|  |  |
| --- | --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ | |
|  | |
| Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Пермский государственный национальный исследовательский университет» | |
|  | |
| Отчет  по лабораторной работе №1  по дисциплине «Инструментальные средства построения баз знаний» | |
|  | Работу выполнил студент  ПМИ-19-2 группы 4курса механико-математического факультета  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.М. Писцов |
|  | Проверил старший преподаватель  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Т.А. Леонтьева  "\_\_\_" апреля 2023 г. |
| Пермь 2023 | |

Оглавление

[Оглавление 2](#_Toc130343752)

[Постановка 3](#_Toc130343753)

[Описание разработанной оболочки ЭС 3](#_Toc130343754)

[Описание класса KnowledgeBase 4](#_Toc130343755)

[Описание класса KnowledgeAcquisitionComponent 6](#_Toc130343756)

[Описание класса WorkingMemory 7](#_Toc130343757)

[Описание класса LogicalInferenceMechanism 8](#_Toc130343758)

[Описание класса ExplanationComponent 9](#_Toc130343759)

[Описание класса ввода-вывода 10](#_Toc130343760)

[Описание базы знаний 10](#_Toc130343761)

[Описание графического интерфейса приложения 11](#_Toc130343762)

[Тестирование механизма логического вывода 39](#_Toc130343763)

[Приложение А 49](#_Toc130343764)

Постановка

Разработать и реализовать оболочку экспертных систем. Использовать базу данных или объектную модель. Тестирование проводить на разработанной ранее экспертной системе.

Описание разработанной оболочки ЭС

Оболочка экспертной системы состоит из следующий компонентов: база знаний, компонента объяснения (объяснительные способности), компонента приобретения знаний, машина логического вывода, а также интерфейса, через который пользователи эксперты взаимодействую с оболочкой. Схема взаимодействия компонентов оболочки ЭС приведена на рисунке 1.

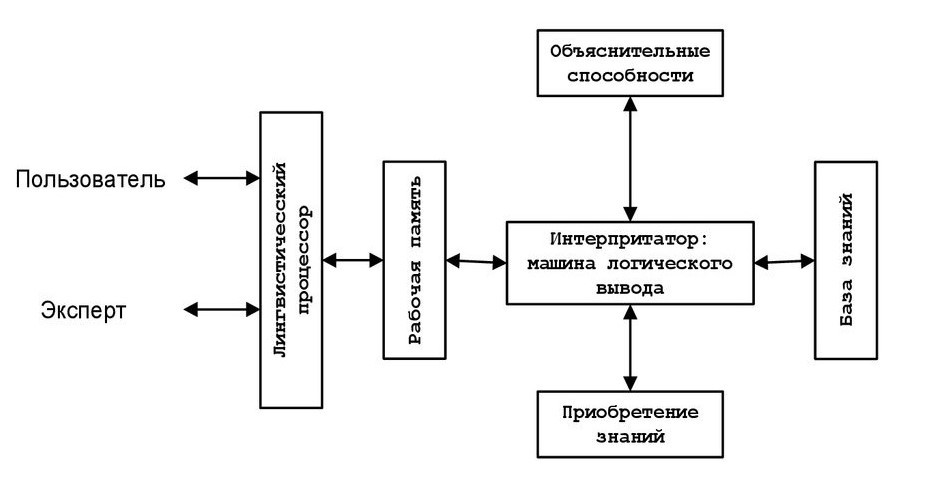


Рисунок 1. Схема взаимодействия компонентов оболочки ЭС

В соответствии со схемой приложение содержит содержит классы KnowledgeAcquisitionComponent получения, добавления, обновления данных с интерфейса; LogicalInferenceMechanism для доказательства цели в процессе консультации путем использования механизма логического вывода; WorkingMemory для хранения выведенных и запрошенных переменных, сработавших правил в процессе консультации; KnowledgeBase для хранения правил, переменных и доменов. Для доступа к правилам, переменным и доменам реализованы соответствующие классы: Rule, Domain, Variable. Компонента объяснения описывается классом ExplanationComponent. Диаграмма разработанных классов представлена на рисунке 2.

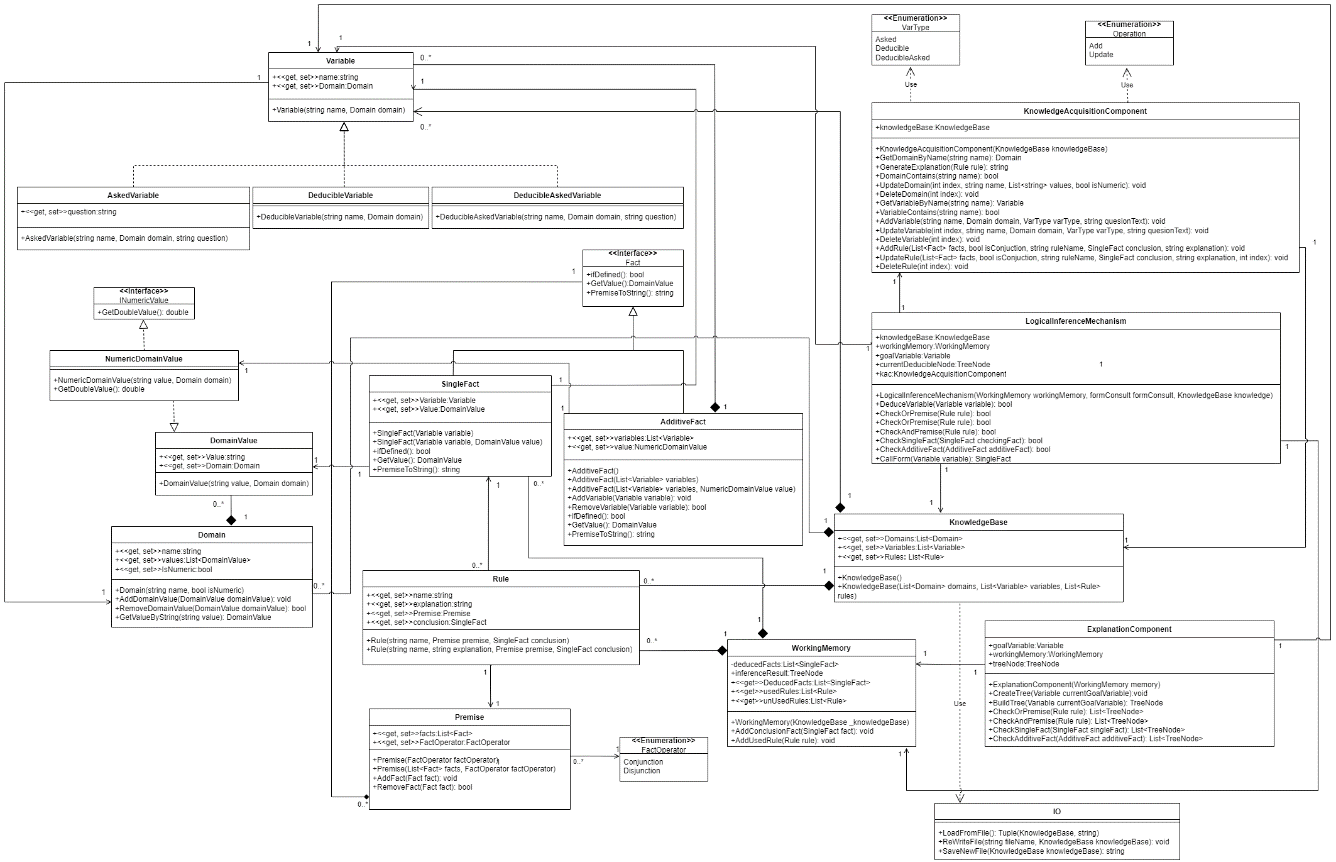


Рисунок 2. Диаграмма классов

Описание класса KnowledgeBase

Используется продукционная модель представления знаний, правило описывает класс Rule. Правило содержит название (name), объяснение (explanation), если оно добавлено пользователем (иначе генерируется на основе посылки и заключения методом GenerateExplanation); посылку, которая описывается классом Premise, а также заключение, которое описывается классом SingleFact.

Посылка должна содержать минимум 1 факт, набор фактов посылки представляет собой список экземпляров классов SingleFact и AdditiveFact, которые реализуют интерфейс Fact. Для удобства реализации список фактов представлен элементами типа Fact, в процессе работы используется конкретный тип элемента (одиночный или состоящий из нескольких переменных типа Variable). Факт из нескольких переменных может подержать значение переменной только типа NumericDomainValue, т.к. ранее спроектированная база знаний содержит факты, значения переменных которых складываются в процессе вывода. Каждая посылка может иметь отношение фактов только через оператор ‘И’ или только через оператор ‘ИЛИ’. Поэтому для создания посылка, содержащей разные операторы отношения необходимо создавать правила-части (в случае правила «факт1 и (факт 2 или факт3)» необходимо создать правило с посылкой «факт 2 или факт3» с заключением, которое затем будет подставлено в изначальное правило). Интерфейс Fact содержит 3 метода:

* ifDefined – означена ли переменная,
* GetValue – получить значение переменной,
* PremiseToString(привести факт к текстовому представлению при выводе посылки.

Переменная факта описывается классами AskedVariable (запрашиваемая переменная), DeducibleVariable(выводимая переменная) и DeducibleAskedVariable (выводимо-запрашиваемая переменная), которые наследуются от базового абстрактного класса Variable. Для удобства переменные факта имеют тип Variable. Запрашиваемая переменная отличается наличием поля question для вывода вопроса в процессе консультации.

Домен представлен классом Domain, который содержит поля name (название домена) и список значений домена (экземпляры классов DomainValue и NumericDomainValue).

Значение домена описывается классом DomainValue и содержит следующие поля: значение (строковое), домен (экземпляр Domain, к которому привязано данное значение).

Класс NumericDomainValue содержит лишь числовое положительное значение (целочисленное, вещественное) и реализует интерфейс INumericValue (содержит метод GetDoubleValue()).

База знаний описывается классом KnowledgeBase и содержит списки правил, переменных и доменов, каждый объект имеет уникальное имя. Класс содержит метод GetDomainByName(для получения экземпляра домена на основе переданного имени.

Описание класса KnowledgeAcquisitionComponent

Каждая форма работы с базой знаний содержит экземпляр класса KnowledgeAcquisitionComponent для обновления правил, переменных, доменов. Класс содержит экземпляр базы знаний KnowledgeBase и следующие методы:

* GenerateExplanation – генерирует описание переданного правила. Используется при отображении используемых фактов в дереве вывода и отображении информации о правиле, если пользователь не задал вручную описание правила;
* DomainContains – проверка наличия переданного домена по параметру в базе знаний;
* AddDomain – на основе переданного названия, списка значений и типа домена создается домен со значениями соответствующего типа (строковые или числовые), затем добавляется в базу знаний. Проверка уникальности имени происходит на форме;
* UpdateDomain – по заданному индексу в базе знаний происходит обновления домена на основе входных данных. Новые значения добавляется в домен, отсутствующие удаляются, при этом остальные значения не меняются (если значения задействованы в правилах, ссылки на объекты не пропадут). Также есть возможность изменить значений домена на числовой или строковый, проверка значения осуществляется на соответствующей форме. В случае изменения типа значения все знания добавляются заново. Т.к. необходимо реализовать поддержку изменения порядка значений домена, после обновления происходит сортировка в порядке значений по параметру;
* DeleteDomain – удаление домена по индексу из базы знаний;
* GetVariableByName – получение переменной из базы знаний по названию;
* VariableContains – проверка наличия переменной в базе знаний по названию;
* AddVariable – добавления новой переменной в базу на основе переданных с интерфейса параметров. В зависимости от параметра типа, создается и добавляется запрашиваемая, выводимая или выводимо-запрашиваемая переменная;
* UpdateVariable – обновление переменной по индексу из базы знаний на основе переданного параметра. Есть возможность изменить тип переменной, в случае изменения типа создается новая переменная, которая заполняется данными из переданных параметров. Контроль уникальности имен происходит на форме;
* DeleteVariable – удаление переменной из базы знаний по индексу;
* AddRule – добавления нового правила в базу знаний на основе переданных параметров;
* UpdateRule – обновление правила в базе знаний по индексу на основе переданных параметров;
* DeleteRule – удаление правила из базы знаний по индексу.

Описание класса WorkingMemory

В процессе доказательства необходимо явно хранить сработавшие правила и выведенные факты, поэтому классе WorkingMemory содержатся соответствующие поля UnusedRules (список неиспользованных правил), и DeducedFacts (выведенные факты). Класса содержит следующие методы:

* AddConclusionFact – добавить факт в список выведенных фактов;
* AddUsedRule – удалить правило из списка неиспользованных.

Описание класса LogicalInferenceMechanism

Механизм логического вывода описан классом LogicalInferenceMechanism. Он имеет поля working memory (рабочая память), goalVariable (целевая переменна, выводимая), TreeNode для описания компоненты объяснения. Класс содержит следующие методы:

* DeduceVariable – вывод текущей подцели. В первую очередь метод запускается с целевой переменной. В процессе обхода неиспользованных правил, проверяется наличие данной переменной в заключении. Если переменная есть, происходить попытка ее вывести. Если в процессе вывода необходимо вывести другую переменную, данный метод вновь вызывается с переменной подцели как параметром. В зависимости от типа оператора посылки вызываются советующие методы CheckOrPremise или CheckAndPremise;
* CheckOrPremise – попытка вывести правило с посылкой, у которой задан оператор ‘ИЛИ’. Проверка будет успешной, если хотя бы 1 факт будет выведен. Для реализации нетерпеливой стратегии тестирования посылки в первую очередь переменные из фактов посылки проверяются на их наличие в рабочей памяти, лишь затем производится запрос запрашиваемый переменных, либо вывод выводимых. На данном этапе происходит учет типа факта: для факта с одной переменной вызывается метод CheckSingleFact, для факта с несколькими переменными вызывается метод CheckAdditiveFact;
* CheckAndPremise – попытка вывести правило с посылкой, у которой задан оператор ‘И’. Проверка будет успешной, если все факты будут выведены. Как и в методе CheckSingleFact реализована нетерпеливая стратегия тестирования посылок. Для работы с разными операторами посылки вызываются соответствующие методы;
* CheckSingleFact – попытка доказать факт. В первую очередь проверяется наличие переменной факта рабочей памяти. Если переменная не найдена в рабочей памяти, проверяется тип переменной. Для запрашиваемой поток доказательства цели останавливается до тех пор, пока не будет получен ответ с формы. Для доказательства выводимой переменной происходит обращение к методу DeduceVariable;
* CheckAdditiveFact – попытка доказать факт из нескольких переменных. Каждая переменная последовательно проверяется на наличие в рабочей памяти, запрашивается или выводится. Факт будет доказан, если сумма всех доказанных переменных будет равняться числовому значению заключения;
* CallForm – заполнение выпадающего списка возможных значений запрашиваемой переменной на форме, а также ожидание события ManualResetEvent, которое активизируется после выбора ответа на форме.

Описание класса ExplanationComponent

Для построения дерева вывода необходимо иметь экземпляр класса WorkingMemory c результатами вывода в процессе консультации и Variable, которая содержит целевую переменную консультации. В случае успешной консультации каждой выводимой переменной соответствует одно правило, а при все запрашиваемые переменные сработавших правил обязательно были запрошены. Если в памяти есть выводимо-запрашиваемая переменная, необходимо узнать каким образом переменная была получена. Если в памяти есть правило, в заключении которого содержится данная переменная, значит она была выведена, иначе запрошена. Класс имеет следующие методы:

* CreateTree – инициализация целевой переменной для текущего класса, запуск обхода правил и фактов из рабочей памяти;
* BuildTree – обход посылки правила, содержащего текущую выводимую (или запрашиваемо-выводимую) переменную;
* CheckOrPremise – поиск и добавление переменной, которая была выведена (или запрошена) в процессе доказательства цели, а также всех подцелей;
* CheckAndPremise – поиск и добавление всех переменных посылки в процессе доказательства цели, а также всех подцелей;
* CheckSingleFact – проверка вывода текущего факта. Если факт не был выведен, возвращается пустой список, иначе список запрошенных переменных и подцелей;
* CheckAdditiveFact – проверка текущего факта из нескольких переменных. Если результат сложения всех переменных не совпадает с результатом заключения правила, возвращается пустой список.

Описание класса ввода-вывода

База знаний хранится как результат сериализации класса KnowledgeBase в виде файла с расширением ‘.es’. Сериализация и десериализация базы знаний происходит в классе IO. Класс содержит следующие методы:

* LoadFromFile – загрузка файла с сериализованной базой знаний из файла;
* ReWriteFile – изменения существующего файла с базой знаний с сохранением названия файла;
* SaveNewFile – создание нового файла с сохраненной базой знаний.

Описание базы знаний

Тестирование разработанной оболочки производилось на базе знаний «Качество образа жизни», разработанной на курсе «Базы знаний и оболочки экспертных систем». Данная экспертная система помогает определить человеку, являются ли его ежедневные привычки результатом плохого самочувствия. В зависимости от уровня образа жизни человеку следует пересмотреть привычки или обратиться к врачу. Список всех правил содержится в приложении А.

Концептуальная модель предметной области разработанной ЭС изображена на рисунке 3.

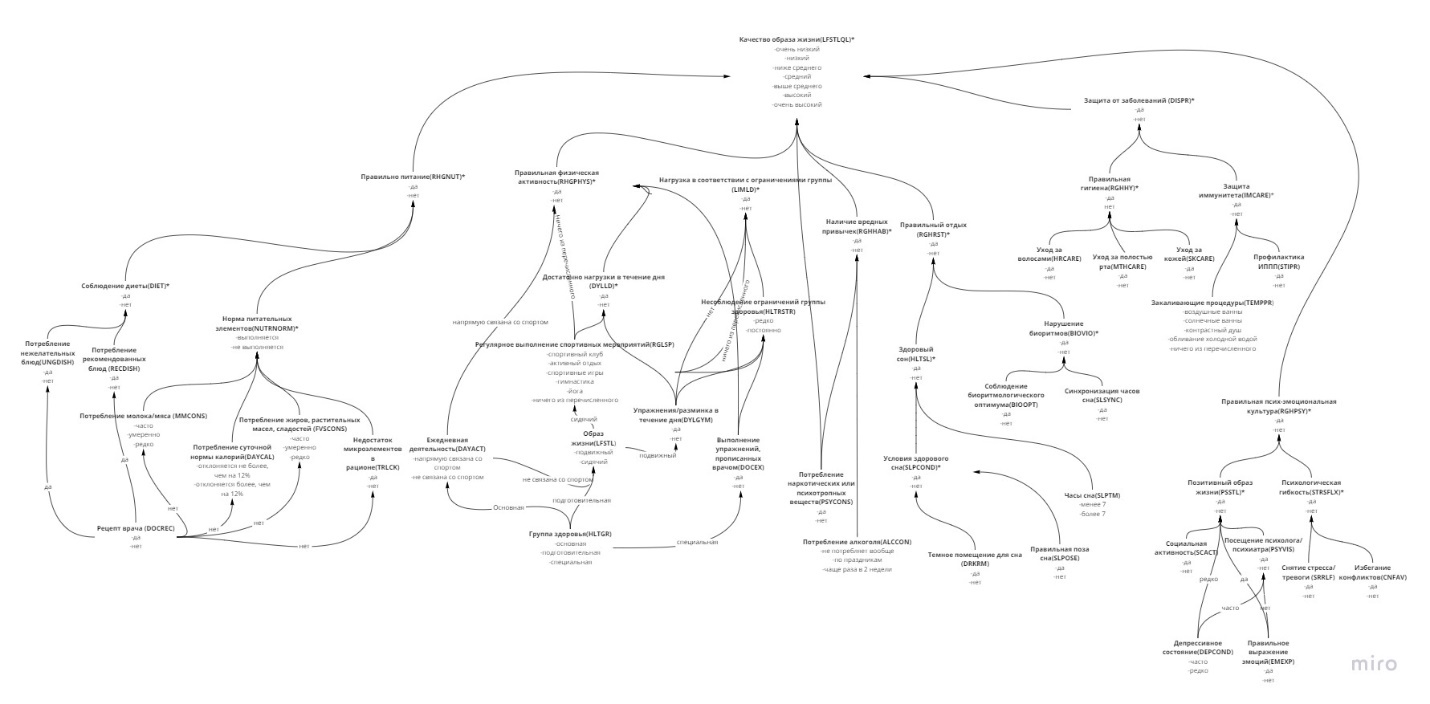


Рисунок 3. Концептуальная модель предметной области

Описание графического интерфейса приложения

Начальный экран оболочки изображен на рисунке 4.

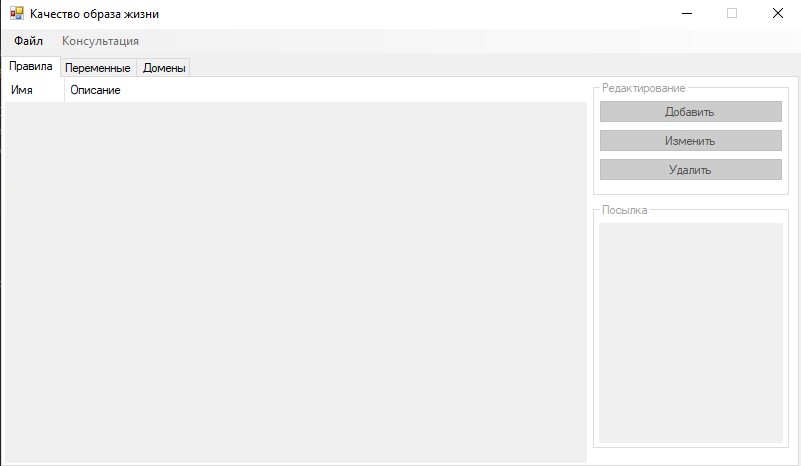


Рисунок 4. Начальный экран оболочки

Меню оболочки содержит 2 пункта: «Файл» и «Консультация». Пункт «Файл» содержит 5 подпунктов:

* Новый – создать новую пустую базу знаний;
* Открыть… - открыть файл с базой знаний;
* Сохранить – обновить существующий файл базы знаний текущей версией с оболочки;
* Сохранить как… - сохранить базу знаний оболочки в новый файл;
* Выход – завершить работу оболочки.

Изначально форма предоставляет возможность открыть файл или создать пустую базу знаний (рисунок 5).

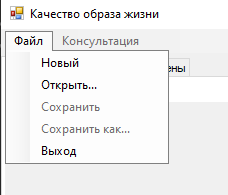


Рисунок 5. Пункт «Файл» при запуске оболочки

После добавление базы знаний становятся доступными остальные элементы, в том числе возможность провести консультацию (рисунок 6).

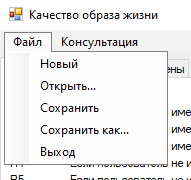


Рисунок 6. Пункт «Файл» после создания/загрузки базы знаний

Пункт «Консультация» содержит 2 подпункта:

* Начать консультацию – открыть формы консультации;
* Показать объяснение – открыть форму объяснения.

Подпункт «Показать объяснение» недоступен до тех пор, пока консультация не будет завершена успешно (рисунок 7).

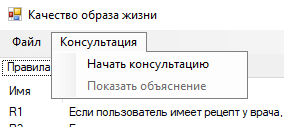


Рисунок 7. Пункт «консультация» до успешного завершения консультации

После загрузки файла с сериализованной базой знаний форма принимает вид, изображенный на рисунке 9.

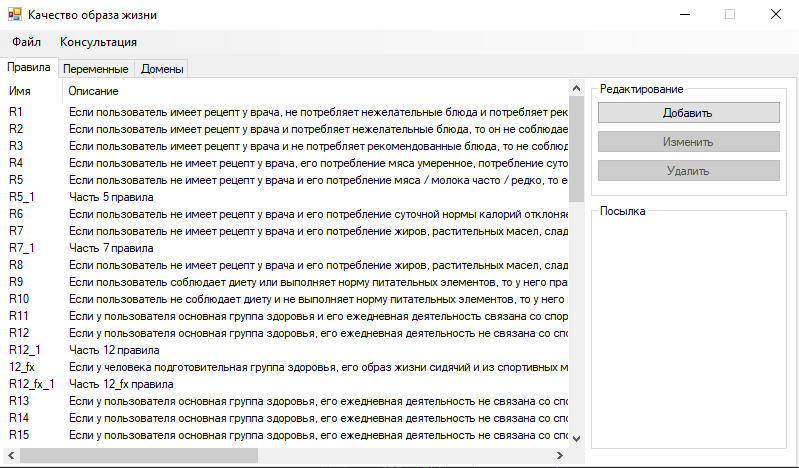


Рисунок 8. Форма после загрузки базы знаний из файла

Ранее заблокированный элемент TabControl содержит 3 пункта: правила, переменные и домены. После загрузки каждый пункт содержит соответствующие элементы, загруженные из правила. На данном экране отображены правила (имя, описание), посылка правила приведения в разделе «Посылка». Посылка появляется при выборе правила. Т.к. порядок правил важен при выводе переменных, элемент ListView, содержащий правил, поддерживает Drag and Drop для изменения порядка правил в локальной базе знаний. Любое изменение базы знаний приводит к обновлению информации на форме. На рисунке 9 изображен пример изменения порядка правил.

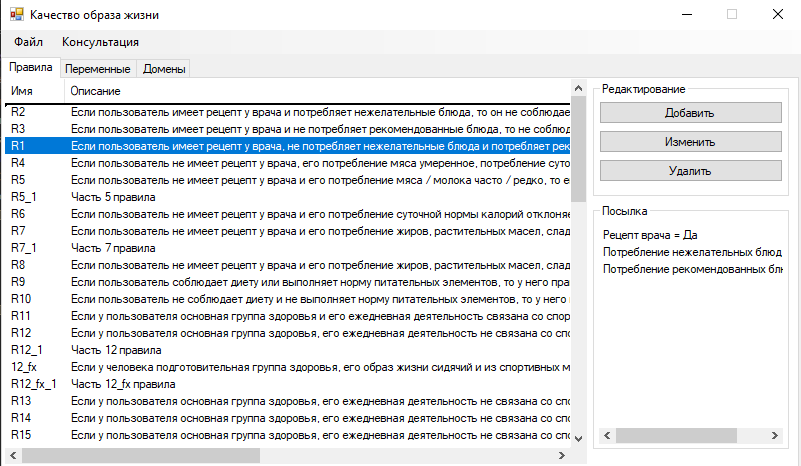


Рисунок 9. Изменение порядка правил в элементе ListView

Для добавления новых правил, обновления и редактирования существующих в разделе «Редактирование» присутствуют кнопки для открытия соответствующих форм/изменения базы знаний. Кнопки «Изменить» и «Удалить» недоступны до тех пор, пока не будет выбрано правило. Добавление правила осуществляется путем добавления элемента в конец списка правил базы знаний. Если на форме выбрано правило, добавление будет происходить сразу после выбранного правила (рисунок 10).

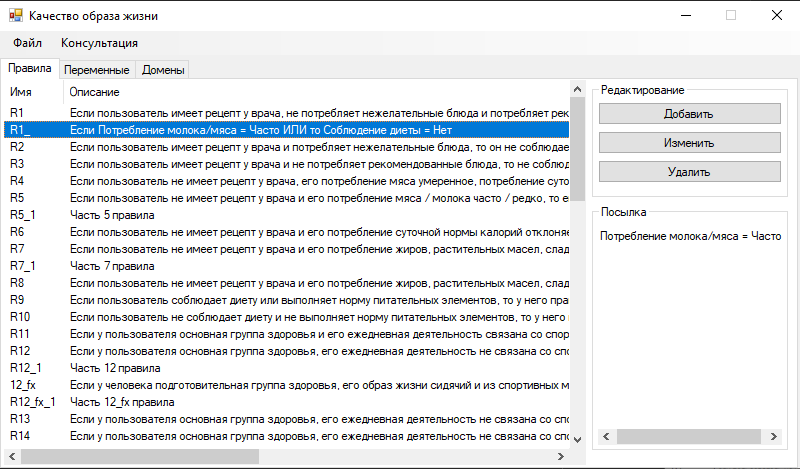


Рисунок 10. Добавление нового правила после выделенного

При редактировании существующего правила открывается форма правила, заполненная значениями выбранного правила. На рисунке 11 изображен пример редактирования правила, названию окна содержит название, соответствующее операции.

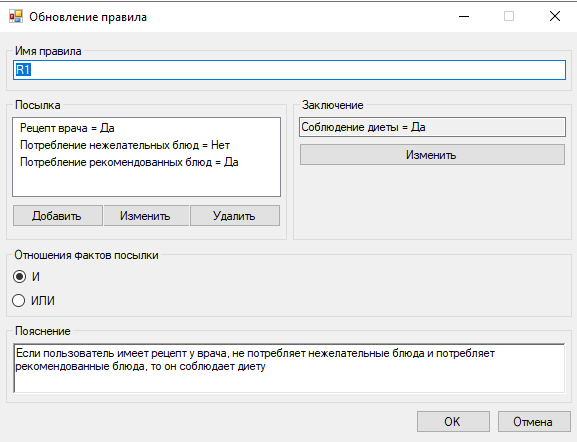


Рисунок 11. Форма обновления существующего правила

На данной форме пользователь может изменить название правила, добавить, изменить, удалить факты посылки. Заключение в соответствии разработанной экспертной системы содержит 1 факт. Также пользователь может изменить оператор отношения между фактами посылки и пояснение к правилу. На рисунке 12 приведен пример формы добавления нового правила, название окна содержит описания соответствующей операции.

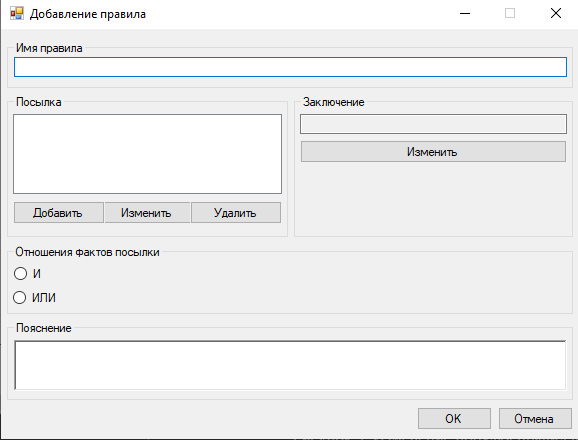


Рисунок 12. Пример добавления нового правила

Правило должно содержать название, при попытке добавить/изменить правило с пустым названием, будет выведено соответствующее сообщение с ошибкой (Рисунок 13).

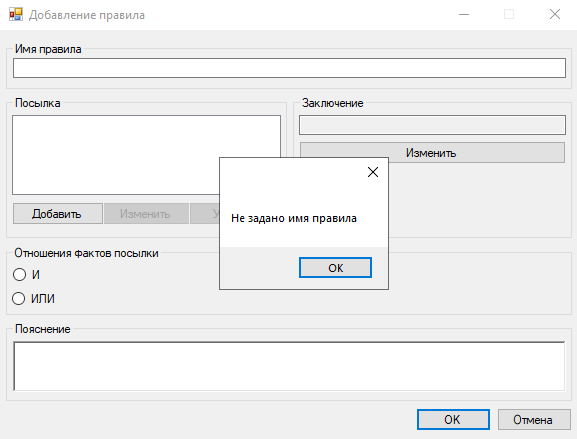


Рисунок 13. Ошибка добавления правила без имени

Посылка правила должно содержать минимум 1 факт. Если один факт не выбран, кнопки «Изменить» и «Удалить» недоступны. При попытке добавить правило без единого факта, появляется соответствующее сообщение об ошибке (рисунок 14).

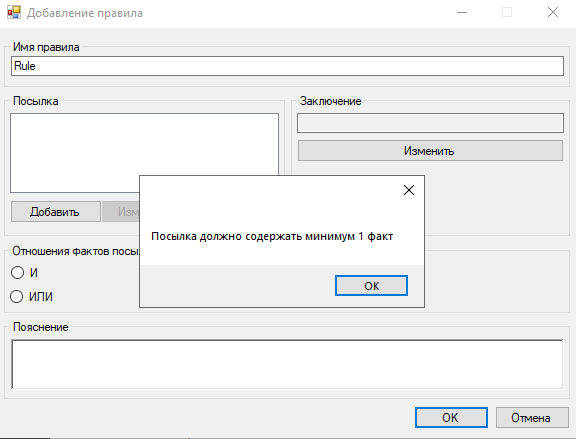


Рисунок 14. Добавление правила с пустой посылкой.

Аналогично правило должно содержать заключение, выбранный оператор отношения фактов посылки. На рисунках 15-16 изображены соответствующие сообщения об ошибках.

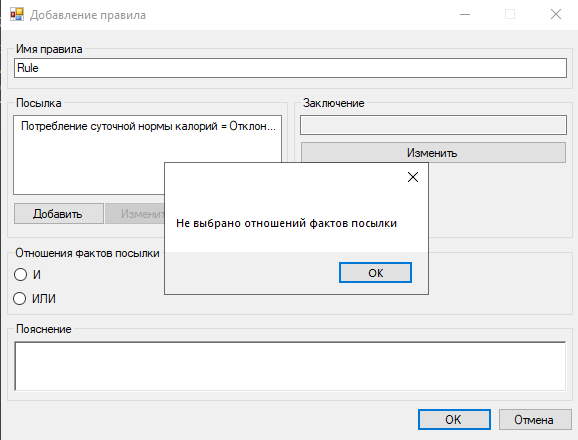


Рисунок 15. Ошибка добавления правила без выбора оператора отношения между фактами посылки

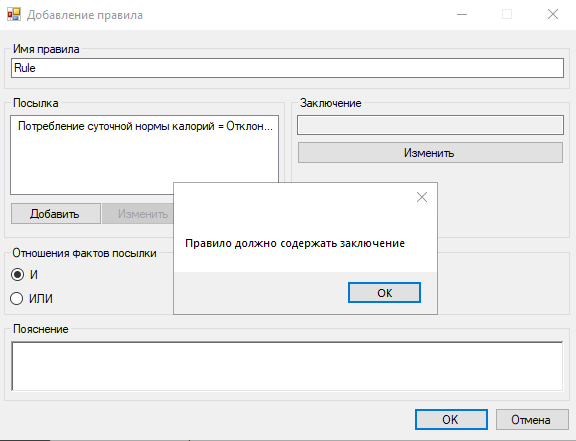


Рисунок 16. Ошибка добавления правила без заключения

При попытке задать правилу имя, которое уже содержится в базе знаний появляется сообщение с соответствующей ошибкой (рисунок 17).

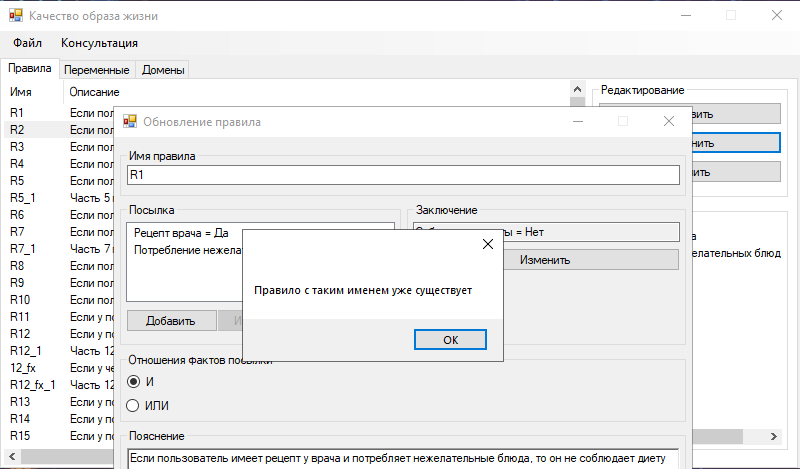


Рисунок 17. Ошибка добавления правила с существующим именем

При изменении/добавлении факта посылки правила открывается соответствующая форма. Форма содержит название, соответствующее операции (рисунки 18-19).

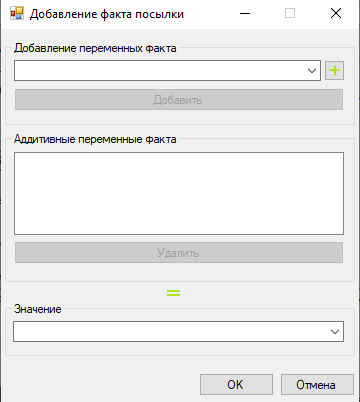


Рисунок 18. Форма добавления факта посылки

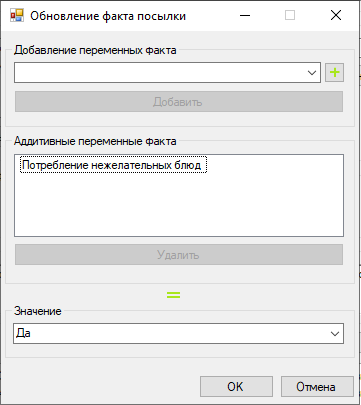


Рисунок 19. Форма обновления факта посылки

Кнопка «Добавить» становится активной только в том случае, если из переменная выбрана. Кнопка «Удалить» становится активной в том случае, если в списке «Аддитивные переменные факта» будет выбрана переменная. При открытии формы список доступных переменных заполняется из базы знаний (рисунок 20).

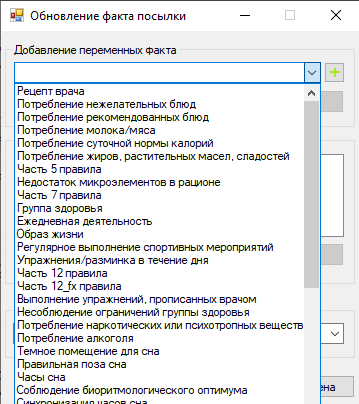


Рисунок 20. Заполнение списка переменных из базы знаний при работе с формой факта посылки

При нажатии на кнопку «+» пользователь может добавить новую переменную в базу знаний, которая сразу будет выбрана в соответствующем элементе (рисунки 20-21).

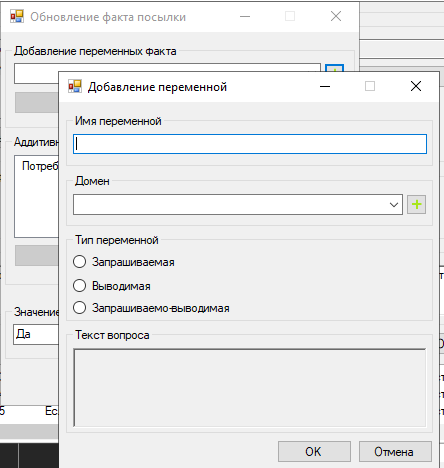


Рисунок 20. Форма добавления переменной, вызванная из формы фактов

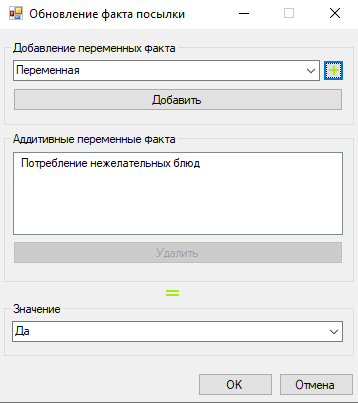


Рисунок 21. Автоматический выбор добавленной переменной на форме фактов

Поддерживается возможность добавления нескольких переменных в факт, однако в этом случае все переменные должны иметь числовой домен. При попытке добавить вторую переменную произойдет проверка обеих (далее - всех) на совпадение домена. На рисунке 22 приведен пример ошибки добавления нескольких переменных.

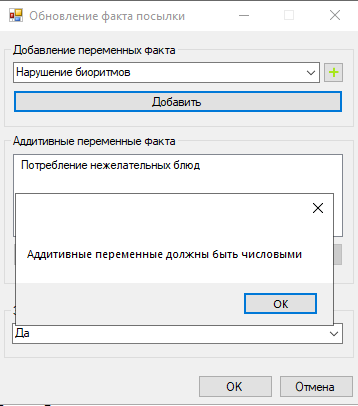


Рисунок 22. Ошибка добавления переменных факта не из одного числового домена

Список «Значения» заполняется возможными значениями домена выбранной переменной. Если переменных несколько, они все принадлежать одному домену. Если пользователь не выбран значение факта, выведется соответствующая ошибка (рисунок 23).

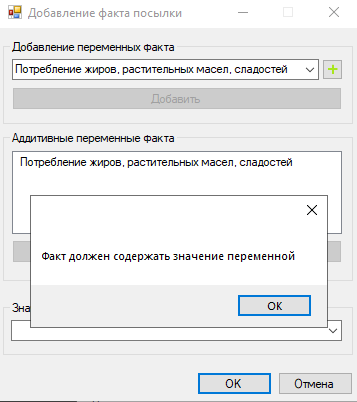


Рисунок 23. Ошибка добавления факта без значения

Раздел с переменными имеет вид, представленный на рисунке 24.

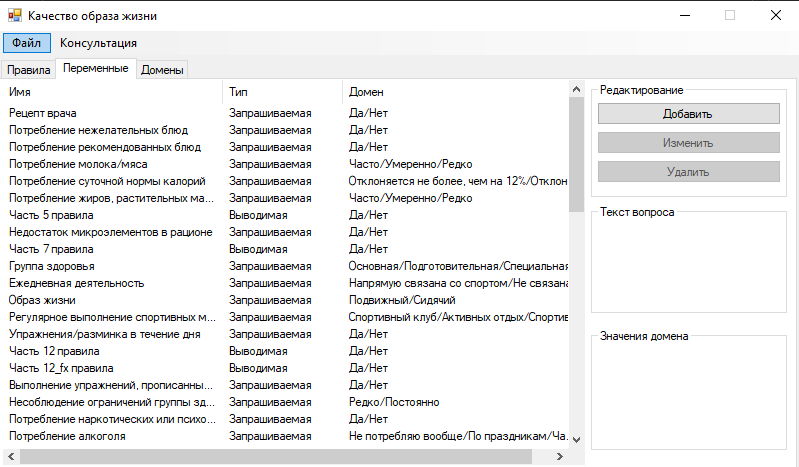


Рисунок 24. Раздел переменных на основной форме

Если ни одна переменная не выбрана, кнопки «Изменить» и «Добавить» недоступны. Форма отображает имя, тип и название домена переменных. Также текст вопроса переменной (если она не выводимая) и список всех значений домена переменных выводится в соответствующие окна справа от списка переменных (рисунок 25).

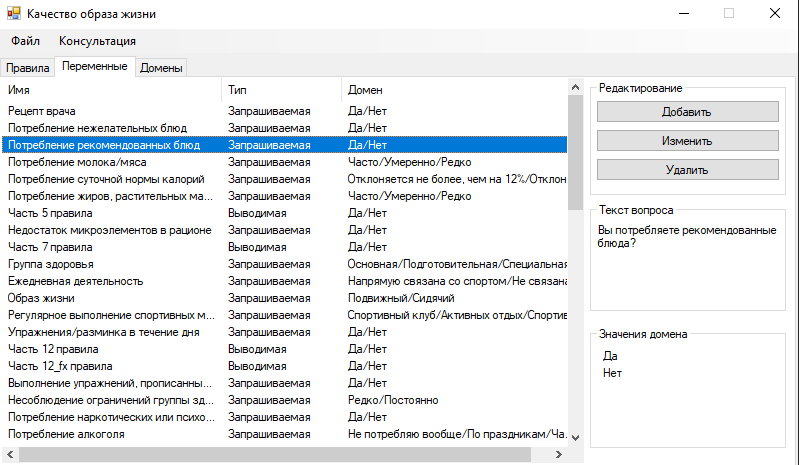


Рисунок 25. Пример отображения информации выбранной переменной

На рисунках 26-27 приведены примеры формы добавления/изменения переменной, в названии формы указана соответствующая операция.

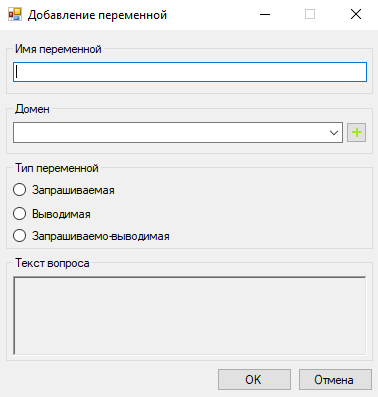


Рисунок 26. Форма добавления переменной.

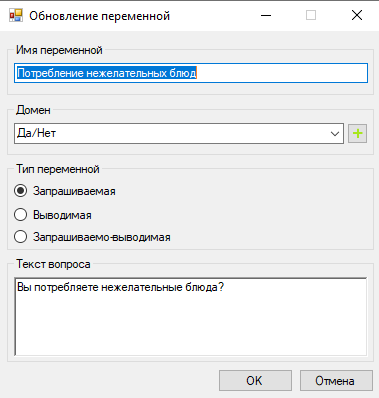


Рисунок 27. Форма обновления переменной

При попытке добавить переменную без имени, с существующим именем, без домена или без типа выводятся окна с соответствующими ошибками (рисунки 28-31)

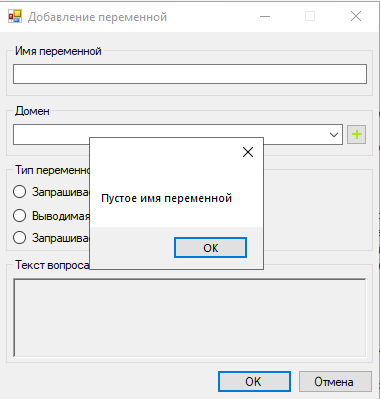


Рисунок 28. Ошибка пустого имени переменной.

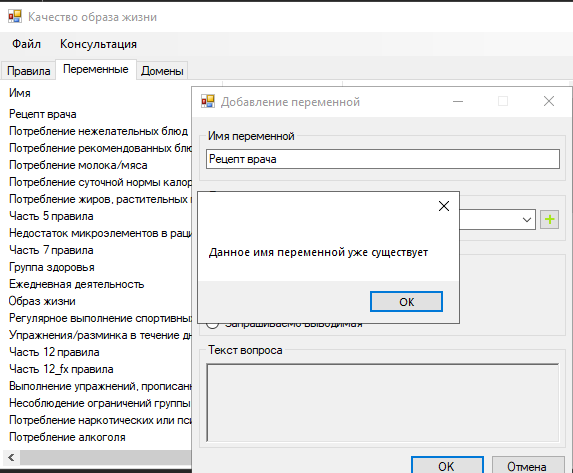


Рисунок 29. Ошибка добавления переменной с существующим имением

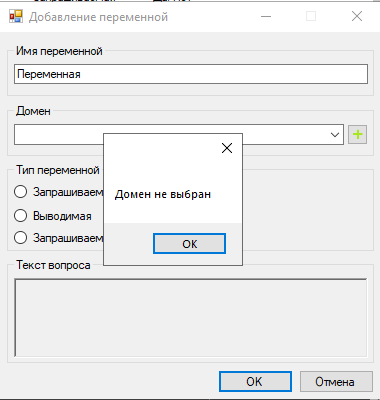


Рисунок 30. Ошибка добавления переменой без домена

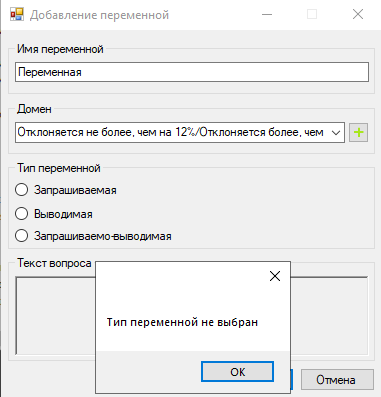


Рисунок 31. Ошибка добавления переменной без типа

Если для запрашиваемой или выводимо-запрашиваемой переменной не указан текста вопроса, он генерируется в формате «Значение переменной имя\_переменной?».

При нажатии на кнопку «+» открывается форма добавления нового домена. В случае успеха добавления домена, он сразу становится выбранным в соответствующем поле (рисунки 32-33).

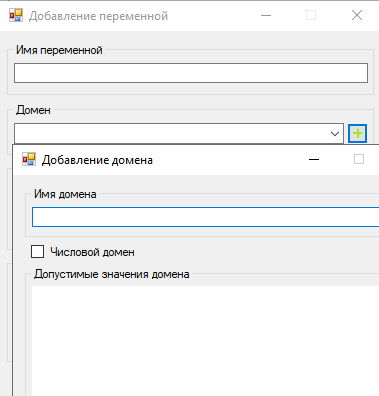


Рисунок 32. Добавление нового домена из формы переменной

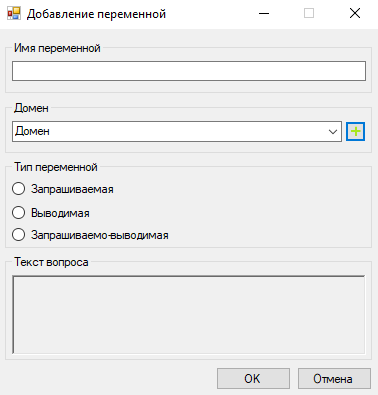


Рисунок 33. Успешно добавленный домен из формы переменной сразу становится выбранным.

Раздел основной формы с доменами имеет вид, изображенный на рисунке 34.

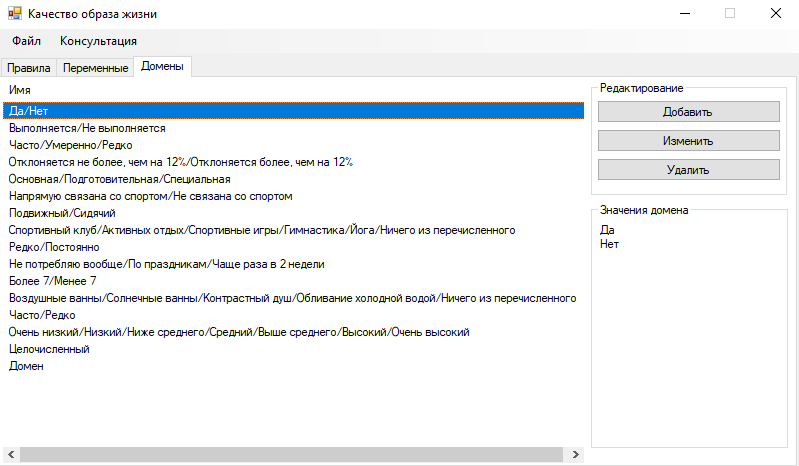


Рисунок 34. Раздел с доменами на основной форме

Раздел содержит список всех доменов из базы знаний, отображается название домена. Справа от списка в разделе «Значения домена» перечислены все значения выбранного домена. Если ни один домен не выбран, раздел «Значения домена» остается пустой, кнопки «Изменить» и «Удалить» не активные (рисунок 35).

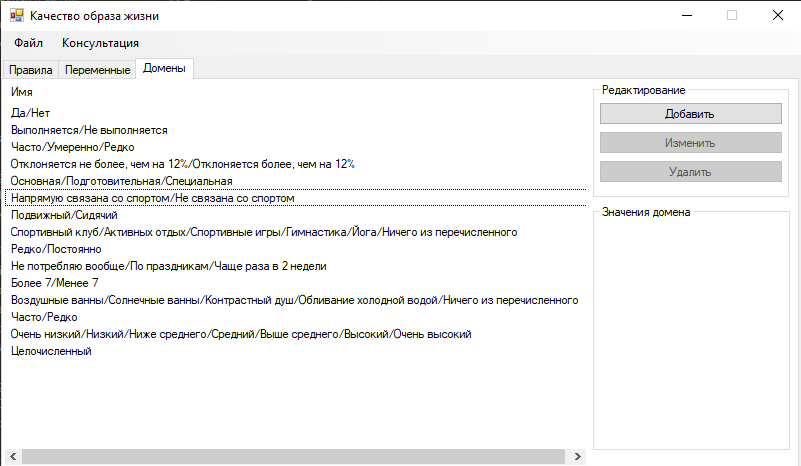


Рисунок 35. Ни один домен на форме не выбран

При добавлении/изменении домена открывается форма домена с соответствующим названием (рисунки 36-37).

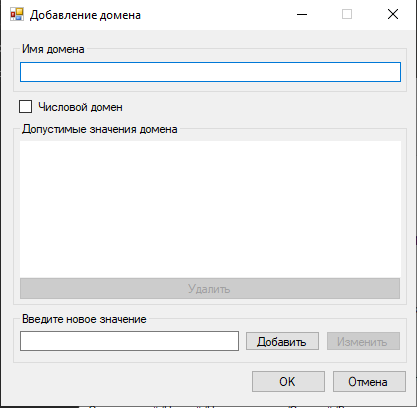


Рисунок 36. Форма добавления домена

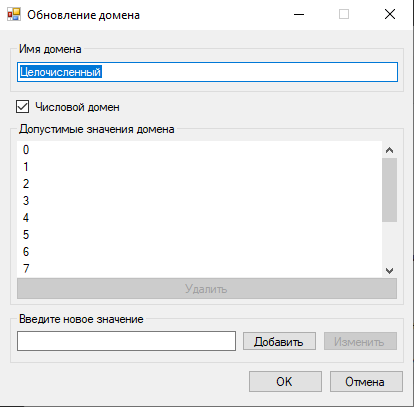


Рисунок 37. Форма обновления домена

При выборе значения домена из списка допустимых значений активизируются кнопки «Удалить» и «Изменить» (Рисунок 38).

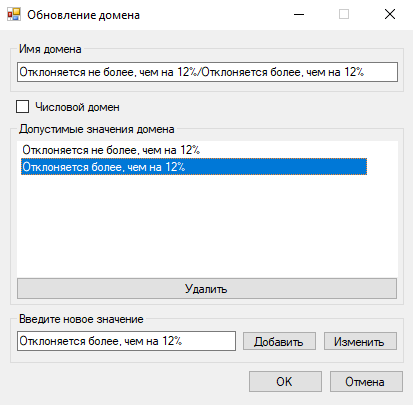


Рисунок 38. Активация кнопок «Удалить» и «Изменить» после выбора значения домена

При попытке сохранить домен без имени или числовой домен с нечисловыми значениями появляются соответствующие ошибки (рисунки 39-40).

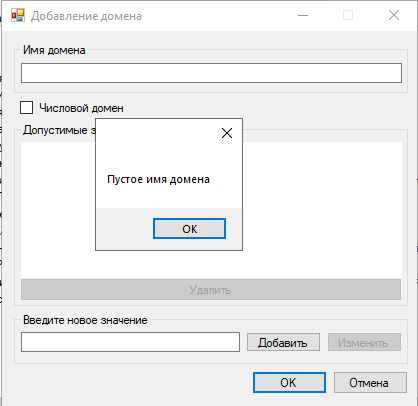


Рисунок 39. Ошибка добавления домена без имени

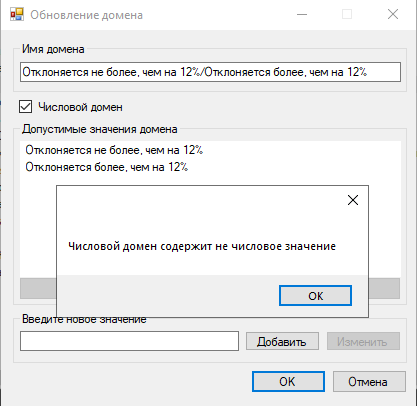


Рисунок 40. Ошибка попытки сохранения числового домена с нечисловыми значениями.

При попытке удалить домен, значение домена или переменную, которые используются в правилах или домена, который привязан к переменной, показывается соответствующее предупреждения (рисунки 41-43).

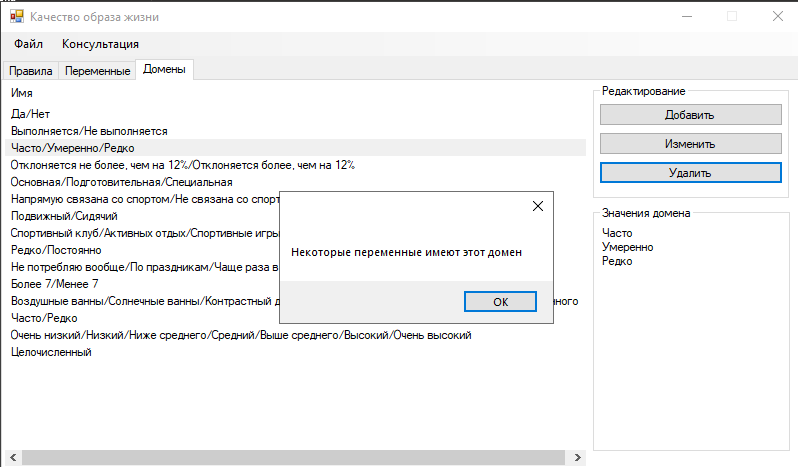


Рисунок 41. Попытка удаления домена, к которому привязан переменная

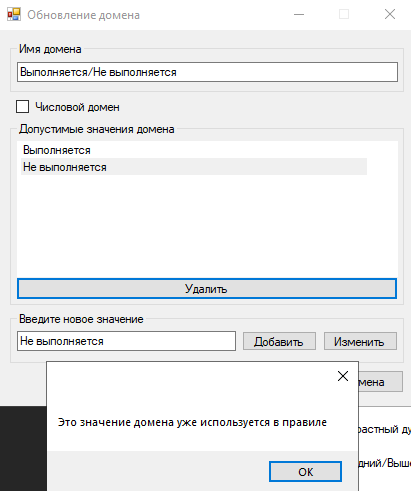


Рисунок 42. Попытка удаления значения домена, которое используется в правиле

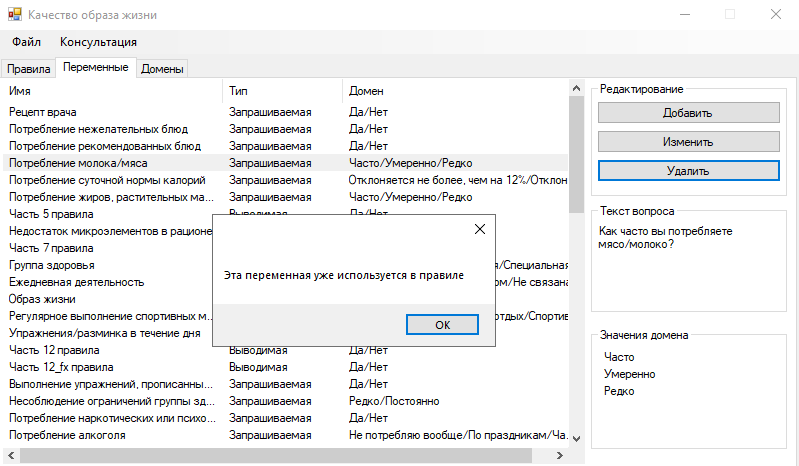


Рисунок 43. Попытка удаления переменной, которая используется в правиле

Форма консультации изображена на рисунке 44.

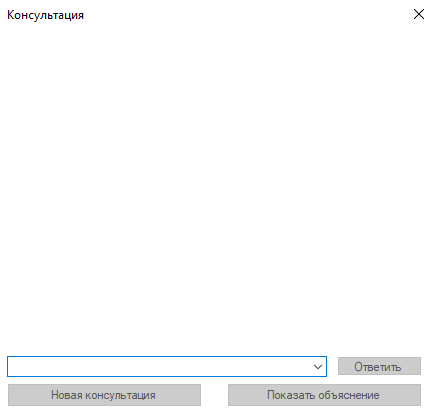


Рисунок 44. Форма консультации

Если ранее консультаций выполнено не было, кнопка «Показать объяснение» будет заблокирована. При открытии данной формы автоматически заполняется список доступных выводимых переменных (рисунок 45).

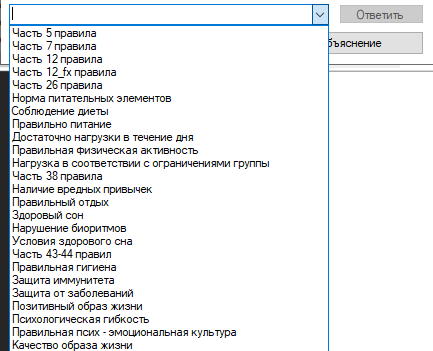


Рисунок 45. Список доступных переменных для консультации

При выборе целевой переменной становится доступна кнопка «Новая консультация». При нажатии на эту кнопку начинается консультация в формате чата. В процессе механизма логического вывода требуется получить значения запрашиваемых и запрашиваемо-выводимых переменных. В таких ситуациях в список, в котором изначально были целевые переменные, выводятся возможные значения запрашиваемой переменной. При выборе возможного значения активируется кнопка «Ответить». Пример консультации приведен на рисунке 46.

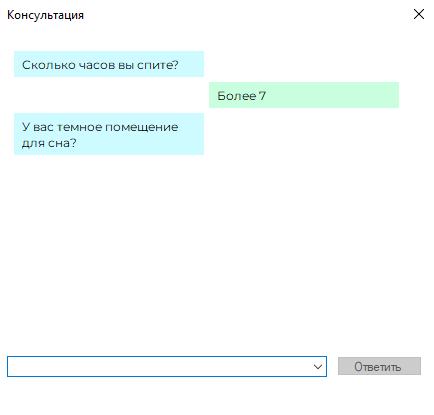


Рисунок 46. Пример вывода переменной «Правильный отдых» в процессе консультации

В случае удачи консультации в чате слева появляется зеленое сообщение с результатом. В случае неуспешной консультации (например, вывод переменной «Соблюдение диеты» при отсутствии рецепта врача) появляется красное сообщение (рисунки 47-48).

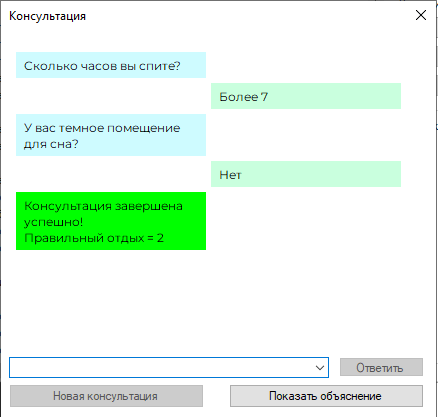


Рисунок 47. Пример успешной консультации

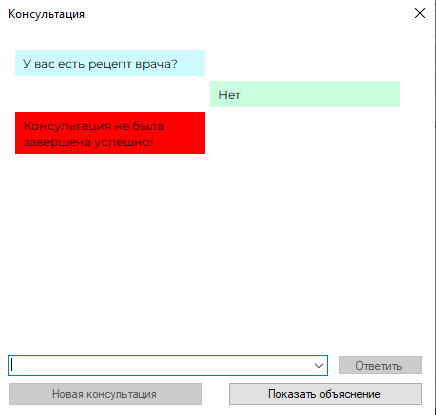


Рисунок 48. Пример неуспешной консультации

При нажатии на кнопку «Показать объяснение» появляется форма «Объяснение». Слева содержится дерево запрашиваемых, выведенных переменных, а также использованные правила в процессе доказательства главной цели и подцелей. Справа содержатся выведенные и запрошенные переменные из рабочей памяти, а также из значения (рисунок 49).

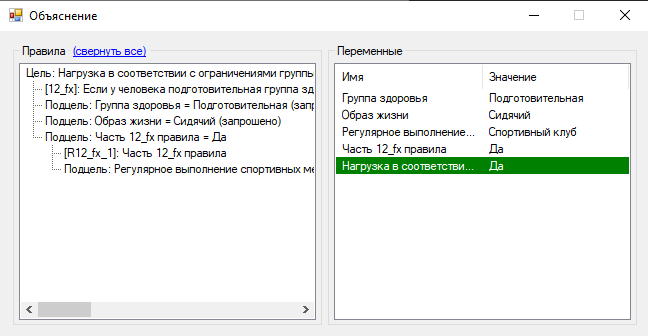


Рисунок 49. Пример формы объяснения после вывода переменной «Нагрузка в соответствии с ограничениями группы».

Также предусмотрена возможность свернуть и развернуть дерево при нажатии на синий элемент с соответствующим текстом над формой с деревом вывода.

Тестирование механизма логического вывода

1. Тест вывода самой короткой ветви. Целевая переменная «Качество образа жизни», имеется 6 правил, требующие выводимых 6 переменных и 1 (R25) правило, требующее 1 выводимую переменную. R25 правило по списку идет раньше всех, выполняется в случае, если переменная «Употребление наркотических или психотропных веществ» равняется «Да». Ожидается, что целевая переменная будет выведена из указанного последнего правила и у будет иметь значение «Очень низкий», результат приведен на рисунках 50-51.

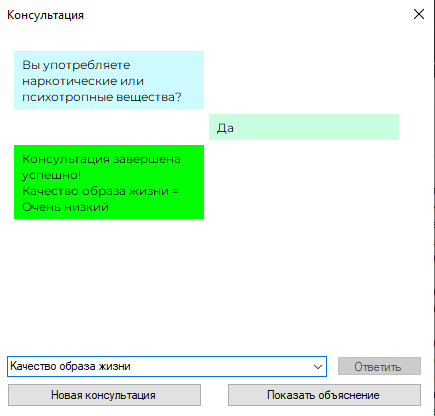


Рисунок 50. Тест 1, короткий пут, чат консультации

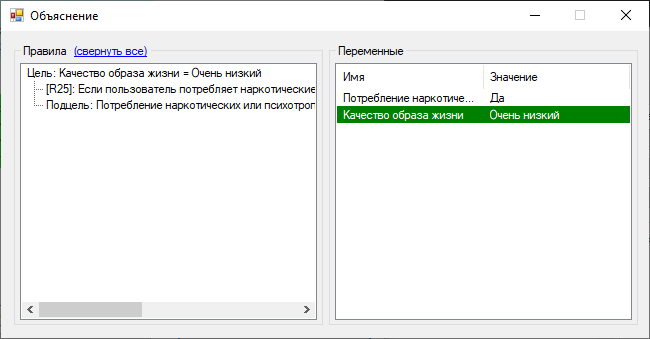


Рисунок 51. Тест 1, короткий путь, объяснение

Результат тест соответствует ожидаемому. Для проверки влияние порядка правил переместим данное правило в конец списка. Теперь должно сработать правило R50 (рисунок 52).

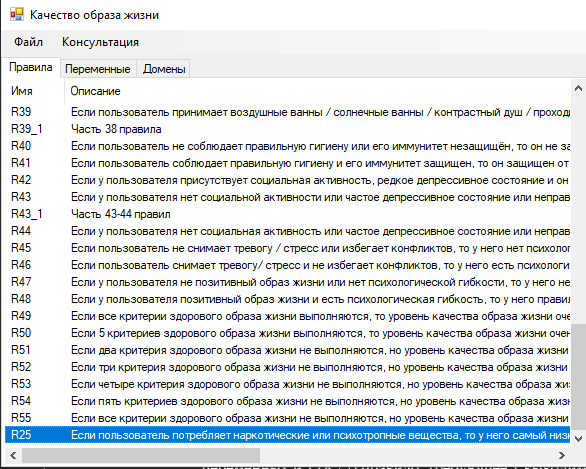


Рисунок 52. Тест 1, изменения порядка правил

На рисунках 53-54 представлены результаты тестирования.

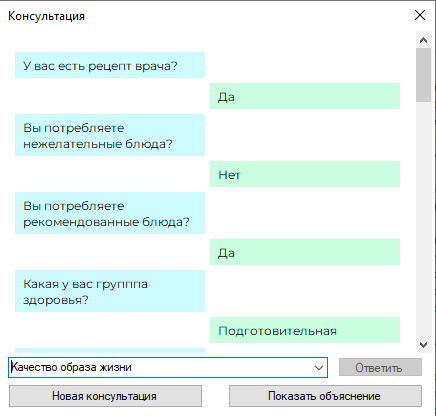


Рисунок 53. Тест 1, изменен порядок, чат

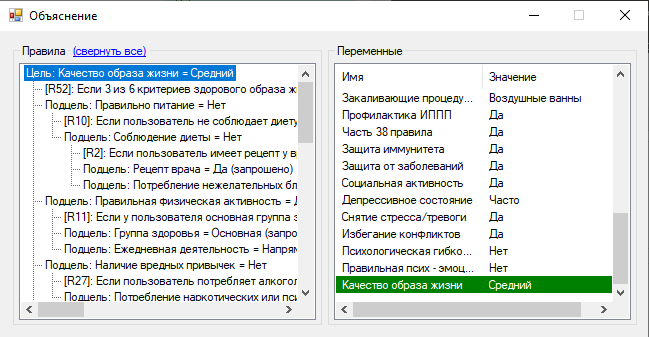


Рисунок 53. Тест 1, изменен порядок правил, объяснение

Результат тестирования соответствует ожидаемому.

1. Невозможность вывода переменной. Целевая переменная – «Соблюдение диеты», может быть выведена лишь в том случае, если пользователь имеет рецепт врача. Ожидается ошибка в процессе выведения переменной, результат представлен на рисунках 54-55.

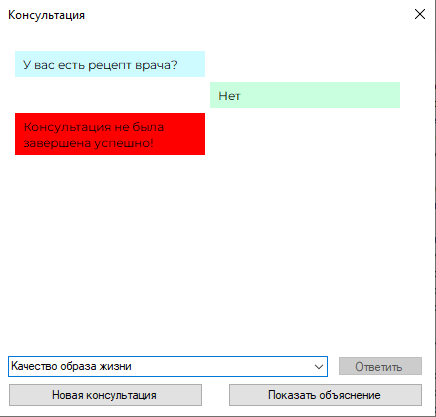


Рисунок 54. Тест 2, чат

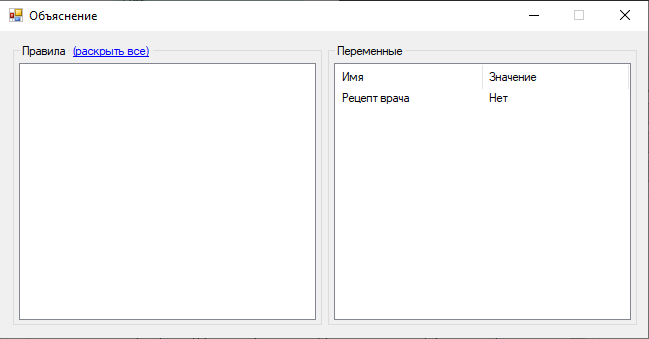


Рисунок 55. Тест 2, объяснение

Результат тестирования соответствует ожидаемому.

1. Тестирование вывода выводимо-запрашиваемой переменной. Созданная база знаний не содержит таких переменных, создадим пустую базу знаний и добавить 1 запрашиваемую, 1 выводим-запрашиваемую переменную, правило, по которому выводимо-запрашиваемая будет выводиться из запрашиваемой. Созданная база знаний приведена на рисунках 56-59.

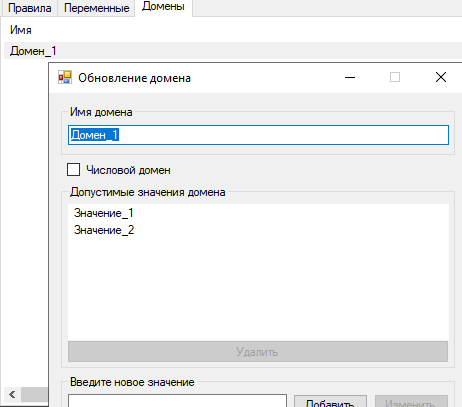


Рисунок 56. Тест 3, Тестовый домен

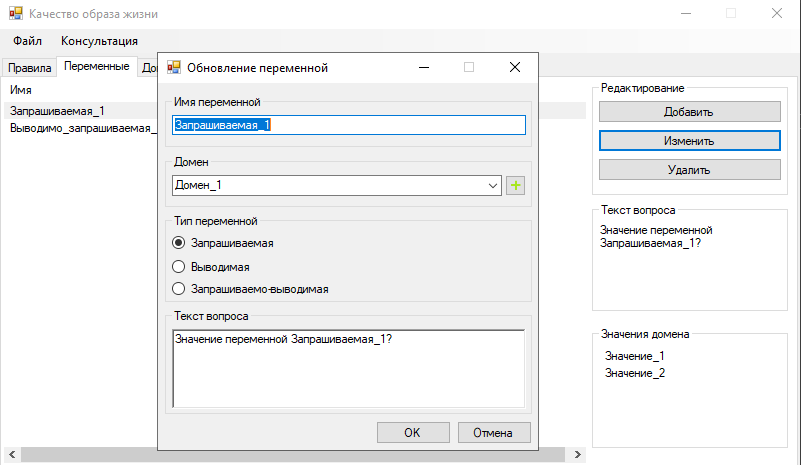


Рисунок 57. Тест 3. Запрашиваемая тестовая переменная

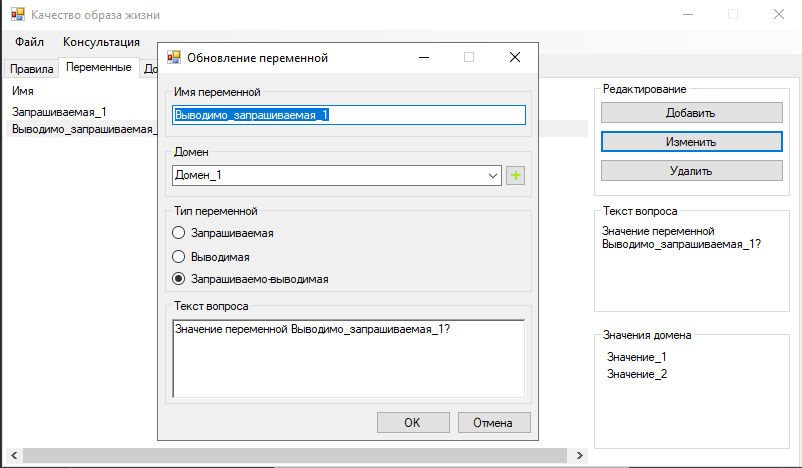


Рисунок 58. Тест 3, Выводимо-запрашиваемая переменная

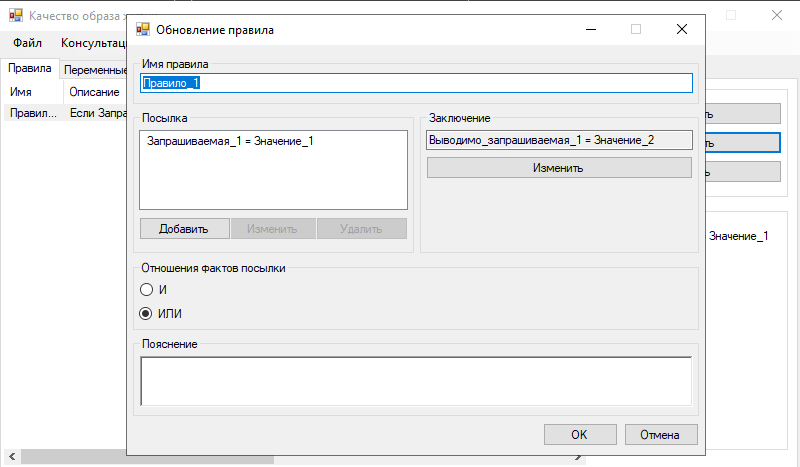


Рисунок 59. Тест 3, Тестовое правило

Ожидается, что сработает тестовое правило, при вводе значения «значение\_1» будет выведена выводимо-запрашиваемая переменная. Результат на рисунках 60-61.

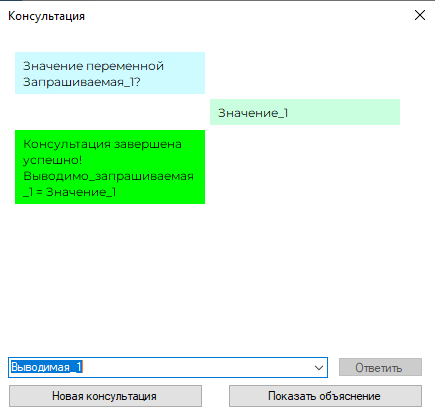


Рисунок 60. Тест 3, чат

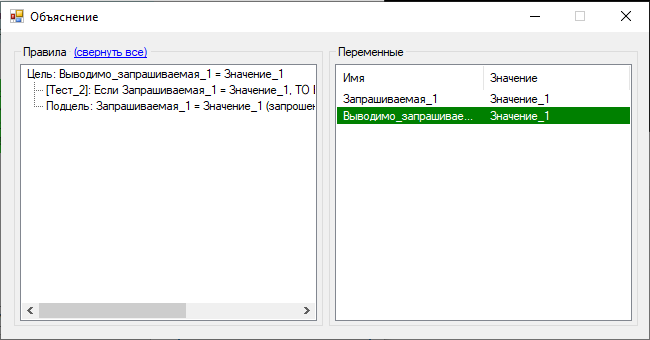


Рисунок 61. Тест 4, объяснение

Результат тестирования соответствует ожидаемому.

1. Тест запроса выводимо-запрашиваемой переменной. На рисунках 62-63 приведена обновленная база знаний для теста.

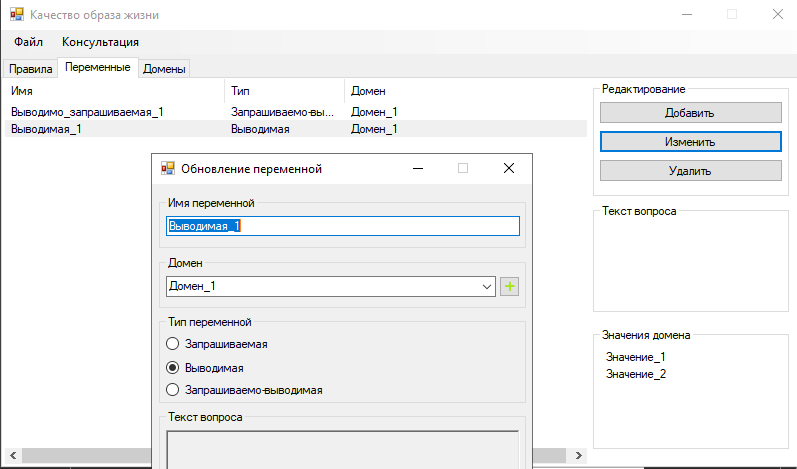


Рисунок 62. Тест 4, выводимая переменная

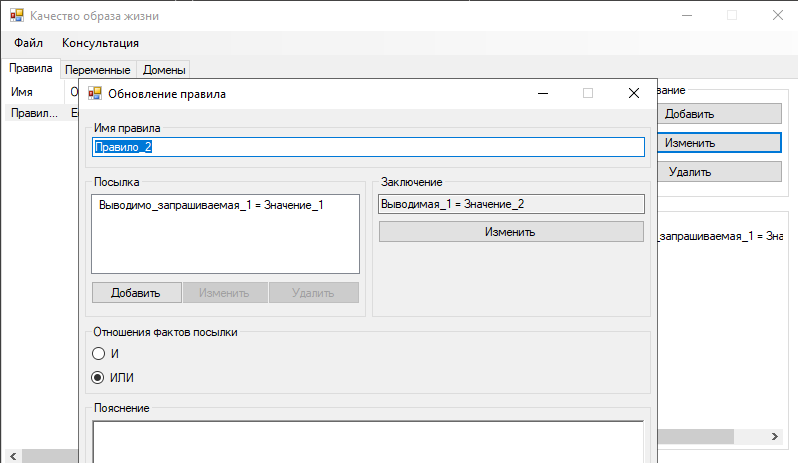


Рисунок 63. Тест 4, тестовое правило

Ожидается, что будет запрошена выводимо-запрашиваемая переменная, сработает добавленное правило и выведется переменная «Выводимая\_1», результаты приведены на рисунка 64-65.

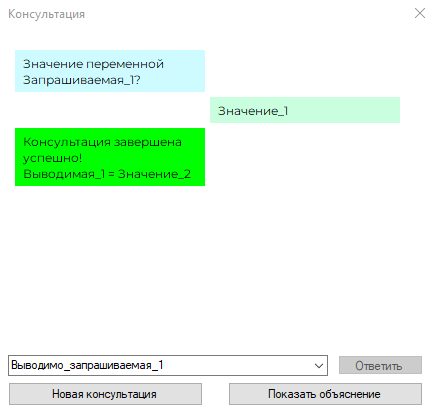


Рисунок 64. Тест 5, чат

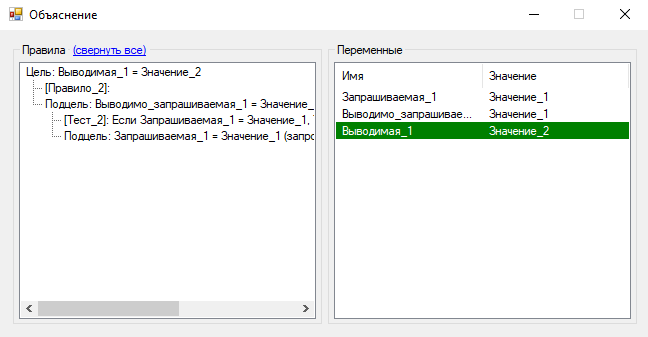


Рисунок 65. Тест 4, объяснение

Результат тестирования соответствует ожидаемому. Все тесты были пройдены успешно.

Приложение А

Правила экспертной системы «Качество образа жизни»:

R1: Если Рецепт врача = Да И Потребление нежелательных блюд = Нет И Потребление рекомендованных блюд = Да, ТО Соблюдение диеты = Да;

R2: Если Рецепт врача = Да И Потребление нежелательных блюд = Да, ТО Соблюдение диеты = Нет;

R3: Если Рецепт врача = Да И Потребление рекомендованных блюд = Нет, ТО Соблюдение диеты = Нет;

R4: Если Рецепт врача = Нет И Потребление молока/мяса = Умеренно И Потребление суточной нормы калорий = Отклоняется не более, чем на 12% И Потребление жиров, растительных масел, сладостей = Редко И Недостаток микроэлементов в рационе = Нет, ТО Норма питательных элементов = Выполняется;

R5: Если Рецепт врача = Нет И Часть 5 правила = Да, ТО Норма питательных элементов = Не выполняется;

R5\_1: Если Потребление молока/мяса = Часто ИЛИ Потребление молока/мяса = Редко, ТО Часть 5 правила = Да;

R6: Если Рецепт врача = Нет И Потребление суточной нормы калорий = Отклоняется более, чем на 12%, ТО Норма питательных элементов = Не выполняется;

R7: Если Рецепт врача = Нет И Часть 7 правила = Да, ТО Норма питательных элементов = Не выполняется;

R7\_1: Если Потребление жиров, растительных масел, сладостей = Умеренно ИЛИ Потребление жиров, растительных масел, сладостей = Часто, ТО Часть 7 правила = Да;

R8: Если Рецепт врача = Нет И Недостаток микроэлементов в рационе = Да, ТО Норма питательных элементов = Не выполняется;

R9: Если Соблюдение диеты = Да ИЛИ Норма питательных элементов = Выполняется, ТО Правильно питание = 1;

R10: Если Соблюдение диеты = Нет ИЛИ Норма питательных элементов = Не выполняется, ТО Правильно питание = 2;

R11: Если Группа здоровья = Основная И Ежедневная деятельность = Напрямую связана со спортом, ТО Правильная физическая активность = 1;

R12: Если Группа здоровья = Основная И Ежедневная деятельность = Не связана со спортом И Образ жизни = Сидячий И Часть 12 правила = Да, ТО Достаточно нагрузки в течение дня = Да;

R12\_1: Если Регулярное выполнение спортивных мероприятий = Спортивный клуб ИЛИ Регулярное выполнение спортивных мероприятий = Активных отдых ИЛИ Регулярное выполнение спортивных мероприятий = Спортивные игры ИЛИ Регулярное выполнение спортивных мероприятий = Гимнастика ИЛИ Регулярное выполнение спортивных мероприятий = Йога, ТО Часть 12 правила = Да;

12\_fx: Если Группа здоровья = Подготовительная И Образ жизни = Сидячий И Часть 12\_fx правила = Да, ТО Нагрузка в соответствии с ограничениями группы = Да;

R12\_fx\_1: Если Регулярное выполнение спортивных мероприятий = Спортивный клуб ИЛИ Регулярное выполнение спортивных мероприятий = Активных отдых ИЛИ Регулярное выполнение спортивных мероприятий = Спортивные игры ИЛИ Регулярное выполнение спортивных мероприятий = Гимнастика ИЛИ Регулярное выполнение спортивных мероприятий = Йога, ТО Часть 12\_fx правила = Да;

R13: Если Группа здоровья = Основная И Ежедневная деятельность = Не связана со спортом И Образ жизни = Сидячий И Регулярное выполнение спортивных мероприятий = Ничего из перечисленного, ТО Достаточно нагрузки в течение дня = Нет;

R14: Если Группа здоровья = Основная И Ежедневная деятельность = Не связана со спортом И Образ жизни = Подвижный И Упражнения/разминка в течение дня = Да, ТО Достаточно нагрузки в течение дня = Да;

R15: Если Группа здоровья = Основная И Ежедневная деятельность = Не связана со спортом И Образ жизни = Подвижный И Упражнения/разминка в течение дня = Нет, ТО Достаточно нагрузки в течение дня = Нет;

R16: Если Группа здоровья = Специальная И Выполнение упражнений, прописанных врачом = Да И Несоблюдение ограничений группы здоровья = Редко, ТО Нагрузка в соответствии с ограничениями группы = Да;

R17: Если Группа здоровья = Специальная И Выполнение упражнений, прописанных врачом = Нет, ТО Нагрузка в соответствии с ограничениями группы = Нет;

R18: Если Группа здоровья = Специальная И Выполнение упражнений, прописанных врачом = Да И Несоблюдение ограничений группы здоровья = Постоянно, ТО Нагрузка в соответствии с ограничениями группы = Нет;

R19: Если Группа здоровья = Подготовительная И Образ жизни = Сидячий И Регулярное выполнение спортивных мероприятий = Ничего из перечисленного, ТО Нагрузка в соответствии с ограничениями группы = Нет;

R20: Если Группа здоровья = Подготовительная И Образ жизни = Подвижный И Упражнения/разминка в течение дня = Да И Несоблюдение ограничений группы здоровья = Редко, ТО Нагрузка в соответствии с ограничениями группы = Да;

R21: Если Группа здоровья = Подготовительная И Образ жизни = Подвижный И Упражнения/разминка в течение дня = Да И Несоблюдение ограничений группы здоровья = Постоянно, ТО Нагрузка в соответствии с ограничениями группы = Нет;

R22: Если Группа здоровья = Подготовительная И Образ жизни = Подвижный И Упражнения/разминка в течение дня = Нет, ТО Нагрузка в соответствии с ограничениями группы = Нет;

R23: Если Достаточно нагрузки в течение дня = Да ИЛИ Нагрузка в соответствии с ограничениями группы = Да, ТО Правильная физическая активность = 1;

R24: Если Достаточно нагрузки в течение дня = Нет ИЛИ Нагрузка в соответствии с ограничениями группы = Нет, ТО Правильная физическая активность = 2;

R25: Если Потребление наркотических или психотропных веществ = Да, ТО Качество образа жизни = Очень низкий;

R26: Если Потребление наркотических или психотропных веществ = Нет И Часть 26 правила = Да, ТО Наличие вредных привычек = 1;

R26\_1: Если Потребление алкоголя = Не потребляю вообще ИЛИ Потребление алкоголя = По праздникам, ТО Часть 26 правила = Да;

R27: Если Потребление наркотических или психотропных веществ = Да ИЛИ Потребление алкоголя = Чаще раза в 2 недели, ТО Наличие вредных привычек = 2;

R28: Если Темное помещение для сна = Нет ИЛИ Правильная поза сна = Нет, ТО Условия здорового сна = Нет;

R29: Если Темное помещение для сна = Да И Правильная поза сна = Да, ТО Условия здорового сна = Да;

R30: Если Часы сна = Менее 7 ИЛИ Условия здорового сна = Нет, ТО Здоровый сон = Нет;

R31: Если Часы сна = Более 7 И Условия здорового сна = Да, ТО Здоровый сон = Да;

R32: Если Cоблюдение биоритмологического оптимума = Нет ИЛИ Синхронизация часов сна = Нет, ТО Нарушение биоритмов = Да;

R33: Если Cоблюдение биоритмологического оптимума = Да И Синхронизация часов сна = Да, ТО Нарушение биоритмов = Нет;

R34: Если Здоровый сон = Нет ИЛИ Нарушение биоритмов = Да, ТО Правильный отдых = 2;

R35: Если Здоровый сон = Да И Нарушение биоритмов = Нет, ТО Правильный отдых = 1;

R36: Если Уход за полостью рта = Нет ИЛИ Уход за полостью рта = Нет ИЛИ Уход за кожей = Нет, ТО Правильная гигиена = Нет;

R37: Если Уход за волосами = Да И Уход за полостью рта = Да И Уход за кожей = Да, ТО Правильная гигиена = Да;

R38: Если Закаливающие процедуры = Ничего из перечисленного ИЛИ Профилактика ИППП = Нет, ТО Защита иммунитета = Нет;

R39: Если Часть 38 правила = Да И Профилактика ИППП = Да, ТО Защита иммунитета = Да;

R39\_1: Если Закаливающие процедуры = Воздушные ванны ИЛИ Закаливающие процедуры = Солнечные ванны ИЛИ Закаливающие процедуры = Контрастный душ ИЛИ Закаливающие процедуры = Обливание холодной водой, ТО Часть 38 правила = Да;

R40: Если Правильная гигиена = Нет ИЛИ Защита иммунитета = Нет, ТО Защита от заболеваний = 2;

R41: Если Правильная гигиена = Да И Защита иммунитета = Да, ТО Защита от заболеваний = 1;

R42: Если Социальная активность = Да И Депрессивное состояние = Редко И Правильное выражение эмоций = Да, ТО Позитивный образ жизни = Да;

R43: Если Часть 43-44 правил = Да И Посещение психолога/психиатра = Да, ТО Позитивный образ жизни = Да;

R43\_1: Если Социальная активность = Нет ИЛИ Депрессивное состояние = Часто ИЛИ Правильное выражение эмоций = Нет, ТО Часть 43-44 правил = Да;

R44: Если Часть 43-44 правил = Да И Посещение психолога/психиатра = Нет, ТО Позитивный образ жизни = Нет;

R45: Если Снятие стресса/тревоги = Нет ИЛИ Избегание конфликтов = Да, ТО Психологическая гибкость = Нет;

R46: Если Снятие стресса/тревоги = Да И Избегание конфликтов = Нет, ТО Психологическая гибкость = Да;

R47: Если Позитивный образ жизни = Нет ИЛИ Психологическая гибкость = Нет, ТО Правильная псих - эмоциональная культура = 2;

R48: Если Позитивный образ жизни = Да И Психологическая гибкость = Да, ТО Правильная псих - эмоциональная культура = 1;

R49: Если Правильно питание + Правильная физическая активность + Наличие вредных привычек + Правильный отдых + Защита от заболеваний + Правильная псих - эмоциональная культура = 6, ТО Качество образа жизни = Очень высокий;

R50: Если Правильно питание + Правильная физическая активность + Наличие вредных привычек + Правильный отдых + Защита от заболеваний + Правильная псих - эмоциональная культура = 7, ТО Качество образа жизни = Очень высокий;

R51: Если Правильно питание + Правильная физическая активность + Наличие вредных привычек + Правильный отдых + Защита от заболеваний + Правильная псих - эмоциональная культура = 8, ТО Качество образа жизни = Выше среднего;

R52: Если Правильно питание + Правильная физическая активность + Наличие вредных привычек + Правильный отдых + Защита от заболеваний + Правильная псих - эмоциональная культура = 9, ТО Качество образа жизни = Средний;

R53: Если Правильно питание + Правильная физическая активность + Наличие вредных привычек + Правильный отдых + Защита от заболеваний + Правильная псих - эмоциональная культура = 10, ТО Качество образа жизни = Ниже среднего;

R54: Если Правильно питание + Правильная физическая активность + Наличие вредных привычек + Правильный отдых + Защита от заболеваний + Правильная псих - эмоциональная культура = 11, ТО Качество образа жизни = Низкий;

R55: Если Правильно питание + Правильная физическая активность + Наличие вредных привычек + Правильный отдых + Защита от заболеваний + Правильная псих - эмоциональная культура = 12, ТО Качество образа жизни = Очень низкий.