# Реферат

Отчёт 82 стр., 15 рис., 1 табл., 22 формулы, 19 источников,4 прил.

Цель работы - разработка системы поддержки принятия решения в выборе профессии для поступления в ВУЗ.

В процессе работы была проанализирована литература, рассмотрены имеющиеся в настоящее время системы тестирования в сфере профессиональной ориентации, написано техническое задание на создание СППР, спроектирована база знаний для СППР, программно реализованы модули СППР.

В результате работы было создано приложение «Система поддержки принятия решения в выборе профессии для поступления в ВУЗ», которое взаимодействует с системой управления базами данных MS Sql Server Compact.

Целевой аудиторией созданной системы являются молодые люди, которые не определились с профессией для поступления в учебное заведение.

СОДЕРЖАНИЕ

[Реферат 1](#_Toc485995912)

[Перечень используемых условных обозначений, сокращений, терминов 3](#_Toc485995913)

[ВВЕДЕНИЕ 4](#_Toc485995914)

[1. Системы поддержки принятия решений 6](#_Toc485995915)

[1.1. Классификация СППР 7](#_Toc485995916)

[1.2. Структура систем поддержки принятия решений 8](#_Toc485995917)

[1.3. Структура экспертной СППР 11](#_Toc485995918)

[1.4. Требования, предъявляемые к СППР 13](#_Toc485995919)

[1.5. СППР в выбранной сфере. 13](#_Toc485995920)

[2. Нейлоровская СППР 18](#_Toc485995921)

[2.1. Теоретические основы Нейлоровской СППР 18](#_Toc485995922)

[2.2. Структура Нейлоровской СППР 19](#_Toc485995923)

[3. Реализация Нейлоровской СППР в выборе специальности 27](#_Toc485995924)

[1. Формулировка задачи 27](#_Toc485995925)

[2. Проектирование базы знаний 27](#_Toc485995926)

[3. Программная реализация 31](#_Toc485995927)

[ЗАКЛЮЧЕНИЕ 38](#_Toc485995928)

[Выводы 39](#_Toc485995929)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 40](#_Toc485995930)

[Приложение 1 42](#_Toc485995931)

[Приложение 2 50](#_Toc485995932)

[Приложение 3 55](#_Toc485995933)

[Приложение 4 64](#_Toc485995934)

# Перечень используемых условных обозначений, сокращений, терминов

**СППР** – система поддержки принятия решения;

**БД** – база данных, это информационная модель, позволяющая упорядоченно хранить данные о группе объектов, обладающих одинаковым набором свойств;

**База знаний** - совокупность знаний, относящихся к некоторой предметной области и формально представленных таким образом, чтобы на их основе можно было осуществлять рассуждения;

**СУБД** – система управления базами данных, программное обеспечение, предназначенное для работы с базами данных;

**ЭС** – экспертная система ‑ комплекс компьютерного программного обеспечения, помогающий человеку принимать обоснованные решения. Экспертные системы используют информацию, полученную заранее от экспертов — людей, которые в какой-либо области являются лучшими специалистами.

**Агрегирование** – процесс сбора, обработки и представления информации в окончательном виде.

**СОЭД** – сервер обработки электронных документов;

**ИСУ** – информационная система управления - набор, находящихся в отношениях и связях друг с другом, образующих определенную целостность и единство, элементов для сбора, обработки и хранения и предоставления информации.

# ВВЕДЕНИЕ

В какой бы ты сфере жизнедеятельности не находился, везде сталкиваешься с проблемой выбора при принятии решения. Принимать решения приходится всегда и везде. В экономике, в бизнесе, в международных делах, даже - какой выбрать шкаф. Решение, которое принимается, должно быть наилучшим из предложенных вариантов. А, чтобы учесть все факторы, аспекты альтернатив необходимо принять к сведению, что без посторонней помощи принять какое-либо решение будет сложно.

Повышение качества профессионального образования – одно из приоритетных направлений современного этапа модернизации образования. Выбор будущей профессии, и, соответственно, вуза для поступления — серьёзный выбор, который приходится делать молодому человеку, выпускнику школы. И, к сожалению, далеко не все студенты к концу обучения довольны сделанным когда-то выбором.

Достаточно большая доля выпускников школы может не знать, кем им быть по профессии. Родители, друзья могут подсказать, но не дать уверенность в правильности выбора. Исходя из такой проблемы, а она масштабна, профессиональная ориентация помогает её решить.

Существенную роль в практической реализации обозначенной выше проблемы играет нахождение самих профориентационных тестов. Исходя из достоверных источников, хороших профориентационных тестов, которые бы учитывали всё: психологические качества человека, его способности, неизвестные самому тестируемому его задатки, - не существует.

Создание системы поддержки принятия решений в данной сфере очень актуальная тема. Нет систем, которые бы отражали полную информацию при выборе и ранжировании альтернатив профессий. Поэтому, в рамках данной работыразрабатывается система поддержки принятия решений в выборе специальности для поступления в ВУЗ.

Целью работы является разработка системы поддержки принятия решения в выборе профессии для поступления в ВУЗ.

На сегодняшний день в сфере профессиональной ориентации разработано немало систем. К примеру, в работе были рассмотрены лишь 2 информационные системы – «Психодиагностика» и «Профориентатор».

Выпускная квалификационная работа состоит из двух основных частей:

1. Теоретическая часть, - в ней отражается информация о системах поддержки принятия решений, какие они бывают; их структура и предъявляемые требования; теоретические основы Нейлоровских систем, которые основаны на теории вероятностей и, также, структура таких систем.
2. Практическая часть, - здесь раскрыта суть ВКР – реализация системы поддержки принятия решений в выборе специальности для поступления в ВУЗ: техническое задание на разработку СППР; проектирование базы знаний; программная реализация.

# Системы поддержки принятия решений

Что же такое система поддержки принятия решения? Из самого названия можно понять, что это такая система, которая помогает принять решение, сделать какой-либо вывод или заключение. Бывают ситуации, когда одному человеку или даже группе людей не в силах произвести выполнение каких-либо задач в ограниченное время для принятия решения, для этого и нужны СППР.

СППР - это автоматизированная система. Человеку свойственно ошибаться, особенно когда выполняешь монотонную работу. Машина же, в свою очередь, не может совершить какую-то оплошность, она производит расчёты по алгоритму.

Предназначением СППР является поддержка многокритериальных решений. А это означает, для того чтобы найти результат, необходимо учитывать множество показателей одновременно – оценка решения по одному критерию не приведёт к правильному заключению.

СППР не производит генерацию правильных решений, а предоставляет данные в соответствующем виде (отчёты, таблицы, графики и т.д.) для изучения анализа.

Учёт множества критериев, их обработка и само решение составляют большой объём данных. В этом состоит информационная сложность.

Основным задачами СППР являются: сбор, хранения и анализ хранимой информации. Но следующие задачи отличают СППР от остальных информационных систем:

1. Оптимизация - выбор наилучшего решения из числа допустимых;
2. Упорядочивание решений по предпочтению.

Принципиальным моментом обеих задач является подбор множества критериев, которые будут составлять основу для оценки и сопоставления возможных решений.

## Классификация СППР

1. На уровне пользователя СППР подразделяются на 3 типа:

* Пассивная

Система, которая помогает процессу принятия решений, но не может вынести предложение, которое решение можно принять.

* Активная

Система принимает участие в разработке правильного решения. Она может сделать предложение, какое решение надо выбрать.

* Кооперативная

Такая СППР предполагает возможность взаимодействия с пользователем. Выдвинутые системой решения пользователь может изменить, дополнить, улучшить, а затем отослать обратно в систему для проверки. Система, усовершенствовав решения, заново отправляет их ЛПР, и так до тех пор, пока не появится согласованное решение.

1. На концептуальном уровне СППР различают:

* Модельно-ориентированные

Характеризуются доступом и манипуляциями с математическими моделями: статистические, финансовые или иные модели.

* Управляемые сообщениями

Групповая СППР. Поддержка работы нескольких пользователей (группы), которые работают над выполнением одной задачей.

* Управляемые данными

Ориентированы на доступ и манипуляции данными. Используют в работе как внутренние данные, так и внешние.

* Управляемые документами

Управляют, производят поиск и манипулирование неструктурированной информацией, которая может быть задана в разных электронных форматах.

* Управляемые знаниями

СППР, обеспечивающие решение задач в виде фактов, правил и процедур.

1. На техническом уровне различают:

* Настольные СППР

Системы, которые по объёму являются небольшими и обслуживаются лишь одним пользовательским компьютером.

* СППР предприятия

Имеют подключение к большим хранилищам информации и обслуживают многих менеджеров организации.

1. По компьютерной поддержке принятия решений:

* Экспертные

Основой систем являются знания экспертов. Компьютерное представление знаний высококвалифицированных специалистов даёт возможность работы с системами людям другой квалификации.

* Советующие

Последовательно взаимодействуют с оператором для выявления причин проблемы ситуации и выдача поэтапных рекомендаций пользователю по решению возникшей проблемы.

## Структура систем поддержки принятия решений

Система поддержки принятия решений - компьютерная система, которая автоматизирована с целью помощи людям, принимающим решение в сложных условиях для полного и объективного анализа предметной деятельности.

СППР состоит из двух частей:

1. Управленческая информационная система – система, с помощью которой выполняются поиск, обработка и хранение информации.
2. Система управления базами данных (СУБД) – совокупность программных и лингвистических средств, которые необходимы для управления созданием баз данных (БД) и их использованием.

Рассмотрим структуру СППР (рисунок 1). Не включая пользователя, у СППР три составляющие: подсистема обработки и хранения данных, подсистема хранения и использования моделей, программная подсистема.

Уровень принятия решений

СППР

Уровень обработки информации

ИСУ

ИСУ

Уровень обработки данных

СОЭД

СОЭД

СОЭД

Рисунок 1.1 - Структура СППР

1. Подсистема обработки и хранения данных.

Данную подсистему характеризуют преимущества построения и использования БД. Она должна обладать возможностями:

* Составление комбинаций данных, полученных из разных источников путём использования функций агрегирования и фильтрации;
* Прибавление и отключение источников данных;
* Построение логической структуры данных в терминах понятных пользователю;
* Использование и манипуляция данными, которые являются неофициальными, для проверки альтернатив пользователя;
* Управление данными с помощью большого ряда функций, которые представляет СУБД;
* Обеспечение независимости БД, входящей в СППР, от других БД.

1. Подсистема хранения и использования моделей

Подсистема обеспечивает доступ к моделям принятия решений. Достигается это путём введения в ИС соответствующих моделей. В результате в СППР будут использоваться методы моделирования ситуаций.

Моделирование должно быть гибким. Из этого следует, что сам процесс должен включать в себя специальный язык моделирования, отдельные программы.

Моделей существует множество, следовательно, они должны храниться в определённом месте. Для этого существует база моделей. Она включает в себя различные модели, блоки и процедуры, используемые для построения моделей.

1. Программная подсистема

В использовании СППР немаловажную роль имеет используемый интерфейс. Он включает в себя систему управления диалогом (СУД), компьютер и пользователя.

Интерфейс должен обладать такими возможностями:

* Манипулирование формами диалога, то есть изменение их в процессе выбора пользователя;
* Передача данных по системе разными способами;
* Получение данных от разного рода оборудования системы в различном формате;
* Обеспечение гибкой поддержки знаний пользователя – помощь, справка, подсказки.

Если рассматривать СППР через призму процессов принятия решений, то можно выделить следующие виды:

1. Информационная – система, которая собирает и анализирует большое количество информации, после чего она может влиять на процесс выбора решения.
2. Модельная – система, предоставляющая использованию математических, статистических, финансовых моделей при принятии решения.
3. Экспертная – система, которая на основании знаний эксперта, выдаёт заключение – «диагноз», даёт совет в принятии решения.

Рассмотрим структуру одного типа СППР – экспертной СППР.

## Структура экспертной СППР

Экспертная СППР (ЭС) — это реализация элемента знаний и опыта эксперта в таком виде, что система самостоятельно может дать совет или принять интеллектуальное решение относительно обрабатываемой функции. Дополнительное свойство — способность, с помощью которой система сама может разъяснить ход своих действий простым языком, понятным пользователю.

СППР, по существу, производят моделирование поведения эксперта при принятии решения в различных предметных областях. Исходя из этого, обязательным требованием создания СППР является то, что должны быть известны личности людей (эксперты), которые справляются с поставленными задачами. После этого происходит изучение их деятельности, в которой они специализируются, для выявления необходимых знаний. Затем формируется база знаний (БЗ), которая представляет собой связанные друг между другом сведения, факты и правила, которые заранее структурированы и интерпретированы.

СППР выполняют следующие функции:

1. Распознавание сложившейся ситуации, её анализ, постановку диагноза и формулирование совета для получения какого-то результата;
2. Выстраивание путей достижения конкретных целей с учётом резервов организации;
3. Пополнение БЗ;
4. Обеспечение простого в использовании интерфейса.

На рисунке 2 представлены блоки, из которых состоят ЭС.

|  |  |
| --- | --- |
| Интерфейс | |
| Эксперта | Пользователя |

Рисунок 1.2 - Блок-схема экспертной системы

Блок ввода вывода данных

Блок приобретения знаний

Блок объяснений

Блок расчётов

База данных

База данных

Блок логических выводов

**База знаний** – совокупность знаний, отражающих знания эксперта в некоторой предметной области, и построенных таким образом, что на их основании можно было делать выводы.

**Блок логических выводов**  - блок, предназначение которого является извлечение знаний из системы, должен работать с данными, которые могут привести к ошибкам. Данный блок является главным, так как с его помощью пользователь генерирует альтернативы и производит решение. Количество диагнозов возможно большое количество, для этого в применение берут способы их иерархического описания, это позволяет облегчить саму структуру и упростить процесс выбора решения.

**Блок объяснений** - отражает путь, по которому система пришла к определённому решению. В экспертных СППР, основанных на правилах, объяснения получают путём отслеживания шагов, которые привели к данному выводу.

**База данных, блоки расчётов и ввода и вывода данных** являются отличительной чертой, присущей только экспертным СППР. Так как наличие данных компонент обусловлено большим объёмом и жёсткостью требований к точности расчётов, необходимых для принятия решений.

**Блок приобретения знаний** отвечает за самообучение экспертной СППР, и, следовательно, за надёжность и точность выдаваемых рекомендаций.

## Требования, предъявляемые к СППР

Ко всем системам есть свои требования. Далее будут перечислены требования, предъявляемые к СППР:

* Хранимые данные должны быть детализированы и обобщены;
* Данные должны быть верными - ошибки не допускаются;
* Формат хранения должен быть единым;
* Избыточные данные допускаются, НО в ограниченных рамках;
* В управлении данными должно допускаться добавление новых данных;
* Данные, которые накапливаются в продолжительный период времени, должны быть доступны все;
* Запросы могут быть произвольными, заранее не оформленными;
* Загрузка процессора только при выполнении запросов на 100%;
* Гибкость и независимость работы пользователей.

## СППР в выбранной сфере.

Современные компьютерные технологии дошли и до сферы профессиональной ориентации. Автоматизация интерпретации позволила автоматизировать само проведение тестирования, обработку и выдачу промежуточных результатов и заключений.

Широкое распространение получили компьютеризированные версии методик, но тут нужно понимать, что эти системы лишь автоматизируют обработку и имитируют выдачу психологических заключений, производя подмену процесса процедурой компиляции совокупности диагностических значений.

Если создавать полномасштабную систему, то без применения искусственного интеллекта не обойтись. Она предполагает под собой, перенос структурированных экспертных знаний в вычислительную машину, моделирование рассуждений эксперта-психолога при объяснении результатов тестирования. Успешность создания такой системы заключается в выбранном специалистом искусственного интеллекта методе извлечения и представления знаний, а также познаний, которые предоставит эксперт-**психолог**.

Работа по созданию экспертных систем в направлении профориентации ведётся уже более 20 лет. Первые публикации на эту тему были скорее теоретические, чем практические.

При поиске существующих экспертных систем в сфере профессиональной ориентации было найдено большое количество работ. Но, описаны будут, лишь, 2 системы.

1. **ЭС «Психодиагностика»**

Проектирование ЭС включает в себя построение концептуальной модели предметной области и модели интеллектуальной деятельности специалиста, в данном случае эксперта-психолога.

Далее идёт работа по созданию базы знаний. Экспертное заключение производится таким образом, что учитывается не особенности клиента, а процедура работы с системой. При прохождении процедуры есть комментарии, в которых отображаются методики, которые были применены при образовании экспертного вывода. То есть система в таком обличии преобразуется как справочно-библиотечная.

Само устройство ЭС «Психодиагностика» состоит из трёх частей: вводный раздел «Запрос», система «Планирование обследования», система «Интерпретация результатов».

Во вводном разделе клиентом заполняется анкета, в которой важны две характеристики – возраст и статус субъекта. Они используются для первичного отбора методик. А также клиенту необходимо выбрать «заказчика», который необходим для уточнения запроса. Эта информация необходима для формирования рекомендаций по предоставлению результатов обследования заинтересованным лицам. Например, если обследуемым является ученик школы, то заказчиками могут быть: администрация школы, учителя, родители.

Система «Планирование обследования» рекомендует пользователю выбор методик, которые подходят ему по возрасту, ограничениям и организации проведения (время, которое отводится на обследование; наличие расходных материалов). В подведённых итогах выводятся методики, сформированные системой, и объяснения, обусловившие выбор той или иной методики.

Вторая система «Интерпретация результатов» после проведённого обследования выводит результаты, а также объяснения полученных итогов по использованным методикам.

В выводе результата самого обследования представляется в краткой форме данные о клиенте (раздел «Запрос»), информация об использованных методиках, рекомендации по формированию психологического диагноза, рекомендации по формированию заключения.

1. **ЭС «Профориентатор»**

Первая версия ЭС была разработана еще в 1997 году. В его основу вошёл набор тестовых заданий, а также набор тестовых шкал, который не дублирует ни один из имеющихся в профориентации инструментов.

Изначально комплекс состоял из двух блоков: «Интересы» и «Способности». Далее в работе стало очевидно, что с помощью этих блоков выбор профессии определяется неточно. В результате добавился ещё один блок личностных качеств.

Целевой аудиторией экспертной системы «Профориентатор» являются учащиеся 7-11 классов и абитуриентов. Она определяет профессиональную направленность, формирует профильные классы, предоставляет выбор средний и высших учебных заведений.

«Профориентатор» включает в себя три блока:

1. Оценка структуры интересов.

Данный блок включает в себя 80 утверждений, каждое из которых относится к разным сферам профессиональной деятельности: наука, искусство, техника, бизнес, знак, общение, природа, риск.

Время, которое даётся на прохождение этого блока не ограничено.

1. Оценка структуры интеллекта.

Из названия блока можно понять, что здесь будет определяться уровень развития способностей (факторов умственной деятельности). Оценивается умение вычислять, логическое мышление, лексика, эрудиция, зрительная и абстрактная логика. А также выводится средний балл «уровня умственного развития», то есть отражается уровень обучаемости и умственной работоспособности.

На выполнение отводится 40 минут.

1. Оценка структуры личности.

Третий блок состоит из 45 вопросов, которые ориентированы на выявление личностных качеств: самоконтроль, активность, эмоциональная стабильность, согласие. Время прохождения не ограничено.

Надёжность системы отражается во внутренней согласованности всех включённых в тест шкал.

Результаты тестирования выводятся в десятибалльной шкале, то есть баллы, полученные в ходе тестирования, подвергаются стандартизации – а для неё требуется знать нормы: среднее значение и стандартное отклонение по каждой шкале. Нормы рассчитываются по возрастным группам.

Также выполняется анализ психометрических показателей. Данная работа заключается в следующем: производится анализ коэффициента корреляции каждого отдельного пункта, распределяются ответы испытуемых, вводятся новые тестовые пункты на место неэффективных, если такие имеются.

Анализ корреляций шкал друг с другом просто необходим для таких экспертных систем. Некоторые шкалы могут быть связаны между собой как положительно, так и отрицательно. И эта функция помогает психологу-консультанту в различных случаях строить гипотезы при недостатке информации о клиенте, замечать и интерпретировать противоречивые результаты и связывать их воедино.

**Выводы**

Исходя из рассмотрения вышеперечисленных систем поддержки принятия решений в сфере профессиональной ориентации «Психодиагностика» и «Профориентатор» можно сделать вывод. Эти системы работают по простым алгоритмам – то есть, чтобы сделать какой-то «диагноз», система находит либо среднее значение, либо производит расчёт по несложным формулам.

Отсюда следует, что создание систем со сложными математическими расчётами необходимо. Использование основ Нейлоровских систем с вероятностным подходом в разработке систем будет отражать наиболее точные результаты, нежели подходы, которые используются в созданных в настоящее время СППР.

# Нейлоровская СППР

## Теоретические основы Нейлоровской СППР

К. Нейлор разработал методику создания экспертной системы на основе Байесовского подхода. Его базой является теория вероятностей и теорема Байеса. Формула Байеса даёт возможность накопления информации, которая поступает из разных источников, для утверждения или отрицания определённой гипотезы.

Пусть существует некая гипотеза G и её априорная вероятность V(G) истинности. Эта вероятность известна изначально. Либо её на предыдущем шаге высчитали, либо она задавалась как исходные данные.

Следующим шагом становится известным некоторое свидетельство S, которое относится к данной известной гипотезе. Формула Байеса (формула 2.1) «уточняет» априорную вероятность, что гипотеза истинна, на основе известной информации о гипотезе G:

1. – вероятность гипотезы при свидетельстве;
2. – вероятность свидетельства при гипотезе;
3. – исходная априорная вероятность гипотезы;
4. – вероятность свидетельства при отсутствии гипотезы;
5. – вероятность обратная вероятности истинности;

Исходя из условий задачи, вероятности под номерами 2, 3, 4 известны первоначально. Вероятность (5) высчитывается по формуле (2.2):

(2.2)

Работа ЭС по Байесу принципиально состоит в следующем. Изначально у нас в наличии имеется априорная вероятность , хранение которой происходит в базе знаний. При получении свидетельства *S*, априорная вероятность пересчитывается по формуле 2.1, и теперь она записывается вместо . Далее, получая новые свидетельства, происходит обновление вычисляемой вероятности. В конце работы такого пересчёта вероятностей гипотез, собрав все свидетельства, касающихся их, выводится заключение («диагноз»), в котором отражена та гипотеза, которая наиболее вероятна.

## Структура Нейлоровской СППР

Нейлоровская диагностирующая система имеет свои отличительные принципы:

1. Имеются максимальный P1(G) и минимальный P2(G) пороги для вероятностей гипотез;
2. Учитываются неопределённости, которые заключаются в реакции пользователей;
3. Вводятся цены свидетельств, которые определяют сценарий проведения диалога с пользователем.

Рассмотрим эти пункты.

* + 1. **Элементы механизма логического вывода**

Первое, что усложнилось в Байесовском подходе, это введение нижнего и верхнего порогов для вероятностей каждой их гипотез.

Если вероятность V(G) при учёте всех свидетельств выше верхнего порога P1(G), то гипотеза G идёт в принятие возможного заключения как основа (формула 2.3).

(2.3)

Если вероятность V(G) меньше нижнего порога, то гипотеза отвергается, становится неправдоподобной (формула 2.4).

(2.4)

Возможна установка границ индивидуально для каждой гипотезы.

Важным моментом в организации диалога с пользователем является учёт неопределённости в ответах на вопросы. Идеальной ситуацией является то, когда можно ответить на вопрос с точностью в 100% «да» или «нет». Но таких ситуаций в жизни мало. Поэтому, реальной ситуацией будет, когда пользователь не сможет точно дать ответ, либо захочет уклониться от ответа.

Для пользователя предоставляется ранжирование оценку ответа по 11-балльной шкале.

* -5 соответствует ответу НЕТ;
* 0 соответствует НЕ ЗНАЮ;
* +5 соответствует ДА.

В итоге каждое из множества свидетельств на основании ответа пользователя будет оцениваться по этой шкале

Исходя из байесовского подхода после обработки каждого из свидетельств S, вычисляется вероятность *V(G/S)* и предыдущая априорная вероятность *V(G)* заменяются ею. В методе, предложенном Нейлором, предлагается метод вычисления не *V(G/S)*, а *V(G/F)*.

Во-первых, ответ с оценкой «5» будет соответствовать вероятности *V(G/S)*, которая вычисляется по формуле Байеса (2.1).

Во-вторых, вариант ответа «-5» соответствует вероятности ), которая может быть найдена из формул 2.5, 2.6, 2.7.

(2.5)*,*

(2.6),

(2.7)*.*

Значения , подразумеваются заданными.

В-третьих, случай, когда F=0 («НЕ ЗНАЮ») не несёт под собой изменение априорной вероятности, поэтому здесь остаётся старое значение вероятности (2.8).

(2.8)

Промежуточные значения *V(G/F)* предполагается вычислять с помощью линейной интерполяции, так как известны характерные точки. Очевидно, для любогополучаются соответствующие значения .

* + 1. **Цены свидетельств – косвенная цепочка рассуждений**

Следующее усложнение, которое было внесено в базовую схему байесовского метода, связано с управлением логического вывода.

База знаний хранит в себе конечные множества гипотез и свидетельств. Каждой из гипотез соответствует своё подмножество свидетельств. Но обработка огромного количества возможных вопросов для выбора и вывода заключения поведёт за собой трудности как для машины, на которой работает эта система, так и для пользователя, он будет уставать от долгой работы и давать неточные ответы.

Поэтому часто используется обратная цепочка рассуждений: от гипотез – к свидетельствам. В данном случае перебор будет вестись не по свидетельствам, а по гипотезам. Такой подход выглядит более продуктивным, нежели предыдущий. Но и здесь неясно, не определён порядок перебора гипотез, а также прилегающих к ним свидетельств.

В Нейлоровской же системе будет применена смешанная цепочка вывода. Она не относится ни к прямой, ни к обратной цепочке рассуждений.

В рассматриваемом смешанном подходе назначаются цены свидетельствам *Si,* которые отражают важность свидетельств в процессе логического вывода. Очевидно, в процессе диалога будут выбираться свидетельства с наибольшими ценами. А в процессе вывода всё время ведётся пересчёт цен свидетельств в зависимости от получаемых ответов. В качестве цены свидетельства *S* будем использовать следующую формулу (2.9):

где:

(2.7)

соответствуют 2.7 и 2.10 соответствуют формулам Байеса, а выражение для вероятности (формула 2.11)

получается из предыдущей формулы заменой *S* на на основании очевидных равенств:

Таким образом, цена свидетельства равна полной сумме возможных изменений вероятностей по всем *n* гипотезам, которые имеются в базе знаний.

Далее следует выбор свидетельства с максимальной ценой, система задаёт соответствующий вопрос пользователю. После получения ответа пересчитываются все вероятности. После формирования нового массива *V(Si)*, заново пересчитываются все цены свидетельств и алгоритм повторяется.

Необходимо запомнить, что вероятности не изменяются в процессе работы алгоритма, меняется лишь массив вероятностей *V(G).*

* + 1. **Правила остановки**

Для окончания работы алгоритма Нейлоровской ЭС необходимы условия остановки. Если не задействовать никакие правила, то исход работы системы ясен, будут перебираться все свидетельства и гипотезы. Вряд ли найдётся такой пользователь, которого надо будет тестировать, используя все свидетельства и гипотезы. Окончание процесса экспертизы происходит намного раньше.

У каждой гипотезы в рассматриваемой системе имеются числа, связанные с ней:

*V(G) –* начальная априорная вероятность истинности гипотезы G;

*Vmax(G) –* текущая максимальная вероятность гипотезы G, предполагая, что все свидетельства будут в её пользу;

*Vmin(G) ‑*  текущая минимальная вероятность гипотезы G, предполагая, что все свидетельства будут против неё;

*P1(G), P2(G) –* максимальный и минимальный пороги, пропорциональны величине *Vmax(G).*

Изначально *Vmax(G)= Vmax(G),* но затем текущая максимальная величина *Vmax(G)* будет уменьшаться.

Выделяются три условия остановки:

1. *Определение наиболее вероятной гипотезы.*

Если в какой-то момент, когда совершается обработка списка свидетельств, во время работы ЭС становится ясным, что для некоторой гипотезы выполняются условия:

*Vmin(Gk) > Vmax(Gi)* для, (2.14)

то, очевидно, что нет смысла продолжать работу ЭС, так как гипотеза *Gk* наиболее вероятная «Диагноз» ‑ *Gk.*

1. *Определение правдоподобной гипотезы.*

Если в некий момент времени для гипотезы *G* выполнилось условие:

*Vmin(G) > P1(G)* (2.15)

где *P1(G)-* максимальный верхний порог. Описываемый критерий остановки в действие приходит чаще других. Обычно пользователь ищет все гипотезы, которые подходят этому условию, поэтому необходимо их проверять ещё дополнительным условием:

*Vmax(G) < P1(G)* (2.16)

Если данное условие не выполняется, то можно продолжать процесс работы ЭС для пополнения перечня правдоподобных итогов.

1. Отсутствие заключения

Факт, что система не может принять решение и выдать заключение. Такая ситуация происходит в некоторый момент времени при следующем условии.

*Vmax(G) < P2(G)* (2.17)

Для реальных ЭС все условия должны быть согласованы между собой и базой знаний. А также при добавлении новых баз знаний необходимо проверять на корректность и логическую непротиворечивость, чтоб работа системы оставалась безошибочной.

* + 1. **Структура базы знаний**

Как мы знаем, база знаний хранит знания о гипотезах и свидетельствах – «диагнозах» и их «симптомах». Формат записи для описания конкретной гипотезы следующий:

Название гипотезы; *V(G)*; N; (i1, ); …; (iN, ).

*V(G) –* исходная вероятность гипотезы;

N – количество свидетельств *S*;

iN – номер свидетельства;

= *–* вероятность выполнения свидетельства для гипотезы;

= - вероятность выполнения свидетельства при неверности данной гипотезы.

* + 1. **Алгоритм логического вывода**

*Шаг первый.*

Формируем массив V(G) априорных вероятностей для имеющихся гипотез. Для этого необходимо посмотреть базу знаний и извлечь из неё вероятности для каждой из гипотез.

*Шаг второй.*

Формируем массив цен свидетельств B(S) по формуле 2.18:

где *j* – номер свидетельства. Очевидно, что в сумму входят только те слагаемые, которые относятся гипотезам , содержащим в своём описании тройки с номером *j.*

*Шаг третий.*

Определить свидетельство *Sm* с максимальной ценой:

.

*Шаг четвёртый.*

Задать вопрос пользователю, соответствующий свидетельству с максимальной ценой (найден на третьем шаге). Ответ пользователя *Fm* должен быть дан по 11-балльной шкале от -5 до +5.

*Шаг пятый.*

Пересчитать массив *V(Gi)* на основе полученного ответа на четвёртом шаге.

*V(Gi)* := *V(Gi/ Fm)* (2.19)

*Шаг шестой.*

На основе полученных результатов на пятом шаге пересчитать массив цен свидетельств *B(Sj).*

*Шаг седьмой.*

Для каждой из множества гипотез *G* вычислить максимальную *Vmax(Gi)* и минимальную *Vmin(Gi)* вероятности.

*Шаг восьмой.*

По формуле 2.20 определить число *Vm:*

(2.20)

*Шаг девятый.*

Проверить по формуле 2.21, существует ли такой номер k, для которого

*Vmax(Gi) > Vm.* (2.21)

Если ДА, то перейти к третьему шагу. Если НЕТ, то выбрать (2.22) гипотезу или гипотезы *Gm:*

(2.22)

*Шаг десятый.*

Выдать результат – гипотезу (-ы) *Gm* и вызывать подсистему, которая выдаёт протокол объяснений с описанием всех выводов, которые сделала ЭС.

# Реализация Нейлоровской СППР в выборе специальности

Для реализации системы поддержки принятия решений в сфере профориентации необходимо написание технического задания, в котором будут сформулированы технико-эксплуатационные требования для создания СППР в выборе специальности для поступления в ВУЗ. Техническое задание представлено в Приложении 1.

Следующим шагом создания будет проектирование основы СППР – базы знаний. Проектирование базы знаний является одной из главных составляющих всей работы над системой, ведь опираясь на неё, мы будем получать какие-либо результаты.

## Формулировка задачи

Задачей выполнения работы является:

1. Проанализировать существующие экспертные системы в сфере профориентации;
2. Написать техническое задание на создание СППР в выборе специальности для поступления в ВУЗ;
3. Спроектировать базу знаний для создаваемой системы;
4. Разработать спроектированную СППР.

## Проектирование базы знаний

Для построения базы знаний необходимо выбрать методики профориентационного тестирования.

Первой из выбранных методик стала методика Йоваши. В ней отражается принадлежность склонностей испытуемого к сферам профессиональной деятельности:

1. Сфера искусства;
2. Сфера работы с людьми;
3. Сфера технических интересов;
4. Сфера умственного труда;
5. Сфера физического труда;
6. Сфера материальных интересов.

Данная методика состоит из 30 вопросов, в каждом из которых надо выбрать одно понятие из двух представленных. В приложении 2 представлены все вопросы и обработка результатов методики Йоваши.

На этом работа системы не закончится, следом за этой методикой будет выполняться методика Филимоновой О.Г., и уже здесь будет работать Нейлоровская система. Так как эта методика состоит из 145 вопросов, то к каждому из них будет прикреплена своя вероятность. Вопросы делятся на 29 направлений (таблица 1), каждое из которых будет пояснено своими профессиями.

Таблица 1. Направления в методике Филимоновой О.Г.

|  |  |
| --- | --- |
| **№** | **Направление** |
| 1. | Биология |
| 2. | География |
| 3. | Геология |
| 4. | Медицина |
| 5. | Легкая и пищевая промышленность |
| 6. | Физика |
| 7. | Химия |
| 8. | Техника, механика |
| 9. | Электротехника, радиотехника, электроника |
| 10. | Обработка материалов (дерево, металл и прочее) |
| 11. | Информационные технологии |
| 12. | Строительство |
| 13. | Транспорт, авиация, морское дело |
| 14. | Психология |
| 15. | Военные специальности |
| 16. | История |
| 17. | Литература, филология |
| 18. | Журналистика, связи с общественностью, реклама |
| 19. | Социология, философия |
| 20. | Педагогика |
| 21. | Право, юриспруденция |
| 22. | Сфера обслуживания |
| 23. | Математика |
| 24. | Экономика, бизнес |
| 25. | Иностранные языки, лингвистика |
| 26. | Изобразительное искусство |
| 27. | Сценическое искусство |
| 28. | Музыка |
| 29. | Физкультура и спорт |

Здесь подсчитываются баллы по тому, как привлекает или отвергает от себя профессия:

1. если очень нравится: +2 балла
2. если нравится, но не так сильно привлекает: +1 балл
3. если нейтрально относитесь: 0 баллов
4. не нравится профессия: -1 балл
5. профессия приводит к отвращению: -2 балла.

Изначально будут задаваться вопросы по каждому из направлений, и сразу будет вестись суммарный подсчёт. По мере набора баллов, будут выбираться те вопросы, которые относятся к приоритетному направлению. В приложении 3 показаны вопросы методики Филимоновой О.Г..

Так как точность работы системы определяет Нейлоровский подход, а его основой является теория вероятностей, то для каждой гипотезы, каждого свидетельства необходимо расставить вероятности – априорные вероятности гипотез, вероятность выполнения свидетельства для гипотезы, вероятность выполнения свидетельства при невыполнении гипотезы. Итак, получается 29 вероятностей для гипотез, и по 145 вероятностей двух видов для каждого из свидетельств. На рисунке 3.1. и 3.2. отображены занесённые в БД вероятности.

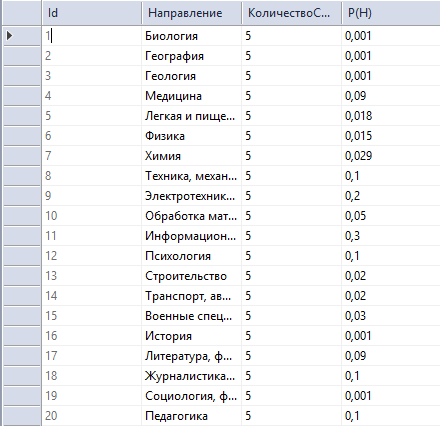


Рисунок 3.1 Априорные вероятности гипотез.

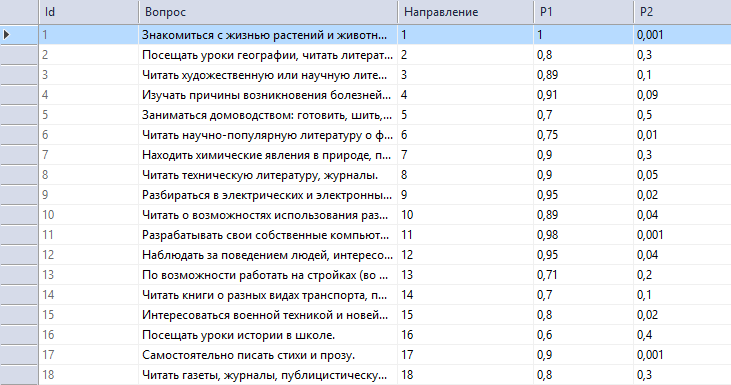


Рисунок 3.2 Вероятности свидетельств при выполнении/невыполнении гипотез (P1 и P2)

База знаний должна быть построена так, что на основе её знаний, можно было сделать рассуждения, то здесь можно пояснить следующим образом. При выдаче вопроса (свидетельства гипотезы) наши вероятности, цены пересчитываются, и на их основе выводится либо следующий вопрос, либо уже сам «диагноз» - заключение.

## Программная реализация

Суть программной реализации состояла в организации работы методик Йоваши и Филимоновой. Для этого изначально была заполнена база данных MS Sql Server, которая используется по умолчанию в Microsoft Visual Studio. Были заполнены 5 таблиц: «Вопросы Йоваши» (рисунок 3.3), «Ответы Йоваши» (рисунок 3.4) , «Результаты Йоваши» (рисунок 3.5), «Вопросы Филимоновой О.Г.» (рисунок 3.6), «Направления Филимоновой О.Г.» (рисунок 3.7).

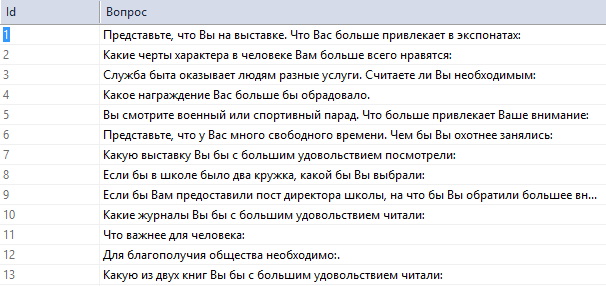


Рисунок 3.3 Таблица «Вопросы»

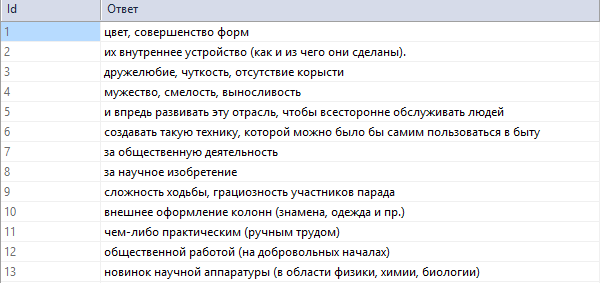


Рисунок 3.4. Таблица «Ответы»

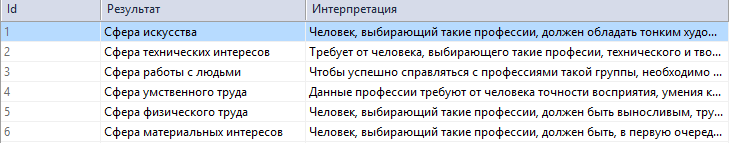


Рисунок 3.5 Таблица «Результаты»

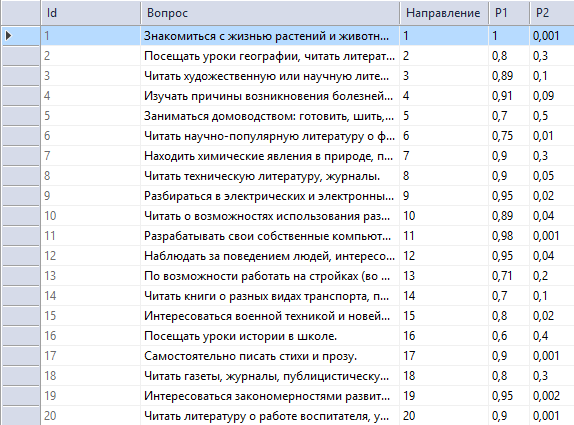


Рисунок 3.6 Вопросы Филимоновой О.Г.

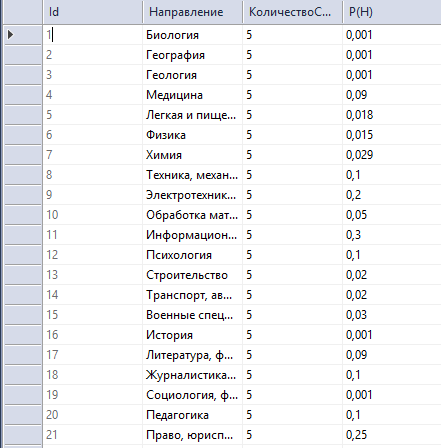


Рисунок 3.7 Направления Филимоновой О.Г.

Затем было создано подключение к базе данных (рисунок 3.8).

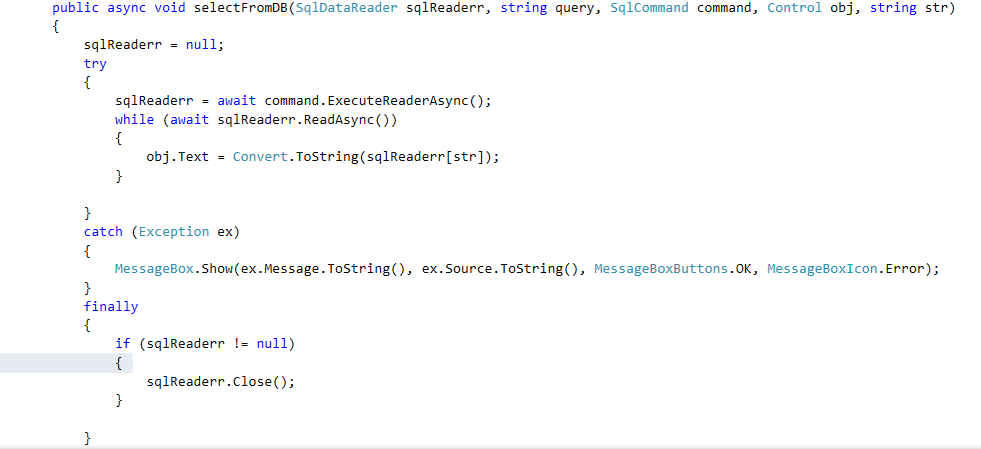


Рисунок 3.8 Функция подключения к базе данных

Чтобы получить данные из базы данных необходимо написать SQL-запросы. Запросы к разным таблицам выглядят аналогично. На рисунке 6 показан запрос к таблице «Результаты».

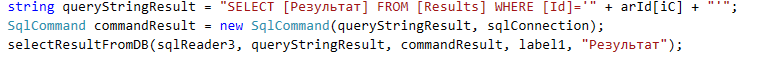


Рисунок 3.9 SQL-запрос к таблице «Результаты»

Реализация системы графически расположена на 2 формах. На первой будет реализована методика Йоваши, на второй методика Филимоновой. При запуске программы выходит форма с приветствием (рисунок 3.9).

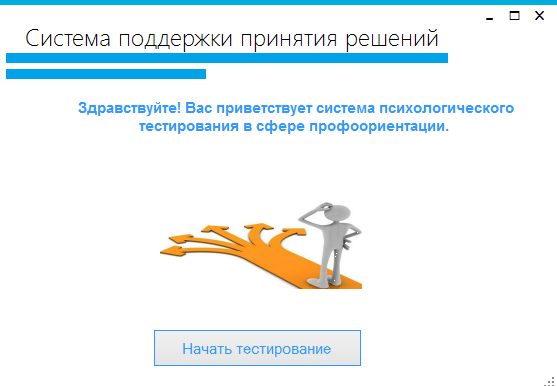


Рисунок 3.10 Форма СППР с приветствием

Графический интерфейс СППР работы с пользователем выглядит следующим образом (рисунок 3.10).

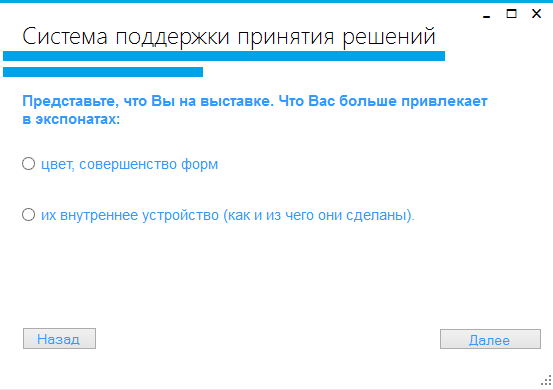


Рисунок 3.10 Графический интерфейс СППР работы с пользователем

Результат работы системы по методике Йоваши представлен на рисунке 3.11. Далее пользователю система предлагает выбор – либо закончить на методике Йоваши тестирование, либо продолжить уже другое тестирование по методике Филимоновой О.Г..Если пользователя удовлетворит результат, то второе тестирование можно не проходит. Напротив, если не удовлетворит результат, то предлагается «Продолжить тестирование».

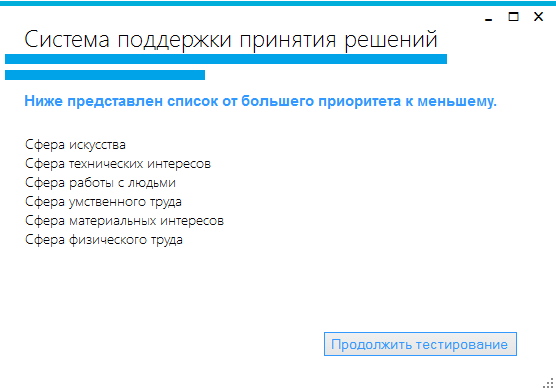


Рисунок 3.11 Результаты работы системы по методике Йоваши

Работа СППР по методике Филимоновой устроена на Нейлоровском подходе, поэтому, на второй форме появляется элемент TrackBar, благодаря которому пользователем будет устанавливаться оценка свидетельству. Слева от элемента TrackBar расположен label, в котором выходит значение ползунка на шкале TrackBar. Весь алгоритм работы описан в подразделе 2.2.5.

Перед прохождением и во время тестирования на форме присутствует памятка, в которой отражено пояснение оценивания свидетельства. На рисунке 3.12 показана форма, с помощью которой реализуется методика Филимоновой.

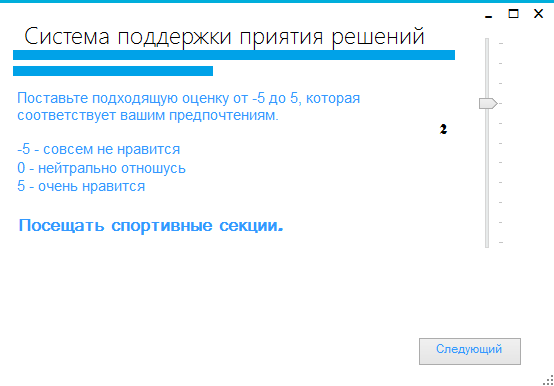


Рисунок 3.12 Форма методики Филимоновой

При работе системы по методике Филимоновой, пользователю СППР задаст максимум лишь часть, максимум вопросов 20, хотя в системе их 145. Больше вопросов система задаст лишь тогда, когда пользователь совсем не определён, не знает свои способности, интересуется всем-всем. Таких людей единицы.

На рисунке 3.14 показан вывод результатов работы системы. На рисунке видно, что система вывела 2 конечных результата. Направления медицина и биология связаны между собой, а места их распределились благодаря вероятностному подходу.

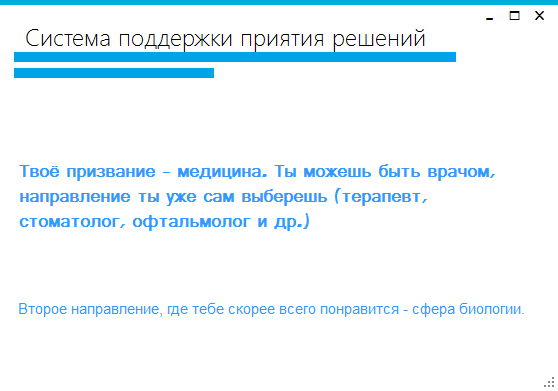


Рисунок 3.13 Вывод результатов работы СППР

# ЗАКЛЮЧЕНИЕ

На сегодняшний день работа в сфере профориентации набирает свои обороты. Создаются новые методики тестирований для молодых людей, разрабатываются различные системы поддержки принятия решений. Но, всё же, системы, которая была бы одной из лучших работ, в настоящее время не существует.

Подводя итог по разработке системы поддержки принятия решения в выборе специальности для поступления в ВУЗ, сделан вывод таков, что созданная СППР является точной в выводе результатов, так как в основе её лежит Нейлоровский подход, в котором основой является формула Байеса. Благодаря вероятностному подходу, предложенному Нейлором алгоритму, можно выделить следующие достоинства метода:

1. Результаты работы точны;
2. Система не перебирает всю свою базу знаний, а лишь ту часть, свидетельства которой попадают под условия выбора по вероятностям
3. В результате работы выводится набор более вероятных «диагнозов».

Системой могут пользоваться молодые люди, которые хотят поступить в учебное заведение, но не знают, в какую сферу им податься.

Особенно стоит отметить тот факт, что внедрение системы способно значительно повлиять на выбор направления или профессии молодым людям. Расширение системы, добавление новых методик подразумевается в дальнейшей разработке данной СППР.

# Выводы

В данной выпускной квалификационной работе выполнены все задачи, которые были поставлены в начале работы:

1. Произведены обзор и анализ литературы о системах поддержки принятия решений, об экспертных СППР, Нейлоровских СППР. Также была найдена информация о существующих в настоящее время системах в сферах профессиональной ориентации.
2. Написано техническое задание на создание СППР в выборе профессии для поступления в ВУЗ.
3. Сформирована база знаний из двух методик Йоваши и Филимоновой О.Г..
4. Программно реализована СППР, в основе которого лежит Нейлоровский подход.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Горбачёва С.М., Стрижко И.И. Профессиональная ориентация учащихся // Молодой учёный. – 2015. – с. 778-781.
2. Кан Л.В., Кузнецова Ю.М. Экспертные системы в области психодиагностики. // Искусственный интеллект и принятие решений 2/2010.
3. Кравченко Т.К. Экспертная система поддержки принятия решений.

// Экономика знаний. Открытое образование. - 6/2010. – с. 147-156.

1. Тамахина О.В. Система профессиональной ориентации школьников как фактор осознанного профессионального выбора (на примере разработки и реализации социального проекта «Твой профиль»).
2. Черноруцкий И.Г. Нейлоровские диагностирующие системы. // Методы принятия решений. – 2005.- с. 309-320.
3. [Электронный ресурс] // Профориентация учащихся на современном этапе развития школы. URL: <http://festival.1september.ru/articles/603612/> (дата обращения: 02.05.2017)
4. [Электронный ресурс] // Методика Л.А. Йоваши.

URL: <http://testoteka.narod.ru/prof/1/12.html> (дата обращения: 05.05.2017)

1. [Электронный ресурс] // «Карта интересов» модификация Филимоновой О.Г. URL: <http://testoteka.narod.ru/prof/1/07.html> (дата обращения: 05.05.2017)
2. [Электронный ресурс] // Системы поддержки принятия решений. URL: <http://bourabai.ru/tpoi/dss.htm> (дата обращения: 10.05.2017)
3. [Электронный ресурс] // Классификация методов поддержки принятия решений. URL: <http://www.ipiran.ru/niap/pages/st_19.pdf> (дата обращения: 10.05.2017)
4. [Электронный ресурс] // Системы поддержки принятия решений. URL: <http://rema44.ru/resurs/study/sppr/sppr01_intro.pdf> (дата обращения: 11.05.2017)
5. [Электронный ресурс] // Методы принятия управленческих решений. URL:<http://studme.org/31908/menedzhment/informatsionnye_sistemy_operativnogo_upravleniya> (дата обращения: .05.2017)
6. [Электронный ресурс] // Модели и методы принятия решений в условиях неопределённости. URL: http://edu.sernam.ru/book\_mmn.php?id=4 (дата обращения: 12.05.2017)
7. [Электронный ресурс] // Интерактивная профориентационная экспертная система. URL: <http://teletesting.ru/modules/tests/?pg=5> (дата обращения: 13.05.2017)
8. [Электронный ресурс] // Комплекс тестирования "ПРОФОРИЕНТАТОР": психометрические и прагматические свойства теста. URL: <http://teletesting.ru/modules/articles/index.php?op=viewarticle&artid=9> (дата обращения: 13.05.2017)
9. [Электронный ресурс] // Методики и алгоритмы поддержки принятия решений в системе профессионального самоопределения. URL: <http://www.dslib.net/upravlenie-socsystem/metodiki-i-algoritmy-podderzhki-prinjatija-reshenij-v-sisteme-professionalnogo.html> (дата обращения: 15.05.2017)
10. [Электронный ресурс] // Информационная система поддержки принятия решений для выпускников бакалавриата. URL: <http://emag.iis.ru/arc/infosoc/emag.nsf/BPA/ac78b3849290b4be44257d6400297cff> (дата обращения: 16.05.2017)
11. [Электронный ресурс] // Структура систем поддержки принятия решений. URL: <http://mydocx.ru/7-29210.html> (дата обращения 30.05.2017).
12. [Электронный ресурс] // Системы поддержки принятия решений (СППР). Общая концепция СППР. URL:<http://economics.studio/voprosyi-menedjmenta-obschie/sistemyi-podderjki-prinyatiya-resheniy-sppr-66880.html> (дата обращения 06.06.2017)

# Приложение 1

**Технические задание на разработку системы поддержки принятия решения в выборе специальности для поступления в ВУЗ**

1. **Общие сведения**
   1. ***Наименование программы***
      1. *Полное наименование программы*

Полное наименование: Система поддержки принятия решения в выборе специальности для поступления в ВУЗ.

* + 1. *Краткое наименование: СППР, система.*
  1. ***Наименование организаций – Заказчика и Разработчика***
     1. *Заказчик:*

Заказчик: ПНИПУ, кафедра Вычислительной математики и механики.

Адрес фактический: г. Пермь, ул. Профессора Поздеева, д.13

Телефон: 8-800-555-35-35

* + 1. *Разработчик:*

Разработчик: Боталова Юлия Евгеньевна

Адрес фактический: г. Пермь, ул. Профессора Дедюкина, д.18-4, к.402

Телефон: 8-950-475-52-56

* 1. ***Основания для проведения работ***

Работа выполняется на основании договора №305 от 30 сентября 2016 года между ПНИПУ, кафедрой ВМиМ и Боталовой Юлией Евгеньевной.

* 1. ***Плановые сроки начала и окончания работы***

Начало: 30.09.2016 г.

Окончание: 21.06.2017 г.

* 1. ***Источники и порядок финансирования***

Требования не предъявляются.

1. **Назначение и цели создания программы.** 
   1. ***Назначение программы***

СППР предназначена для повышения оперативности и качества принимаемого решения в выборе профессии для выпускников школы, абитуриентов.

Основным назначением СППР является автоматизация разных психологических методик с применением Нейлоровских диагностирующих систем.

* 1. ***Цель создания программы***

Цели создания системы:

1. Сокращение времени на выполнение, сбора информации и вывода результатов.
2. Повышение качества (полноты и достоверности) результатов
3. **Требования к системе**
   1. ***Требования к системе в целом***
      1. ***Требования к структуре и функционированию программы***
         1. ***Требования по диагностированию программы***

Требования не предъявляются.

* + - 1. ***Перечень подсистем, их назначение и основные характеристики***

1. Подсистема сбора и обработки информации.
2. Подсистема хранения данных.
   * + 1. ***Требования к характеристикам взаимосвязей со смежными системами***

Источником данных для системы должна быть СУБД Microsoft SQL Server Compact.

* + - 1. ***Требования к режимам функционирования системы***

Система должна поддерживать следующий режим функционирования системы:

* Основной режим, в котором подсистемы СППР выполняют все свои функции.
  + 1. ***Требование к численности и квалификации персонала:*** 
       1. ***Требования к численности персонала***

В состав персонала, необходимого для обеспечения эксплуатации программы необходимо выделение ответственных лиц:

1. Руководитель эксплуатирующего подразделения:

На всём протяжении функционирования СППР обеспечивает общее руководство группой сопровождения.

1. Администратор подсистемы сбора, обработки и загрузки данных:

На всём протяжении функционирования СППР обеспечивает подготовку и загрузку данных из хранилища данных.

1. Администратор подсистемы хранения данных:

На всём протяжении функционирования СППР обеспечивает распределение дискового пространства, оптимизацию производительности.

* + - 1. ***Требования к квалификации персонала***

К квалификации персонала, эксплуатирующего СППР, предъявляются следующие требования:

* Конечный пользователь – знания и навыки работы в оконном интерфейсе;
* Администратор подсистемы сбора, обработки и загрузки данных:

Знание методологии проектирования хранилищ данных; знание СУБД; знание языка запросов SQL.

* Администратор подсистемы хранения данных:

Знания СУБД.

* + - 1. ***Требования к режимам работы персонала***

Система реализуется на персональных компьютерах, поэтому требования к организации труда и режима отдыха при работе с программой должны устанавливаться с учетом этого типа вычислительной техники.

* + 1. ***Показатели назначения***

Время отклика системы не должно превышать 5 сек.

* + 1. ***Требования к надежности***

Надёжность должна обеспечиваться за счёт соблюдения правил эксплуатации и технического обслуживания программно-аппаратных средств.

* + 1. ***Требование к эргономике и технической эстетике:***

В части внешнего оформления должно быть обеспечено наличие русскоязычного интерфейса пользователя.

В части оформления:

* Интерфейсы систем должны быть типизированы
* Должно быть обеспечено наличие русскоязычного интерфейса пользователя

В части диалога с пользователем:

* При возникновении ошибок в работе программы на экран монитора должно выводиться сообщение с наименованием ошибки.
  + 1. ***Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов программы***

Условия эксплуатации, а также виды и периодичность обслуживания технических средств системы должны соответствовать требованиям по эксплуатации, техническому обслуживанию, ремонту и хранению, изложенным в документации завода-изготовителя (производителя) на них.

* + 1. ***Требования к защите информации от несанкционированного доступа***
* Защита информации об используемых и выдаваемых данных обеспечивается операционной системой и техническими средствами заказчика
* Программно-аппаратные средства защиты не должны существенно ухудшать основные функциональные характеристики СППР (надёжность, быстродействие).
  + 1. ***Требования по сохранности информации при авариях***

Требования не предъявляются.

* + 1. ***Требования к стандартизации и унификации***

Программа должна работать под управлением операционной системы Windows 7, 8, 10.

* + 1. ***Дополнительные требования***

Требования не предъявляются.

* + 1. ***Требования безопасности***

Требования не предъявляются.

* 1. ***Требования к функциям (задачам), выполняемым системой:***
* При возникновении ошибок происходит вывод определённого окна с указанием ошибки
* Выполнение вывода результата Нейлоровской диагностирующей системы.
  1. ***Требования к видам обеспечения.***
     1. ***Требования к видам математического обеспечения:***

Требования не предъявляются.

* + 1. ***Требования к информационному обеспечению***
       1. Требования к составу, структуре и способам организации данных в системе.

Структура хранения данных СППР должна состоять из следующих областей:

* Область временного хранения данных
* Область постоянного хранения данных
  + - 1. Требования к информационному обмену между компонентами системы

Требования не предъявляются.

* + - 1. Требования к информационной совместимости со смежными системами

Требования не предъявляются.

* + - 1. Требования по использованию классификаторов, унифицированных документов и классификаторов

Требования не предъявляются.

* + - 1. Требования по применению систем управления базами данных.

Для реализации хранения данных должна использоваться СУБД MS SQL Server Compact.

* + - 1. Требования к структуре процесса сбора, обработки, передачи данных в системе и представлению данных

Требования не предъявляются.

* + - 1. Требования к защите данных от разрушений при авариях и сбоях в электропитании программы

Требования не предъявляются.

* + 1. ***Требования к лингвистическому обеспечению***
* При реализации программы должен применяться язык высокого уровня C#.
* При написании запросов к базе данных должен применяться язык SQL.
* Для организации диалога программы с пользователем должен быть реализован интерфейс на русском языке.
  + 1. ***Требования к программному обеспечению***

Программа должна иметь возможность установки на ОС Windows 7 и более поздние версии.

* + 1. ***Требования к техническому обеспечению***

Требования не предъявляются.

* + 1. ***Требования к метрологическому обеспечению***

Требования не предъявляются.

* + 1. ***Требования к организационному обеспечению***

Требования не предъявляются.

* + 1. ***Требования к методическому обеспечению***

В состав нормативно-правового и методического обеспечения системы должны входить следующий стандарт ГОСТ 34.602-89. Техническое задание на создание автоматизированной системы.

* + 1. ***Требования к патентной чистоте***

Работа программы происходит на операционной системе MS Windows 8.1, покупка лицензии на данное программное обеспечение выполняется заказчиком.

1. **Состав и содержание работ по созданию программы**

Работы по созданию программы выполняются в 5 этапов:

* Написание технического задания (срок: 14.11.16)
* Проектирование базы знаний (БЗ) (срок: 21.04.17)
* Разработка интерфейсной части (срок: 10.06.17)
* Реализация (срок: 15.06.17)
* Тестирование (срок: 20.06.17)

Конкретные сроки выполнения стадий и этапов разработки и создания программы определяются планом выполнения работ, являющимся неотъемлемой частью Договора на выполнение работ по настоящему техническому заданию.

1. **Порядок контроля и приёмки системы**
   1. ***Виды и объём испытаний программы***

Виды и объём испытаний не предусмотрены.

* 1. ***Требования к приемке работ по стадиям***

Приемка работ проводится комиссией в составе: представитель от заказчика и один представитель от исполнителя.

1. **Требования к составу и содержанию работ по подготовке объекта к автоматизации к вводу программы в действие**

Требования не предъявляются.

1. **Требования к документированию**

Требования не предъявляются.

1. **Источники разработки**

Настоящее Техническое Задание разработано на основе следующих документов и информационных материалов:

1. Договор №305 от 30 сентября 2016 года между ПНИПУ, кафедрой ВМиМ и Боталовой Юлией Евгеньевной.

ГОСТ 34.602-89. Техническое задание на создание автоматизированной системы.

# Приложение 2

**Вопросы и обработка результатов методики Йовайши**

1. Представьте, что Вы на выставке. Что Вас больше привлекает в экспонатах:

а) цвет, совершенство форм;

б) их внутреннее устройство (как и из чего они сделаны).

1. Какие черты характера в человеке Вам больше всего нравятся:   
   а) дружелюбие, чуткость, отсутствие корысти;

б) мужество, смелость, выносливость.

1. Служба быта оказывает людям разные услуги. Считаете ли Вы необходимым:   
   а) и впредь развивать эту отрасль, чтобы всесторонне обслуживать людей;   
   б) создавать такую технику, которой можно было бы самим пользоваться в быту.
2. Какое награждение Вас больше бы обрадовало?

а) за общественную деятельность;

б) за научное изобретение.

1. Вы смотрите военный или спортивный парад. Что больше привлекает Ваше внимание:

а) сложность ходьбы, грациозность участников парада;

б) внешнее оформление колонн (знамена, одежда и пр.).

1. Представьте, что у Вас много свободного времени. Чем бы Вы охотнее занялись:   
   а) чем-либо практическим (ручным трудом);

б) общественной работой (на добровольных началах).

1. Какую выставку Вы бы с большим удовольствием посмотрели:   
   а) новинок научной аппаратуры (в области физики, химии, биологии);   
   б) новых продовольственных товаров.
2. Если бы в школе было два кружка, какой бы Вы выбрали:

а) музыкальный;

б) технический.

1. Если бы Вам предоставили пост директора школы, на что бы Вы обратили большее внимание:

а) на сплоченность коллектива;

6) на создание необходимых удобств.

1. Какие журналы Вы бы с большим удовольствием читали:   
   а) литературно-художественные;

б) научно-популярные.

1. Что важнее для человека:

а) создавать себе благополучный, удобный быт;

б) жить без некоторых удобств, но иметь возможность пользоваться сокровищницей искусства, создавать искусство.

1. Для благополучия общества необходимо:

а) техника;

б) правосудие.

1. Какую из двух книг Вы бы с большим удовольствием читали:

а) о развитии науки в нашей стране;

б) о достижениях спортсменов нашей страны.

1. В газете две статьи разного содержания. Какая из них вызвала бы у Вас большую заинтересованность:

а) о машине нового типа

б) о новой научной теории.

1. Какая из двух работ на свежем воздухе Вас больше бы привлекала:   
   а) работа, связанная с постоянными передвижениями (агроном, лесничий, дорожный мастер);

б) работа с машинами.

1. Какая, на Ваш взгляд, задача школы важнее:   
   а) подготовить учащихся к работе с людьми, чтобы они могли помогать другим, создавать материальные блага;

б) подготовить учащихся к практической деятельности, к умению создавать материальные блага.

1. Что, на Ваш взгляд, следует больше ценить у участников самодеятельности:   
   а) то, что они несут людям искусство и красоту;

б) то, что они выполняют общественно полезную работу.

1. Какая, на Ваш взгляд, область деятельности человека в дальнейшем будет иметь доминирующее значение:

а) физика;

б) физическая культура.

1. Что обществу принесет больше пользы:

а) забота о благосостоянии граждан;

б) изучение поведения людей.

1. Какого характера научную работы Вы бы выбрали:

а) работу с книгами в библиотеке;

б) работу на свежем воздухе в экспедиции.

1. Представьте, что Вы - профессор университета. Чему Вы отдали бы предпочтение в свободное от работы время:

а) занятиям по литературе;

б) опытам по физике, химии.

1. Вам предоставляется возможность совершить путешествие в разные страны. В качестве кого Вы охотнее поехали бы:

а) как известный спортсмен на международные соревнования;   
б) как известный специалист по внешней торговле с целью покупки необходимых товаров для нашей страны.

1. Какие лекции слушали бы Вы с большим удовольствием:   
   а) о выдающихся художниках;

б) о выдающихся ученых.

1. Что Вас больше привлекает при чтении книг:

а) яркое изображение смелости и храбрости героев;

б) прекрасный литературный стиль.

1. Вам предоставляется возможность выбора профессии. Какой из них Вы бы отдали предпочтение:

а) работе малоподвижной, но связанной с созданием новой техники;   
б) физической культуре или другой работе, связанной с движением.

1. Какими выдающимися учеными Вы больше интересуетесь:   
   а) Поповым и Циолковским;

б) Менделеевым и Павловым.

1. Как Вам кажется, на что следовало бы в школе обратить большее внимание:   
   а) на спорт, так как это нужно для укрепления здоровья;

б) на успеваемость учащихся, так как это необходимо для будущего.

1. Что бы Вас больше заинтересовало в печати:

а) сообщение о состоявшейся художественной выставке;

б) известие о прошедшем митинге в защиту прав человека.

1. Если бы Вам представилась возможность занять определенный пост, какой бы Вы выбрали:

а) главного инженера завода;

б) директора универмага.

1. Как Вы считаете, что важнее:

а) много знать;

б) создавать материальные блага.

**Обработка результатов и интерпретация**

В заполненном листе ответов в каждом столбце подсчитывается количество баллов. Результаты записываются под каждым столбцом, который соответствует определенной сфере профессиональных интересов: 1-й столбец - сфера искусства

2-й столбец - сфера технических интересов

3-й столбец - сфера работы с людьми

4-й столбец - сфера умственного труда

5-й столбец - сфера физического труда

6-й столбец - сфера материальных интересов

Анализируя полученные данные, необходимо выделить столбцы, содержащие наибольшее количество баллов, и эти сферы деятельности можно считать предпочитаемыми испытуемым. Столбцы, содержащие наименьшее количество баллов, отражают сферы деятельности, отвергаемые испытуемым.

# Приложение 3

**Вопросы и обработка результатов методики «Карта интересов» Филимоновой О.Г.**

1. Знакомиться с жизнью растений и животных.
2. Посещать уроки географии, читать литературу по географии.
3. Читать художественную или научную литературу о геологических экспедициях.
4. Изучать причины возникновения болезней и пути их лечения.
5. Заниматься домоводством: готовить, шить, приводить в порядок мебель и одежду.
6. Читать научно-популярную литературу о физических открытиях.
7. Находить химические явления в природе, проводить опыты по химии.
8. Читать техническую литературу, журналы.
9. Разбираться в электрических и электронных схемах.
10. Читать о возможностях использования различных материалов (медь, олово, сосна и т.д.)
11. Разрабатывать свои собственные компьютерные программы.
12. Наблюдать за поведением людей, интересоваться причинами их поступков.
13. По возможности работать на стройках (во время летних каникул).
14. Читать книги о разных видах транспорта, профессиях, связанных с транспортными перевозками.
15. Интересоваться военной техникой и новейшими разработками в этой области.
16. Посещать уроки истории в школе.
17. Самостоятельно писать стихи и прозу.
18. Читать газеты, журналы, публицистическую литературу.
19. Интересоваться закономерностями развития общества.
20. Читать литературу о работе воспитателя, учителя, о деятельности педагогов.
21. Смотреть телепередачи о раскрытии преступлений.
22. Работать в сфере обслуживания.
23. Читать дополнительную литературу по математике.
24. Знакомиться с компьютерными программами по бухгалтерскому учету.
25. Посещать курсы иностранных языков.
26. Смотреть фильмы и читать литературу о художниках и их творчестве.
27. Знакомиться с жизнью выдающихся мастеров сцены, кино.
28. Участвовать в музыкальных смотрах-конкурсах.
29. Посещать спортивные секции.
30. Изучать ботанику, зоологию, биологию, экологию.
31. Знакомиться с географическими особенностями различных стран по описаниям и картам.
32. Посещать геологические музеи.
33. Посещать уроки анатомии и физиологии человека.
34. Знакомиться с работой предприятий легкой промышленности (швейное, обувное, текстильное и др.).
35. Изучать физические явления и законы.
36. Читать научно-популярную литературу об открытиях в области химии и деятельности выдающихся химиков.
37. Знакомиться с новейшими достижениями современной техники (смотреть телепередачи, искать статьи в журналах).
38. Разбираться в устройстве электроприборов, видео- и радиоаппаратуры и электрических машин.
39. Посещать кружки по обработке дерева, металлов.
40. Читать книги о программировании, журналы о новых компьютерных изобретениях и технологиях.
41. Читать научно-популярную литературу по психологии.
42. Выполнять наброски различных зданий, проектировать постройки.
43. Интересоваться историей развития разных видов транспорта.
44. Участвовать в военных сборах, играть в военные игры.
45. Читать книги по истории, узнавать новое об исторических событиях, деятелях науки и культуры.
46. Работать с литературными источниками, библиографическими справочниками.
47. Смотреть ежедневные выпуски новостей по телевизору, обсуждать со взрослыми и сверстниками проблемы общественной жизни.
48. Посещать факультативные занятия по обществознанию.
49. Давать объяснения товарищам, как выполнять учебное задание, если они не могут сделать его сами.
50. Участвовать в разработке уставов, положений, нормативных документов для школы.
51. Помогать клиентам выбирать покупку в магазине.
52. Читать научно-популярную литературу об открытиях в области математики, о жизни выдающихся математиков.
53. Интересоваться стоимостью товаров, пытаясь понять вопросы ценообразования, заработной платы, организации труда.
54. Учить второй иностранный язык дополнительно.
55. Читать книги об изобразительном искусстве.
56. Посещать творческие вечера-встречи с актерами.
57. Знакомиться с жизнью и творчеством выдающихся музыкантов, с вопросами теории музыкального искусства.
58. Играть в спортивные игры, участвовать в спортивных соревнованиях.
59. Собирать информацию о животных, растениях, микроорганизмах.
60. Смотреть телепередачи о разных странах.
61. Знакомиться и составлять описание геологических объектов (минералов, слоев земли и т.п.).
62. Читать и смотреть фильмы о врачах и достижениях в области медицины.
63. Знакомиться с новыми технологиями, используемыми в пищевой и легкой промышленности.
64. Проводить опыты по физике.
65. Знакомиться с новыми технологиями в химическом производстве, с получением новых веществ и материалов.
66. Разбирать, ремонтировать, интересоваться устройством различных механизмов (часов, бытовых приборов, швейных машин).
67. Ремонтировать радиоприборы и аппаратуру.
68. Смотреть телевизионные передачи об исследовании и применении различных веществ и материалов.
69. Заниматься информатикой дополнительно.
70. Посещать психологический кружок.
71. Участвовать в ремонтных работах дома, в школе, помогать в строительстве домов, построек.
72. Смотреть телепередачи, фильмы о профессиях, связанных с транспортными перевозками.
73. Ходить на экскурсии в места военной славы, посещать военные музеи.
74. Смотреть исторические фильмы, телепередачи.
75. Посещать факультативные занятия по литературе и русскому языку.
76. Обсуждать текущие дела и события в классе и школе.
77. Проводить опросы общественного мнения, брать интервью, выявлять тенденции в развитии разных явлений общественной жизни.
78. Проводить время с маленькими детьми (читать им книги, что-либо им рассказывать, помогать им в чем-либо).
79. Выяснять причины противоправного поведения людей.
80. Наблюдать за работой продавца, повара, официанта, читать литературу и смотреть телепередачи о конкурсах в сфере обслуживания.
81. Посещать дополнительные занятия по математике.
82. Читать сообщения на экономические темы в печати.
83. Читать книги на иностранном языке дополнительно.
84. Посещать художественные выставки.
85. Посещать театры.
86. Слушать оперную музыку, посещать концерты симфонической музыки.
87. Посещать спортивные соревнования, следить за их ходом, слушать и смотреть радио- и телепередачи о спорте.
88. Смотреть телепередачи о животных и растениях.
89. Самостоятельно составлять географические карты, накапливать и собирать различные географические сведения, изучать закономерности.
90. Участвовать в геологических экспедициях.
91. Знакомиться с работой медсестры, врача или фармацевта.
92. Опробовать рецепты приготовления пищи.
93. Заниматься в физическом кружке или посещать факультативные занятия по физике.
94. Решать сложные задачи по химии и участвовать в химических олимпиадах.
95. Моделировать самолеты, ракеты, корабли, машины и пр.
96. Проектировать и собирать электрические и радиоприборы (радиоаппаратуру, датчики температуры, движения, домофоны, системы сигнализации и т.п.).
97. Мастерить изделия из дерева, металла и других материалов, собирать детали своими руками.
98. Посещать уроки информатики в школе.
99. Интересоваться проявлениями характера человека, правилами взаимодействия людей.
100. Смотреть телевизионные передачи о строительстве («Квартирный вопрос» и т.д.).
101. Интересоваться новыми разработками в мире транспорта (новинки автомобилестроения, новые изобретения в авиации, в железнодорожном транспорте и т.п.)
102. Читать книги, смотреть фильмы на военную тему, знакомиться с историей крупных сражений, войн и судьбами великих полководцев.
103. Посещать исторические музеи, ездить на экскурсии по историческим местам.
104. Писать классные или домашние сочинения по литературе.
105. Взаимодействовать с людьми: убеждать, разъяснять, организовывать и т.п.
106. Читать литературу по философии, социологии.
107. Выполнять работу воспитателя, замещать уроки в младших классах.
108. Читать юридическую литературу, интересоваться историей права.
109. Обеспечивать семью продуктами, организовывать питание во время похода, поездки, путешествия.
110. Решать сложные задачи по математике.
111. Интересоваться вопросами экономической географии.
112. Смотреть фильмы на иностранном языке без перевода и пытаться понять их.
113. Рисовать дома для удовольствия.
114. Выступать на сцене перед зрителями.
115. Заниматься в музыкальной школе, студии, в хоре, музыкальном кружке.
116. Читать книги о здоровом образе жизни, о спорте, выдающихся спортсменах.
117. Разводить растения, ухаживать за животными, посещать выставки и участвовать в них.
118. Участвовать в географических экспедициях, походах.
119. Собирать коллекции минералов.
120. Ухаживать за больными, оказывать им помощь.
121. Придумывать новые рецепты приготовления пищи, конструировать новые модели одежды.
122. Решать сложные задачи по физике, участвовать в физических олимпиадах.
123. Знакомиться с разными возможностями применения химических знаний (фармацевтика, криминалистика, промышленность и т.п.)
124. Разбираться в технических чертежах и схемах, чертить или составлять чертежи самому.
125. Читать и смотреть телепередачи о достижениях в области электроники и радиотехники.
126. Знакомиться с информацией о новых технологиях в области материаловедения.
127. Работать на компьютере, использовать Интернет в целях поиска дополнительной информации.
128. Помогать знакомым преодолевать трудности, решать жизненные проблемы, выслушивать, успокаивать.
129. Читать книги о строительных работах («Как построить дом?» и т.д.)
130. Заниматься в клубах авиамоделирования, автолюбителей, парусного моделирования, быть членом дайвинг-клуба.
131. Встречаться с участниками военных действий, слушать их рассказы.
132. Знакомиться с историческими закономерностями, посещать олимпиады по истории.
133. Читать книги или смотреть передачи о поэтах и писателях.
134. Писать статьи в газеты, журналы.
135. Интересоваться историей философской мысли.
136. Обсуждать с кем-либо вопросы воспитания детей и подростков.
137. Читать книги, смотреть фильмы о работе милиции.
138. Заботиться о порядке в вещах, о красивом виде помещения, в котором учитесь, работаете, живете.
139. Участвовать в математических олимпиадах.
140. Вести расчеты своих доходов, расходов, а также доходов и расходов своей семьи.
141. Общаться с иностранцами на разных языках, работать переводчиком.
142. Посещать студию изобразительного искусства, художественную школу.
143. Заниматься в театральной студии.
144. Слушать классическую музыку.
145. Регулярно смотреть Олимпийские игры, спортивные соревнования.

Таблица 2. Обработка результатов методики Филимоновой О.Г.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№** | **Направление** | **Номера вопросов** | | | | |
|  | Биология | 1 | 30 | 59 | 88 | 117 |
|  | География | 2 | 31 | 60 | 89 | 118 |
|  | Геология | 3 | 32 | 61 | 90 | 119 |
|  | Медицина | 4 | 33 | 62 | 91 | 120 |
|  | Легкая и пищевая промышленность | 5 | 34 | 63 | 92 | 121 |
|  | Физика | 6 | 35 | 64 | 93 | 122 |
|  | Химия | 7 | 36 | 65 | 94 | 123 |
|  | Техника, механика | 8 | 37 | 66 | 95 | 124 |
|  | Электротехника, радиотехника, электроника | 9 | 38 | 67 | 96 | 125 |
|  | Обработка материалов (дерево, металл и прочее) | 10 | 39 | 68 | 97 | 126 |
|  | Информационные технологии | 11 | 40 | 69 | 98 | 127 |
|  | Психология | 12 | 41 | 70 | 99 | 128 |
|  | Строительство | 13 | 42 | 71 | 100 | 129 |
|  | Транспорт, авиация, морское дело | 14 | 43 | 72 | 101 | 130 |
|  | Военные специальности | 15 | 44 | 73 | 102 | 131 |
|  | История | 16 | 45 | 74 | 103 | 132 |
|  | Литература, филология | 17 | 46 | 75 | 104 | 133 |
|  | Журналистика, связи с общественностью, реклама | 18 | 47 | 76 | 105 | 134 |
|  | Социология, философия | 19 | 48 | 77 | 106 | 135 |
|  | Педагогика | 20 | 49 | 78 | 107 | 136 |
|  | Право, юриспруденция | 21 | 50 | 79 | 108 | 137 |
|  | Сфера обслуживания | 22 | 51 | 80 | 109 | 138 |
|  | Математика | 23 | 52 | 81 | 110 | 139 |
|  | Экономика, бизнес | 24 | 53 | 82 | 111 | 140 |
|  | Иностранные языки, лингвистика | 25 | 54 | 83 | 112 | 141 |
|  | Изобразительное искусство | 26 | 55 | 84 | 113 | 142 |
|  | Сценическое искусство | 27 | 56 | 85 | 114 | 143 |
|  | Музыка | 28 | 57 | 86 | 115 | 144 |
|  | Физкультура и спорт | 29 | 58 | 87 | 116 | 145 |

# Приложение 4

namespace СППР

{

partial class Form1

{

/// <summary>

/// Required designer variable.

/// </summary>

private System.ComponentModel.IContainer components = null;

/// <summary>

/// Clean up any resources being used.

/// </summary>

/// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>

protected override void Dispose(bool disposing)

{

if (disposing && (components != null))

{

components.Dispose();

}

base.Dispose(disposing);

}

#region Windows Form Designer generated code

/// <summary>

/// Required method for Designer support - do not modify

/// the contents of this method with the code editor.

/// </summary>

private void InitializeComponent()

{

System.ComponentModel.ComponentResourceManager resources = new System.ComponentModel.ComponentResourceManager(typeof(Form1));

this.label1 = new System.Windows.Forms.Label();

this.radioButton2 = new System.Windows.Forms.RadioButton();

this.radioButton3 = new System.Windows.Forms.RadioButton();

this.button1 = new System.Windows.Forms.Button();

this.button2 = new System.Windows.Forms.Button();

this.button3 = new System.Windows.Forms.Button();

this.panel1 = new System.Windows.Forms.Panel();

this.panel3 = new System.Windows.Forms.Panel();

this.radioButton1 = new System.Windows.Forms.RadioButton();

this.pictureBox1 = new System.Windows.Forms.PictureBox();

((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginInit();

this.SuspendLayout();

//

// label1

//

this.label1.Font = new System.Drawing.Font("Microsoft Himalaya", 11F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(204)));

this.label1.ForeColor = System.Drawing.SystemColors.Highlight;

this.label1.Location = new System.Drawing.Point(23, 91);

this.label1.Name = "label1";

this.label1.Size = new System.Drawing.Size(477, 42);

this.label1.TabIndex = 0;

this.label1.Text = "Вопрос";

//

// radioButton2

//

this.radioButton2.Font = new System.Drawing.Font("Microsoft Himalaya", 11F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));

this.radioButton2.ForeColor = System.Drawing.SystemColors.Highlight;

this.radioButton2.Location = new System.Drawing.Point(26, 191);

this.radioButton2.Name = "radioButton2";

this.radioButton2.Size = new System.Drawing.Size(474, 51);

this.radioButton2.TabIndex = 2;

this.radioButton2.TabStop = true;

this.radioButton2.UseVisualStyleBackColor = true;

this.radioButton2.Visible = false;

//

// radioButton3

//

this.radioButton3.Font = new System.Drawing.Font("Microsoft Himalaya", 11F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));

this.radioButton3.ForeColor = System.Drawing.SystemColors.Highlight;

this.radioButton3.Location = new System.Drawing.Point(26, 248);

this.radioButton3.Name = "radioButton3";

this.radioButton3.Size = new System.Drawing.Size(474, 55);

this.radioButton3.TabIndex = 3;

this.radioButton3.TabStop = true;

this.radioButton3.UseVisualStyleBackColor = true;

this.radioButton3.Visible = false;

this.radioButton3.CheckedChanged += new System.EventHandler(this.radioButton3\_CheckedChanged);

//

// button1

//

this.button1.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));

this.button1.ForeColor = System.Drawing.SystemColors.Highlight;

this.button1.Location = new System.Drawing.Point(26, 329);

this.button1.Name = "button1";

this.button1.Size = new System.Drawing.Size(75, 23);

this.button1.TabIndex = 4;

this.button1.Text = "Назад";

this.button1.UseVisualStyleBackColor = true;

this.button1.Visible = false;

this.button1.Click += new System.EventHandler(this.button1\_Click);

//

// button2

//

this.button2.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));

this.button2.ForeColor = System.Drawing.SystemColors.Highlight;

this.button2.Location = new System.Drawing.Point(425, 329);

this.button2.Name = "button2";

this.button2.Size = new System.Drawing.Size(75, 23);

this.button2.TabIndex = 5;

this.button2.Text = "Далее";

this.button2.UseVisualStyleBackColor = true;

this.button2.Visible = false;

this.button2.Click += new System.EventHandler(this.button2\_Click);

//

// button3

//

this.button3.Font = new System.Drawing.Font("Microsoft Sans Serif", 11F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));

this.button3.ForeColor = System.Drawing.SystemColors.Highlight;

this.button3.Location = new System.Drawing.Point(154, 329);

this.button3.Name = "button3";

this.button3.Size = new System.Drawing.Size(209, 38);

this.button3.TabIndex = 6;

this.button3.Text = "Начать тестирование";

this.button3.UseVisualStyleBackColor = true;

this.button3.Click += new System.EventHandler(this.button3\_Click);

//

// panel1

//

this.panel1.BackgroundImage = ((System.Drawing.Image)(resources.GetObject("panel1.BackgroundImage")));

this.panel1.Location = new System.Drawing.Point(7, 53);

this.panel1.Name = "panel1";

this.panel1.Size = new System.Drawing.Size(442, 10);

this.panel1.TabIndex = 7;

//

// panel3

//

this.panel3.BackgroundImage = ((System.Drawing.Image)(resources.GetObject("panel3.BackgroundImage")));

this.panel3.Location = new System.Drawing.Point(7, 69);

this.panel3.Name = "panel3";

this.panel3.Size = new System.Drawing.Size(200, 10);

this.panel3.TabIndex = 9;

//

// radioButton1

//

this.radioButton1.Font = new System.Drawing.Font("Microsoft Himalaya", 11F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));

this.radioButton1.ForeColor = System.Drawing.SystemColors.Highlight;

this.radioButton1.Location = new System.Drawing.Point(26, 136);

this.radioButton1.Name = "radioButton1";

this.radioButton1.Size = new System.Drawing.Size(474, 60);

this.radioButton1.TabIndex = 1;

this.radioButton1.TabStop = true;

this.radioButton1.UseVisualStyleBackColor = true;

this.radioButton1.Visible = false;

//

// pictureBox1

//

this.pictureBox1.BackgroundImage = ((System.Drawing.Image)(resources.GetObject("pictureBox1.BackgroundImage")));

this.pictureBox1.BackgroundImageLayout = System.Windows.Forms.ImageLayout.Stretch;

this.pictureBox1.Location = new System.Drawing.Point(161, 202);

this.pictureBox1.Name = "pictureBox1";

this.pictureBox1.Size = new System.Drawing.Size(202, 87);

this.pictureBox1.TabIndex = 10;

this.pictureBox1.TabStop = false;

//

// Form1

//

this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.ClientSize = new System.Drawing.Size(559, 391);

this.Controls.Add(this.pictureBox1);

this.Controls.Add(this.panel3);

this.Controls.Add(this.panel1);

this.Controls.Add(this.button3);

this.Controls.Add(this.button2);

this.Controls.Add(this.button1);

this.Controls.Add(this.radioButton3);

this.Controls.Add(this.radioButton2);

this.Controls.Add(this.radioButton1);

this.Controls.Add(this.label1);

this.Name = "Form1";

this.Text = "Система поддержки принятия решений";

this.TransparencyKey = System.Drawing.Color.Gray;

this.FormClosing += new System.Windows.Forms.FormClosingEventHandler(this.Form1\_FormClosing);

this.Load += new System.EventHandler(this.Form1\_Load);

((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndInit();

this.ResumeLayout(false);

}

#endregion

private System.Windows.Forms.Label label1;

private System.Windows.Forms.RadioButton radioButton2;

private System.Windows.Forms.RadioButton radioButton3;

private System.Windows.Forms.Button button1;

private System.Windows.Forms.Button button2;

private System.Windows.Forms.Button button3;

private System.Windows.Forms.Panel panel1;

private System.Windows.Forms.Panel panel3;

private System.Windows.Forms.RadioButton radioButton1;

private System.Windows.Forms.PictureBox pictureBox1;

}

}

using System;

using System.Collections.Generic;

using System.Data.OleDb;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using MetroFramework.Components;

using MetroFramework.Forms;

using MetroFramework.Fonts;

#pragma warning disable CS0105 // Using directive appeared previously in this namespace

using System.Collections.Generic;

#pragma warning restore CS0105 // Using directive appeared previously in this namespace

namespace СППР

{

public partial class Form1 : MetroForm

{

SqlConnection sqlConnection;

int iQ, iA1 = 1;

int iA2 = 2;

int sI, sT, sU, sM, sL, sF = 0;

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

label1.TextAlign = ContentAlignment.MiddleCenter;

label1.Text = "Здравствуйте! Вас приветствует система психологического тестирования в сфере профоориентации. ";

label1.Location = new Point(60, 96);

string connectionString = @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\user\Desktop\НИРс\СППР\СППР\СППР\Database1.mdf;Integrated Security=True; MultipleActiveResultSets=True";

sqlConnection = new SqlConnection(connectionString);

sqlConnection.Close();

}

public async void selectFromDB(SqlDataReader sqlReaderr, string query, SqlCommand command, Control obj, string str)

{

sqlReaderr = null;

try

{

sqlReaderr = await command.ExecuteReaderAsync();

while (await sqlReaderr.ReadAsync())

{

obj.Text = Convert.ToString(sqlReaderr[str]);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString(), ex.Source.ToString(), MessageBoxButtons.OK, MessageBoxIcon.Error);

}

finally

{

if (sqlReaderr != null)

{

sqlReaderr.Close();

}

}

}

public async void selectResultFromDB(SqlDataReader sqlReaderr, string query, SqlCommand command, Control obj, string str)

{

sqlReaderr = null;

try

{

sqlReaderr = await command.ExecuteReaderAsync();

while (await sqlReaderr.ReadAsync())

{

obj.Text = obj.Text + '\n' + '\n' + Convert.ToString(sqlReaderr[str]);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString(), ex.Source.ToString(), MessageBoxButtons.OK, MessageBoxIcon.Error);

}

finally

{

if (sqlReaderr != null)

{

sqlReaderr.Close();

}

}

}

private async void button3\_Click(object sender, EventArgs e)

{

pictureBox1.Visible = false;

iQ = 1;

button1.Visible = true;

button2.Visible = true;

button3.Visible = false;

radioButton1.Visible = true;

radioButton2.Visible = true;

label1.Location = new Point(42, 21);

label1.TextAlign = ContentAlignment.MiddleLeft;

string connectionString = @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\user\Desktop\НИРс\СППР\СППР\СППР\Database1.mdf;Integrated Security=True; MultipleActiveResultSets=True";

sqlConnection = new SqlConnection(connectionString);

await sqlConnection.OpenAsync();

SqlDataReader sqlReader = null;

string queryStringQuestion = "SELECT [Вопрос] FROM [Questions] WHERE [Id]=1";

SqlCommand commandQuestion = new SqlCommand(queryStringQuestion, sqlConnection);

selectFromDB(sqlReader, queryStringQuestion, commandQuestion, label1, "Вопрос");

SqlDataReader sqlReader1 = null;

string queryStringAnswer1 = "SELECT [Ответ] FROM [Answers] WHERE [Id]=1";

SqlCommand commandAnswer1 = new SqlCommand(queryStringAnswer1, sqlConnection);

selectFromDB(sqlReader1, queryStringAnswer1, commandAnswer1, radioButton1, "Ответ");

SqlDataReader sqlReader2 = null;

string queryStringAnswer2 = "SELECT [Ответ] FROM [Answers] WHERE [Id]=2";

SqlCommand commandAnswer2 = new SqlCommand(queryStringAnswer2, sqlConnection);

selectFromDB(sqlReader2, queryStringAnswer2, commandAnswer2, radioButton2, "Ответ");

}

private void Form1\_FormClosing(object sender, FormClosingEventArgs e)

{

if (sqlConnection != null && sqlConnection.State != ConnectionState.Closed)

{

sqlConnection.Close();

}

}

static double[] BubbleSort(double[] mas)

{

double temp;

for (int i = 0; i < mas.Length; i++)

{

for (int j = i + 1; j < mas.Length; j++)

{

if (mas[i] < mas[j])

{

temp = mas[i];

mas[i] = mas[j];

mas[j] = temp;

}

}

}

return mas;

}

private void button2\_Click(object sender, EventArgs e)

{

if (radioButton1.Checked)

{

if (iQ == 1 || iQ == 8 || iQ == 10 || iQ == 17 || iQ == 21 || iQ == 23 || iQ == 28)

{

sI = sI + 1;

}

if (iQ == 6 || iQ == 12 || iQ == 14 || iQ == 25 || iQ == 26)

{

sT = sT + 1;

}

if (iQ == 2 || iQ == 4 || iQ == 9 || iQ == 16)

{

sL = sL + 1;

}

if (iQ == 7 || iQ == 13 || iQ == 18 || iQ == 20 || iQ == 30)

{

sU = sU + 1;

}

if (iQ == 5 || iQ == 15 || iQ == 22 || iQ == 24 || iQ == 27)

{

sF = sF + 1;

}

if (iQ == 3 || iQ == 11 || iQ == 19 || iQ == 29)

{

sM = sM + 1;

}

}

if (radioButton2.Checked)

{

if (iQ == 5 || iQ == 11 || iQ == 24)

{

sI = sI + 1;

}

if (iQ == 1 || iQ == 3 || iQ == 8 || iQ == 15 || iQ == 29)

{

sT = sT + 1;

}

if (iQ == 6 || iQ == 17 || iQ == 19 || iQ == 23 || iQ == 28)

{

sL = sL + 1;

}

if (iQ == 4 || iQ == 10 || iQ == 14 || iQ == 21 || iQ == 26)

{

sU = sU + 1;

}

if (iQ == 2 || iQ == 13 || iQ == 18 || iQ == 20 || iQ == 25)

{

sF = sF + 1;

}

if (iQ == 7 || iQ == 9 || iQ == 16 || iQ == 22 || iQ == 17 || iQ == 30)

{

sM = sM + 1;

}

}

radioButton1.Checked = false;

radioButton2.Checked = false;

iQ = iQ + 1;

iA1 = iA1 + 2;

iA2 = iA2 + 2;

if (iQ == 31)

{

label1.AutoSize = true;

button2.AutoSize = true;

button1.Visible = false;

button2.Location = new Point(367, 272);

button2.Text = "Продолжить тестирование";

button3.Visible = false;

radioButton1.Visible = false;

radioButton2.Visible = false;

int[] arResults;

arResults = new int[6];

arResults[0] = sI;

arResults[1] = sT;

arResults[2] = sL;

arResults[3] = sU;

arResults[4] = sF;

arResults[5] = sM;

double[] arResultsId, arId, arIdEx;

arResultsId = new double[6];

arId = new double[6];

arIdEx = new double[6];

for (int n = 0; n <= 5; n++)

{

arResultsId[n] = arResults[n];

arIdEx[n] = -1;

}

BubbleSort(arResultsId);

int m, k = 0;

for (int i = 0; i <= 5; i++)

{

for (int j = 0; j <= 5; j++)

{

m = 0;

if (arResultsId[i] == arResults[j])

{

for (int ii = 0; ii <= 5; ii++)

{

if (j + 1 != arId[ii])

{

m = m + 1;

}

}

if (m == 6)

{

arId[i] = j + 1;

k = k + 1;

break;

}

}

}

}

label1.Text = null;

label1.Text = "Ниже представлен список от большего приоритета к меньшему.";

SqlDataReader sqlReader3 = null;

for (int iC = 0; iC <= 5; iC++)

{

string queryStringResult = "SELECT [Результат] FROM [Results] WHERE [Id]='" + arId[iC] + "'";

SqlCommand commandResult = new SqlCommand(queryStringResult, sqlConnection);

selectResultFromDB(sqlReader3, queryStringResult, commandResult, label1, "Результат");

}

return;

}

SqlDataReader sqlReader = null;

string queryString = "SELECT [Вопрос] FROM [Questions] WHERE [Id]='" + iQ + "'";

SqlCommand command = new SqlCommand(queryString, sqlConnection);

selectFromDB(sqlReader, queryString, command, label1, "Вопрос");

SqlDataReader sqlReader1 = null;

string queryStringAnswer1 = "SELECT [Ответ] FROM [Answers] WHERE [Id]='" + iA1 + "'";

SqlCommand commandAnswer1 = new SqlCommand(queryStringAnswer1, sqlConnection);

selectFromDB(sqlReader1, queryStringAnswer1, commandAnswer1, radioButton1, "Ответ");

SqlDataReader sqlReader2 = null;

string queryStringAnswer2 = "SELECT [Ответ] FROM [Answers] WHERE [Id]='" + iA2 + "'";

SqlCommand commandAnswer2 = new SqlCommand(queryStringAnswer2, sqlConnection);

selectFromDB(sqlReader2, queryStringAnswer2, commandAnswer2, radioButton2, "Ответ");

if (iQ == 32)

{

Form1 f1 = new Form1();

Form2 f2 = new Form2();

f1.Visible = false;

f1.Close();

f2.Show();

}

}

private void button1\_Click(object sender, EventArgs e)

{

while (iQ != 1)

{

iQ = iQ - 1;

iA1 = iA1 - 1;

iA2 = iA2 - 1;

SqlDataReader sqlReader = null;

string queryString = "SELECT [Вопрос] FROM [Questions] WHERE [Id]='" + iQ + "'";

SqlCommand command = new SqlCommand(queryString, sqlConnection);

selectFromDB(sqlReader, queryString, command, label1, "Вопрос");

SqlDataReader sqlReader1 = null;

string queryStringAnswer1 = "SELECT [Ответ] FROM [Answers] WHERE [Id]='" + iA1 + "'";

SqlCommand commandAnswer1 = new SqlCommand(queryStringAnswer1, sqlConnection);

selectFromDB(sqlReader1, queryStringAnswer1, commandAnswer1, radioButton1, "Ответ");

SqlDataReader sqlReader2 = null;

string queryStringAnswer2 = "SELECT [Ответ] FROM [Answers] WHERE [Id]='" + iA2 + "'";

SqlCommand commandAnswer2 = new SqlCommand(queryStringAnswer2, sqlConnection);

selectFromDB(sqlReader2, queryStringAnswer2, commandAnswer2, radioButton2, "Ответ");

}

}

}

}

namespace СППР

{

partial class Form2

{

/// <summary>

/// Required designer variable.

/// </summary>

private System.ComponentModel.IContainer components = null;

/// <summary>

/// Clean up any resources being used.

/// </summary>

/// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>

protected override void Dispose(bool disposing)

{

if (disposing && (components != null))

{

components.Dispose();

}

base.Dispose(disposing);

}

#region Windows Form Designer generated code

/// <summary>

/// Required method for Designer support - do not modify

/// the contents of this method with the code editor.

/// </summary>

private void InitializeComponent()

{

System.ComponentModel.ComponentResourceManager resources = new System.ComponentModel.ComponentResourceManager(typeof(Form2));

this.trackBar1 = new System.Windows.Forms.TrackBar();

this.label6 = new System.Windows.Forms.Label();

this.button1 = new System.Windows.Forms.Button();

this.button2 = new System.Windows.Forms.Button();

this.label3 = new System.Windows.Forms.Label();

this.label4 = new System.Windows.Forms.Label();

this.label5 = new System.Windows.Forms.Label();

this.label1 = new System.Windows.Forms.Label();

this.label2 = new System.Windows.Forms.Label();

this.panel3 = new System.Windows.Forms.Panel();

this.panel2 = new System.Windows.Forms.Panel();

((System.ComponentModel.ISupportInitialize)(this.trackBar1)).BeginInit();

this.SuspendLayout();

//

// trackBar1

//

this.trackBar1.LargeChange = 11;

this.trackBar1.Location = new System.Drawing.Point(479, 32);

this.trackBar1.Name = "trackBar1";

this.trackBar1.Orientation = System.Windows.Forms.Orientation.Vertical;

this.trackBar1.Size = new System.Drawing.Size(45, 226);

this.trackBar1.SmallChange = 0;

this.trackBar1.TabIndex = 0;

this.trackBar1.Scroll += new System.EventHandler(this.trackBar1\_Scroll);

//

// label6

//

this.label6.AutoSize = true;

this.label6.Font = new System.Drawing.Font("Matura MT Script Capitals", 11F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.label6.ForeColor = System.Drawing.SystemColors.Highlight;

this.label6.Location = new System.Drawing.Point(15, 217);

this.label6.Name = "label6";

this.label6.Size = new System.Drawing.Size(57, 20);

this.label6.TabIndex = 1;

this.label6.Text = "Вопрос";

this.label6.Click += new System.EventHandler(this.label6\_Click);

//

// button1

//

this.button1.Font = new System.Drawing.Font("Microsoft Himalaya", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.button1.ForeColor = System.Drawing.SystemColors.Highlight;

this.button1.Location = new System.Drawing.Point(404, 304);

this.button1.Name = "button1";

this.button1.Size = new System.Drawing.Size(104, 29);

this.button1.TabIndex = 2;

this.button1.Text = "Следующий";

this.button1.UseVisualStyleBackColor = true;

this.button1.Click += new System.EventHandler(this.button1\_Click);

//

// button2

//

this.button2.Font = new System.Drawing.Font("Microsoft Himalaya", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.button2.ForeColor = System.Drawing.SystemColors.Highlight;

this.button2.Location = new System.Drawing.Point(404, 265);

this.button2.Name = "button2";

this.button2.Size = new System.Drawing.Size(104, 32);

this.button2.TabIndex = 3;

this.button2.Text = "Предыдущий";

this.button2.UseVisualStyleBackColor = true;

this.button2.Click += new System.EventHandler(this.button2\_Click);

//

// label3

//

this.label3.AutoSize = true;

this.label3.Font = new System.Drawing.Font("Microsoft Himalaya", 11F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));

this.label3.ForeColor = System.Drawing.SystemColors.Highlight;

this.label3.Location = new System.Drawing.Point(16, 142);

this.label3.Name = "label3";

this.label3.Size = new System.Drawing.Size(171, 17);

this.label3.TabIndex = 5;

this.label3.Text = "-5 - совсем не нравится";

//

// label4

//

this.label4.AutoSize = true;

this.label4.Font = new System.Drawing.Font("Microsoft Himalaya", 11F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));

this.label4.ForeColor = System.Drawing.SystemColors.Highlight;

this.label4.Location = new System.Drawing.Point(16, 161);

this.label4.Name = "label4";

this.label4.Size = new System.Drawing.Size(177, 17);

this.label4.TabIndex = 6;

this.label4.Text = "0 - нейтрально отношусь";

//

// label5

//

this.label5.AutoSize = true;

this.label5.Font = new System.Drawing.Font("Microsoft Himalaya", 11F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));

this.label5.ForeColor = System.Drawing.SystemColors.Highlight;

this.label5.Location = new System.Drawing.Point(16, 179);

this.label5.Name = "label5";

this.label5.Size = new System.Drawing.Size(136, 17);

this.label5.TabIndex = 7;

this.label5.Text = "5 - очень нравится";

//

// label1

//

this.label1.AutoSize = true;

this.label1.Font = new System.Drawing.Font("Matura MT Script Capitals", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.label1.Location = new System.Drawing.Point(438, 121);

this.label1.Name = "label1";

this.label1.Size = new System.Drawing.Size(0, 22);

this.label1.TabIndex = 8;

//

// label2

//

this.label2.Font = new System.Drawing.Font("Microsoft Himalaya", 11F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));

this.label2.ForeColor = System.Drawing.SystemColors.Highlight;

this.label2.Location = new System.Drawing.Point(16, 91);

this.label2.Name = "label2";

this.label2.Size = new System.Drawing.Size(392, 52);

this.label2.TabIndex = 4;

this.label2.Text = "Поставьте подходящую оценку от -5 до 5, которая соответствует вашим предпочтениям" +

". ";

//

// panel3

//

this.panel3.BackgroundImage = ((System.Drawing.Image)(resources.GetObject("panel3.BackgroundImage")));

this.panel3.Location = new System.Drawing.Point(15, 68);

this.panel3.Name = "panel3";

this.panel3.Size = new System.Drawing.Size(200, 10);

this.panel3.TabIndex = 10;

//

// panel2

//

this.panel2.BackgroundImage = ((System.Drawing.Image)(resources.GetObject("panel2.BackgroundImage")));

this.panel2.Location = new System.Drawing.Point(15, 52);

this.panel2.Name = "panel2";

this.panel2.Size = new System.Drawing.Size(442, 10);

this.panel2.TabIndex = 11;

//

// Form2

//

this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 14F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.ClientSize = new System.Drawing.Size(593, 349);

this.Controls.Add(this.panel2);

this.Controls.Add(this.panel3);

this.Controls.Add(this.label1);

this.Controls.Add(this.label5);

this.Controls.Add(this.label4);

this.Controls.Add(this.label3);

this.Controls.Add(this.label2);

this.Controls.Add(this.button2);

this.Controls.Add(this.button1);

this.Controls.Add(this.label6);

this.Controls.Add(this.trackBar1);

this.Font = new System.Drawing.Font("Modern No. 20", 8.249999F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.Name = "Form2";

this.Padding = new System.Windows.Forms.Padding(20, 65, 20, 22);

this.Text = "Система поддержки приятия решений";

this.Load += new System.EventHandler(this.Form2\_Load);

((System.ComponentModel.ISupportInitialize)(this.trackBar1)).EndInit();

this.ResumeLayout(false);

this.PerformLayout();

}

#endregion

private System.Windows.Forms.TrackBar trackBar1;

private System.Windows.Forms.Label label6;

private System.Windows.Forms.Button button1;

private System.Windows.Forms.Button button2;

private System.Windows.Forms.Label label3;

private System.Windows.Forms.Label label4;

private System.Windows.Forms.Label label5;

private System.Windows.Forms.Label label1;

private System.Windows.Forms.Label label2;

private System.Windows.Forms.Panel panel3;

private System.Windows.Forms.Panel panel2;

}

}

using System;

using System.Collections.Generic;

using System.Data.OleDb;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using MetroFramework.Components;

using MetroFramework.Forms;

using MetroFramework.Fonts;

namespace СППР

{

public partial class Form2 : MetroForm

{

static double[] Prices = new double[145];

double[] P1 = new double[145];

double[] P2 = new double[145];

double[] PH = new double[145];

double R;

int n = 0;

int h;

int pQ = 0;

int[] pQuests = new int[145];

int d = 0;

public int idMax;

double PM;

static double[] pMin = new double[29];

static double[] pMax = new double[29];

static double[] pricess = new double[145];

private Form1 form1;

public Form2()

{

InitializeComponent();

}

public Form2(Form1 form1)

{

this.form1 = form1;

}

private void trackBar1\_Scroll(object sender, EventArgs e)

{

double trb = -5 + trackBar1.Value;

label1.Text = trb.ToString();

label1.Visible = true;

}

private void Form2\_Load(object sender, EventArgs e)

{

button1.Text = "Начать";

// button2.Visible = false;

List <double> p11 = new List<double>();

List<double> p22 = new List<double>();

List<double> phh = new List<double>();

string connectionString = @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\user\Desktop\НИРс\СППР\СППР\СППР\Database1.mdf;Integrated Security=True; MultipleActiveResultSets=True";

SqlConnection sqlConnection = new SqlConnection(connectionString);

using (sqlConnection)

{

sqlConnection.Open();

SqlDataReader sqlReader3 = null;

SqlCommand commandAnswer3 = new SqlCommand("SELECT [P1] FROM [QuestionsFil]", sqlConnection);

sqlReader3 = commandAnswer3.ExecuteReader();

while (sqlReader3.Read())

{

p11.Add(sqlReader3.GetFloat(0));

}

SqlDataReader sqlReader4 = null;

SqlCommand commandAnswer4 = new SqlCommand("SELECT [P2] FROM [QuestionsFil]", sqlConnection);

sqlReader4 = commandAnswer4.ExecuteReader();

while (sqlReader4.Read())

{

p22.Add(sqlReader4.GetFloat(0));

}

SqlDataReader sqlReader5 = null;

SqlCommand commandAnswer5 = new SqlCommand("SELECT [P(H)] FROM [Directions]", sqlConnection);

sqlReader5 = commandAnswer5.ExecuteReader();

while (sqlReader5.Read())

{

phh.Add(sqlReader5.GetFloat(0));

}

p11.CopyTo(P1);

p22.CopyTo(P2);

phh.CopyTo(PH);

}

sqlConnection.Close();

}

static double[] Price(double[] p1, double[] p2, double[] pH)

{

int j = 0;

for (int i = 0; i <= 144; i++)

{

Prices[i] = System.Math.Abs(((p1[i] \* pH[j]) / ((p1[i] \* pH[j]) + (p2[i] \* (1 - pH[j]))) - (((1 - p1[i]) \* pH[j])) / (1 - (p1[i] \* pH[j]) - (p2[i] \* (1 - pH[j])))));

j = j + 1;

if (j == 29)

{

j = 0;

}

}

for (int i=0; i<145; i++)

{

pricess[i] = Prices[i];

}

return (Prices);

}

public void selectFromDB(SqlDataReader sqlReaderr, string query, SqlCommand command, Control obj, string str)

{

sqlReaderr = null;

try

{

sqlReaderr = (command.ExecuteReader());

while (sqlReaderr.Read())

{

obj.Text = Convert.ToString(sqlReaderr[str]);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString(), ex.Source.ToString(), MessageBoxButtons.OK, MessageBoxIcon.Error);

}

finally

{

if (sqlReaderr != null)

{

sqlReaderr.Close();

}

}

}

private void Neylor(double[] P1, double[] P2, double[] PH, double R)

{

n = 5 - d;

double priceMax;

double[] priceMaxAr = new double[145];

int i;

i = 0;

Price(P1, P2, PH);

priceMaxAr = BubbleSort(pricess);

priceMax = pricess[0];

for (int j = 0; j<145 && i<145; j++)

{

while (Prices[i] != priceMax)

{

i = i + 1;

}

for (int jj = 0; jj < 145; jj++)

{

if (i == pQuests[jj] && i<145)

{

i = i + 1;

break;

}

}

}

idMax = i;

pQuests[pQ] = idMax;

pQ = pQ + 1;

}

static double[] ProbabilityMax(double[] p1, double[] p2, double[] ph)

{

double[] pHHer = new double[30];

double pE, pnE;

int k = 0;

int l = 0;

double[] phE = new double[30];

while (l != 29)

{

for (int j = 0; k < 145; j++)

{

pE = (p1[k] \* ph[l]) + (p2[k] \* (1 - ph[l]));

pnE = 1 - pE;

phE[j] = (p1[k] \* ph[l]) / pE;

k = k + 29;

}

pHHer = BubbleSort(phE);

pMax[l] = pHHer[0];

l = l + 1;

k = l;

}

return(pMax);

}

static double[] BubbleSort(double[] mas)

{

double temp;

for (int i = 0; i < mas.Length; i++)

{

for (int j = i + 1; j < mas.Length; j++)

{

if (mas[i] < mas[j])

{

temp = mas[i];

mas[i] = mas[j];

mas[j] = temp;

}

}

}

return mas;

}

static double[] ProbabilityMin(double[] p1, double[] p2, double[] ph)

{

double[] pHH = new double[30];

double pE, pnE;

int k = 0;

int l = 0;

double[] phE = new double[30];

while (l != 29)

{

for (int j = 0; k < 145; j++)

{

pE = (p1[k] \* ph[l]) + (p2[k] \* (1 - ph[l]));

pnE = 1 - pE;

phE[j] = (p1[k] \* ph[l]) / pE;

k = k + 29;

}

pHH = BubbleSort(phE);

pMin[l] = pHH[pHH.Length-1];

l = l + 1;

k = l;

}

return (pMin);

}

private void button1\_Click(object sender, EventArgs e)

{

double[] pMaxHH = new double[145];

button1.Text = "Следующий";

//button2.Visible = false;

R = -5 + trackBar1.Value;

//int h = 0;

int j = 0;

//int f = 1;

double pMaxH;

string connectionString = @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\user\Desktop\НИРс\СППР\СППР\СППР\Database1.mdf;Integrated Security=True; MultipleActiveResultSets=True";

SqlConnection sqlConnection = new SqlConnection(connectionString);

//sqlConnection.Open();

/\*SqlDataReader sqlReader = null;

string queryString = "SELECT [Вопрос] FROM [Directions] WHERE [Id]='" + f + "'";

SqlCommand command = new SqlCommand(queryString, sqlConnection);

selectFromDB(sqlReader, queryString, command, label6, "Направление");

f = f + 1;\*/

Neylor(P1, P2, PH, R);

while(h != 29)

{

sqlConnection.Open();

idMax = idMax + 1;

SqlDataReader sqlReader = null;

string queryString = "SELECT [Вопрос] FROM [QuestionsFil] WHERE [Id]='" + idMax + "'";

SqlCommand command = new SqlCommand(queryString, sqlConnection);

selectFromDB(sqlReader, queryString, command, label6, "Вопрос");

return;

}

if (h == 29)

{

pMaxHH = BubbleSort(pMax);

pMaxH = pMax[0];

while (pMax[j] != pMaxH)

{

j = j + 1;

}

sqlConnection.Open();

SqlDataReader sqlReader1 = null;

string queryString1= "SELECT [Направление] FROM [Directions] WHERE [Id]='" + j + "'";

SqlCommand command1 = new SqlCommand(queryString1, sqlConnection);

selectFromDB(sqlReader1, queryString1, command1, label6, "Направление");

}

}

static double[] ProbabilityOfHypotheses(double[] pH, double[] p1, double[] p2, double R, int idmax)

{

double[] pHH = new double[145];

double pHH2, linInterpol, pE, pnE, phE, phne;

int k;

k = idmax;

for (int i = 0; i<=144; i++)

{

if (i == idmax)

{

if (R == 0)

{

pHH[i] = pH[i];

continue;

}

pE = (p1[k] \* pH[k]) + (p2[k] \* (1 - pH[k]));

pnE = 1 - pE;

phE = (p1[k] \* pH[k]) / pE;

phne = ((1 - p1[k]) \* pH[k]) / (1 - pE);

pHH2 = (phE \* pE) + (phne \* pnE);

if (R == 5)

{

pHH[i] = pHH2;

}

if (R == -5)

{

pHH[i] = phE;

}

linInterpol = (((pHH2 - pH[i]) / 10) \* (R + 5)) + pH[i];

pHH[i] = linInterpol;

}

}

return (pHH);

}

private void button2\_Click(object sender, EventArgs e)

{

double[] newPH;

newPH = ProbabilityOfHypotheses(PH, P1, P2, R, idMax);

Price(P1, P2, newPH);

pMax = ProbabilityMax(P1, P2, PH);

pMin = ProbabilityMin(P1, P2, PH);

PM = pMin.Max();

for (int i = 0; i < 29; i++)

{

if (pMax[i] <= PM)

{

h = h + 1;

}

}

}

}

}