

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**

(Н И У « Б е л Г У »)

ФАКУЛЬТЕТ МАТЕМАТИКИ И ЕСТЕСТВЕННОНАУЧНОГО ОБРАЗОВАНИЯ

КАФЕДРА ИНФОРМАТИКИ, ЕСТЕСТВЕННОНАУЧНЫХ ДИСЦИПЛИН И МЕТОДИК
ПРЕПОДАВАНИЯ

**СОЗДАНИЕ ЭКСПЕРТНОЙ СИСТЕМЫ ДЛЯ АНАЛИЗА
КНИГООБЕСПЕЧЕННОСТИ ШКОЛЬНОЙ БИБЛИОТЕКИ**

Выпускная квалификационная работа
обучающегося по направлению подготовки
44.03.05 Педагогическое образование, профиль Информатика и иностранный
язык (английский)
очной формы обучения, группы 02041305
Косухиной Алены Николаевны

Научный руководитель
к. ф.-м. н., доцент
Беляева И.Н.

БЕЛГОРОД 2018

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 ВВЕДЕНИЕ В ТЕХНОЛОГИЮ СОЗДАНИЯ ЭКСПЕРТНЫХ СИСТЕМ.....	5
1.1 Понятие экспертная система	5
1.2 Описание процесса работы школьной библиотеки.....	8
1.3 Обзор существующих средств создания экспертных систем	14
2 РАЗРАБОТКА ЭКСПЕРТНОЙ СИСТЕМЫ	19
2.1 Язык программирования Prolog.....	19
2.2 Возможности SWI-Prolog	27
2.3 Создание графической оболочки.....	31
2.4 Демонстрация экспертной системы для анализа книгообеспеченности школьной библиотеки.....	40
2.5 Инструкция пользователю	49
ЗАКЛЮЧЕНИЕ	60
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	61

ВВЕДЕНИЕ

Современный мир невозможно представить без компьютеров. Они используются в разнообразных сферах человеческой деятельности, таких как промышленная индустрия, военная сфера, образование и других. Использование компьютеров в сфере образования позволяет применять новейшие технологии в учебном процессе, тем самым облегчая и упрощая его. В организациях, которые осуществляют образовательную деятельность, принято формировать библиотеки, позволяющие обеспечить реализацию образовательных программ. Большое внимание стоит уделить школьным библиотекам. Для автоматизации работы библиотекаря с библиотечным фондом и расчета коэффициента книгообеспеченности учебными изданиями образовательной деятельности можно использовать экспертные системы, предназначенные для этих целей. Экспертная система для анализа книгообеспеченности школьной библиотеки позволит ускорить, автоматизировать и облегчить процесс работы школьного библиотекаря.

Экспертные системы представляют собой программы, способные выступать в роли экспертов-консультантов в некоторой конкретной предметной области. Экспертные системы позволяют расширить область применения знаний, основанных на мнении авторитетных специалистов, сэкономить на оплате труда консультанту по данному вопросу, принять решение исходя из определенных данных и параметров. Экспертные системы используются для расширения и усиления профессиональных возможностей их пользователей. Традиционно экспертные системы применяются при интерпретации данных, диагностике, мониторинге, проектировании, прогнозировании, планировании и обучении [19].

ЦЕЛЬ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ – создание экспертной системы для анализа книгообеспеченности школьной библиотеки.

В соответствии с этой целью были сформированы следующие **ЗАДАЧИ**:

- Выявить основные потребности школьной библиотеки.

- Изучить принцип расчета коэффициента книгообеспеченности.
- Изучить основные средства создания экспертных систем.
- Изучить основные возможности языка логического программирования Prolog.

- Разработать экспертную систему для анализа книгообеспеченности школьной библиотеки.

ОБЪЕКТ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ – деятельность школьной библиотеки.

ПРЕДМЕТ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ – экспертная система для анализа книгообеспеченности школьной библиотеки.

Научная новизна состоит в разработке анализирующей экспертной системе.

1 ВВЕДЕНИЕ В ТЕХНОЛОГИЮ СОЗДАНИЯ ЭКСПЕРТНЫХ СИСТЕМ

1.1 Понятие экспертная система

В 20 веке сформулировали понятие искусственного интеллекта (ИИ) и предложили несколько определений этого понятия. Одним из первых определений было следующее: «Способ заставить компьютеры думать как люди». Популярность этого определения связана с тем, что подобные взгляды пропагандируются во многих научно-фантастических фильмах. Однако исток этого определения – знаменитый тест Тьюринга, который был предложен британским математиком и одним из первых исследователей в сфере компьютерных наук Аланом Тьюрингом (Alan Turing). В этом тесте экспериментатор пытался определить, кем является «лицо», с которым он производил обмен сообщениями с помощью дистанционной клавиатуры, человеком или компьютерной программой. Объект, который успешно проходил такой тест, считался обладающим сильным искусственным интеллектом. Данный термин распространяется специалистами, считающими, что ИИ должен базироваться на строгом логическом основании, в отличие от подхода, называемого ими слабым искусственным интеллектом, базирующимся на искусственных нейронных сетях, генетических алгоритмах и эволюционных методах [12].

Первая программа, которая прошла знаменитый тест Тьюринга, была написана в ходе проведения психологических экспериментов Стивеном Вейценбаумом в 1967 году [17]. В данный момент уровень знаний в этой области значительно увеличился, а методы взаимодействия с экспериментаторами стали наиболее совершенными.

Экспертные системы разработали как научно-исследовательские инструментальные средства в 60-х годах 20 столетия. Их рассматривали в качестве ИИ специального типа, который предназначен для успешного

решения сложных задач в узкой предметной области, примером такой области может быть медицинская диагностика заболеваний. Традиционная задача создания программы ИИ общего назначения, способная найти решение любой проблемы без конкретных знаний в определенной предметной области, оказалась слишком сложной. В начале 80-х годов прошлого столетия произошло коммерческое внедрение экспертных систем, и с того момента экспертные системы получили очень широкое распространение. В современном мире экспертные системы применяются в технике, науке, бизнесе, на производстве, а также в большинстве других сфер, в которых имеется вполне определенная предметная область [12].

Основной смысл выражения «вполне определенный» заключается в том, что эксперт способен определить этапы рассуждений, с помощью которых может быть решена любая задача из данной предметной области, а это означает, что аналогичные действия могут быть выполнены и с использованием экспертной системы [12].

В экспертных системах широко используются специальные знания для решения задач на уровне человека-эксперта. Термином эксперт принято обозначать личность, обладающую экспертными знаниями в определенной сфере. Это значит, что эксперт имеет знания и навыки, которые являются неизвестными или недоступными большинству людей. Эксперт способен более эффективно решать те задачи, которые большинство людей не может решить или решает менее эффективно. Первые разработанные экспертные системы содержали исключительно экспертные знания [25]. Но в настоящее время термин «экспертная система» достаточно часто применяется по отношению к любой системе, использующей технологию экспертных систем [12].

В качестве знаний в экспертных системах могут использоваться либо экспертные знания, либо общедоступные знания, которые могут быть получены из журналов, книг или от хорошо осведомленных людей. Но стоит

заметить, что обычные знания рассматриваются как понятия более низкого уровня, по сравнению с экспертными знаниями [12].

Экспертные системы – программы для компьютера, способные воспроизводить процесс решения проблемы человеком-экспертом [18].

Экспертная система – программа, использующая знания экспертов (специалистов) о некоторой конкретной узкоспециализированной предметной области и в пределах этой области способная принимать решения на уровне эксперта-профессионала [19].

Экспертная система – программа для компьютера, оперирующая со знаниями в определенной предметной области с целью выработки советов или решения проблем [24].

Знания эксперта относятся только к одной предметной области, которая представляет собой специальную проблемную область (медицина, наука и техника, финансы и т.д.), в которой может хорошо решать задачи только определенный эксперт [12]. Предназначение экспертных систем и людей-экспертов состоит в использовании их как экспертов в одной предметной области. Например, нельзя считать, что эксперт в области математики будет обладать экспертными знаниями, относящимися к области медицины.

Область знаний эксперта – это знания эксперта, касающиеся решения конкретных задач [12]. Например, медицинская экспертная система, которая предназначена для диагностирования кардиологических заболеваний, должна обладать большим объемом знаний об определенных симптомах, которые вызваны заболеваниями сердца. В этом случае областью знаний является медицина, а сами знания представляют собой сведения о заболеваниях, симптомах и методов их лечения. Связь между предметной областью и областью знаний можно увидеть на рисунке 1.



Рисунок 1 – Связь между предметной областью и областью знаний

Так же следует отметить преимущества экспертных систем:

- Повышенная доступность;
- Стабильность;
- Возможность получения экспертных знаний из многих источников;
- Повышенная надежность;
- Разъяснение;
- Быстрый отклик;
- Постоянно верный, лишенный эмоций и подробный ответ при любых условиях;
- Допустимость использования в качестве интеллектуальной обучающей программы;
- Возможность использования в качестве интеллектуальной базы данных [12].

1.2 Описание процесса работы школьной библиотеки

Школьная библиотека представляет собой структурное подразделение современного образовательного учреждения, а кроме этого информационное пространство, в котором гарантирован равноправный и открытый доступ к качественным источникам информации на любых носителях [27]. Как правило, школьная библиотека состоит из абонементов (место, где выдают книги), книгохранилища и читального зала. Книгохранилище может быть отдельным или совмещенным с абонементом. Читальный зал в маленьких библиотеках состоит из 2-3 столов для чтения. Так же в школьной библиотеке имеются каталоги и картотеки, которые могут в бумажном или электронном вариантах [23].

Целью школьной библиотеки является обеспечение учебного процесса необходимой литературой, а также увеличение уровня информационной культуры участников образовательного процесса. К задачам школьной библиотеки принято относить:

- обеспечение учебно-воспитательного процесса и самообучения посредством библиотечно-библиографического и информационного обслуживания учащихся и педагогов;
- формирование у обучающихся способностей самостоятельного библиотечно-информационного пользователя;
- развитие традиционных и овладение новыми библиотечными технологиями [29].

Для осуществления ключевых задач библиотека [29]:

1. Создает фонд библиотечно-информационных ресурсов школы:
 - составляет фонд учебными, педагогическими, художественными, справочными, научными, и научно-популярными документами на классических и нестандартных носителях информации;
 - осуществляет организацию, размещение, и сохранность документов.
2. Формирует информационную продукцию:

- реализует аналитико-синтетическую переработку информации;
- организует и ведет справочно-библиографический аппарат: каталоги (алфавитный, систематический), электронный каталог, картотеки (систематическую картотеку статей, тематические картотеки), базы данных согласно профилю среднего учебного заведения;

- создает библиографические пособия рекомендательного характера (обзоры, указатели, списки и т.п.);

- реализует уведомление пользователей о новой информационной продукции.

3. Реализовывает разграниченное библиотечно-информационное обслуживание обучающихся и сотрудников школы:

- предоставляет информационные ресурсы на разнообразных носителях на основе исследования их интересов и информационного спроса;

- формирует условия для реализации самостоятельности в процессе обучения, познавательной, творческой деятельности с опорой на коммуникацию; содействует в формировании навыков самообучения;

- организует обучение навыкам независимого библиотечного пользователя и потребителя информации, способствует интеграции комплекса знаний, умений и навыков работы с книгами и информацией;

- оказывает информационную помощь при решении задач, которые возникают во время учебной, самообразовательной и досуговой деятельности;

- организует различные массовые мероприятия, которые ориентированы на развитие общей и читательской культуры личности, а также содействует развитию критического мышления;

- помогает сотрудникам школы в организации образовательного процесса и интересного досуга обучающихся.

4. Реализовывает дифференцированное библиотечно-информационное обслуживание педагогических работников:

- выявляет информационные потребности и удовлетворяет запросы, связанные с обучением, воспитание и здоровьем детей;
- обнаруживает информационные потребности и удовлетворяет запросы в области педагогических инноваций и новых технологий;
- формирует банк педагогической информации как основы единой информационной службы школы, осуществляет накопление, систематизацию информации по предметам, разделам и темам;
- организует доступ к банку педагогической информации на любых носителях;
- организует просмотр электронных версий педагогических изданий;
- осуществляет текущее информирование (дни информации, обзоры новых поступлений и публикаций), информирование руководства школы по вопросам управления образовательным процессом;
- содействует деятельности педагогических работников в области создания информационных продуктов (документов, Web-страниц, баз данных и т.п.);
- способствует проведению занятий по формированию информационной культуры;
- представляет собой базу для проведения практических занятий по работе с различными информационными ресурсами.

5. Реализует дифференцированное библиотечно-информационное обслуживание родителей (или иных законных представителей) обучающихся:

- удовлетворяет запросы пользователей и сообщает о новых поступлениях в библиотеку;
- консультирует по вопросам учебных изданий для обучающихся.

Важным параметром работы школьной библиотеки является её книгообеспеченность. Существует несколько подходов к определению коэффициента книгообеспеченности. Согласно первому подходу,

коэффициент книгообеспеченности – это количество экземпляров учебников, приходящихся на одного учащегося. Согласно второму подходу, коэффициент книгообеспеченности представляет собой степень удовлетворения библиотекой потребности читателя в учебной литературе. Он определяется, как соотношение имеющегося к требуемому, ресурса к потребности [26].

Осуществляться расчет коэффициента книгообеспеченности может следующим образом:

1. По одной книге. Данная формула применяется в том случае, если книга используется в одном триместре (четверти) или период использования книги во внимание при расчете не принимается.

2. По одной книге за два семестра (четверти). Формула используется в случае изменения в том или ином триместре количества выделенных на дисциплину экземпляров книги, а также количества учащихся, изучающих данную дисциплину.

3. По нескольким книгам. Эта формула применяется, если по расчетному показателю используется не одна, а несколько книг.

Теперь подробнее о расчете коэффициента книгообеспеченности, который осуществляется по одной книге. Коэффициент обеспеченности книгой представляет собой частное от деления количества экземпляров данной книги на количество учащихся, изучающих дисциплину или дисциплины, по которым используется книга:

$$K = P \div S, \quad (1)$$

где P — количество имеющихся в библиотеке экземпляров данной книги;

S — количество учащихся, использующих данную книгу [15].

Если книга используется в двух триместрах, то коэффициент обеспеченности определяется как частное от деления среднего количества экземпляров данной книги на среднее число учащихся, которые используют книгу в данных триместрах:

$$K = P_{\text{ср}} \div S_{\text{ср}}. \quad (2)$$

В формуле (2) $P_{\text{ср}}$ — среднее количество экземпляров книги и $S_{\text{ср}}$ — среднее количество учащихся, которые используют данную книгу, вычисляются следующим образом

$$P_{\text{ср}} = (P_1 + P_2) \div 2, \quad (3)$$

$$S_{\text{ср}} = (S_1 + S_2) \div 2, \quad (4)$$

где P_1 — количество экземпляров книги в первом триместре, P_2 — количество экземпляров той же книги во втором триместре, S_1 — количество учащихся, которые используют данную книгу в первом триместре, S_2 — количество учащихся, которые используют данную книгу во втором триместре.

Подробнее расчете коэффициента книгообеспеченности по нескольким книгам. Его принято определять, как сумму экземпляров используемых книг, деленная на сумму числа учащихся по каждой книге. Для n наименований коэффициент книгообеспеченности равен:

$$K = P_1 + P_2 + \dots + P_n \div S_1 + S_2 + \dots + S_n, \quad (5)$$

где P_1, P_2, P_n — количество экземпляров каждой книги, S_1, S_2, S_n — количество учащихся, использующих каждую книгу.

Например, рассмотрим расчет коэффициента книгообеспеченности по дисциплине. Допустим, что есть 3 наименования: $a=21$ экз.; $b=24$ экз.; $c=13$ экз. Число учащихся = 24.

Для начала решим первым способом, который был описан выше. Для этого определим коэффициенты книгообеспеченности (1) по каждой книге:

$$K_a = 21:24 = 0,88$$

$$K_b = 24:24 = 1$$

$$K_c = 13:24 = 0,54$$

Затем суммируем коэффициенты книгообеспеченности по каждой книге:

$$(0,88+1+0,54) = 2,42$$

После этого разделим полученную сумму на число используемых наименований:

$$2,42:3 = 0,81$$

Теперь решим эту же задачу вторым способом, т.е. вычислим коэффициент обеспеченности по формуле (2). Для этого суммируем количество учебной литературы, которая используется по дисциплине:

$$21 + 24 + 13 = 58 \text{ экз.}$$

Затем суммируем количество учащихся, использующих каждое наименование:

$$24 + 24 + 24 = 72 \text{ человек.}$$

После этого делим сумму экземпляров используемой учебной литературы на суммарное количество читателей:

$$58:72 = 0,81.$$

В результате мы видим, что оба метода являются точными. Чаще всего для анализа книгообеспеченности литературой учебного процесса используется первый способ.

1.3 Обзор существующих средств создания экспертных систем

Раньше создания и проектирования экспертной системы требовало больших затрат в плане времени (до нескольких лет) и людей (несколько десятков). В настоящее время имеется ряд средств, ускоряющих процесс создания. Эти средства называются инструментальными или инструментарием. Использование инструментальных средств создания экспертных систем позволяет сократить время, затраченное на создания, в 3-5 раз.

Инструментальное средство разработки экспертных систем представляет собой язык программирования, который используется инженером знаний или (и) программистом для построения экспертной системы. Данный инструмент отличается от обычных языков

программирования тем, что обеспечивает удобные способы представления сложных высокоуровневых понятий [14].

По своему назначению и функциональным возможностям инструментальные программы, которые применяются при проектировании экспертных систем, можно разделить на четыре достаточно большие категории:

1. Оболочки экспертных систем;
2. Языки программирования высокого уровня;
3. Среда программирования, поддерживающая несколько парадигм;
4. Дополнительные модули [14].

Если рассматривать инструментальные средства создания экспертных систем, то наиболее популярными языками программирования являются LISP и PROLOG, а среди оболочек экспертных систем KEE, CENTAUR, G2 и GDA, AT_ТЕХНОЛОГИЯ, которые предоставляют в распоряжение разработчика широкий набор для комбинирования систем представления знаний, языков программирования, объектов и процедур.

LISP происходит от английского LISt Processing, что переводится как «обработка списков». LISP является семейством языков программирования, которое основано на представлении программы системой линейных списков символов, являющихся основной структурой данных языка. Он считается вторым старейшим языком программирования высокого уровня после Fortran [2].

Язык LISP был предложен Дж. Маккарти в 1960 году. Он был ориентирован на разработку задач численного характера. Основной областью применения данного языка является обработка списков. Большим достоинством языка LISP является его функциональная направленность, т. е. программирование ведется с помощью функций [2].

Доказательством простоты использования LISP является его широкое применение в исследовательских работах по программированию сложных методов обработки знаний. Большая часть программ искусственного

интеллекта, которые являются значимыми, запрограммирована на LISP или на языке более высокого уровня, основанном на LISP [9]. Если говорить о недостатках данного языка, то к ним можно отнести сложность освоения и высокую стоимость средств разработки [16].

Проанализируем язык логического программирования PROLOG. Программа на этом языке состоит из набора фактов, некоторых отношений между объектами данных (фактами) и набором правил (образцами отношений между объектами базы данных). Правила и факты вводятся в базу данных и применяются с целью установления того, какие подстановки для переменных в запросе (именуемые унификацией) согласуются с информацией в базе данных. Язык PROLOG, как интерпретатор, приглашает пользователя вводить данные. Пользователь набирает запрос или имя функции. В результате выводится значение (истина – YES, или ложь – NO) этого запроса, а также возможные значения переменных запроса, присвоение которых делает запрос истинным (т.е. унифицирует запрос). Несмотря на то, что выполнение программы на языке PROLOG основывается на спецификации предикатов, оно напоминает выполнение программ на языках LISP или ML. К минусам данного языка принято причислять отсутствие механизма прямого вывода [30].

Следует обратить внимание так же и на программную среду для разработки экспертных систем CLIPS. Наименование языка CLIPS – сокращение от английского C Language Integrated Production System. Этот язык был разработан в Центре космических исследований NASA в середине 80-х годов 20-го столетия и во многом похож с языками, которые созданы на базе LISP.

В CLIPS применяются 2 основные конструкции: правила и факты. Факты могут быть упорядоченными и неупорядоченными, их принято называть шаблонами или фреймами. Мощный инструмент создания правил включает в себя возможность задания приоритета правил, которая даёт возможность пользователю устанавливать преимущества для своих правил. К

главным плюсам данного языка относят свободное распространение, мультиплатформенность, полную открытую документацию и объектно-ориентированное расширение CLIPS Object-Oriented Language (COOL) [12].

Еще одна оболочка для создания экспертной системы «Рапана» (ЭС «Рапана»), которая может применяться как для создания простых баз знаний (БЗ) с целью локализованного использования, так и быть основой для решения глобальных задач любой сложности.

База знаний основана на реляционной модели, а её реализация с помощью современных средств СУБД обеспечивает масштабируемость, значительную скорость обработки информации, возможность доступа к различным видам данных. Количество задач хранящихся в одной БЗ практически неограниченно, а количество баз знаний на одном компьютере ограничено только ёмкостью жесткого диска. Одной из особенностей ЭС «Рапана» является легкость формирования базы знаний. Введение данных с помощью специальной программы Когнитограф возможен в двух вариантах: первый похож на работу в программах типа материального учета, а второй – визуальный. Еще одна особенность – визуальная пошаговая отладка, предоставляющая создателю БЗ возможность видеть все, что происходит в ЭС в каждый момент её выполнения. ЭС «Рапана» используется как на локальных компьютерах, так и в режиме on-line. Машина вывода ЭС «Рапана» использует прямой метод вывода, т. е. поиск идет не от цели, а от данных [11].

Также из программ оболочек интерес представляет средство на базе JAVA: Corvid Exsys. Эта система нацелена на различные классы пользователей в зависимости от их подготовки в области программирования и искусственного интеллекта. Она имеет развитый интерфейс с современными СУБД и электронными таблицами, средства сбора статистики и т. д. Система может работать на разных вычислительных платформах под управлением различных операционных систем, а также имеет поддержку сетевой конфигурации.

Экспертная система Exsys представляет собой интеллектуальную систему, которую можно использовать для разработки базы знаний в любой сфере. При этом знания представляются в виде продукционных правил. Следует отметить, что в систему включены средства отладки и тестирования программы, редактирования для модификации знаний и данных [20].

Особенность системы Exsys – это возможность вывода информации из других правил, которая позволяет разбивать сложные задачи на более простые подзадачи. Также в Exsys реализованы механизмы и прямого, и обратного вывода. В системе Exsys, как и в любой другой экспертной системе, существует возможность объяснения полученных результатов. Если возникает необходимость выяснить весь ход вычислений, то это можно выполнить вплоть до исходных данных [20].

Во время процесса поиска решений система Exsys способна выполнять ряд вычислительных задач, но некоторые задачи выходят за рамки возможностей системы. Поэтому Exsys обладает возможностью вызова внешних программ для выполнения вычислений. Данные из внешней программы могут передавать данные Exsys через БД или spread sheet. Exsys способна вызывать практически все MSDOS-программы в качестве внешних программ [13].

2 РАЗРАБОТКА ЭКСПЕРТНОЙ СИСТЕМЫ

2.1 Язык программирования Prolog

Любой существующий язык программирования ориентирован на определенный круг задач, в процессе решения которых он является наиболее эффективным. Проекты, которые связаны с созданием систем искусственного интеллекта, являются наиболее типичными для языка Prolog. В задачах, которые связаны с манипулированием на естественном языке, также часто используется Prolog. Данный язык обладает достаточно сильными средствами, которые позволяют получать сведения из баз данных и знаний. Стоит заметить, что его методы поиска принципиально отличаются от классических [28].

Prolog является языком программирования, который основывается логике предикатов, а не на алгоритме. В том случае, если программа на алгоритмическом (процедурном) языке представляет собой последовательность инструкций, выполняющихся в установленном порядке, то Prolog, являясь декларативным языком, содержит только описание задачи, а Prolog-машина осуществляет поиск решения и при этом руководствуется только данным описанием и использует механизм поиска с возвратом и унификацию [28].

Создание языка Prolog началось в 1970 г. Аланом Кулмероэ (Alain Colmerauer) и Филиппом Русселом (Philippe Roussel) в университете города Марсель во Франции. Они хотели создать язык, делающий логические заключения на основе заданного текста. Частичная мотивация их работы заключалась в желании применить использование логики в качестве декларативного языка представления знаний с процессуальным подходом к представлению знаний. Этот подход считался популярным в Северной Америке в конце шестидесятых и начале семидесятых годов 20 века.

Название Prolog Филипп Руссел выбрал как аббревиатуру от французского PROgrammationen LOGique (PROLOG). В 1972 году была завершена первая реализация этого языка с использованием компилятора Николауса Вирта «Algol-W». В 1973 году были заложены основы современного языка Prolog. Распространение этого языка среди людей, занимающихся логическим программированием, происходило в основном в следствие личных контактов, а не через коммерческое продвижение продукта [28].

В университете Эдинбурга в 1977 году Д. Уоррен и Ф. Перейра создают очень результативный компилятор языка Prolog для электронно-вычислительной машины DEC-10, этим они переводят в практическую плоскость методы логического программирования. Первым и единственным стандартом языка фактически стала данная реализация Prolog, также известная как «эдинбургская версия». Как правило, в том случае если современная Prolog-система не поддерживает эдинбургский Prolog, то в ее состав входит подсистема, переводящая Prolog-программу в «эдинбургский» вид.

Позднее в 1980 году в Великобритании К. Кларк и Ф. Маккейб создали версию Prolog для персональных электронно-вычислительных машин. А в 1996 году международной организацией по стандартизации (ISO – International Organization for Standardization) была разработана рабочая группа ISO/IEC JTC1/SC22AVG17 по языку программирования Prolog и принят стандарт ISO/IEC 13211-1 (Part1: General core). У этого стандарта существует две корректировки в 2007 и 2012 годах. Помимо этого, в 2000 году был принят дополнительный стандарт ISO/IEC 13211-2 (Part 2: Modules), который типизирует использование модулей в Prolog.

Утверждения языка одновременно являются и записями, которые подобны записям в базе данных, и правилами, которые несут в себе способы их обработки, потому что равнозначность представления программы и данных (декларативность) – базовый принцип языка. В сочетании эти качества приводят к следующему: по мере работы Prolog-системы знания

накапливаются. При этом накапливаются как данные, так и правила. Поэтому Prolog-системы считают естественной средой для накопления базы знаний, которая в свою очередь является важным компонентом интеллектуальной системы. Они предназначены для поиска способов решения проблем из некоторой предметной области, опираясь на записи базы знаний и на пользовательское описание ситуации [28].

Простые базы знаний можно использовать для создания экспертных систем, хранящих данные в организации: статей технического обеспечения, руководств, документации. Основной целью создания подобных баз является помощь менее опытным людям в поиске уже имеющегося описание способа решения некой проблемы. Говоря о сферах использования языка Prolog, следует сразу отметить, что он недостаточно приспособлен для решения задач, которые связаны с обработкой графики, а также вычислениями или численными методами. Но при этом данный язык может с успехом применяться в компьютерной алгебре, которая, в отличие от численных методов, занимается реализацией аналитических методов решения математических задач на компьютере, а также предполагает, что исходные сведения, как и результаты полученные в процессе решения, сформулированы в аналитическом виде [8]. Основными областями применения Prolog являются следующие направления:

- создание быстрых прототипов прикладных программ;
- координация производственных процессов;
- разработка динамических реляционных баз данных;
- перевод с одного языка на другой;
- создание естественно-языковых интерфейсов;
- реализация экспертных систем и оболочек экспертных систем;
- создание пакетов символьных вычислений;
- доказательства теорем;

- интеллектуальные системы, в которых применяются возможности языка Prolog по обеспечению дедуктивного вывода для проверки различных теорий.

Язык Prolog используется и в ряде других сфер, например, при решении задач формирования сложных расписаний. Он используется в разных системах, но обычно не в качестве основного языка, а в качестве языка для разработки некоторой части системы. Достаточно часто Prolog используют для написания функций взаимодействия с базами данных [28].

Prolog применяется также в сложных поисковых системах. Эти системы выполняют не только поиск, но и играют роль некой «отвечающей системы» – программного комплекса, умеющего получать информацию из баз данных и большой выборки текстовых файлов, а затем вести диалог с пользователем, отвечая, в обычном понимании этого слова, на его вопросы.

Применение Prolog имеет место и при написании новых специфичных языков программирования. Примером является функциональный язык Eerland, построенный на основе Prolog. Можно сказать, что Eerland является усовершенствованием Prolog для некоторых специфических целей, связанных с задачами реального времени [28].

Теоретической основой языка Prolog является раздел символической логики, который называют исчислением предикатов. Предикат представляет собой логическую функцию, которая выражает некоторое отношение между своими аргументами и принимает значение «истина», если это отношение имеется, или «ложь», если оно отсутствует.

В отличие от процедурных языков, в которых программист должен описать процедуру решений шаг за шагом, при использовании Prolog достаточно описать задачу и основные правила её решения. Язык Prolog имеет ряд свойств, которыми не обладают процедурные языки:

- Механизм вывода с поиском и возвратом;
- Встроенные механизмы сопоставления с образцом;

- Простая, но при этом выразительная структура данных с возможностью её изменения [28].

Prolog считается декларативным языком, т.е. обладая необходимыми фактами и правилами, он может применять дедуктивные выводы для решения задач программирования. Prolog-программа содержит описание проблемы, представляющие собой набор логических утверждений в форме фактов (Максим любит Аню. Федор любит бокс.) или в форме правил (Максим любит X, если Федор любит X (Максим любит тоже самое, что любит и Федор)).

На основании имеющихся правил и фактов Prolog делает выводы. Имея два вышеперечисленных факта и одно правило, он придет к такому выводу: «Максим любит бокс». А если будет задан следующий запрос: «Кто любит бокс?», то он найдет все решения этой задачи. При этом Prolog автоматически управляет решением задачи, стремясь в период выполнения программы найти все вероятные наборы значений, удовлетворяющие запросу. Необходимо заметить, что поиск выполняется по всей базе данных, которая ранее была введена в систему.

При объединении набора фактов с целью совместного их использования при решении некоторой конкретной задачи используется термин «база данных». Термин «база знаний» часто используется в том случае, если кроме фактов в описании предметной области содержатся еще и правила.

В языке Prolog применяется метод поиска с возвратом, позволяющий ему в случае нахождения одного решения пересмотреть все сделанные предположения снова для того, чтобы выяснить: приводит новое значение переменной к еще одному решению или нет. Процесс освоения синтаксиса языка Prolog проще, чем освоение синтаксиса традиционных языков программирования [28].

Программа на языке Prolog состоит из большого количества предложений. Каждое предложение может быть одно из 3-х типов:

1) Факт — это фраза без условий, утверждение о том, что соблюдается некоторое отношение. Он записывается как имя предиката, за которым следует заключенный в скобки список аргументов. Например:

loves('Максим', 'Анна').

loves('Федор', 'бокс').

2) Правило — это факт, истинность которого зависит от истинности других фактов. Состоит из тела и головы, разделенных знаком :- , который читается как «если» и соответствует импликации. Голова правила является предикатом, а тело правила — последовательностью предикатов, разделенных запятыми. Правило должно заканчиваться точкой, а запятая в теле правила означает конъюнкцию (логическое И). Например:

loves('Максим',X):- loves('Федор',X).

3) Запрос — это последовательность предикатов, которые разделены запятыми или точкой с запятой и завершаются точкой. На естественном языке запятая соответствует союзу «и», что на языке математической логики обозначает конъюнкцию. Точка с запятой соответствует союзу «или» и обозначает дизъюнкцию. Предикат запроса называется целью. Простые запросы, которые не содержат никаких переменных, допускают лишь два возможных ответа: «true» или «false». Если ответ «true», тогда говорят, что запрос завершился успехом, цель достигнута. Использование переменных в запросах позволяет задавать более сложные вопросы. Например:

?- loves('Максим','бокс').

true.

?- loves(Who,'Анна').

Who = 'Максим' .

?- loves(Man,'бокс').

Man = 'Федор' ;

Man = 'Максим'.

?- loves(X, 'Анна'),loves(X,'бокс').

X = 'Максим'.

Процедура представляет собой несколько правил, заголовки которых содержат одинаковые предикаты. Так, например, два правила реализуют процедуру нахождения наибольшего из двух чисел и использует одинаковый предикат $\text{maxi}/3$, вида $\text{maxi}(\text{число1}, \text{число2}, \text{максимальное_число})$:

$\text{maxi}(X, Y, X) :- X \geq Y.$

$\text{maxi}(X, Y, Y) :- X < Y.$

Принято считать, что между всеми правилами одной процедуры неявно присутствует соединительный союз «или», другими словами все правила процедуры дизъюнктивно связаны между собой [6].

Принятие стандарта языка Prolog послужило стимулом для создания многих независимых реализаций данного языка. Существуют как коммерческие, так и бесплатно предоставляемые широкому кругу пользователей реализации. В настоящее время существует около 26 реализаций языка Prolog. На рисунке 2 представлено сравнение основных характеристик реализаций Prolog.

Платформа		Особенности								Инструмент			Механизм
Наименование	Операционная система	Графика	Компилятор	Unicode	ООП	Управление ОС	Автон. исполнение	Интерфейс к С	Интерфейс к Java	Интерпретатор	Отладчик	Профайлер	Синтаксис Prolog
BProlog	U, W, M		+		+	+	+	+	+	+	+	+	ISO
JProlog	JVM, A	+		+		через Java	+	через Java	+	+	+		ISO
Ciao	U,W,M		+		+	+	+	+	+	+	+	+	ISO_R
GNU Prolog	U,W,M		+			+	+	+		+	+		ISO
JLog	JVM	+	+						+	+			ISO
JScriptLog	Браузер									+			ISO
jTrolog	JVM			+					+	+	+		ISO
LPA-Prolog	W	+	+	+	+	+	+	+	+	+	+	+	EP_R
Open Prolog	Mac OS										+		
Poplog	L, U, W	+	+			+	+	+		+	+		EP_R
SICStus Prolog	U,L,W,M	+	+	+	+	+	+	+	+	+	+	+	ISO
SWI-Prolog	U,L,W,M	+	+	+		+	+	+	+	+	+	+	ISO, EP
tuProlog	JVM, A	+		+				+	+	+	+		ISO
Visual Prolog	W	+	+	+	+	+	+	+			+	+	
XSB Prolog	L,W,S,M		+			+	+	+	+	+	+	+	ISO
PEA-Prolog	L,W,S,M		+	+		+	+	+	+	+	+		EP, ISO

Условные обозначения: U – ОС Unix, W – ОС Windows, M – Mac OS X, L – ОС Linux, S – ОС Solaris, A – ОС Android, JVM – Java Virtual Machine, ISO – ISO Prolog, ISO_R – ISO Prolog с расширениями, EP – Edinburgh Prolog, EP_R – Edinburgh Prolog с расширениями

Рисунок 2 – Сравнение основных характеристик реализаций языка Prolog

Следует обратить внимание на реализацию SWI-Prolog. Данная реализация, ориентированная в первую очередь на образование и научные исследования, является стабильной и бесплатной. Этот продукт достаточно популярен в настоящее время.

Разрабатывается SWI-Prolog в университете Амстердама с 1987 года. Jan Wielemaker – его создатель и основной разработчик [28]. Почти как все реализации, SWI-Prolog в основном следует известной «эдинбургской версии», но при этом включает в себя частично реализованные особенности

ISO Prolog. К наиболее важным особенностям последних версий SWI-Prolog необходимо отнести:

- Богатые библиотеки предикатов;
- Возможность написания логических модулей для веб-приложений;
- Поддержку GUI для XPCЕ и PreEmacs;
- Встроенную командную строку;
- Работу с файлами.

SWI-Prolog включает в себя развитые средства разработчика, включая интегрированную среду разработки (англ. Integrated Development Environment - IDE) с графическим отладчиком и профилировщиком, и обширную документацию. Кроме базовых функций языка, платформа реализует также ODBC, многопоточность, GUI, unit-тестирование, интерфейс к языку программирования Java и так далее. Следует заметить, что SWI-Prolog имеет встроенный собственный веб-сервер и работает на Unix, Macintosh, Windows и Linux платформах [28].

2.2 Возможности SWI-Prolog

SWI-Prolog является свободно-распространяемой реализацией (диалектом) языка программирования Prolog. SWI-Prolog формируется с 1987 года, сегодня он широко используется в исследовательских и образовательных целях, а также в коммерческих предложениях. Создателем и главным разработчиком SWI-Prolog является Ян Вьелемакер (Jan Wielemaker) [4]. Наименование SWI происходит от Sociaal-Wetenschappelijke Informatica, что в переводе с голландского «социально-научная информатика», первоначального названия группы в Амстердамском университете, где работает Вьелемакер [21].

SWI-Prolog дает возможность создавать приложения любой направленности, включая Web-приложения и параллельные вычисления,

однако главным направлением использования является разработка экспертных систем, программ обработки естественного языка, обучающих программ, интеллектуальных игр и т.п.

Программирование в SWI-Prolog возможно в разных вариантах:

- с помощью стандартной offline-среды программирования, которая продемонстрирована на рисунке 3;
- с помощью offline-среды программирования SWI-Prolog-Editor, продемонстрированной на рисунке 4;
- с помощью стандартной online-среды программирования SWI-Prolog, изображенной на рисунке 5 [21].

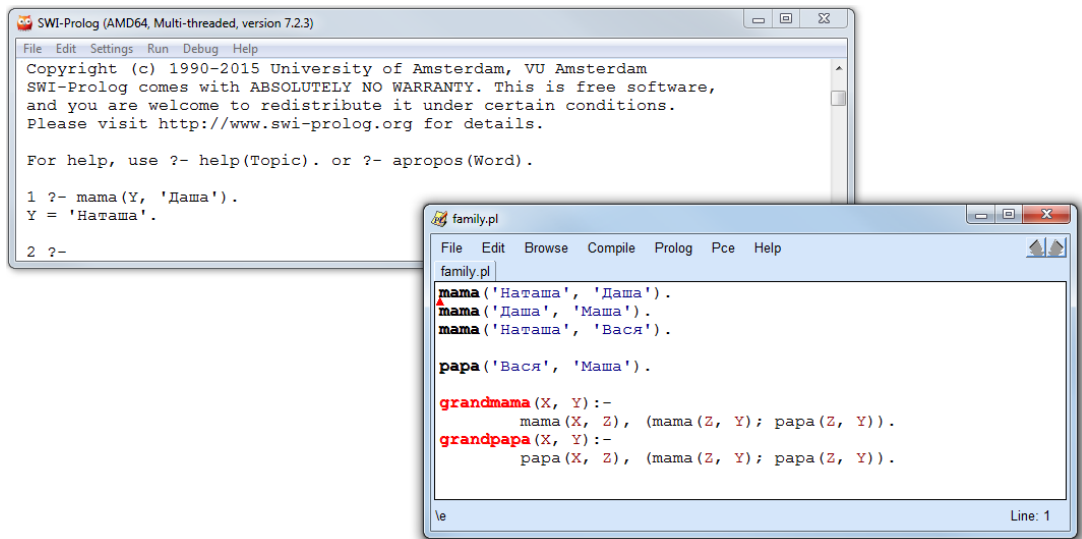


Рисунок 3 – Offline-среда программирования SWI-Prolog

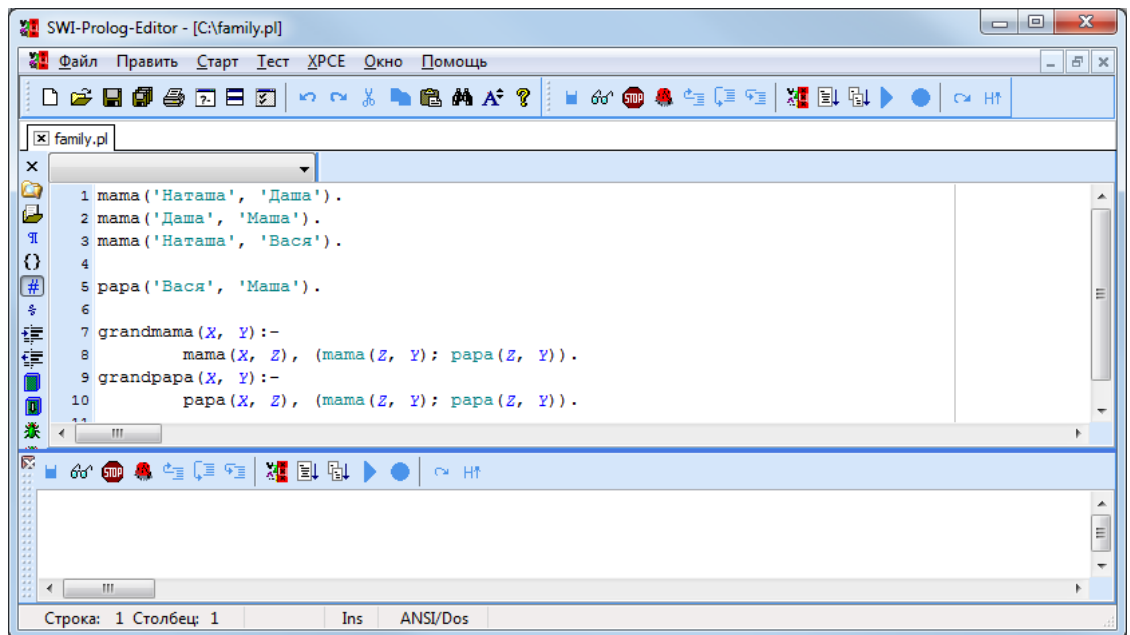


Рисунок 4 – Offline-среды программирования SWI-Prolog-Editor

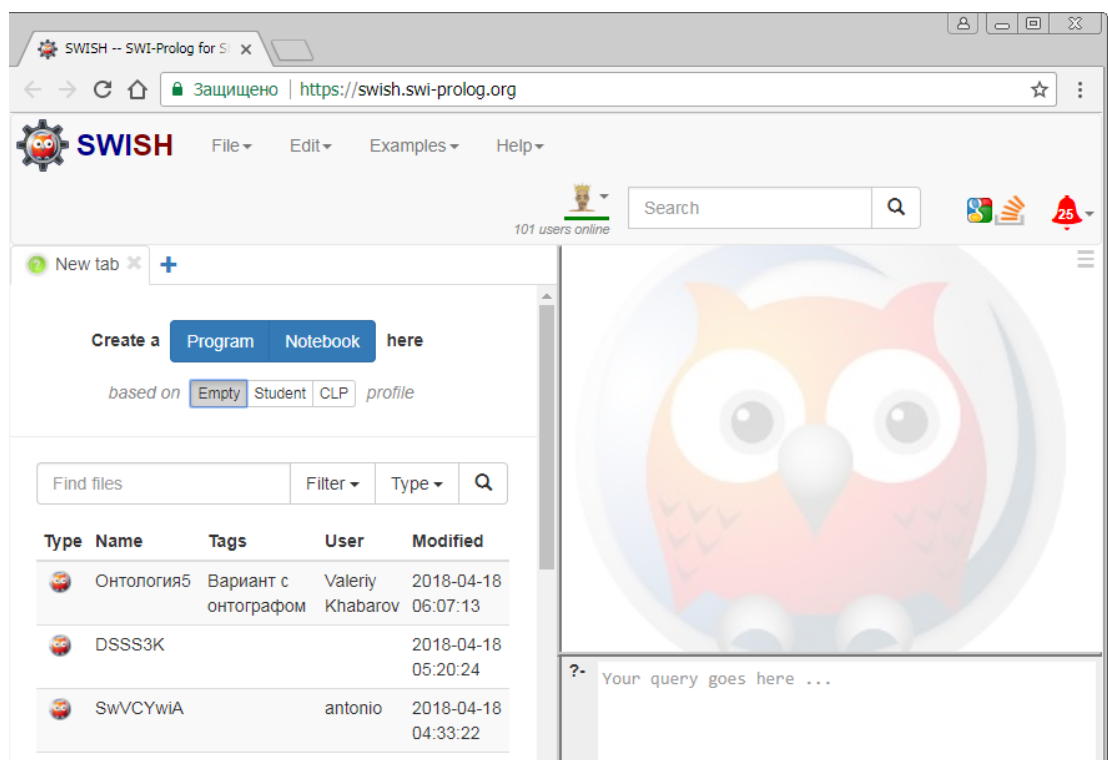


Рисунок 5 – Стандартная online-среда программирования SWI-Prolog

РсеЕмас — это встроенный в среду SWI-Prolog текстовый редактор. Он поддерживает автоматические отступы, подсветку синтаксиса, полную проверку синтаксиса путем вызова парсера SWI-Prolog, предупреждения о

singleton переменных и поиск определений предикатов на основании исходного кода в Prolog-базе данных [3].

Система SWI-Prolog включает средства объектно-ориентированной библиотеки XPCE (X-windows Process Control Engineering), которая предназначена для создания графического пользовательского интерфейса [1]. Для SWI-Prolog ряд средств библиотеки оформлен в виде предикатов, поэтому вызов некоторых методов классов, представленных в библиотеке, имеет вид пролог-целей [3].

Рассмотрим часть базовых возможностей этой библиотеки, которые позволяют создать диалоговое окно со стандартными элементами графического интерфейса: кнопки, области для ввода/вывода текста и чисел, области для вывода графических фигур [7].

Классы библиотеки XPCE:

- `dialog` – диалоговое окно;
- `button` – кнопка;
- `int_item` – поле для ввода/вывода чисел;
- `text_item` – поле для ввода/вывода строки;
- `text` – класс текст, предназначенный для вывода текста;
- `size(width, height)` – размер объекта(ширина, высота);
- `message` – класс предназначенный для создания и отправки сообщения объекту;
- `picture` – область для вывода графических объектов;
- `point(X,Y)` – точка с координатами X и Y;
- `box(width, height)` – прямоугольник заданной ширины;
- `circle(diameter)` – окружность заданного диаметра;
- `line(start_x, start_y, end_x, end_y, arrows:={none,first,second,both})` – отрезок линии с началом и концом в заданных точках, `arrows` – наличие стрелки на концах линии.

- `dialog_group` – создание группы объектов диалогового окна для их последующего совместного вывода в диалоговое окно [7].

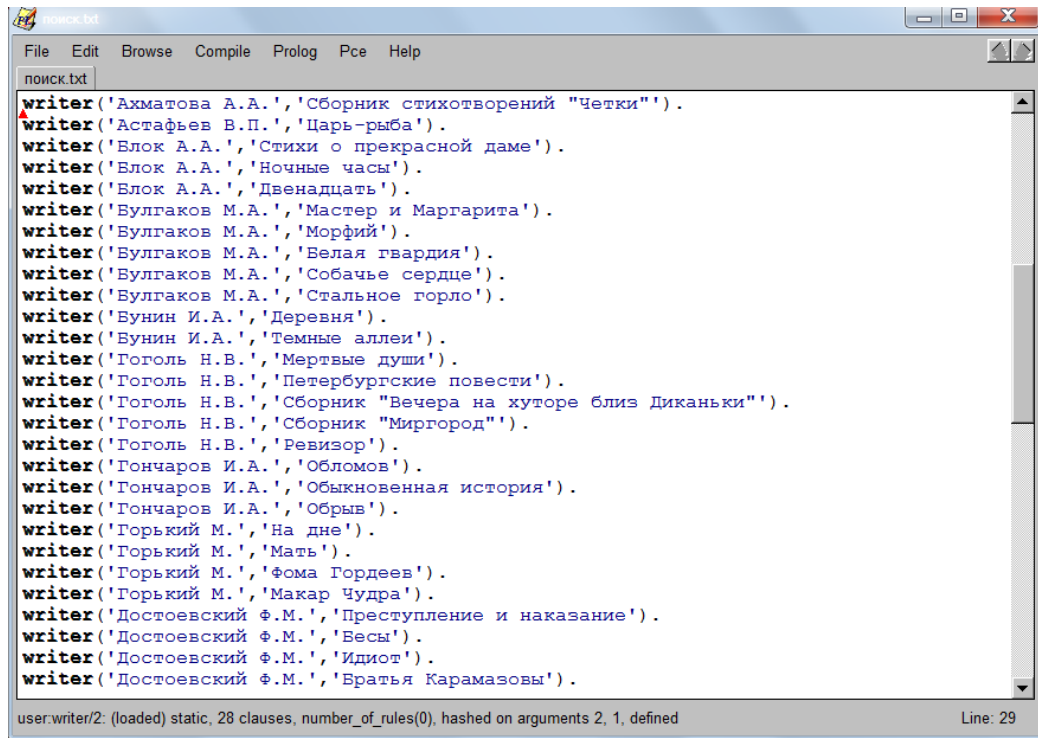
SWI-Prolog позволяет создавать экспертные системы, решающие ряд задач, а так же графический интерфейс созданных систем.

2.3 Создание графической оболочки

При создании экспертной системы важно создать простой и ясный графический интерфейс, который позволит пользователю работать с той или иной ЭС. На основании беседы с библиотекарем и изучения основных документов контролирующей деятельность школьной библиотеки и её обеспечение были сформулированы требования к системе.

Экспертная система для анализа книгообеспеченности школьной библиотеки будет выполнять следующие функции: поиск книг и анализ литературы. Функция поиска книги будет выдавать информацию о том, где в библиотеке находится необходимая книга, а также отображает все книги данного автора. Функция анализа литературы производит анализ книгообеспеченности школьной библиотеки: достаточно ли книг, словарей и других учебных материалов на определенное количество обучающихся, позволит рассчитать процент книгообеспеченности, а также вывести информацию о том, какой процент расходов на учебные материалы покрывает школьный бюджет. Также данная ЭС позволит увидеть недостающее количество учебных материалов на стандартный учебный класс из 20 человек.

Для реализации первой функции необходимо создать базу знаний, в которой содержится информация об авторе, названии книги или произведения, а также её нахождение в библиотеке (полка, стеллаж) и количестве экземпляров. С помощью предиката `write (X, Y)`, где `X` – автор, `Y` – название, заносим в базу знаний книги находящиеся в библиотеке. На рисунке 6 представлен фрагмент базы знаний, записанный с помощью этого предиката.



```
File Edit Browse Compile Prolog Pce Help
поиск.txt
writer('Ахматова А.А.', 'Сборник стихотворений "Четки"').
writer('Астафьев В.П.', 'Царь-рыба').
writer('Блок А.А.', 'Стихи о прекрасной даме').
writer('Блок А.А.', 'Ночные часы').
writer('Блок А.А.', 'Двенадцать').
writer('Булгаков М.А.', 'Мастер и Маргарита').
writer('Булгаков М.А.', 'Морфий').
writer('Булгаков М.А.', 'Белая гвардия').
writer('Булгаков М.А.', 'Собачье сердце').
writer('Булгаков М.А.', 'Стальное горло').
writer('Бунин И.А.', 'Деревня').
writer('Бунин И.А.', 'Темные аллеи').
writer('Гоголь Н.В.', 'Мертвые души').
writer('Гоголь Н.В.', 'Петербургские повести').
writer('Гоголь Н.В.', 'Сборник "Вечера на хуторе близ Диканьки"').
writer('Гоголь Н.В.', 'Сборник "Миргород"').
writer('Гоголь Н.В.', 'Ревизор').
writer('Гончаров И.А.', 'Обломов').
writer('Гончаров И.А.', 'Обыкновенная история').
writer('Гончаров И.А.', 'Обрыв').
writer('Горький М.', 'На дне').
writer('Горький М.', 'Мать').
writer('Горький М.', 'Фома Гордеев').
writer('Горький М.', 'Макар Чудра').
writer('Достоевский ф.М.', 'Преступление и наказание').
writer('Достоевский ф.М.', 'Бесы').
writer('Достоевский ф.М.', 'Идиот').
writer('Достоевский ф.М.', 'Братья Карамазовы').

user:writer/2: (loaded) static, 28 clauses, number_of_rules(0), hashed on arguments 2, 1, defined
Line: 29
```

Рисунок 6 – Фрагмент записи авторов в базе знаний

С помощью предиката book (X, Y, Z, R), где X – название книги или произведения, Y – стеллаж и Z – полка расположения X, R – количество экземпляров, также заносим данные в базу знаний, фрагмент которой продемонстрирован на рисунке 7.


```

book('Царь-рыба','1','1','5').
book('Сборник стихотворений "Четки"', '1','1','4').
book('Стихи о прекрасной даме','1','2','5').
book('Ночные часы','1','3','6').
book('Двенадцать','1','3','4').
book('Мастер и Маргарита','1','4','6').
book('Морфий','1','5','2').
book('Белая гвардия','1','5','7').
book('Собачье сердце','1','6','4').
book('Стальное горло','1','6','4').
book('Деревня','1','6','3').
book('Темные аллеи','1','8','10').
book('Мертвые души','2','1','11').
book('Петербургские повести','2','2','3').
book('Сборник "Вечера на хуторе близ Диканьки"', '2','2','6').
book('Сборник "Миргород"', '2','3','5').
book('Ревизор','2','3','6').
book('Обломов','2','4','4').
book('Обыкновенная история','2','4','5').
book('Обрыв','2','4','3').
book('На дне','2','5','10').
book('Мать','2','6','3').
book('Фома Гордеев','2','6','8').
book('Макар Чудра','2','8','4').
book('Преступление и наказание','3','2','5').
book('Весы','3','2','3').
book('Идиот','3','2','2').
book('Братья Карамазовы','3','3','7').

```

Рисунок 7 – Фрагмент записи расположения книг в базе знаний

Для отображения данной информации в графическом виде необходимо создать диалоговое окно. Для этого записываем: `new(Dialog, dialog('СБ-1'))`. Далее в диалоговом окне создаем меню, в котором можно выбрать автора и произведение: `send_list(Dialog,append, [new(N1,menu('Выберите автора', cycle)), new(N2, menu('Выберите произведение/книгу',cycle))]`.

Также заносим данные для отображения в созданные меню. Фрагмент кода программы представлен на рисунке 8.

```

send_list(N1, append, ['_', 'Астафьев В.П.', 'Ахматова А.А.', 'Блок А.А.', 'Булгаков М.А.', 'Бунин И.А.', 'Гоголь Н.В.', 'Гончаров И.А.', 'Горький М.', 'Достоевский ф.М.']),
▲ send_list(N2, append, ['_', 'Сборник стихотворений "Четки"', 'Стихи о прекрасной даме', 'Ночные часы', 'Двенадцать', 'Мастер и Маргарита', 'Морфий', 'Белая гвардия', 'Собачье сердце', 'Стальное горло', 'Деревня', 'Темные аллеи', 'Мертвые души', 'Петербургские повести', 'Сборник "Вечера на хуторе близ Диканьки"', 'Сборник "Миргород"', 'Ревизор', 'Обломов', 'Обыкновенная история', 'Обрыв', 'На дне', 'Мать', 'фома Гордеев', 'Макар Чудра', 'Преступление и наказание', 'Весы', 'Идиот', 'Братья Карамазовы']]).

```

Рисунок 8 – Фрагмент кода для отображению в меню

Далее создаем кнопки «Выход» и «Поиск», которые позволяют выйти из системы или вывести необходимую информацию соответственно. Также для кнопок описываются функции, которые они выполняют:

```
button (выход, message(Dialog, destroy)),  
button (поиск, and (message(@prolog,  
output,MyQuest,  
N1?selection,  
N2?selection),  
message(MyQuest,destroy))).
```

Для более приятного интерфейса вставляем картинку: `bitmap('12.jpg')`. Затем создаем окно для вывода информации (`new (MyQuest,list_browser)`) и задаем размеры и параметры шрифта:

```
send(MyQuest,alignment, center),  
send(MyQuest,font,font(times,bold,15)),  
send(MyQuest,size,size(30,6)),  
send(Dialog,open(point(50,9))).
```

Далее указываем действия, которые должны быть выполнены для вывода необходимой информации. Фрагмент кода с данной информацией представлен на рисунке 9.

В результате получаем диалоговое окно, в котором можно выбрать автора, произведение или книгу и вывести информацию о том, где она расположена в библиотеке. Данное окно продемонстрировано на рисунке 10.

```
поиск.bt
File Edit Browse Compile Prolog Pce Help
поиск.txt
output(FrmList,N1,N2):-
send(FrmList,clear),
(
(writer(N1,N2), book(N2,M,N,Y), send_list(FrmList,append,'Стеллаж'),
send_list(FrmList,append,M), send_list(FrmList,append,'Полка'),
send_list(FrmList,append,N),
send_list(FrmList,append,'Количество экземпляров'), send_list(FrmList,append,Y));

(send_list(FrmList,append,'Произведения:'), writer(N1,X),
send_list(FrmList,append,X)),

fail.

user:output/3: (loaded) static, 1 clause, number_of_rules(1), defined
Line: 84
```

Рисунок 9 – Фрагмент кода программы

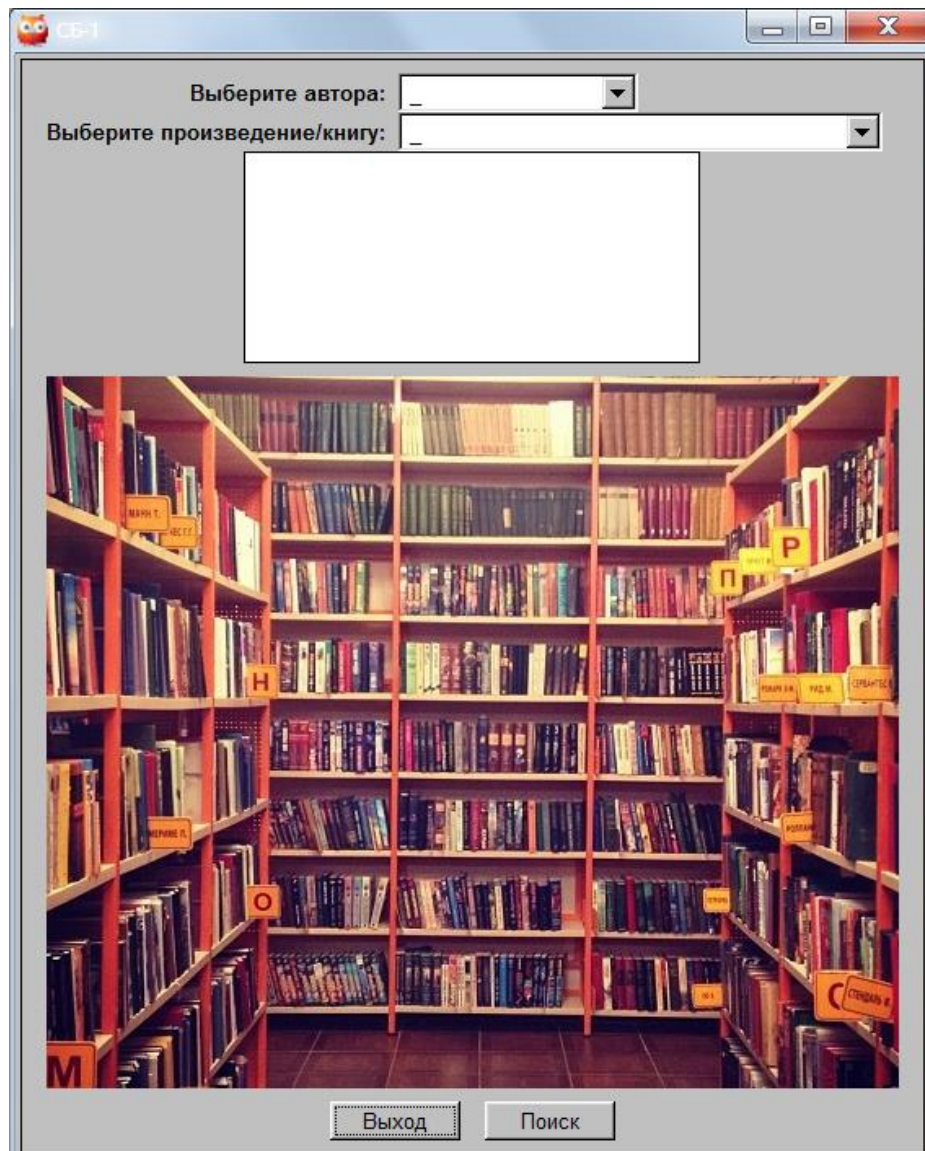


Рисунок 10 – Графический интерфейс поиска книг

Аналогичным образом создаем объекты диалогового окна для анализа литературы. В нем необходимо создать область для ввода количества учеников, область выбора класса, области для вывода информации о книгообеспеченности библиотеки и процент расходов на учебные материалы, покрываемые бюджетом школы, а также кнопки «Анализ», «Поиск», «Посчитать», «Описание» и «Закрыть». При нажатии на кнопку «Поиск» выводится информация о количестве учебных материалов на стандартный класс из 20 человек. Кнопка «Анализ» выводит коэффициент книгообеспеченности учебными материалами. При нажатии на кнопку «Посчитать» выводится информация о расходах на учебные материалы для обучающихся и процент, который покрывает школьный бюджет. Кнопка «Описание» выводит правило, по которому система производила поиск и расчет информации.

В базе знаний записываем информацию об учебных материалах и их количестве для каждого класса с помощью предиката $\text{book}(X, Y, Z, R)$, где X – класс, Y – количество учебных материалов, Z – наименование учебного материала, R – наименование предмета, к которому относится учебный материал. Фрагмент базы знаний, который содержит информацию, записанную при помощи этого предиката представлен на рисунке 11. Также вносится информация о цене учебных материалов разного типа, которая расположена на рисунке 12.

```
analizlit - копия.pl
File Edit Browse Compile Prolog Pce Help
analizlit - копия.pl
book(2, 19, 'учебников', 'История').
book(2, 22, 'учебников', 'Обществознание').
book(2, 20, 'учебников', 'Литература').

book(3, 20, 'учебников', 'Английский язык').
book(3, 19, 'рабочих тетрадей', 'Английский язык').
book(3, 18, 'учебников', 'Русский язык').
book(3, 15, 'рабочих тетрадей', 'Русский язык').
book(3, 22, 'учебников', 'Математика').
book(3, 20, 'учебников', 'Окружающий мир').
book(3, 27, 'рабочих тетрадей', 'Окружающий мир').
book(3, 19, 'учебников', 'История').
book(3, 22, 'учебников', 'Обществознание').
book(3, 20, 'учебников', 'Литература').

book(4, 20, 'учебников', 'Английский язык').
book(4, 15, 'рабочих тетрадей', 'Английский язык').
book(4, 20, 'учебников', 'Русский язык').
book(4, 15, 'рабочих тетрадей', 'Русский язык').
book(4, 22, 'учебников', 'Математика').
book(4, 20, 'учебников', 'Окружающий мир').
book(4, 27, 'рабочих тетрадей', 'Окружающий мир').
book(4, 19, 'учебников', 'История').
book(4, 22, 'учебников', 'Обществознание').
book(4, 20, 'учебников', 'Литература').

book(5, 20, 'учебников', 'Английский язык').
book(5, 15, 'рабочих тетрадей', 'Английский язык').
book(5, 22, 'словарей', 'Английский язык').
book(5, 20, 'учебников', 'Русский язык').
book(5, 20, 'дидактических материалов', 'Русский язык').
book(5, 15, 'рабочих тетрадей', 'Русский язык').
book(5, 14, 'ВПР', 'Русский язык').
```

Рисунок 11 – Фрагмент базы знаний анализа литературы

```
money('справочники', 70).
money('учебники', 160).
money('сборники заданий', 170).
money('дидактические пособия', 90).
money('орфографические словари', 170).
money('словари', 90).
money('атласы', 160).
money('рабочие тетради', 100).
money('карты', 73).
```

Рисунок 12 – Фрагмент базы знаний о цене учебных материалов

На рисунках 13-14 представлены фрагменты кода программы, а на рисунке 15 продемонстрирован графический интерфейс данной функции.

```

analizlit - копия.pl
File Edit Browse Compile Prolog Pce Help
analizlit - копия.pl
    button(посчитать, and(message(@prolog,
        результат3,
        MyList2,
        N1?selection),
        message(MyWin, destroy))),

    button(заккрыть, message(MyWin, destroy)),
    new(text('Примечания:
коэффициент>1 - все учащиеся обеспечены данным видом литературы
коэффициент<1 - данного вида литературы недостаточно'))
]),
    send(MyList, alignment, center),
    send(MyList2, alignment, center),
    send_list(D, append, [выберите_из_списка,
        1,
        2,
        3,
        4,
        5,
        6,
        7,
        8,
        9,
        10,
        11]),
    send(MyWin, default_button, поиск),
    send(AA, font, font(times, italic, 14)),
    send(MyList,font, font(times, italic, 14)),
    send(MyList2,font, font(times, italic, 14)),

    send(MyList, size, size(60, 10)),
    send(MyList2, size, size(60, 10)),
    send(MyWin, size, size(500, 610)),
    send(MyWin, open).

```

Рисунок 13 – Фрагмент кода программы графического интерфейса для анализа литературы

```

analizlit - копия.pl
File Edit Browse Compile Prolog Pce Help
analizlit - копия.pl
результат1(FrmList,W,X):-
send(FrmList, clear),
send(FrmList, append,'Коэффициент обеспеченности литературой учебного процесса:
'),
book(X,Y,Z,S),Z=(_),S=(_),X=(_),Y=(_),
send(FrmList, append,Z),
send(FrmList, append,'по предмету'),
send(FrmList, append,S),
send(FrmList, append,Y/W),
send(FrmList, append,'
'),
fail.

результат2(FrmList,X):-
send(FrmList, clear),
send(FrmList, append,'На стандартный класс из 20 человек не хватает:
'),
book(X,Y,Z,S),Y<15,X=(_),
send(FrmList, append,Z),
send(FrmList, append,'по предмету'),
send(FrmList, append,S),
send(FrmList, append,'
'),
fail.

результат3(FrmList,X):-
send(FrmList, clear),
send(FrmList, append,'Расходы на учебные материалы
для учащихся (в рублях):
'),

```

send/2: Built-in foreign predicate Line: 304

Рисунок 14 – Фрагмент программы для вывода результатов анализа

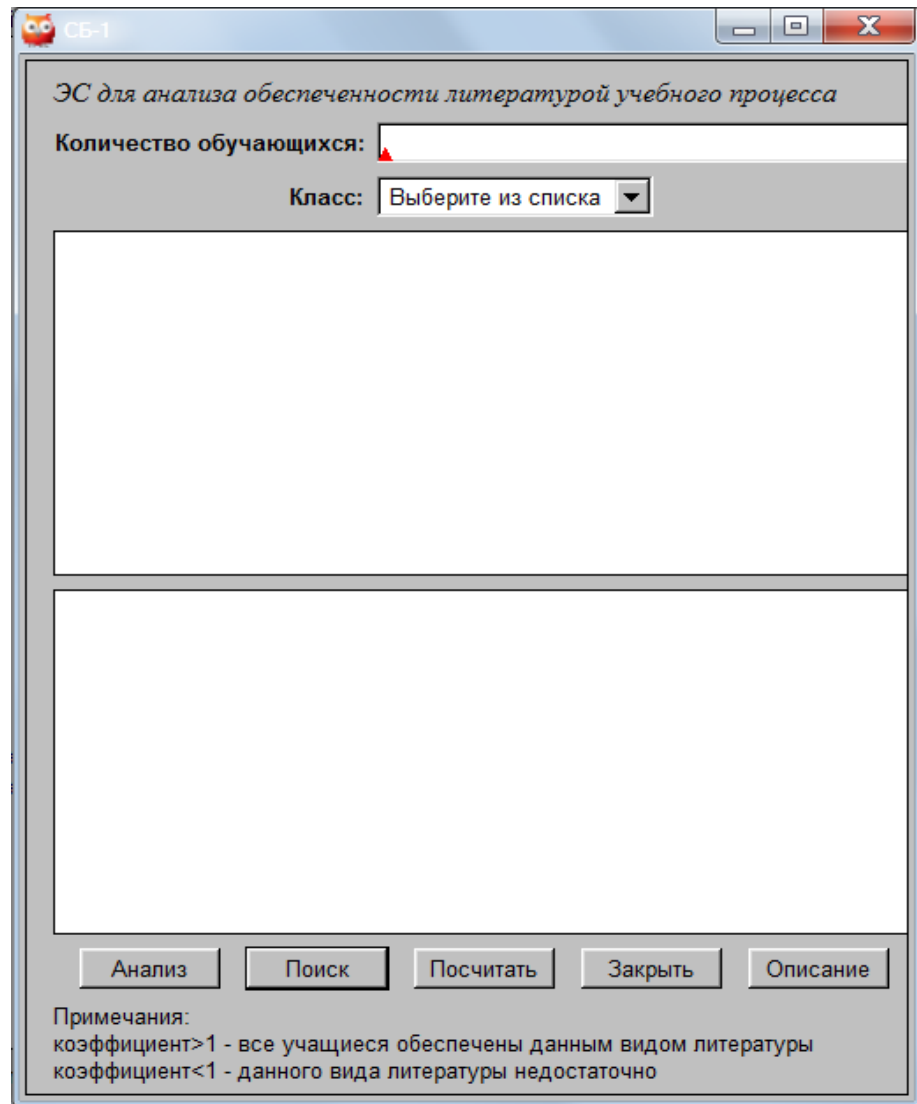


Рисунок 15 – Графический интерфейс анализа литературы

Для того чтобы библиотекаря было удобнее работать с системой и выбирать нужную функцию создадим главную страницу, на которой будет отображено название школы и две кнопки «Анализ литературы» и «Поиск книги». Это окно системы продемонстрировано на рисунке 16. Данные кнопки отрывают диалоговые окна, позволяющие выполнить ранее описанные действия.



Рисунок 16 – Главное окно системы

2.4 Демонстрация экспертной системы для анализа книгообеспеченности школьной библиотеки

В созданной системе для анализа книгообеспеченности школьной библиотеки содержатся книги, находящиеся в ней. В этот список входят основные произведения школьной литературы, а также учебники по всем предметам для всех классов. Далее будут продемонстрированы основные данные, хранящиеся в системе.

Произведения А.А. Ахматовой которые содержатся в системе, можно увидеть на рисунке 17. Основные произведения Булгакова М.А. из школьной программы расположены на рисунке 18.

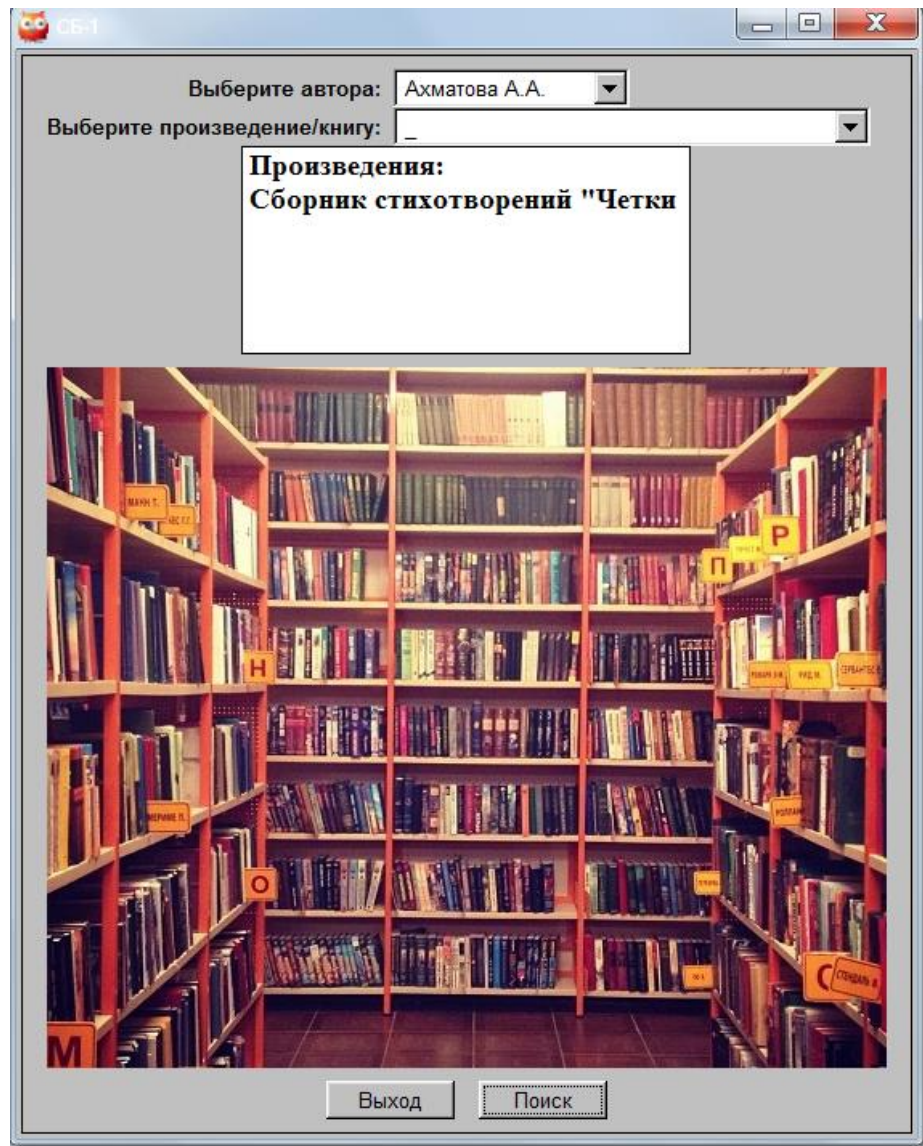


Рисунок 17 – Произведения Ахматовой в системе

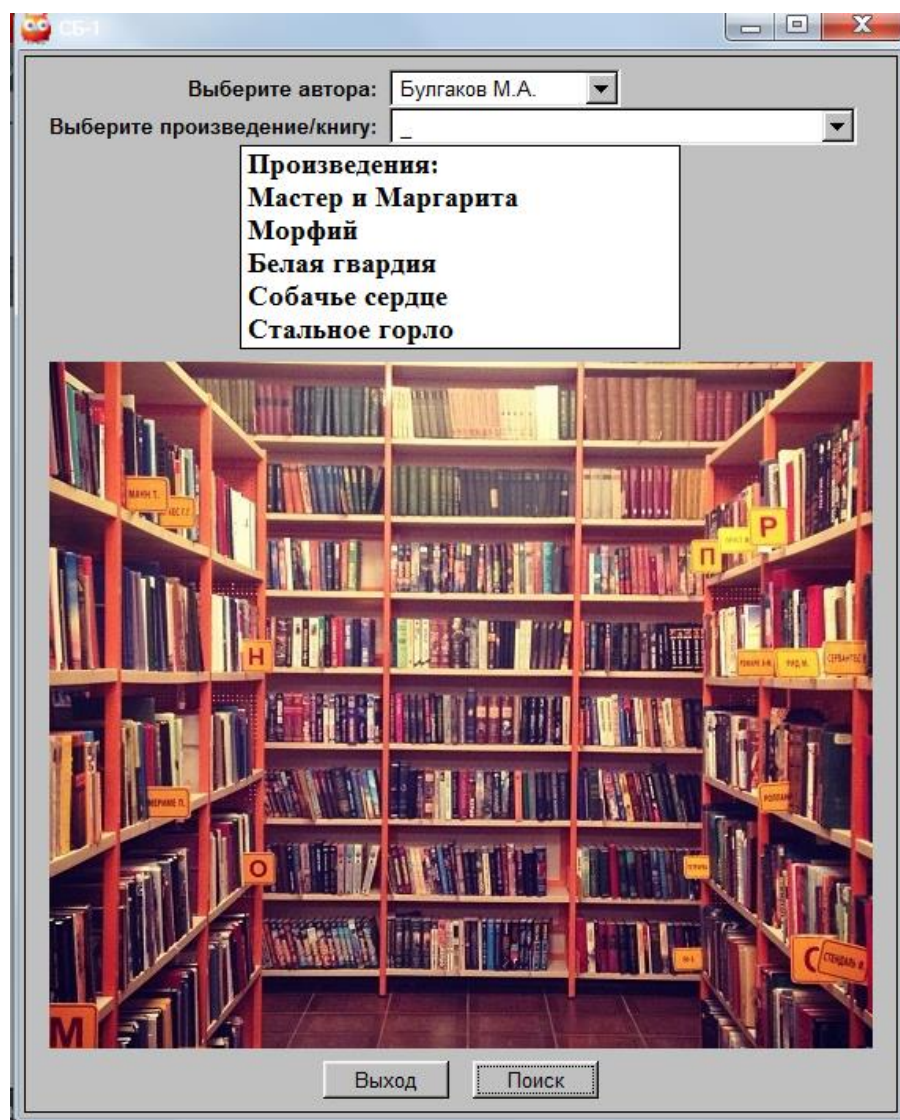


Рисунок 18 – Произведения Булгакова в системе

Важное место в школьной программе по литературе занимают произведения Бунина, Горького, Достоевского и Гоголя, которые расположены на рисунках 19 – 22 соответственно.

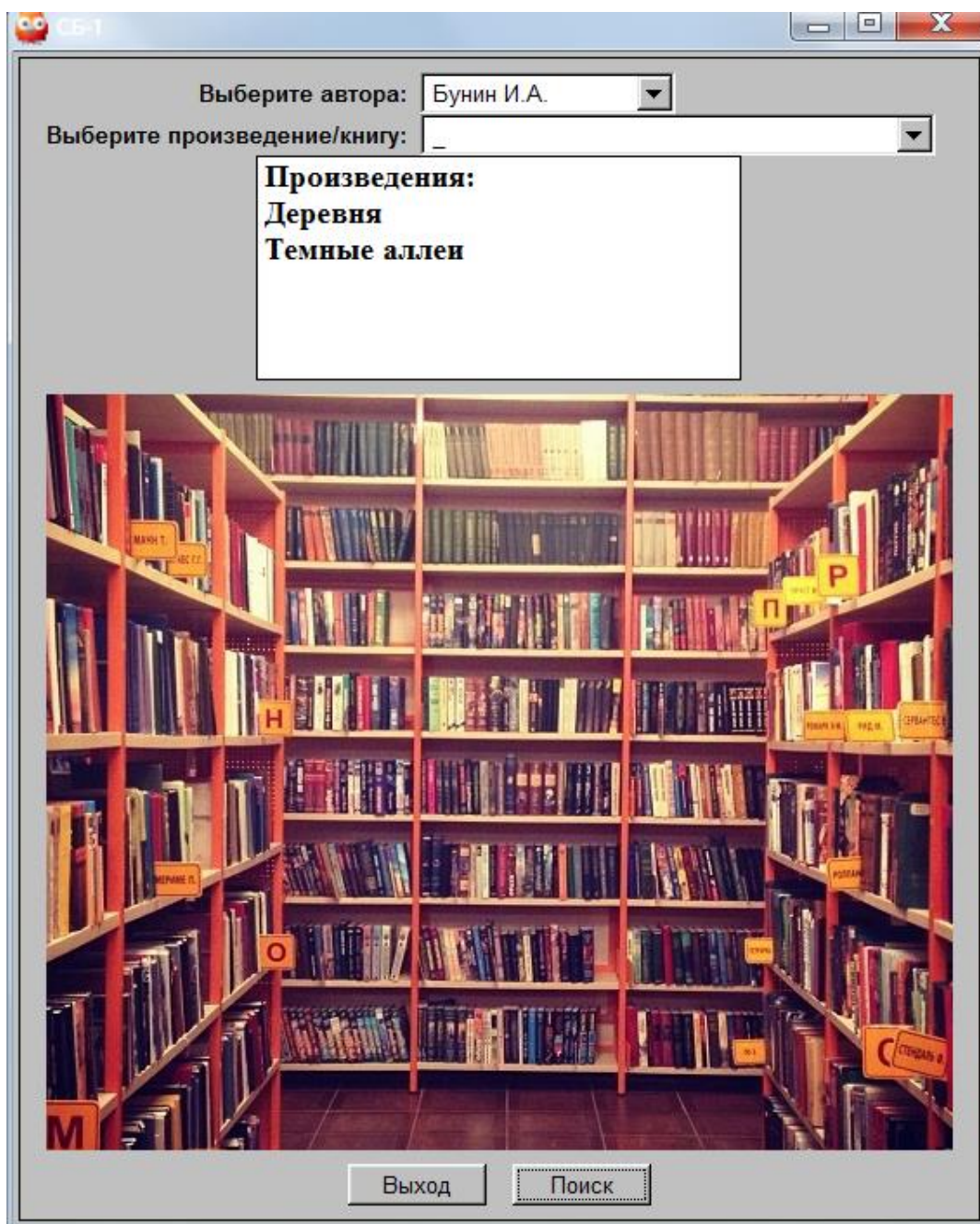


Рисунок 19 – Произведения Бунина в системе

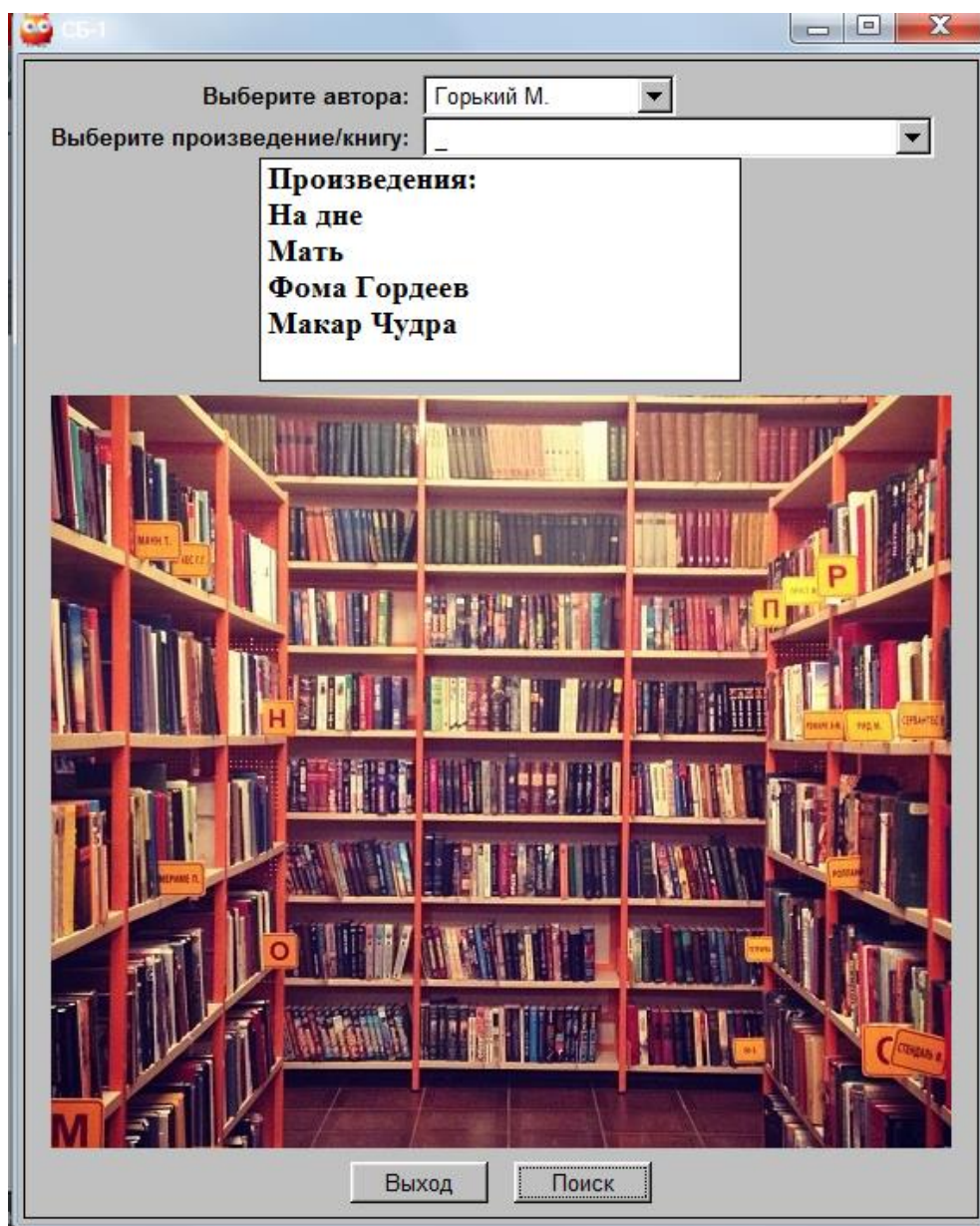


Рисунок 20 – Произведения Горького в системе

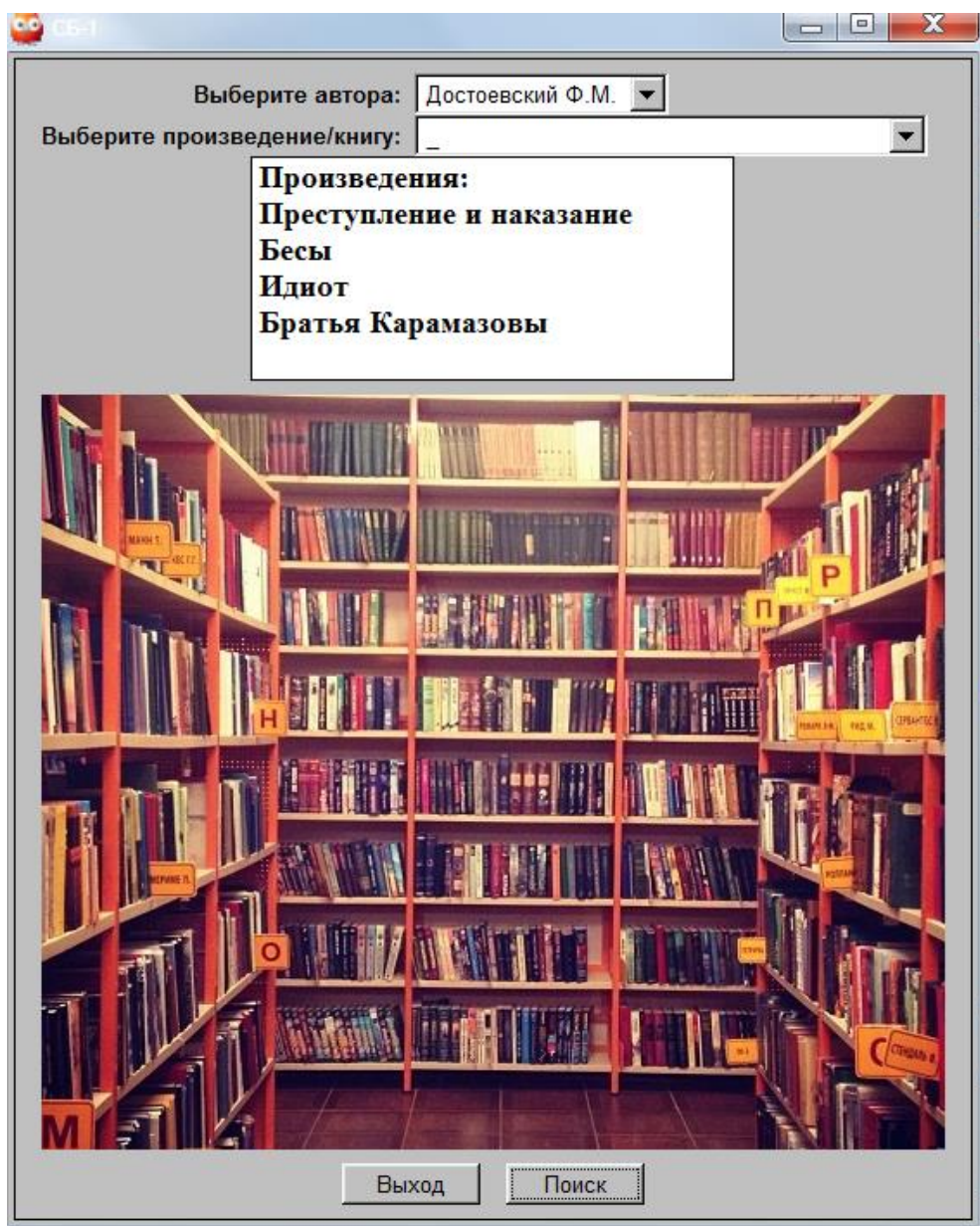


Рисунок 21 – Произведения Достоевского в системе

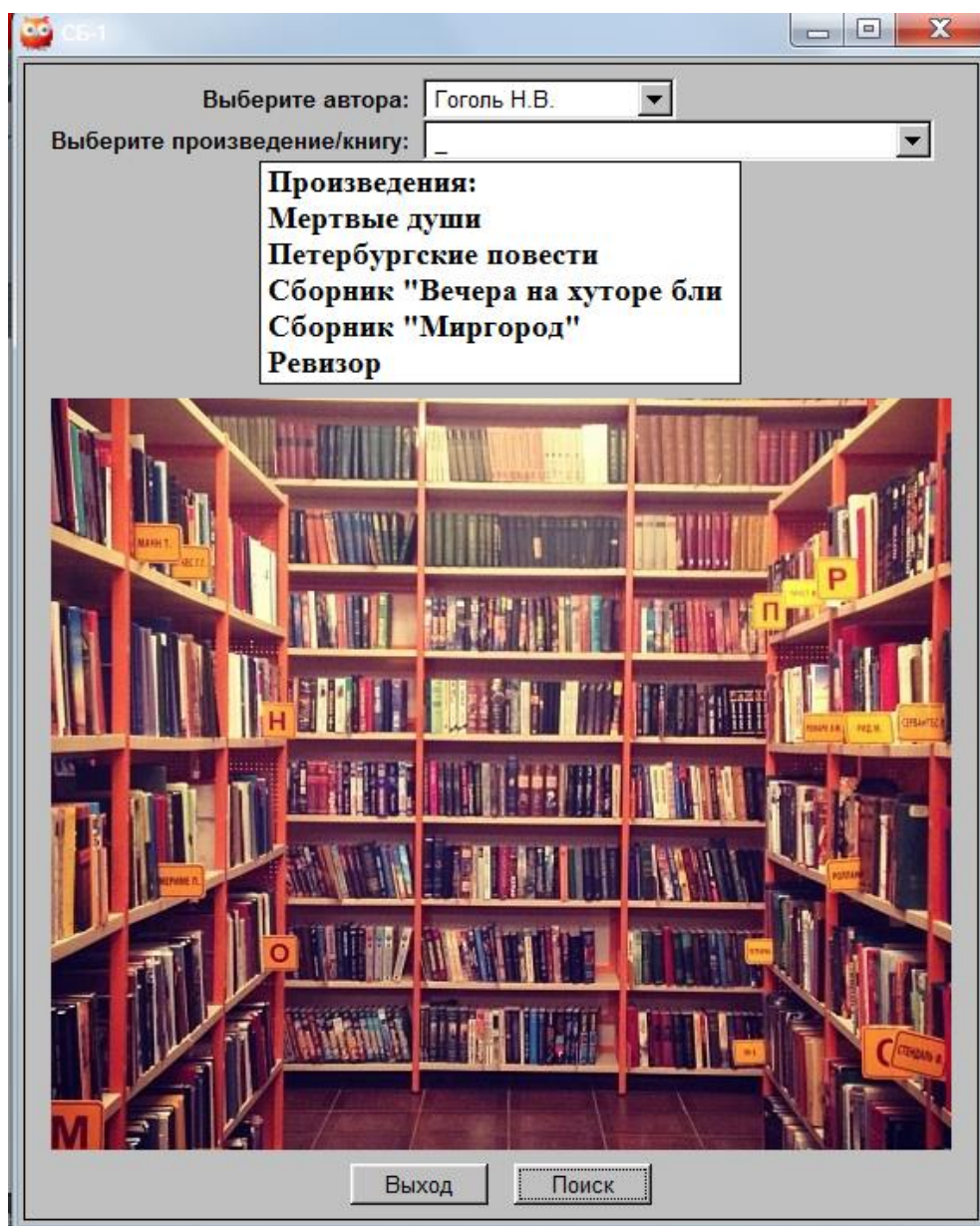


Рисунок 22 – Произведения Гоголя в системе

Также разработанная экспертная система для анализа книгообеспеченности школьной библиотеки производит расчет коэффициента книгообеспеченности учебного материала. Выбрав класс и введя количество учащихся, пользователь ЭС может получить данный анализ. На рисунке 23 выведен расчет коэффициента книгообеспеченности для 8 класса на 25 человек. На основании результата можно сделать вывод, что недостаточно учебных материалов по таким предметам как английский язык, геометрия, биология и химия.

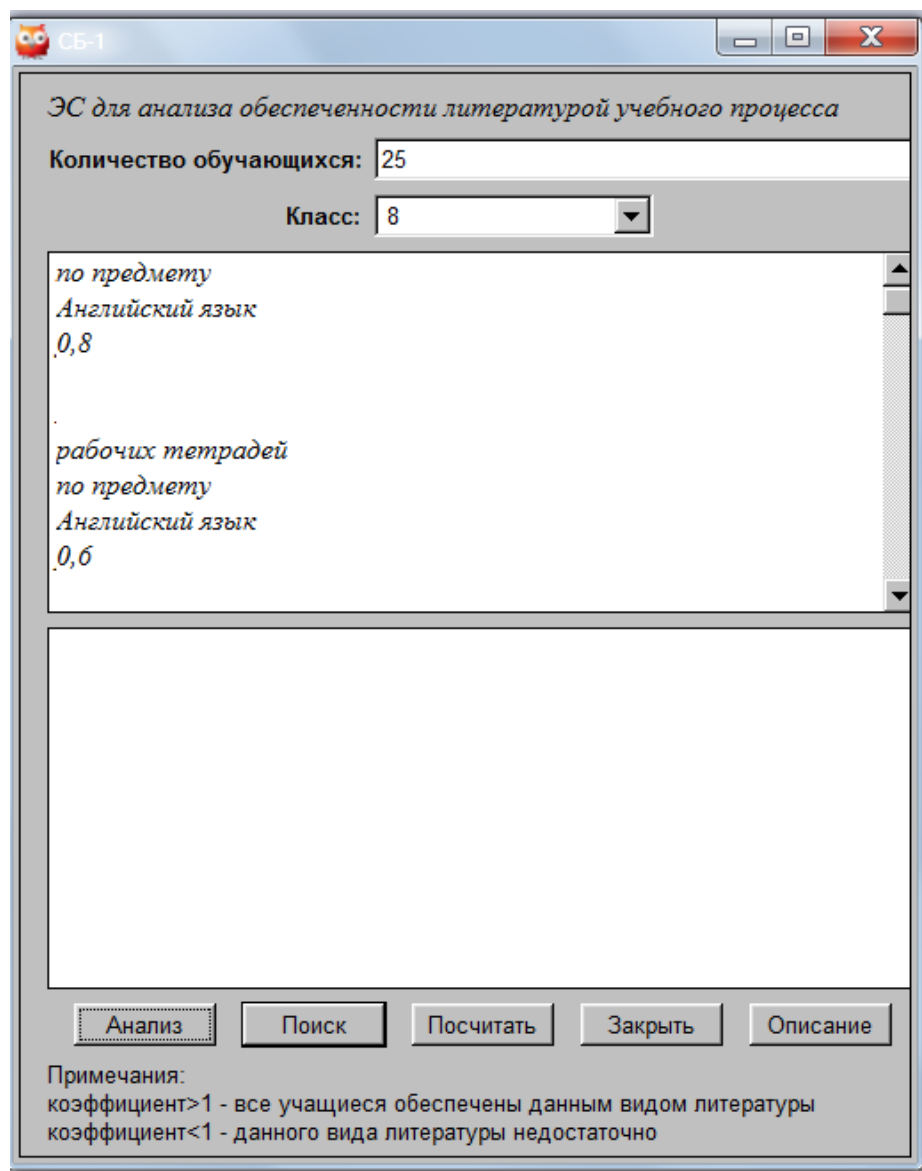


Рисунок 23 – Коэффициент книгообеспеченности для 8 класса

На рисунке 24 расположена информация о коэффициенте книгообеспеченности для 10 класса, состоящего из 20 человек. На основании расчета пользователь делает вывод, что учебников по всем предметам в данном случае достаточно, а рабочих тетрадей по английскому языку недостаточно.

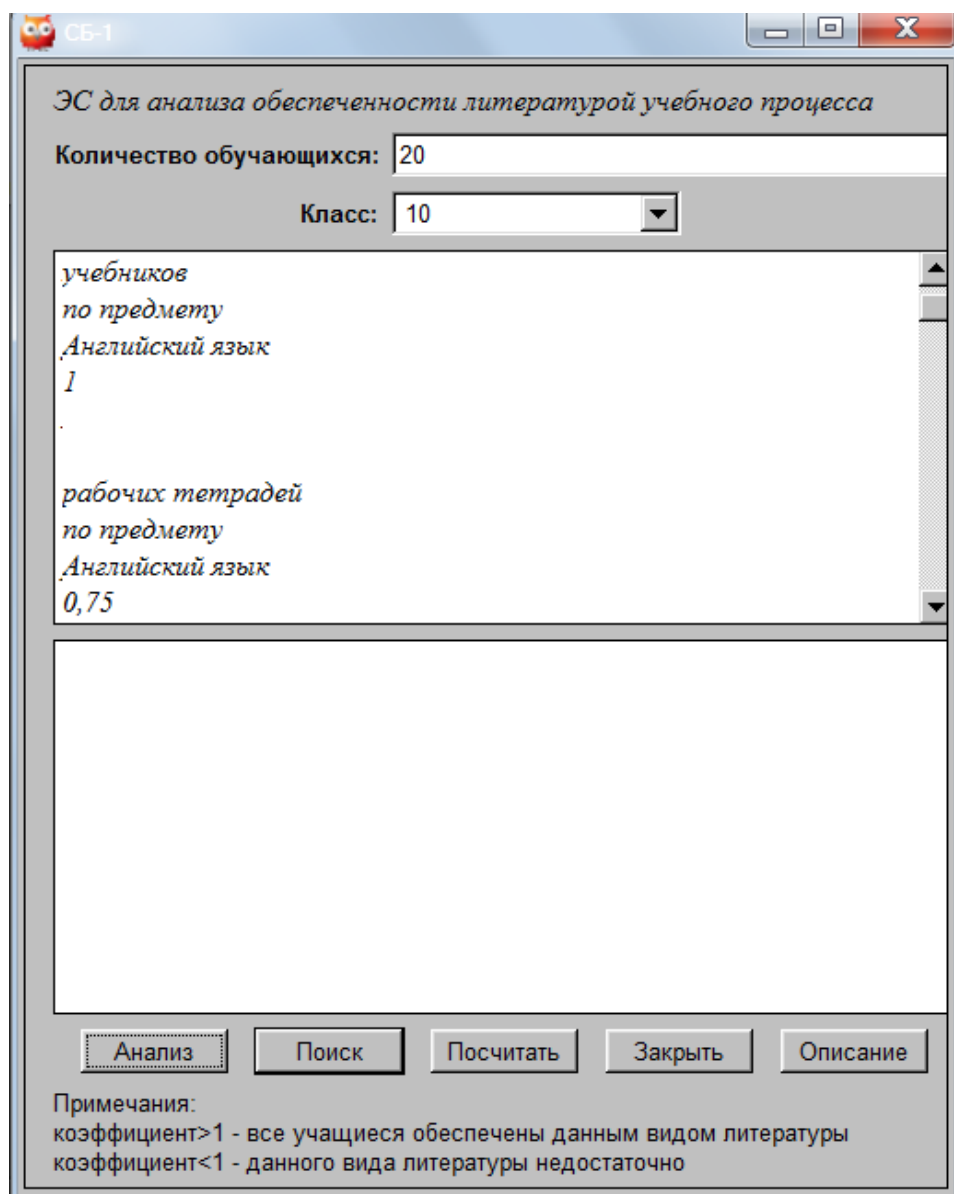


Рисунок 24 – Коэффициент книгообеспеченности для 10 класса

На рисунке 25 расположена информация о коэффициенте книгообеспеченности для 11 класса, состоящего из 27 человек. На основании расчета пользователь делает вывод, что учебников по многим предметам не достаточно. Коэффициент 0,592593 по геометрии говорит о низкой обеспеченности литературой учебного процесса по данному предмету.

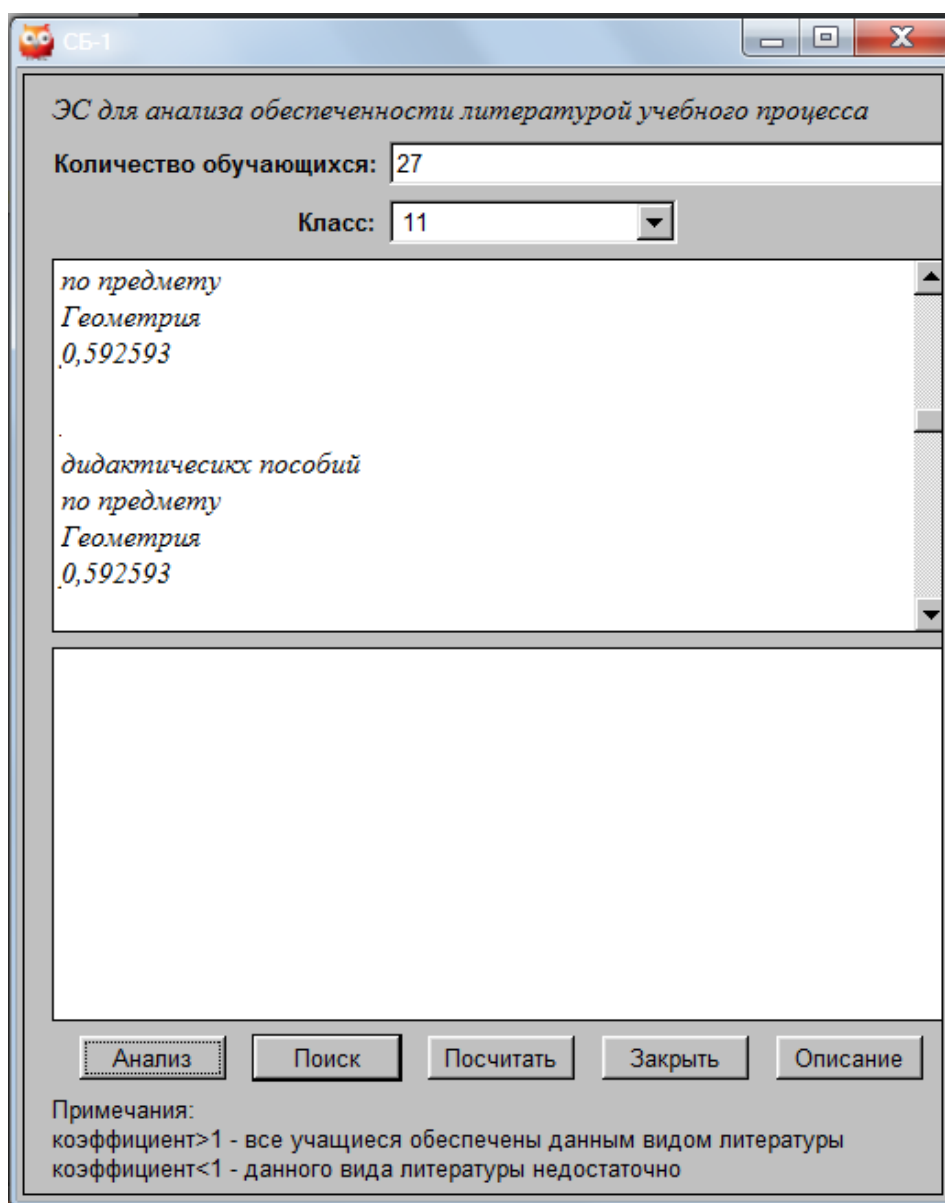


Рисунок 25 – Коэффициент книгообеспеченности для 11 класса

2.5 Инструкция пользователю

Данная инструкция предназначена для библиотекаря школы МГОУ «Шебекинская гимназия-интернат» и содержит описание методики работы в системе. ЭС позволяет найти требуемые книги, произвести анализ книгообеспеченности литературой учебного процесса на основании введенного количества обучающихся и выбранного класса, проверить какой процент расходов на учебные материалы покрывает школьный бюджет.

Целью реализации ЭС является создание информационной базы для управления информацией в рамках школьной библиотеки библиотекарем и школьниками, посещающими её. В рамках обозначенной цели разработанная ЭС позволяет решить следующие задачи:

- сокращение времени на анализ книгообеспеченности;
- сокращение времени на поиск необходимой книги;
- сокращение времени на консультирование школьников о наличии литературы в книгохранилище.

При открытии ЭС перед пользователем главная страница системы, на которой расположены кнопки «Поиск книги» и «Анализ литературы». Эта страница продемонстрирована на рисунке 26.



Рисунок 26 – Главная страница ЭС

Чтобы найти книгу пользователь нажимает на «Поиск книги», отрывается соответствующее окно, продемонстрированное на рисунке 27. Затем можно увидеть все книги какого-либо автора, имеющиеся в библиотеке. Для этого в поле «Выберите автора» пользователь из списка выбирает автора и нажимает кнопку «Поиск». Пример этого действия можно увидеть на рисунке 28. А результат изображен на рисунке 29: выведен список книг выбранного автора.

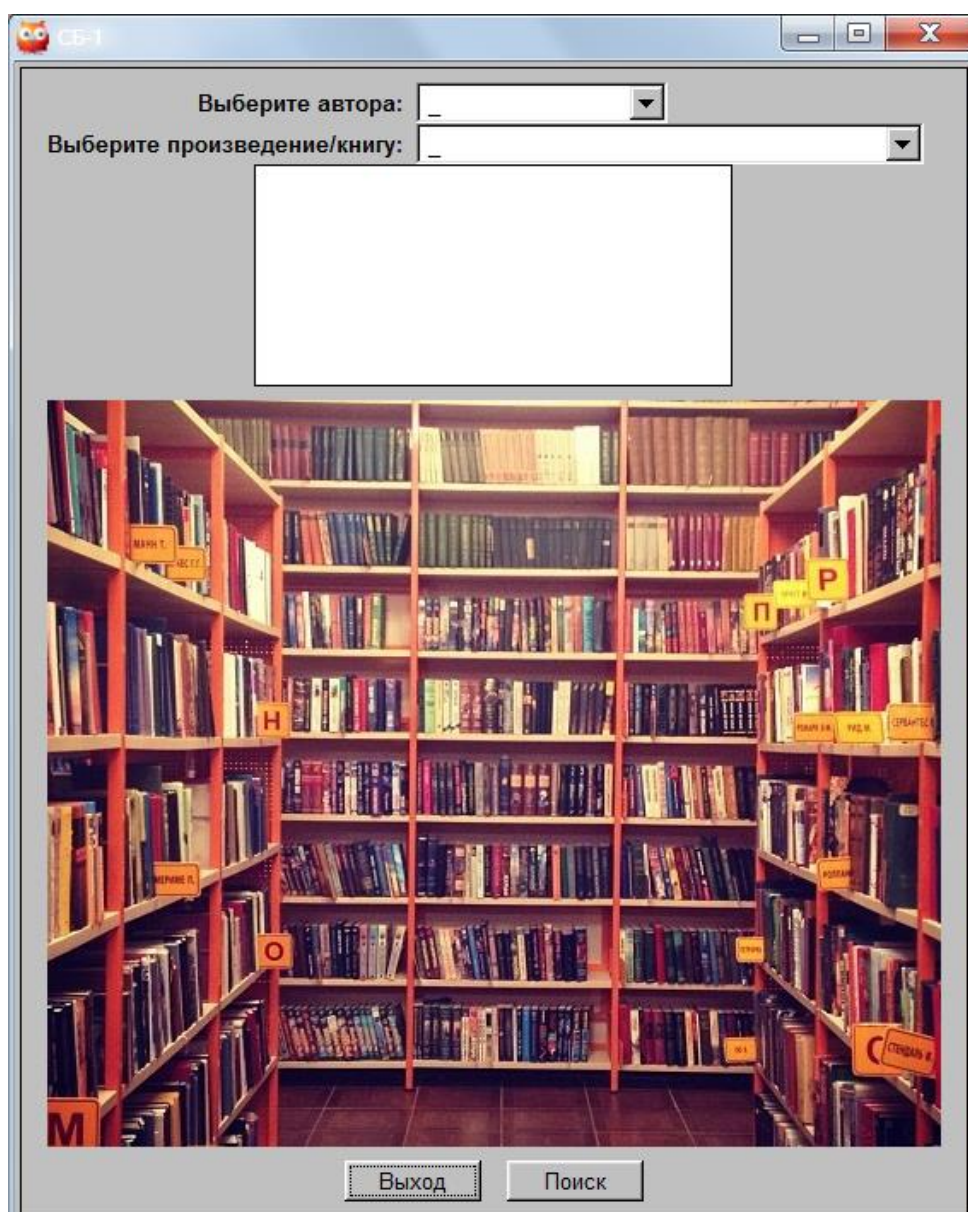


Рисунок 27 – Поиск литературы

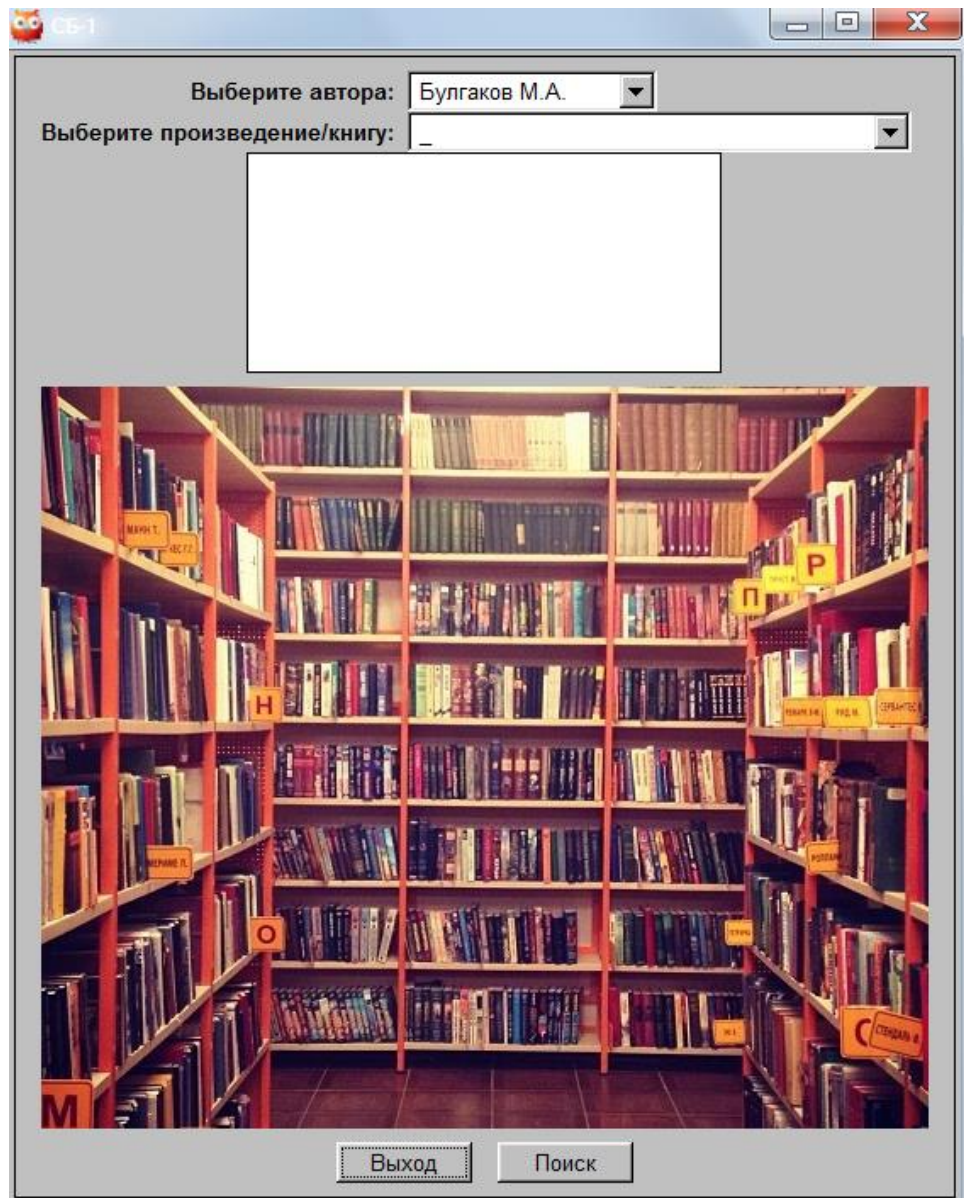


Рисунок 28 – Выбор автора

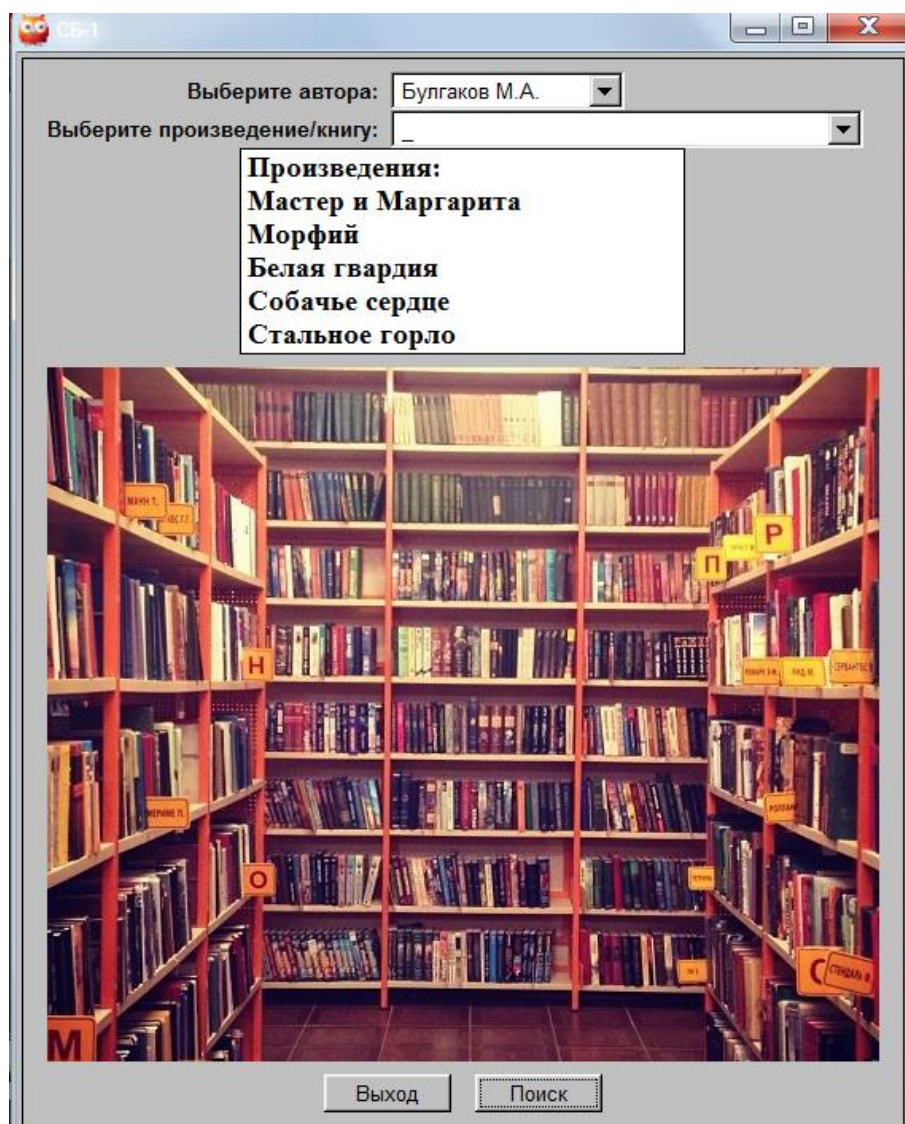


Рисунок 29 – Результат поиска по автору

Если необходимо найти, где расположена определенная книга, в этом же окне, изображенном на рисунке 28 необходимо выбрать автора и в поле «Выберите книгу/ произведение» выбрать необходимую книгу и нажать на кнопку «Поиск». В результате будет выведена информация о нахождении книги в библиотеке (полка, стеллаж) и количестве экземпляров, как на рисунке 30.

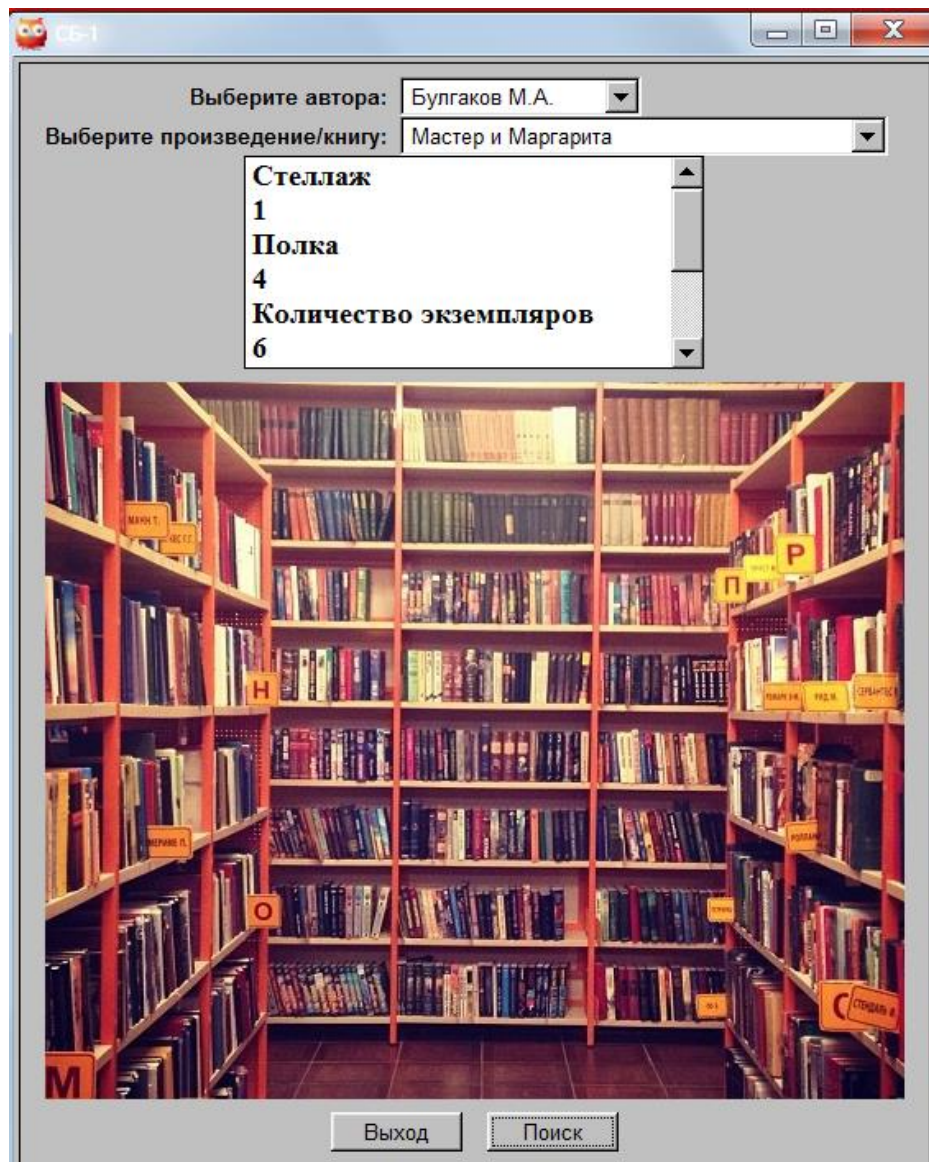


Рисунок 30 – Результат поиска по автору и произведению

Для того чтобы произвести анализ имеющейся учебной литературы в школьной библиотеке необходимо на главной странице, которую можно увидеть на рисунке 26, нажать кнопку «Анализ литературы». В результате откроется соответствующее окно, как на рисунке 31.

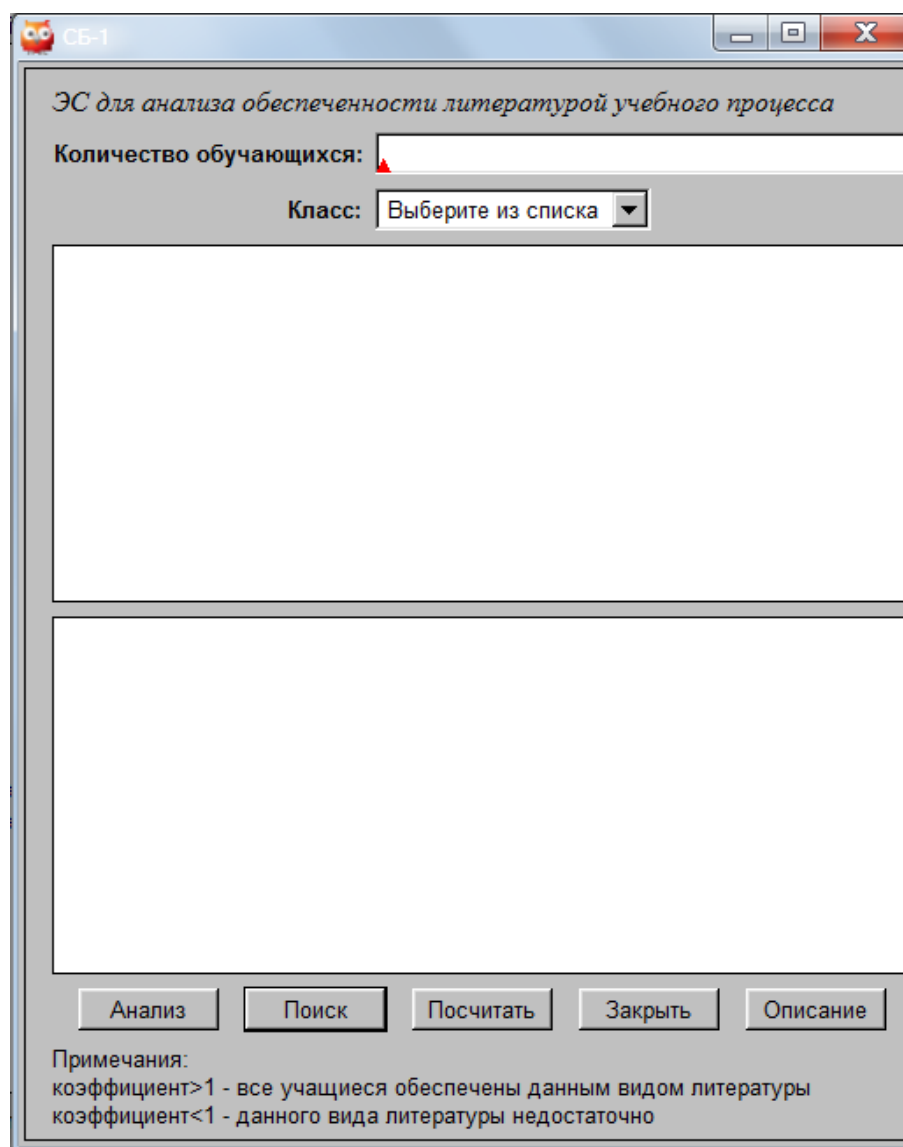


Рисунок 31 – Окно анализа литературы

Чтобы узнать каких учебников недостаточно для стандартного класса из 20 человек в поле «Класс» необходимо выбрать класс и нажать кнопку «Поиск». Пример выполнения этого действия можно увидеть на рисунке 32. На нем можно увидеть, что в поле вывода находится список недостающих учебных материалов с указанием предмета, к которому они относятся.

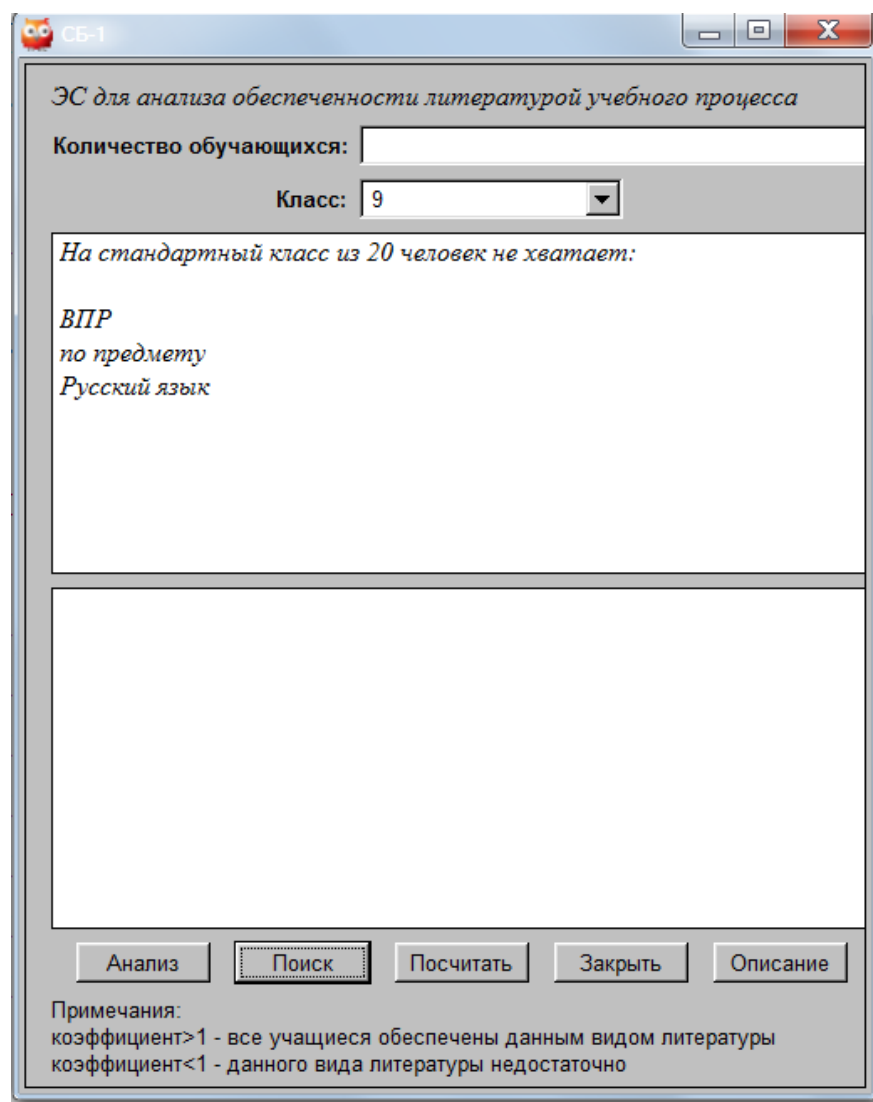


Рисунок 32 – Недостающие учебные материалы

Если необходимо проанализировать книгообеспеченность учебного процесса на определенное количество обучающихся, нужно ввести количество в поле «Количество обучающихся» и выбрать класс в поле «Класс», затем нажать «Анализ». В результате будет выведен коэффициент книгообеспеченности учебными материалами по каждому предмету в выбранном классе. Результат можно увидеть на рисунках 33-34. Следует заметить, что если коэффициент > 1, тогда все можно сделать вывод, что учащиеся обеспечены данным видом литературы. Если же коэффициент < 1 – данного вида литературы недостаточно.

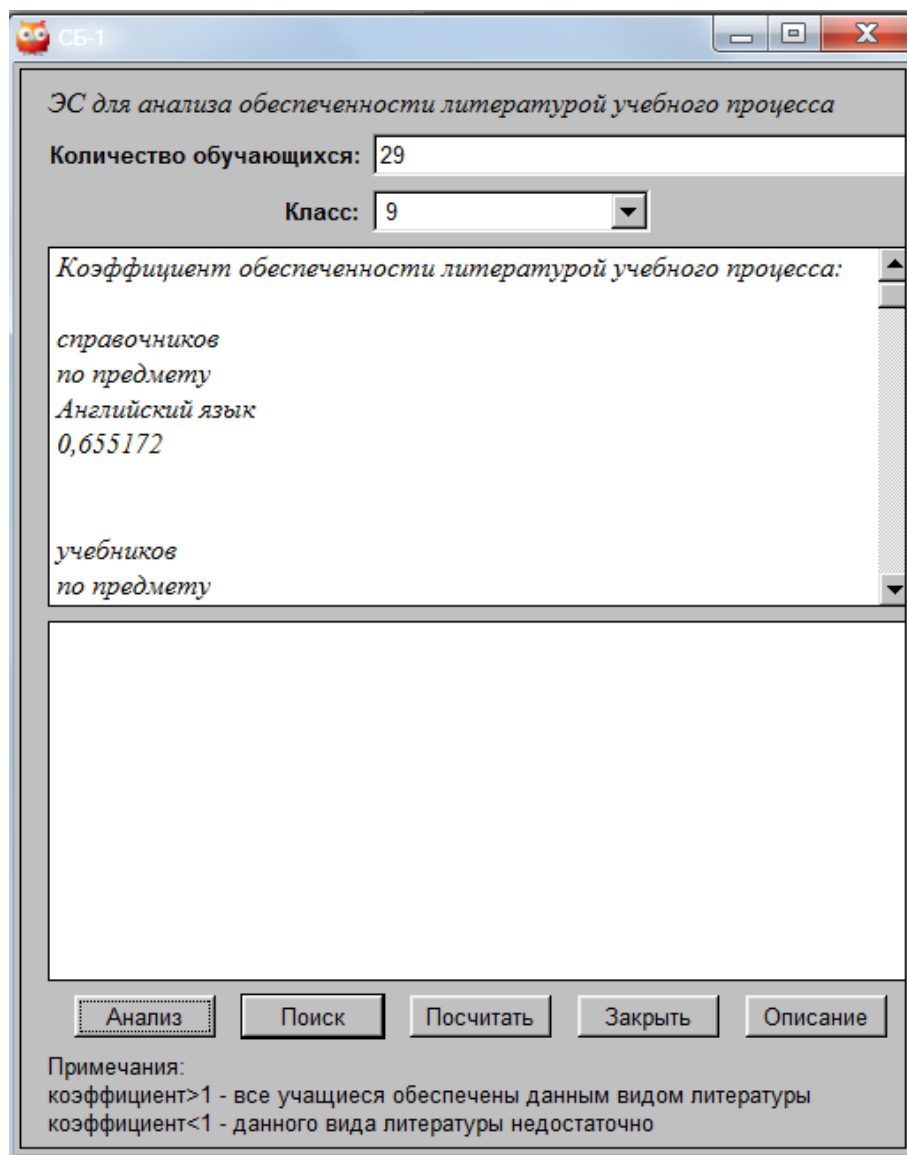


Рисунок 33 – Результат анализа книгообеспеченности

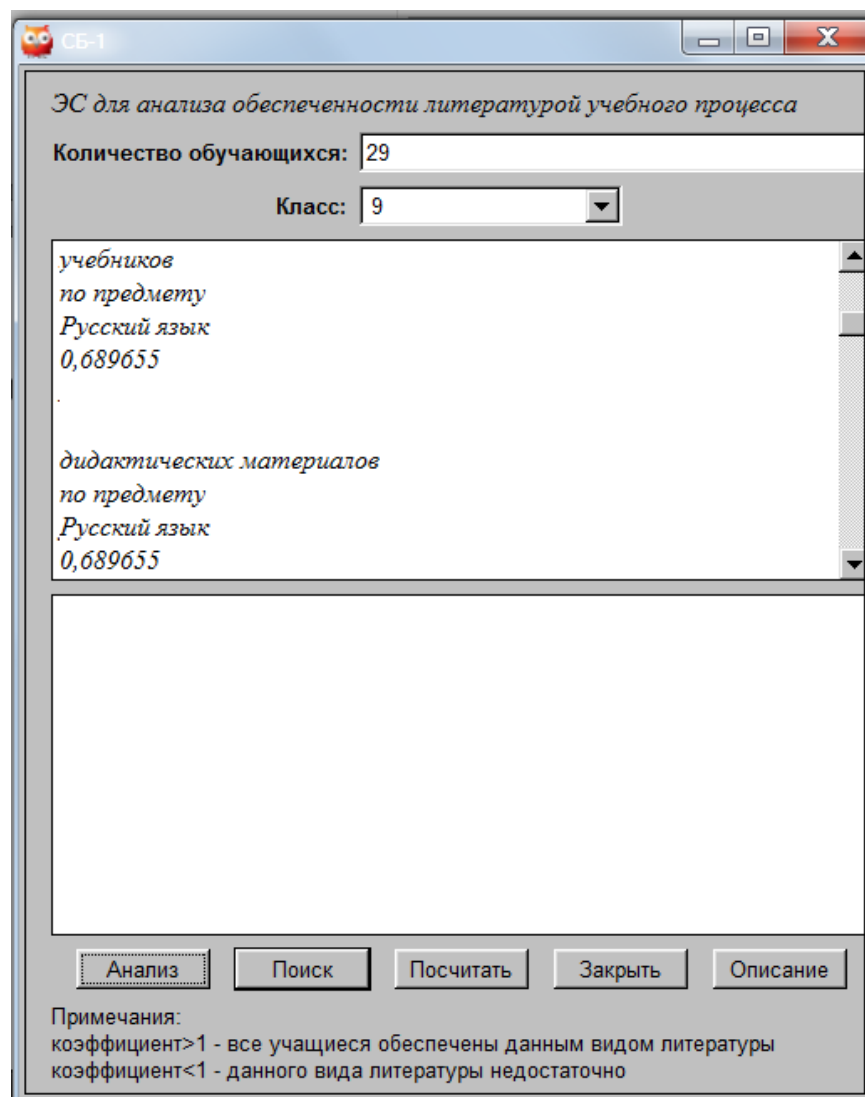


Рисунок 34 – Пример результата анализа книгообеспеченности

Чтобы посчитать, какой процент расходов на учебные материалы покрывает школьный бюджет необходимо нажать «Посчитать». В результате будет выведена информация о затратах на определенный вид учебных материалов и процентах затрат, которые покрываются школьным бюджетом. Пример выполнения данного действия изображен на рисунке 35.

При нажатии на кнопку «Описание» будет выведена информация о том, как считается коэффициент книгообеспеченности и затраты на учебные материалы. Для того, чтобы добавить книгу в систему библиотекарю необходимо открыть текстовый файл «baza_knig.txt» и ввести данные по образцу, указав автора книги, её название и расположение в хранилище школьной библиотеки (стеллаж и полка). Если возникает необходимость

удалить книгу из системы по причине утери, списания или какой-либо другой, необходимо открыть тот же текстовый файл, что и при добавлении, найти нужную книгу и удалить данные.

При необходимости изменить данные о количестве учебников необходимо открыть файл «baza_book.txt» и отредактировать необходимые данные (количество экземпляров учебников, название предмета, название других учебных материалов, цену учебных материалов, бюджет школы, которые тратятся на библиотеку).

ЭС для анализа обеспеченности литературой учебного процесса

Количество обучающихся: 29

Класс: 9

Коэффициент обеспеченности литературой учебного процесса:

справочников
по предмету
Английский язык
0,655172

учебников
по предмету

Расходы на учебные материалы
для учащихся (в рублях):

справочники
2030

бюджет школы покрывает(в %)
73,8916

учебники

Анализ Поиск **Посчитать** Закреть Описание

Примечания:
коэффициент>1 - все учащиеся обеспечены данным видом литературы
коэффициент<1 - данного вида литературы недостаточно

Рисунок 35 – Расходы на учебные материалы

ЗАКЛЮЧЕНИЕ

В процессе выполнения выпускной квалификационной работы было выяснено, что использование компьютеров в современном мире позволяет автоматизировать и упростить работу многих людей в различных областях человеческой деятельности. Экспертные системы позволяют реализовать эту функцию. Они способны заменить эксперта в какой-либо области для решения задач. Отличительной особенностью экспертных систем является возможность объяснения решения. А также возможность поиска решений на основании не полных данных.

Изучение литературы о различных средствах создания экспертных систем позволило проанализировать возможности этих средств. Для создания ЭС для анализа книгообеспеченности школьной библиотеки был выбран язык логического программирования Prolog. Также были рассмотрены основные возможности SWI-Prolog.

Для создания экспертной системы для анализа книгообеспеченности школьной библиотеки, были выявлены основные потребности библиотеки, а также изучена информация об используемых подходах к расчету книгообеспеченности и о подходе, применяемом в данной школьной библиотеке. Также были изучены документы, регламентирующие деятельность школьной библиотеки.

Созданная в результате экспертная система позволила автоматизировать работу школьного библиотекаря, тем самым сократив время, затрачиваемое на поиск книг в библиотеке, анализ учебных материалов и консультирование школьников о наличии книг и произведений того или иного автора, хранящихся в библиотеке, а также позволило рассчитывать затраты на недостающие материалы. На основании этого можно сделать вывод, что задачи выполнены, а цель достигнута.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Introduction // SWI-Prolog. 2013. URL: http://www.swi-prolog.org/pldoc/doc_for?object=manual (дата обращения 15.03.2018).
2. LISP// Энциклопедия языков программирования. 2017. – URL: <http://progopedia.ru/language/lisp/> (Дата обращения 10.12.2017).
3. SWI-Prolog // Национальная библиотека им. Н.Э. Баумана. 2016. URL: <https://ru.bmstu.wiki> (Дата обращения 17.01.2018).
4. Wielemaker J. XPCSE/Prolog Course Notes // SWI-Prolog. 2010. URL: <http://www.swi-prolog.org/download/xpce/doc/coursenotes/coursenotes.pdf> (Дата обращения 15.03.2018).
5. Андрейчиков А.В., Андрейчикова О.Н. Интеллектуальные информационные системы – М.: Наука, 2004. – 424 с.
6. Большакова Е.И. Программирование на языке пролог// Информатика и ЭВМ. Логическое программирование. 2009. URL: <http://www.studmed.ru> (Дата обращения 7.01.2018).
7. Большакова Е.И., Груздев Н.В. Программирование на языке Пролог: учебное пособие. – М.: МАКС Пресс, 2013. – 112 с.
8. Братко И. Алгоритмы искусственного интеллекта на языке Prolog – М.: Вильямс, 2001. – 640 с.
9. Городняя Л. Введение в введение в программирование на языке Лисп. – Новосибирск: НГУ, 2005. – 93 с.
10. ГОСТ 7.1-2003 Библиографическая запись. Библиографическое описание – Взамен ГОСТ 7.1-84, ГОСТ 7.16-79, ГОСТ 7.40-82; введен 02.06.2003. – М.: Изд-во стандартов, 2004. – 169 с.
11. Гужавин С.А. Экспертная система «Рапана»// Средство быстрой разработки экспертных систем «Рапана». 2008. – URL: <http://esrapana.narod.ru/rapana.pdf> (Дата обращения 11.12.2017).
12. Джарратано Дж., Райли Г. Экспертные системы. Принципы разработки и программирования – М.: Вильямс, 2007. – 1152 с.

13. Еремеев А.П., Чибизова Н.В. Методическое пособие по курсу «Экспертные системы» – М.: Издательство МЭИ, 2002. – 100 с.
14. Инструментальные средства разработки и проектирования экспертных систем// Портал искусственного интеллекта. 2009-2017. URL: <http://www.aiportal.ru/articles/expert-systems/design-tools.html> (Дата обращения 11.12.2017).
15. Лигун Т.А. Книгообеспеченность в библиотеке образовательного учреждения: задачи и решения // Информационный портал школьных библиотек России. 2007. URL: http://rusla.ru/rsba/provision/provision-method/tech_work_bibl (Дата обращения 15.11.2017).
16. Лисп – язык программирования искусственного интеллекта// LISP. 2012. – URL: <http://lisp2d.net/rus/teach/d.html> (дата обращения 09.12.2017).
17. Луценко Е.В. Интеллектуальные информационные системы: учебное пособие – Краснодар: КубГАУ, 2006. – 645 с.
18. Марселлус Д. Программирование экспертных систем на Турбо Прологе: Пер. с англ. – М.: Финансы и статистика, 1994. – 256 с.
19. Муромцев Д.И. Введение в технологию экспертных систем. – СПб.: СПб ГУ ИТМО, 2005. – 93 с.
20. Муромцев Д.И. Оболочка экспертных систем Exsys Corvid: Методическое пособие. – СПб.: СПбГУ ИТМО, 2006. – 69 с.
21. Назначение и основные возможности SWI-Prolog. // Интеллектуальные информационные системы. 2015. URL: <https://sites.google.com> (Дата обращения 17.01.2018).
22. Нейлор К. Как построить свою экспертную систему – М.: Энергоатомиздат, 1991. – 288 с.
23. Положение о библиотеке общеобразовательного учреждения: Приказ от 23.03.2004 №14-51-70/13/ Положение о библиотеке. – 2004. – 7 с.
24. Попов Э.В. Экспертные системы: Решение неформализованных задач в диалоге с ЭВМ. – М.: Наука, 1987. – 288 с.

25. Рассел С., Норвиг П. Искусственный интеллект. Современный подход – М.: Вильямс, 2007. – 1410 с.
26. Старовойтова О.Р., Плескачевская С.М. Справочник школьного библиотекаря/ под ред. Ю.Н. Столярова – М.: Школьная библиотека, 2006. – 448с.
27. Федеральный закон от 29.12.2012 г. № 273-ФЗ (ред. от 07.03.2018) «Об образовании в Российской Федерации», 2012. – 404 с.
28. Хабаров С.П. Интеллектуальные информационные системы. PROLOG- язык разработки интеллектуальных и экспертных систем: учебное пособие – СПб.: СПбГЛТУ, 2013.- 138 с.
29. Школьная библиотека: задачи и функции // Академик. 2010 URL: <https://dic.academic.ru/dic.nsf/ruwiki/1346368> (Дата обращения 7.12.2017)
30. Шрайнер П.А. Основы программирования на языке Пролог. Курс лекций. – М.: ИНТУИТ, 2005. – 176 с.