитываются по формуле:

$$НР = (\Phi OT + C_n + A + \mathcal{E}) \cdot 0,5. \quad (4.13)$$

Накладные расходы согласно формуле 4.13 составят:

$$НР = (289958,9 + 28705,93 + 39630,3 + 354,3) \cdot 0,5 = 179324,72 \text{ тенге.}$$

Таким образом, в соответствии с формулой 4.1, суммарные затраты по разработке программного продукта составляют:

$$C_6 = 289958,9 + 28705,93 + 39630,3 + 354,3 + 179324,72 = 537974,2 \text{ тенге.}$$

Смета затрат по разработке базы знаний для мобильного приложения представлена в таблице 4.6 и на рисунке 4.1.

Таблица 4.6 – Стоимость разработки базы знаний для мобильного приложения

Наименование затрат	Сумма, тенге	Структура затрат, %
ФОТ	289958,9	53,9
Социальный налог	28705,93	5,34
Амортизационные отчисления	39630,3	7,37
Затраты на электроэнергию	354,3	0,07
Накладные расходы	179324,72	33,3
Итого	537974,2	100

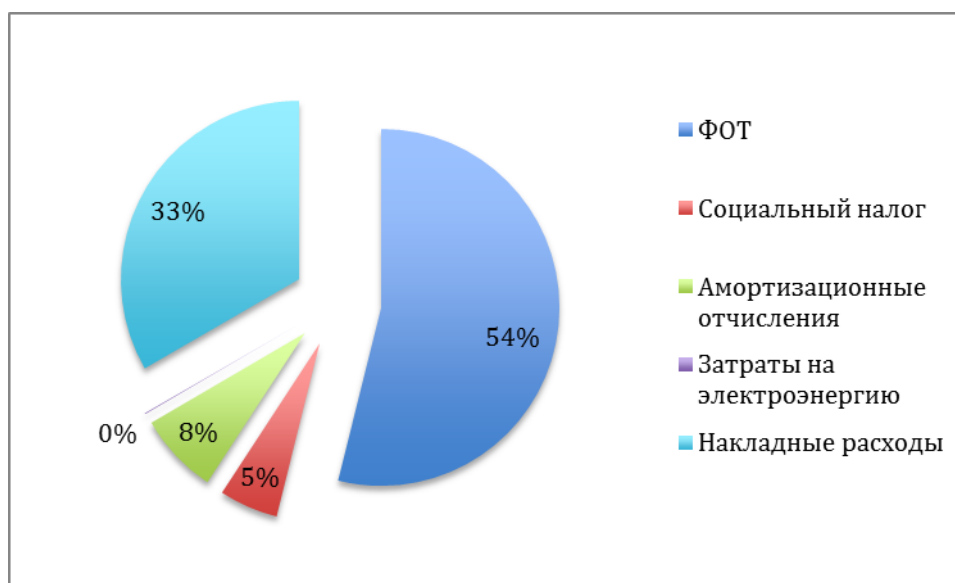


Рисунок 4.1 – Структура затрат по разработке базы знаний для мобильного приложения

#### 4.7 Цена реализации

Цена реализации продукта складывается из его стоимости и прибыли:

$$C = C_6 + П, \quad (4.14)$$

где  $C_6$  – стоимость продукта;

$П$  – прибыль.

При определении первоначальной цены следует задать уровень рентабельности (20%) для реализации программного продукта:

$$C_{\Pi} = C_{\sigma} \cdot (1 + P/100), \quad (4.15)$$

где P – рентабельность.

$$C_{\Pi} = 537974,2 \cdot (1 + 0,2) = 645569 \text{ тенге.}$$

Цена реализации готовой продукции рассчитывается по формуле:

$$C_P = C_{\Pi} + НДС, \quad (4.16)$$

где НДС – налог на добавочную стоимость.

НДС рассчитывается по формуле:

$$НДС = C_{\Pi} \cdot 0,12, \quad (4.17)$$

$$\begin{aligned} НДС &= 645569 \cdot 0,12 = 77468,3 \text{ тенге.} \\ C_P &= 645569 + 77468,3 = 723037,3 \text{ тенге.} \end{aligned}$$

#### **4.8 Оценка научно-технической результативности и социальной эффективности**

В настоящее время экспертные системы используются для решения различных типов задач в самых разнообразных проблемных областях.

В результате экономического расчета, затраты на разработку программного продукта составили 537974,2 тенге. Основной статьей расходов являются заработная плата работников, которые составляют 53,9% от всех затрат. На втором месте накладные расходы, которые составляют 33,3% от всех затрат. Цена реализации программного продукта – 723037,3 тенге.

Учитывая уникальность данного программного продукта и развитость торговой инфраструктуры в Казахстане, программный продукт будет очень конкурентоспособен. Возможно внедрение в магазины компьютерной техники, с целью частично заменить труд консультантов магазина. На данный момент существует множество магазинов компьютерной техники заинтересованных в автоматизации работы магазина.

Все чаще возникает потребность автоматизации труда, а также использования различных программных обеспечений с целью экономии времени и ресурсов. Посредством искусственного интеллекта стало возможным частично заменить знания человека в той или иной области знаний, для это в свою очередь необходимо наделить компьютер знаниями эксперта.



## **5 Безопасность жизнедеятельности**

### **5.1 Анализ условий труда в помещении серверной**

Работа серверной связана с присутствием на рабочем месте человека – оператора, вопросы охраны труда будут рассматриваться исходя из убеждений обеспечения неопасных критерий труда и критерий труда сохраняющих здоровье оператора, пользователя. При работе в помещении где размещается экспертная система по реализации компьютеров следует выделить последующие вредные и небезопасные причины:

- метеорологические условия среды (микроклимат);
- аномальное освещение;
- высокий уровень шума;
- опасность поражения электрическим током;
- пожароопасность.

Высокая температура содействует быстрому утомлению оператора, может привести к перегреву организма, вызывающего термический удар. Низкая температура может вызвать местное либо общее остывание организма, стать предпосылкой простудного заболевания. От каждого персонального компьютера исходит электромагнитное излучение: низкочастотное и радиочастотное. Проблема электромагнитного излучения, исходящего от персональных компьютеров, встает достаточно остро ввиду нескольких причин:

1. Компьютер имеет сразу два источника излучения (монитор и системный блок).

2. Пользователь ПК практически лишен возможности работать на расстоянии.

3. Очень длительное время воздействия.

Источником излучения в помещении с установленной ИС является монитор. При повышенном электромагнитном излучении у человека появляется головная боль, повышенная утомляемость, что снижает сосредоточенность работающего к работе, его внимание.

Электрическое оборудование в помещении считается одним из первых источников небезопасных моментов, потому что главное множество оснащения в нем электронное. Персонал подвержен завышенному риску удара электронным током. Поражение током имеет может случиться от неизолированной проводки, от корпуса системного блока, при неосмотрительной работы с оборудованием, его разборкой и т.п. Воздействие электронного тока на живой организм очень сильные. Проходя сквозь организм человека ток изготавливает тепловое, электронное, механическое и биологическое несоблюдение внутренних биоэлектрических процессов.

Персонал серверной наиболее подвержен воздействию статических вредных факторов. Чтобы не перенапрягать глаза оператор должен проводить у монитора не более 4 ч. в сутки.

Пожароопасность в помещении главным образом представлена оголенными токоведущими частями электропроводки, коротким замыканием проводки, перегрузки электросети, статическим электричеством. Что касается, причин возникновения пожара, не связанных с электричеством, то сюда можно отнести: неправильное устройство и эксплуатация отопительных систем (использование обогревателей), неисправность вентиляционных систем, неосторожное обращение с огнем персонала и др.

## **5.2 Расчет кондиционирования помещения серверной**

При работе серверного оборудования выделяется достаточно большое количество тепла. В замкнутом пространстве серверных комнат это может привести к повышению температуры воздуха до критической, и привести к остановке или сбою в работе серверного оборудования и как следствие, к сбою в работе всей компьютерной сети.

Учитывая стоимость размещенного оборудования, к системам кондиционирования предъявляются особые требования надежности и отказоустойчивости. Таким образом, для кондиционирования серверных помещений необходимо обеспечить резервирование системы охлаждения.

Основные требования к системам охлаждения:

- оборудование обеспечивает поддержание климатических параметров помещения 24 часа в сутки;
- возможность эксплуатации в зимний период, при температуре окружающей среды до -40 градусов;
- обеспечить высокую надежность, а соответственно резервирование системы;
- возможность поддержания необходимого уровня влажности.

При этом выполняются следующие требования:

- расчет тепловыделения размещаемого оборудования (серверы, сетевое оборудование, ИБП);
- помещение оборудуется системой кондиционирования воздуха необходимой мощности;
- поддержание влажности воздуха 30-55%, без конденсации влаги;
- поддержание заданной температуры.

Система кондиционирования воздуха, помимо выполнения задач вентиляции и отопления, позволяет создать благоприятный микроклимат в летний жаркий период года, благодаря использованию в своем составе фреоновой холодильной машины. Задача кондиционирования воздуха состоит в поддержании таких параметров воздушной среды, при которых каждый человек благодаря своей индивидуальной системе автоматической терморегуляции организма чувствовал бы себя комфортно, т.е. не замечал

влияния этой среды. Кондиционирование обеспечит соответствие климата в рабочем помещении нормативам .

Количество приточного воздуха  $L_{пр}$ , м<sup>3</sup>/ч определяем по формуле

$$L_{пр} = \frac{Q_{изб}}{c \cdot \rho_{пр} \cdot (t_{вып} - t_{пр})} , \quad (5.1)$$

где  $Q_{изб}$  – избыточное выделение явной теплоты, к Дж/ч;

$c$  – удельная теплоемкость воздуха при постоянном давлении, равная  $c=1$ к Дж/кг;

$\rho_{пр}$  – плотность поступающего в помещение воздуха, равная  $1,2$ кг/м<sup>3</sup>;

$t_{вып}$  – температура удаляемого из помещения воздуха за пределы рабочей или обслуживаемой зоны, °С;

$t_{пр}$  – температура приточного воздуха, °С.

Температура удаляемого воздуха  $t_{вып}$ , °С, определяется по формуле

$$t_{вып} = t_{рз} + \Delta t \cdot (h_{вп} - z), \quad (5.2)$$

где  $t_{рз}$  – температура в рабочей зоне, которая не должна превышать допустимую по нормам ( $t_{рз} \leq t_{доп}$ ), 0С;

$\Delta t$  – температурный градиент по высоте помещения ( $\Delta t=0,5-1,5$ ), °С;

$h_{вп}$  – расстояние от пола до центра вытяжных проемов (кондиционера), м;

$H$  – высота рабочей зоны, м.

Поскольку расчет производится для теплого периода года, то примем

$t_{рз} = 27$ 0С.

Внутренняя часть кондиционера расположена на высоте  $h_{вп}=2,5$  м

$$t_{вып} = 27 + 1,2 \cdot (2,5 - 3) = 26,4 \text{ } ^\circ\text{C}.$$

Температура приточного воздуха  $t_{пр}$  при наличии избытка явной теплоты должна быть на 5–7 0С ниже температуры воздуха в рабочей зоне

$$t_{пр} = 26,4 - 7 = 19,4 \text{ } ^\circ\text{C} .$$

Величину избыточного выделения явной теплоты ОИЗБ находят на основании баланса теплоты в помещении по формуле

$$O_{изб} = \Sigma Q - \Sigma Q_{ух}, \quad (5.3)$$

где  $\Sigma Q$  – суммарное количество поступающей в помещение явной теплоты;

$\Sigma Q_{ух}$  – суммарное количество уходящей из помещения теплоты (за счет тепло потерь ограждениями, нагрева поступающего в помещение воздуха и т. п.). Основными источниками избыточного тепла являются светильники, люди



и др. Кроме того, необходимо учитывать тепlopоступления от солнечной радиации.

Тепловыделения от искусственного освещения  $Q_2$ , рассчитывают, предполагая, что практически вся затрачиваемая энергия, в конечном счете, преобразуется в тепло, по формуле

$$Q_2 = 1000 * N, \quad (5.4)$$

где  $N$  – расходуемая мощность светильников, кВт

$$Q_2 = 1000 * 0,08 = 80 \text{ кВт.}$$

Тепловыделения от людей  $Q_3$  определяют по формуле

$$Q_3 = n * q_{\text{ч}}, \quad (5.5)$$

где  $n$  – число работающих;

$q_{\text{ч}}$  – количество тепла, выделяемое одним человеком.

$$Q_3 = 1 * 145 = 145 \text{ Вт.}$$

Количество тепла, поступающего в помещение от солнечной радиации  $Q_{\text{ост.рад}}$ , определяют по формуле

$$Q_{\text{ост.рад}} = F_{\text{ост}} * q_{\text{ост}} * A_{\text{ос}}, \quad (5.6)$$

для покрытий

$$Q_{\text{п.рад}} = F_{\text{п}} * q_{\text{п}} * A_{\text{п}}, \quad (5.7)$$

где  $F_{\text{ост}}$  и  $F_{\text{п}}$  – площадь поверхности и покрытия,  $\text{м}^2$ ;

$q_{\text{ост}}$  и  $q_{\text{п}}$  – тепlopоступления через  $1\text{м}^2$  поверхности остекления и поверхности покрытия, при коэффициенте теплопередачи, равном  $\text{Вт}/\text{м}^2 \cdot ^\circ\text{С}$ ,  $\text{Вт}/\text{м}^2$ ;

$A_{\text{ост}}$  – коэффициент остекления;

$k_{\text{п}}$  – коэффициент теплопередачи покрытия,  $\text{Вт}/\text{м}^2 \cdot ^\circ\text{С}$ .

Значение  $q_{\text{ост}}$  в зависимости от географической ориентации поверхности и характеристики окон или фонарей принимается в пределах 70 – 210, а коэффициента  $A_{\text{ост}}$  в зависимости от вида остекления и его солнцезащитных свойств – в пределах 0,25–1,25, средние значения тепlopоступления от солнечной радиации через покрытие в зависимости от географической широты и вида покрытия принимают в пределах 6–24

$$F_{\text{ост}} = 1,5 * 1,2 * 2 = 3,6 \text{ м}^2.$$

Окно рабочего помещения направлено строго на восток, поэтому примем значение  $q_{\text{ост}}$  равным  $140 \text{ Вт}/\text{м}^2 \cdot ^\circ\text{С}$ . Примем  $A_{\text{ост}} = 0,35$ ,

$$Q_{\text{ост.рад}} = 3,6 * 140 * 0,35 = 176,4 \text{ Вт.}$$

Среднее значение теплоступления для покрытия с учетом географической широты примем равным

$$Q_{\text{п.рад}}=18 \text{ Вт.}$$

Потери тепла из помещения  $Q_{yx}$ , кВт через стены двери, окна оценивают ориентировочно по формуле

$$Q_{yx}=\lambda*S*(t_{\text{вып}}-t_{\text{пр}})\delta, \quad (5.8)$$

где  $\lambda$  – теплопроводность стен, Вт/м.<sup>0</sup>С;

$S$  – площадь, м<sup>2</sup>;

$\delta$  – толщина стен, м.

Стены рабочего помещения изготовлены из тяжелого бетона М600, теплопроводность которого равна 1,2 Вт/м.<sup>0</sup>С. Толщина стен  $\delta = 0,5$  м

$$Q_{yx} = \frac{1,2*24*(26,4-19,4)}{0,5} = 403,2 \text{ Вт.}$$

Вычислим суммарное количество поступающей в помещение явной теплоты

$$\Sigma Q=Q_2+Q_3+Q_{\text{ост.рфд}}+Q_{\text{п.рад}}, \quad (5.9)$$

$$\Sigma Q=80000+145+176,4+18=80339,4 \text{ Вт.}$$

Величина избыточного выделения явной теплоты

$$Q_{\text{изб}}=80339,4-403,2=79936,2 \text{ Вт.}$$

Вычислим количество приточного воздуха

$$L_{\text{пр}} = \frac{79936,21}{1*1,2*(26,4 - 19,4)} = 9516,012 \text{ м}^3/\text{ч.}$$

В соответствии с расчетами система кондиционирования должна обладать следующими параметрами:

- оборудование обеспечивает поддержание климатических параметров помещения 24 часа в сутки;

- возможность эксплуатации в зимний период, при температуре окружающей среды до -40 градусов;

- обеспечить высокую надежность, а соответственно резервирование системы;

- возможность поддержания необходимого уровня влажности.

Самое главное требование к системе кондиционирования серверных и подобных помещений — надежность. Исходя из расчетов для серверной нам необходим прецизионный кондиционер (рисунок 5.1), имеющий ресурс минимум в 5-6 раз больше, чем у бытового кондиционера. Прецизионный кондиционер - это автономный кондиционер шкафного типа для точного поддержания параметров воздуха по температуре, относительной влажности, подвижности воздуха в обслуживаемом помещении.



Рисунок 5.1 - прецизионный кондиционер для серверной

### **5.3 Расчет мер защиты от поражения электрическим током**

Заземление – устранение опасности поражения током в случае прикосновения к корпусу электроустановки и другим нетоковедущим металлическим частям, оказавшимся под напряжением вследствие замыкания на корпус и по другим причинам. Как известно, контур заземления состоит из вертикальных заземлителей, горизонтальных заземлителей и заземляющего проводника. Вертикальные заземлители вбиваются в почву на определенную глубину. Горизонтальные заземлители соединяют между собой вертикальные заземлители. Заземляющий проводник соединяет контур заземления непосредственно с электрощитом. Размеры и количество этих заземлителей, расстояние между ними, удельное сопротивление грунта – все эти параметры напрямую зависят от сопротивления заземления.

Работу необходимо проводить в соответствии с правилами технической эксплуатации электроустановок. Классификация помещений по опасности поражения электрическим током, по квалификации персонала и напряжению сетей. Также необходимы вводный и периодический инструктажи по технике безопасности при работе с электрическими приборами, соблюдать трудовую дисциплину, правильно организовать рабочее место.

Шины защитного заземления расположены в местах, доступных для проверки и сохранности. Технические меры защиты. Для защиты от прикосновения к токоведущим частям оборудования и приборов используют изоляцию, недоступное расположение токоведущих частей и ограждение. Для

защиты от поражения электрическим током при прикосновении к металлическим частям оборудования, которые могли случайно оказаться под напряжением, выполнено защитное заземление корпуса установки,  $R < 4$  Ом.

Сопротивление одиночного заземлителя определяется по формуле

$$R_{\text{оз}} = 0,366 * \rho * (\lg(2 * L / d) + 0,5 * \lg[(2 * 4 * t + L) / (4 * t * L)]) / L, \quad (5.10)$$

где  $R_{\text{оз}}$  – сопротивление одиночного заземлителя, Ом;

$\rho$  – удельное сопротивление грунта, Ом.м;

$L$  – длина заземлителя, м;

$t$  – глубина заложения заземлителя, м;

$d$  – диаметр заземлителя, м.

В используемой установке применяются заземлители из угловой стали, для которых определяется эквивалентный диаметр

$$d_{\text{экв}} = 0,95 * b, \quad (5.11)$$

где  $b$  – ширина сторон уголка, м

$$d_{\text{экв}} = 0,95 * 0,05,$$

$$d_{\text{экв}} = 0,0475 \text{ м.}$$

Формула (5.10) для заземлителя из угловой стали примет вид

$$R_{\text{оз}} = 0,366 * \rho * (\lg(2 * L / d_{\text{экв}}) + 0,5 * \lg((2 * 4 * t + L) / (2 * 4 * t * L))) / L, \quad (5.12)$$

$$R_{\text{оз}} = 0,366 * 50 (\lg(2 * 2,5 / 0,0475) + 0,5 * \lg((2 * 4 * 70 + 2,5) / (2 * 4 * 70 * 2,5))) / 2,5.$$

Из формулы 5.12 следует

$$R_{\text{оз}} = 17,5 \text{ Ом.}$$

Количество заземлителей определяется по формуле

$$n = R_{\text{оз}} / R_{\text{зн}}, \quad (5.13)$$

где  $n$  – число заземлителей, шт;

$R_{\text{оз}}$  – сопротивление одиночного заземлителя, Ом;

$R_{\text{зн}}$  – сопротивление заземлителя по нормам (4 Ом).

Расстояние между заземлителями рассчитывается по формуле

$$a = 2 * L, \quad (5.14)$$

где  $a$  – расстояние между заземлителями, м;

$L$  – длина заземлителя, м;

$$a = 2 * 2,5 = 5 \text{ м;}$$

Число заземлителей с учетом их взаимного экранирования определяется по формуле

$$n_3 = n/\eta_{эз}, \quad (5.15)$$

где  $n_3$  – число заземлителей с учетом их взаимного экранирования, шт.;

$n$  – число заземлителей без учета их взаимного экранирования, шт.;

$\eta_{эз}$  – коэффициент использования заземлителей учитывающий их взаимное экранирование

$$n_3 = 4/0,88,$$

$$n_3 = 5 \text{ шт.}$$

Длина заземляющих проводников определяется по формуле

$$L_n = 1,05 * a * n_3, \quad (5.16)$$

где  $L_n$  – длина заземляющих проводников, м;

$a$  – расстояние между заземлителями;

$n_3$  – число заземлителей с учетом их взаимного экранирования.

$$L_n = 1,05 * 5 * 5; L_n = 26,25 \text{ м.}$$

Сопротивление заземляющего проводника находится по формуле

$$R_{пп} = 0,36 * \rho * (\lg(2 * L_n / b * t)) / L, \quad (5.17)$$

где  $R_{пп}$  – сопротивление заземляющего проводника, Ом;

$L_n$  – длина заземляющих проводников, м;

$b$  – ширина полосы заземляющего проводника, м;

$t$  – глубина заложения заземлителя, м.

$$R_{пп} = 0,366 * 50 * (\lg(2 * 26,25 / 0,05 * 0,7)) / 2,5,$$

$$R_{пп} = 3,89 \text{ Ом.}$$

Действительное сопротивление заземляющего проводника с учетом взаимного экранирования заземлителей и заземляющих проводников находится по формуле

$$R_{пп} = R_{пп} / \eta_n, \quad (5.18)$$

где  $R_{пп}$  – действительное сопротивление заземляющего проводника, Ом;

$R_{пп}$  – сопротивление заземляющего проводника, Ом;

$\eta_n$  – коэффициент использования заземляющего проводника.

$$R_{пп} = 3,89 / 0,89,$$

$$R_{пп} = 4,37 \text{ Ом.}$$

Сопротивление растеканию тока всего заземляющего устройства находится по формуле

$$R_{зп} = R_{оз}R_{пп} / (R_{оз}*\eta_n + R_{пп}*\eta_{эз}*n), \quad (5.19)$$

где  $R_{зп}$  – сопротивление растеканию тока всего заземляющего устройства, Ом .

$$R_{зп} = 17,5*4,37 / (17,5*0,89 + 4,37*0,88.4),$$

$$R_{зп} = 2,47 \text{ Ом.}$$

Действительное число заземлителей определяется по формуле

$$n = R_{оз} / \eta_{эз} \cdot R_{зп}, \quad (5.20)$$

где  $n$  – действительное число заземлителей, шт

$$n = 17,5 / (0,88 * 2,47) = 8 \text{ шт.}$$

Схема заземляющего устройства на рисунке 5.2.

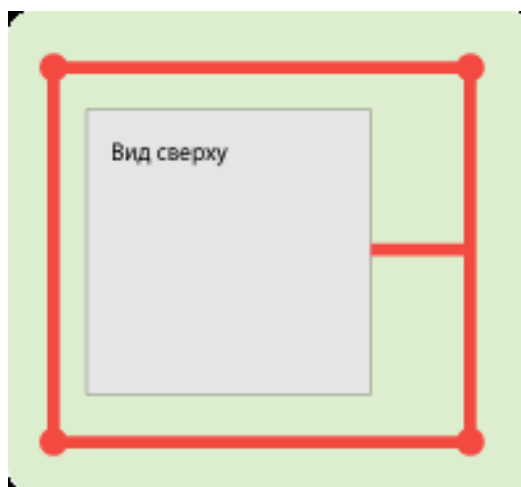


Рисунок 5.2 – Схема заземляющего устройства (ЗУ)

Вывод. Проектируемый заземлитель состоит из 8 вертикальных стержней из угловой стали длиной 2,5 м с эквивалентным диаметром 0,047 м, расположенных друг от друга на расстоянии 5 м и стального горизонтального электрода длиной 26,25 м, заглубленных в землю на 0,7 м. Общее сопротивление грунтового заземлителя 2,47 Ом, что обеспечивает безопасность работы на оборудовании. Учитывая стоимость размещенного оборудования, к системам кондиционирования предъявляются особые требования надежности и отказоустойчивости. Таким образом, для

кондиционирования серверных помещений необходимо обеспечить резервирование системы охлаждения.

### **Заключение**

Результатом дипломной работы является разработанная база знаний для мобильного приложения, осуществляющего консультацию при покупке компьютерной техники. В ходе выполнения дипломного проекта был выполнен обзор предметной области, на основе которой в дальнейшем созданы классы базы знаний, а также был проанализирован рынок компьютерной техники. После выбора тематики и категории были проанализированы все подобные приложения, выявлены все недостатки конкурентов и пожелания пользователей.

Непосредственно, была изучена технология создания базы знаний с помощью редактора онтологий, разработаны правила базы знаний с помощью редактора. На последнем этапе разработки была произведена связка базы знаний с мобильным приложением.

База знаний включает в себя более 30 правил. А также является легко расширяемой, поэтому в дальнейшем возможно увеличение числа правил и добавление нового функционала в существующую программу.

После внедрения в магазины, исходя из потребностей покупателей, возможно внесение поправок в само мобильное приложение, что влечет за собой появление новых функций, знаний и фактов в базе знаний.





## Список использованной литературы

1. Джексон П. Введение в экспертные системы. 3-е издание. - М.: Издательский дом "Вильямс", 2001. - 624 с.
2. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. СПб: Питер 2000.
3. Попов Э.В., Фоминых И.Б., Кисель Е.Б., Шамот М.Д. Статические и динамические экспертные системы // Москва, Финансы и статистика, 1996.
4. Вебсайт – <https://ru.wikipedia.org/>
5. Вебсайт – <http://www.intuit.ru/>
6. Вебсайт – <http://www.alser.kz/>
7. Вебсайт – <http://sibac.info/conf/tech/xl/39974>
8. Рыбина Г.В. Автоматизированное построение интегрированных экспертных систем для статических проблемных областей (проект АТ-технология): некоторые итоги и результаты // КИИ'98 Шестая национальная конференция по искусственному интеллекту с международным участием, 1998.
9. A Practical Guide To Building OW: Ontologies Using Protégé 4 and CO-ODE Tools: практический справочник / Matthew Horridge – The University of Manchester: 2009. – 109 с.
10. Ontology Development 101: A Guide to creating Your First Ontology: практический справочник / Natalya F. Noy, Deborah L. McGuinness – Stanford University: 2001. – 25 с.
11. Онтологический инжиниринг знаний в системе Protégé: методическое пособие / Д.И. Муромцев – СПб:СПб ГУ ИТМО: 2007. – 62 с.
12. Овдей О.М. Обзор инструментов инженерии онтологий / О.М. Овдей, Г.Ю. Проскудина // Журнал ЭБ. – 2004 – №4.
13. Формирование прикладных онтологий / Гаврилова Т.А. Труды XX национальной конференции по Искусственному Интеллекту с международным участием - КИ-2006, т. 2 - Москва: изд-во Физматлит, 2006 г.
14. Байзакова А.А., Санатова Т.С. Охрана труда: Методические указания к выполнению расчетно-графических работ. – 2015
15. Белов С.В., Ильницкая А.В., Козьяков А.Ф. и др. Безопасность жизнедеятельности. Учебник для вузов. – М.: Высш.шк., 2005. – 448 с.
16. Хакимжанов Т.Е. Сборник задач по охране труда и безопасности жизнедеятельности: Учебное пособие для вузов. – Алматы: Эверо, 2007. – 274с.
17. Сивков В.П., Смирнов С.Г., Козьяков А.Ф. и др. Сборник типовых расчетов по курсу «Охрана труда». – М.: МВТУ, 1979. – 79
18. Абикинова А.А., Методические указания к выполнению раздела «Пожарная профилактика» - Алматы: АИЭС, 2009.

19. СНиП РК 4.02–42–2006 Отопление, вентиляция и кондиционирование. – Астана: Издательство стандартов, 2007.

20. Стюарт Рассел, Питер Норвиг. Искусственный интеллект: современный подход 2-е изд.: пер. с англ. –М.: Изд.дом «Вильямс», 2006. – 1408с.: ил. Парал.тит.англ.

21. Муромцев Д.И. Онтологический инжиниринг знаний в системе Protégé. – СПб: СПбГУИТМО, 2007.-62с.

22. Веб-сайт [http://sherdim.rsu.ru/pts/semantic\\_web/REC-owl-guide-20040210\\_ru.html](http://sherdim.rsu.ru/pts/semantic_web/REC-owl-guide-20040210_ru.html) OWL, язык веб-онтологий. Руководство. Рекомендация W3C 10 февр. 2004.: пер. с англ. – Д. Щербин, 2004.

## Приложение А

Листинг кода связки мобильного приложения и базы знаний

```
<?php
header("Access-Control-Allow-Origin: *");
header("Content-Type: application/json; charset=UTF-8");
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "devices";

#Создание подключения к бд
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

# Запрос БД
$sql
="SELECT
cpm.m_cpu_freqMin_ghz,dev.dev_prod,dev.dev_model,prod.prod_name,cpm.m_cp
u_model,
          rm.m_ram_freq_ghz,          rm.m_ram_memory_gb,
pd.prod_name,vdm.m_vid_model,vdm.m_vid_memory_gb,vdm.m_vid_by
te,dev.dev_price_tg, DIS.dis_diag_inch, DIS.dis_res_x,
DIS.dis_res_y, DIS.dis_matrix, DIS.dis_type, DIS.dis_sensor, STG.m_stor_type,
STG.m_stor_memory_gb, AUD.m_aud_technology, OS.os_name, OS.os_bit,
PRTS.m_po_descrip, dev.dev_img from device dev join cpu_m cpm on
cpm.m_cpu_id=dev.m_cpu_id join
ram_m rm on rm.m_ram_id=dev.m_ram_id join videocard_m vdm on
vdm.m_vid_id=dev.m_vid_id join producer prod on prod.prod_id=cpm.prod_id join
producer pd on pd.prod_id=vdm.prod_id join display DIS on
DIS.dis_id=dev.dis_id join storage STG on
STG.m_stor_id=dev.m_stor_id join audio AUD on AUD.m_aud_id=dev.m_aud_id
join os OS on OS.os_id=dev.os_id join ports PRTS on
PRTS.m_po_id=dev.m_po_id where dev.dev_type='Laptop';
$result = $conn->query($sql);
$records=array();
# Переменные для формулы
$ram_memory=array();
$cpu_freq=array();
```

```

$ram_freq=array();
$vid_memory=array();
$vid_bus=array();

```

*Продолжение приложения А*

```

$vyvod=array();
# ВЫВОД с базы данных
if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        $records[]=$row;
    }
}

# КОНСТАНТЫ(знания)
# OFFICE LAPTOP
$ram_memory_const = 4;
$ram_freq_const = 1.3;
$cpu_freq_const= 1.3;
$vid_memory_const = 2;
$vid_bus_const = 16;
$office_step
=
(($ram_memory_const/3)+$ram_freq_const+$cpu_freq_const+$vid_memory_const
+$vid_bus_const)/5;
# ЦИКЛ ВЫБОРКИ

for ($i = 0; $i < count($records); $i++) {
    $ram_memory[$i] = ($records[$i]['m_ram_memory_gb']);
    $cpu_freq[$i] = ($records[$i]['m_cpu_freqMin_ghz']);
    $ram_freq[$i] = ($records[$i]['m_ram_freq_ghz']);
    $vid_memory[$i] = ($records[$i]['m_vid_memory_gb']);
    $vid_bus[$i] = ($records[$i]['m_vid_byte']);
    $out_office[$i] = (((($ram_memory_const + $ram_memory[$i])/3) +
($ram_freq_const + $ram_freq[$i]) + ($cpu_freq_const + $cpu_freq[$i]) +
($vid_memory_const + $vid_memory[$i]) + ($vid_bus_const + $vid_bus[$i]))/10;
    if ($office_step>=$out_office[$i]) {
        $vyvod[$i] = $records[$i];
    }
}
echo json_encode(array_values($vyvod));

```

```
$conn->close();  
?>
```

### *Продолжение Приложения А*

```
<?php  
header("Access-Control-Allow-Origin: *");  
header("Content-Type: application/json; charset=UTF-8");  
$servername = "localhost";  
$username = "root";  
$password = "";  
$dbname = "devices";  
  
// Create connection  
$conn = new mysqli($servername, $username, $password, $dbname);  
// Check connection  
if ($conn->connect_error) {  
    die("Connection failed: " . $conn->connect_error);  
}  
  
# Запрос БД  
$sql = "SELECT  
cpm.m_cpu_freqMin_ghz,dev.dev_prod,dev.dev_model,prod.prod_name,cpm.m_cp  
u_model,          rm.m_ram_freq_ghz,          rm.m_ram_memory_gb,  
pd.prod_name,vdm.m_vid_model,vdm.m_vid_memory_gb,vdm.m_vid_byte,dev.de  
v_price_tg, DIS.dis_diag_inch, DIS.dis_res_x, DIS.dis_res_y, DIS.dis_matrix,  
DIS.dis_type, DIS.dis_sensor, STG.m_stor_type, STG.m_stor_memory_gb,  
AUD.m_aud_technology, OS.os_name, OS.os_bit, PRTS.m_po_descrip from  
device dev join cpu_m cpm on cpm.m_cpu_id=dev.m_cpu_id join ram_m rm on  
rm.m_ram_id=dev.m_ram_id          join          videocard_m          vdm          on  
vdm.m_vid_id=dev.m_vid_id join producer prod on prod.prod_id=cpm.prod_id join  
producer pd on pd.prod_id=vdm.prod_id join display DIS on DIS.dis_id=dev.dis_id  
join storage STG on STG.m_stor_id=dev.m_stor_id join audio AUD on  
AUD.m_aud_id=dev.m_aud_id join os OS on OS.os_id=dev.os_id join ports PRTS  
on PRTS.m_po_id=dev.m_po_id where dev.dev_type='Laptop';  
$result = $conn->query($sql);  
$records=array();  
# Переменные для формулы  
$ram_memory=array();  
$cpu_freq=array();  
$ram_freq=array();  
$vid_memory=array();
```

```

$vid_bus=array();
$vyvod=array();
# Вывод с базы данных
if ($result->num_rows > 0) {

```

*Продолжение Приложения А*

```

// output data of each row
while($row = $result->fetch_assoc()) {
    $records[]=$row;
}
}

# Константы(знания)
$ram_memory_const = 8;
$ram_freq_const = 1.8;
$cpu_freq_const= 2.4;
$vid_memory_const = 2;
$vid_bus_const = 16;

$m_media_g_step =
(($ram_memory_const/3)+$ram_freq_const+$cpu_freq_const+$vid_memory_const
+$vid_bus_const)/5;
# Цикл выборки
    #echo "[";
    for ($i = 0; $i < count($records); $i++) {
        $ram_memory[$i] = ($records[$i]['m_ram_memory_gb']);
        $cpu_freq[$i] = ($records[$i]['m_cpu_freqMin_ghz']);
        $ram_freq[$i] = ($records[$i]['m_ram_freq_ghz']);
        $vid_memory[$i] = ($records[$i]['m_vid_memory_gb']);
        $vid_bus[$i] = ($records[$i]['m_vid_byte']);
        $out[$i] = (((($ram_memory_const + $ram_memory[$i])/3) +
($ram_freq_const + $ram_freq[$i]) + ($cpu_freq_const + $cpu_freq[$i]) +
($vid_memory_const + $vid_memory[$i]) + ($vid_bus_const + $vid_bus[$i]))/10;
        if ($m_media_g_step<$out[$i] && $records[$i]['dis_res_x']>=1920) {

            # $vyvod = json_encode($records);
            $vyvod[$i] = $records[$i];
            #echo ($vyvod);
            #echo ",";

            #file_put_contents('devices.json', $json_file);

```

```

    }
}    echo json_encode(array_values($vyvod));
#echo "]]";

```

```
$conn->close();
```

### *Продолжение Приложения А*

```
?>
```

```
<?php
```

```

header("Access-Control-Allow-Origin: *");
header("Content-Type: application/json; charset=UTF-8");
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "devices";

```

```
// Create connection
```

```
$conn = new mysqli($servername, $username, $password, $dbname);
```

```
// Check connection
```

```

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

```

```
# Запрос БД
```

```

$sql = "SELECT
cpm.m_cpu_freqMin_ghz,dev.dev_prod,dev.dev_model,prod.prod_name,cpm.m_cp
u_model,
rm.m_ram_freq_ghz,
rm.m_ram_memory_gb,
pd.prod_name,vdm.m_vid_model,vdm.m_vid_memory_gb,vdm.m_vid_byte,dev.de
v_price_tg, DIS.dis_diag_inch, DIS.dis_res_x, DIS.dis_res_y, DIS.dis_matrix,
DIS.dis_type, DIS.dis_sensor, STG.m_stor_type, STG.m_stor_memory_gb,
AUD.m_aud_technology, OS.os_name, OS.os_bit, PRTS.m_po_descrip from
device dev join cpu_m cpm on cpm.m_cpu_id=dev.m_cpu_id join ram_m rm on
rm.m_ram_id=dev.m_ram_id join videocard_m vdm on
vdm.m_vid_id=dev.m_vid_id join producer prod on prod.prod_id=cpm.prod_id join
producer pd on pd.prod_id=vdm.prod_id join display DIS on DIS.dis_id=dev.dis_id
join storage STG on STG.m_stor_id=dev.m_stor_id join audio AUD on
AUD.m_aud_id=dev.m_aud_id join os OS on OS.os_id=dev.os_id join ports PRTS
on PRTS.m_po_id=dev.m_po_id where dev.dev_type='Laptop' and dev.m_aud_id
!= 5";
$result = $conn->query($sql);
$records=array();
# Переменные для формулы
$ram_memory=array();

```

```

$cpu_freq=array();
$ram_freq=array();
$vid_memory=array();
$vid_bus=array();
$vyvod=array();
# Вывод с базы данных

```

*Продолжение Приложения А*

```

if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        $records[]=$row;
    }
}
echo json_encode(array_values($records));
#echo "]";

```

```

$conn->close();
?>

```

```

<?php
header("Access-Control-Allow-Origin: *");
header("Content-Type: application/json; charset=UTF-8");

```

```

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "devices";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

```

```

# Запрос БД
$sql = "SELECT
cpm.m_cpu_freqMin_ghz,dev.dev_prod,dev.dev_model,prod.prod_name,cpm.m_cp
u_model,          rm.m_ram_freq_ghz,          rm.m_ram_memory_gb,
pd.prod_name,vdm.m_vid_model,vdm.m_vid_memory_gb,vdm.m_vid_byte,dev.de
v_price_tg, DIS.dis_diag_inch, DIS.dis_res_x, DIS.dis_res_y, DIS.dis_matrix,
DIS.dis_type, DIS.dis_sensor, STG.m_stor_type, STG.m_stor_memory_gb,

```



```
AUD.m_aud_technology, OS.os_name, OS.os_bit, PRTS.m_po_descrip,
dev.dev_img from device dev join cpu_m cpm on cpm.m_cpu_id=dev.m_cpu_id
join ram_m rm on rm.m_ram_id=dev.m_ram_id join videocard_m vdm on
vdm.m_vid_id=dev.m_vid_id join producer prod on prod.prod_id=cpm.prod_id join
producer pd on pd.prod_id=vdm.prod_id join display DIS on DIS.dis_id=dev.dis_id
join storage STG on STG.m_stor_id=dev.m_stor_id join audio AUD on
```

### *Продолжение Приложения А*

```
AUD.m_aud_id=dev.m_aud_id join os OS on OS.os_id=dev.os_id join ports PRTS
on PRTS.m_po_id=dev.m_po_id where dev.dev_type='Laptop';
$result = $conn->query($sql);
$records=array();
# Переменные для формулы
$ram_memory=array();
$cpu_freq=array();
$ram_freq=array();
$vid_memory=array();
$vid_bus=array();
$vyvod=array();
# Вывод с базы данных
if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        $records[]=$row;
    }
}

# Константы(знания)
# IT LAPTOP
$ram_memory_const_it = 6;
$ram_freq_const_it = 1.3;
$cpu_freq_const_it = 2;
$vid_memory_const_it = 1;
$vid_bus_const_it = 16;

# OFFICE LAPTOP
$ram_memory_const = 4;
$ram_freq_const = 1.1;
$cpu_freq_const= 1.3;
$vid_memory_const = 1;
```

```
$vid_bus_const = 16;
```

```
$it_step =  
((($ram_memory_const_it/3)+$ram_freq_const_it+$cpu_freq_const_it+$vid_memory_const_it+$vid_bus_const_it)/5;  
# ЦИКЛ ВЫБОРКИ  
$office_step=2.54;
```

*Продолжение Приложения А*

```
for ($i = 0; $i < count($records); $i++) {  
    $ram_memory[$i] = ($records[$i]['m_ram_memory_gb']);  
    $cpu_freq[$i] = ($records[$i]['m_cpu_freqMin_ghz']);  
    $ram_freq[$i] = ($records[$i]['m_ram_freq_ghz']);  
    $vid_memory[$i] = ($records[$i]['m_vid_memory_gb']);  
    $vid_bus[$i] = ($records[$i]['m_vid_byte']);  
    $out_it[$i] = (((($ram_memory_const_it + $ram_memory[$i])/3) +  
($ram_freq_const_it + $ram_freq[$i]) + ($cpu_freq_const_it + $cpu_freq[$i]) +  
($vid_memory_const_it + $vid_memory[$i]) + ($vid_bus_const_it +  
$vid_bus[$i]))/10;  
    $out_office[$i] = (((($ram_memory_const + $ram_memory[$i])/3) +  
($ram_freq_const + $ram_freq[$i]) + ($cpu_freq_const + $cpu_freq[$i]) +  
($vid_memory_const + $vid_memory[$i]) + ($vid_bus_const + $vid_bus[$i]))/10;  
    if ($it_step>=$out_it[$i] && $out_office[$i]>$office_step) {  
        $vyvod[$i] = $records[$i];  
    }  
}  
echo json_encode(array_values($vyvod));
```

```
$conn->close();
```

```
?>
```

```
<?php
```

```
header("Access-Control-Allow-Origin: *");  
header("Content-Type: application/json; charset=UTF-8");  
$servername = "localhost";  
$username = "root";  
$password = "";  
$dbname = "devices";
```

```
// Create connection
```

```
$conn = new mysqli($servername, $username, $password, $dbname);
```

```
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

# Запрос БД
$sql = "SELECT
cpm.m_cpu_freqMin_ghz,dev.dev_prod,dev.dev_model,prod.prod_name,cpm.m_cp
u_model, rm.m_ram_freq_ghz, rm.m_ram_memory_gb,
```

*Продолжение Приложения А*

```
pd.prod_name,vdm.m_vid_model,vdm.m_vid_memory_gb,vdm.m_vid_byte,dev.de
v_price_tg, DIS.dis_diag_inch, DIS.dis_res_x, DIS.dis_res_y, DIS.dis_matrix,
DIS.dis_type, DIS.dis_sensor, STG.m_stor_type, STG.m_stor_memory_gb,
AUD.m_aud_technology, OS.os_name, OS.os_bit, PRTS.m_po_descrip from
device dev join cpu_m cpm on cpm.m_cpu_id=dev.m_cpu_id join ram_m rm on
rm.m_ram_id=dev.m_ram_id join videocard_m vdm on
vdm.m_vid_id=dev.m_vid_id join producer prod on prod.prod_id=cpm.prod_id join
producer pd on pd.prod_id=vdm.prod_id join display DIS on DIS.dis_id=dev.dis_id
join storage STG on STG.m_stor_id=dev.m_stor_id join audio AUD on
AUD.m_aud_id=dev.m_aud_id join os OS on OS.os_id=dev.os_id join ports PRTS
on PRTS.m_po_id=dev.m_po_id where dev.dev_type='Laptop';
$result = $conn->query($sql);
$records=array();
# Переменные для формулы
$ram_memory=array();
$cpu_freq=array();
$ram_freq=array();
$vid_memory=array();
$vid_bus=array();
$vyvod=array();
# Вывод с базы данных
if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        $records[]=$row;
    }
}

# Константы(знания)
$ram_memory_const = 8;
```

```

$ram_freq_const = 1.8;
$cpu_freq_const= 2.4;
$vid_memory_const = 2;
$vid_bus_const = 16;
#$gaming_step=1;
$gaming_step
=
(($ram_memory_const/3)+$ram_freq_const+$cpu_freq_const+$vid_memory_const
+$vid_bus_const)/5;
for ($i = 0; $i < count($records); $i++) {
    $ram_memory[$i] = ($records[$i]['m_ram_memory_gb']);

```

*Продолжение Приложения А*

```

$cpu_freq[$i] = ($records[$i]['m_cpu_freqMin_ghz']);
$ram_freq[$i] = ($records[$i]['m_ram_freq_ghz']);
$vid_memory[$i] = ($records[$i]['m_vid_memory_gb']);
$vid_bus[$i] = ($records[$i]['m_vid_byte']);
    $out[$i] = (((($ram_memory_const + $ram_memory[$i])/3) +
($ram_freq_const + $ram_freq[$i]) + ($cpu_freq_const + $cpu_freq[$i]) +
($vid_memory_const + $vid_memory[$i]) + ($vid_bus_const + $vid_bus[$i]))/10;
    if ($gaming_step<$out[$i] && $records[$i]['dis_res_x']<=1920) {

        $vyvod[$i] = $records[$i];
    }
}    echo json_encode(array_values($vyvod));

```

```

$conn->close();
?>

```

```

<?php
header("Access-Control-Allow-Origin: *");
header("Content-Type: application/json; charset=UTF-8");
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "devices";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

```

```

# Запрос БД
$sql1="set @price=0;";
$sql = "select *,@price from pc PC join cpu CPU on CPU.cpu_id=PC.cpu_id join
cpu_cooler CPUC on CPUC.coo_id = PC.coo_id
join pc_case PSC on PSC.case_id=PC.case_id join power_supply PWS on
PWS.ps_id=PC.ps_id join hdd HDD on HDD.hdd_id=PC.hdd_id
join ssd SSD on SSD.ssd_id=PC.ssd_id join ram RAM on RAM.ram_id=PC.ram_id
join videocard VD on VD.vid_id=PC.vid_id
join motherboard MTH on MTH.mb_id=PC.mb_id where CPU.cpu_freq_ghz
BETWEEN 2.2 and 2.9 and @price := CPU.cpu_price_tg + CPUC.coo_price_tg +

```

*Продолжение Приложения А*

```

PSC.case_price_tg          +          PWS.ps_price_tg
+RAM.ram_price_tg+VD.vid_price_tg+MTH.mb_price_tg";
$conn->query($sql1);
$result = $conn->query($sql);
$records=array();
# Переменные для формулы
$ram_memory=array();
$cpu_freq=array();
$ram_freq=array();
$vid_memory=array();
$vid_bus=array();
$vyvod=array();
# Вывод с базы данных
if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        $records[]=$row;
    }
}

    echo json_encode(array_values($records));
    #echo "];";

$conn->close();
?>

<?php
header("Access-Control-Allow-Origin: *");
header("Content-Type: application/json; charset=UTF-8");

```

```

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "devices";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

```

### *Продолжение Приложения А*

```

# Запрос БД
$sql1="set @price=0;";
$sql = "select *,@price from pc PC join cpu CPU on CPU.cpu_id=PC.cpu_id join
cpu_cooler CPUC on CPUC.coo_id = PC.coo_id join pc_case PSC on
PSC.case_id=PC.case_id join power_supply PWS on PWS.ps_id=PC.ps_id join
hdd HDD on HDD.hdd_id=PC.hdd_id join ssd SSD on SSD.ssd_id=PC.ssd_id join
ram RAM on RAM.ram_id=PC.ram_id join videocard VD on
VD.vid_id=PC.vid_id join motherboard MTH on MTH.mb_id=PC.mb_id where
CPU.cpu_freq_ghz BETWEEN 2.2 and 3.6 and @price := CPU.cpu_price_tg +
CPUC.coo_price_tg + PSC.case_price_tg + PWS.ps_price_tg
+RAM.ram_price_tg+VD.vid_price_tg+MTH.mb_price_tg";
$conn->query($sql1);
$result = $conn->query($sql);
$records=array();
# Переменные для формулы
$ram_memory=array();
$cpu_freq=array();
$ram_freq=array();
$vid_memory=array();
$vid_bus=array();
$vyvod=array();
# ВЫВОД с базы данных
if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        $records[]=$row;
    }
}
}

```

```

        echo json_encode(array_values($records));
        #echo "】";

$conn->close();
?>

<?php
header("Access-Control-Allow-Origin: *");
header("Content-Type: application/json; charset=UTF-8");
$servername = "localhost";
$username = "root";
$password = "";

```

### *Продолжение Приложения А*

```

$dbname = "devices";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

# Запрос БД
$sql1="set @price=0;";
$sql = "select *,@price from pc PC join cpu CPU on CPU.cpu_id=PC.cpu_id join
cpu_cooler CPUC on CPUC.coo_id = PC.coo_id join pc_case PSC on
PSC.case_id=PC.case_id join power_supply PWS on PWS.ps_id=PC.ps_id join
hdd HDD on HDD.hdd_id=PC.hdd_id join ssd SSD on SSD.ssd_id=PC.ssd_id join
ram RAM on RAM.ram_id=PC.ram_id join videocard VD on
VD.vid_id=PC.vid_id join motherboard MTH on MTH.mb_id=PC.mb_id where
MTH.mb_type='office-it' and
    @price := CPU.cpu_price_tg + CPUC.coo_price_tg + PSC.case_price_tg +
PWS.ps_price_tg +RAM.ram_price_tg+VD.vid_price_tg+MTH.mb_price_tg";
$conn->query($sql1);
$result = $conn->query($sql);
$records=array();
# Переменные для формулы
$ram_memory=array();
$cpu_freq=array();
$ram_freq=array();
$vid_memory=array();

```

```

$vid_bus=array();
$vyvod=array();
# ВЫВОД с базы данных
if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        $records[]=$row;
    }
}

echo json_encode(array_values($records));
#echo "】";

$conn->close();

```

### *Продолжение Приложения А*

```

?>
<?php
header("Access-Control-Allow-Origin: *");
header("Content-Type: application/json; charset=UTF-8");
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "devices";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

# Запрос БД
$sql = "SELECT
cpm.m_cpu_freqMin_ghz,dev.dev_prod,dev.dev_model,prod.prod_name,cpm.m_cp
u_model,
rm.m_ram_freq_ghz,
rm.m_ram_memory_gb,
pd.prod_name,vdm.m_vid_model,vdm.m_vid_memory_gb,vdm.m_vid_byte,dev.de
v_price_tg, DIS.dis_diag_inch, DIS.dis_res_x, DIS.dis_res_y, DIS.dis_matrix,
DIS.dis_type, DIS.dis_sensor, STG.m_stor_type, STG.m_stor_memory_gb,
AUD.m_aud_technology, OS.os_name, OS.os_bit, PRTS.m_po_descrip from
device dev join cpu_m cpm on cpm.m_cpu_id=dev.m_cpu_id join ram_m rm on
rm.m_ram_id=dev.m_ram_id join videocard_m vdm on

```



```

vdm.m_vid_id=dev.m_vid_id join producer prod on prod.prod_id=cpm.prod_id join
producer pd on pd.prod_id=vdm.prod_id join display DIS on DIS.dis_id=dev.dis_id
join storage STG on STG.m_stor_id=dev.m_stor_id join audio AUD on
AUD.m_aud_id=dev.m_aud_id join os OS on OS.os_id=dev.os_id join ports PRTS
on PRTS.m_po_id=dev.m_po_id where dev.dev_type='Monoblock';
$result = $conn->query($sql);
$records=array();
# Переменные для формулы
$ram_memory=array();
$cpu_freq=array();
$ram_freq=array();
$vid_memory=array();
$vid_bus=array();
$vyvod=array();
# Вывод с базы данных
if ($result->num_rows > 0) {

```

*Продолжение Приложения А*

```

// output data of each row
while($row = $result->fetch_assoc()) {
    $records[]=$row;
}
}
# Константы(знания)
$ram_memory_const = 8;
$ram_freq_const = 1.8;
$cpu_freq_const= 2.4;
$vid_memory_const = 2;
$vid_bus_const = 16;
#$gaming_step=1;
$gaming_step =
((($ram_memory_const/3)+$ram_freq_const+$cpu_freq_const+$vid_memory_const
+$vid_bus_const)/5;
# Цикл выборки
#echo "[";
for ($i = 0; $i < count($records); $i++) {
    $ram_memory[$i] = ($records[$i]['m_ram_memory_gb']);
    $cpu_freq[$i] = ($records[$i]['m_cpu_freqMin_ghz']);
    $ram_freq[$i] = ($records[$i]['m_ram_freq_ghz']);
    $vid_memory[$i] = ($records[$i]['m_vid_memory_gb']);
    $vid_bus[$i] = ($records[$i]['m_vid_byte']);

```

```

        $out[$i] = ((($ram_memory_const + $ram_memory[$i])/3) +
($ram_freq_const + $ram_freq[$i]) + ($cpu_freq_const + $cpu_freq[$i]) +
($vid_memory_const + $vid_memory[$i]) + ($vid_bus_const + $vid_bus[$i]))/10;
        if ($gaming_step < $out[$i]) {
            echo $gaming_step;
            # $vyvod = json_encode($records);
            $vyvod[$i] = $records[$i];
            # echo ($vyvod);
            # echo ",";

            # file_put_contents('devices.json', $json_file);
        }
    }
    echo json_encode(array_values($vyvod));
    # echo "]" ;

$conn->close();
?>
<?php
header("Access-Control-Allow-Origin: *");

```

### *Продолжение Приложения А*

```

header("Content-Type: application/json; charset=UTF-8");
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "devices";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

# Запрос БД
$sql = "SELECT
cpm.m_cpu_freqMin_ghz,dev.dev_prod,dev.dev_model,prod.prod_name,cpm.m_cp
u_model,
rm.m_ram_freq_ghz,
rm.m_ram_memory_gb,
pd.prod_name,vdm.m_vid_model,vdm.m_vid_memory_gb,vdm.m_vid_byte,dev.de
v_price_tg, DIS.dis_diag_inch, DIS.dis_res_x, DIS.dis_res_y, DIS.dis_matrix,
DIS.dis_type, DIS.dis_sensor, STG.m_stor_type, STG.m_stor_memory_gb,
AUD.m_aud_technology, OS.os_name, OS.os_bit, PRTS.m_po_descrip,
dev.dev_img from device dev join cpu_m cpm on cpm.m_cpu_id=dev.m_cpu_id

```

```

join ram_m rm on rm.m_ram_id=dev.m_ram_id join videocard_m vdm on
vdm.m_vid_id=dev.m_vid_id join producer prod on prod.prod_id=cpm.prod_id join
producer pd on pd.prod_id=vdm.prod_id join display DIS on DIS.dis_id=dev.dis_id
join storage STG on STG.m_stor_id=dev.m_stor_id join audio AUD on
AUD.m_aud_id=dev.m_aud_id join os OS on OS.os_id=dev.os_id join ports PRTS
on PRTS.m_po_id=dev.m_po_id where dev.dev_type='Monoblock';
$result = $conn->query($sql);
$records=array();
# Переменные для формулы
$ram_memory=array();
$cpu_freq=array();
$ram_freq=array();
$vid_memory=array();
$vid_bus=array();
$vyvod=array();
# Вывод с базы данных
if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        $records[]=$row; } }

```

*Продолжение Приложения А*

```

# Константы(знания)
# OFFICE LAPTOP
$ram_memory_const = 4;
$ram_freq_const = 1.1;
$cpu_freq_const= 1.8;
$vid_memory_const = 1;
$vid_bus_const = 16;
$office_step =
(($ram_memory_const/3)+$ram_freq_const+$cpu_freq_const+$vid_memory_const
+$vid_bus_const)/5;
# Цикл выборки

for ($i = 0; $i < count($records); $i++) {
    $ram_memory[$i] = ($records[$i]['m_ram_memory_gb']);
    $cpu_freq[$i] = ($records[$i]['m_cpu_freqMin_ghz']);
    $ram_freq[$i] = ($records[$i]['m_ram_freq_ghz']);
    $vid_memory[$i] = ($records[$i]['m_vid_memory_gb']);
    $vid_bus[$i] = ($records[$i]['m_vid_byte']);
    $out_office[$i] = (((($ram_memory_const + $ram_memory[$i])/3) +
($ram_freq_const + $ram_freq[$i]) + ($cpu_freq_const + $cpu_freq[$i]) +
($vid_memory_const + $vid_memory[$i]) + ($vid_bus_const + $vid_bus[$i]))/10;

```

```

    if ($office_step>=$out_office[$i]) {
        $vyvod[$i] = $records[$i];
    }
}
echo json_encode(array_values($vyvod));

$conn->close();
?>
<?php
header("Access-Control-Allow-Origin: *");
header("Content-Type: application/json; charset=UTF-8");
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "devices";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error); }

```

### *Продолжение Приложения А*

```

# Запрос БД
$sql = "SELECT
cpm.m_cpu_freqMin_ghz,dev.dev_img,dev.dev_prod,dev.dev_model,prod.prod_name,
cpm.m_cpu_model,      rm.m_ram_freq_ghz,      rm.m_ram_memory_gb,
pd.prod_name,vdm.m_vid_model,vdm.m_vid_memory_gb,vdm.m_vid_byte,dev.dev_price_tg,
DIS.dis_diag_inch, DIS.dis_res_x, DIS.dis_res_y, DIS.dis_matrix,
DIS.dis_type, DIS.dis_sensor, STG.m_stor_type, STG.m_stor_memory_gb,
AUD.m_aud_technology, OS.os_name, OS.os_bit, PRTS.m_po_descrip from
device dev join cpu_m cpm on cpm.m_cpu_id=dev.m_cpu_id join ram_m rm on
rm.m_ram_id=dev.m_ram_id      join      videocard_m      vdm      on
vdm.m_vid_id=dev.m_vid_id join producer prod on prod.prod_id=cpm.prod_id join
producer pd on pd.prod_id=vdm.prod_id join display DIS on DIS.dis_id=dev.dis_id
join storage STG on STG.m_stor_id=dev.m_stor_id join audio AUD on
AUD.m_aud_id=dev.m_aud_id join os OS on OS.os_id=dev.os_id join ports PRTS
on PRTS.m_po_id=dev.m_po_id where dev.dev_type='Monoblock' and
dev.m_aud_id != 5";
$result = $conn->query($sql);
$records=array();
# Переменные для формулы
$ram_memory=array();

```

```
$cpu_freq=array();
$ram_freq=array();
$vid_memory=array();
$vid_bus=array();
$vyvod=array();
# Вывод с базы данных
if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        $records[]=$row;
    }
}
echo json_encode(array_values($records));
#echo "];

$conn->close();
?>
```