СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

АСПТ - автоматическая система пожаротушения

АУП – административно-управленческий персонал

БД – база данных

ВНИИПО - всероссийский научно-исследовательский институт противопожарной обороны

ГГц – гигагерц

ИМ – информационная модель

ИС – информационная система

КСБ – комплексная система безопасности

ЛВС – локальная вычислительная сеть

МЧС – министерство чрезвычайных ситуаций

ОКУД - общероссийский классификатор управленческой документации

ОПС - охранно-пожарная сигнализация

ОС – операционная система

ПК – персональный компьютер

ПКО – проектно-конструкторский отдел

ПО – программное обеспечение

СКУД - система контроля и управления доступом

СКФУ - Северо-Кавказский Федеральный Университет

СМНУ - специализированное монтажно-наладочное управление

СПИ - система передачи извещений

СУБД - система управления базами данных

ТК – технический комитет

ТСОН - телевизионная система охранного наблюдения

УСД – унифицированная система документации

ЧД – чистый доход

ЧДД - чистый дисконтированный доход

ЭВМ – электронно-вычислительная машина

ADSL - asymmetric Digital Subscriber Line

BSS - basic Service Set

CASE - computer-Aided Software Engineering

DFD - data flow diagramming

DSL - dictionary Specification Language

ESS - extended Service Set

GB – gigabyte

HDD - hard disk drive

IDEF - integrated definition

ISO - International Organization for Standardization

KDD - knowledge discovery in databases

LAN - local area network

RAM - random access memory

SDD - solid-state disk

USB - universal serial bus

WAN - wide area network

WDDM - Windows Display Driver Model

xDSL - digital subscriber line

ВВЕДЕНИЕ

Последние несколько десятилетий наблюдается значительный рост объема информации практически во всех сферах общества. Современные исследования показывают, что объем сгенерированных данных составил 2,8 зеттабайт в 2012 году, до 2020 г. прогнозируется рост до 40 зеттабайт (ZB). Экономика, финансы, политика, культура, технологии, каждая из этих сфер с огромной скоростью накапливает и обрабатывает информацию [1]. Процесс накопления, обработки и использования данных в знания с каждым годом все больше ускоряется. По утверждению многих ученых, каждые десять лет объем информации увеличивается в два раза. Исходя из этого факта и возникает необходимость эффективно хранить, распределять и обрабатывать накопленные данные.

Современное общество проходит так называемый процесс информатизации, при этом темпы этого процесса растут. Еще десять лет назад для работы с данными требовалось наличие стационарных компьютеров, сейчас же вычисления можно проводить на мобильных компьютерах, позволяющие не только получать доступ к мировым источникам, но даже контролировать процессы, обрабатывать данные в облаке, иметь место при управлении крупными проектами и зачастую даже заменять персональный компьютер во многих ситуациях. Информатизация общества приняла глобальный социальный характер, особенностью которого накопление, является сбор, продуцирование, обработка, хранение, передача и использование информации. Высокий уровень обслуживания в области информации поддерживает доступность любого члена общества к различным источникам достоверной информации, а также представление информации в визуальном формате, за счет которого повышается существенность используемых данных.

Информационные средства расширяют возможности работы с данными, позволяя не только хранить их, но и позволяют эти данные анализировать и на основе этих результатов принимать решения различного характера.

Информатизация общества давно перенесла задачу анализа данных на плечи вычислительной техники, это намного быстрее и удобнее, чем решать эту задачу человеческими ресурсами. Разработано множество аналитических систем, специального программного обеспечения, веб-сервисов и прочего, которые позволяют анализировать полученные данные. Но остается вопрос, затрагивающий их эффективность и недостатки, в частности касательно представлению конечных результатов под пользователя.

У любого аналитического программного обеспечения есть ряд своих проблем или недостатков, будь то невозможность работы с пропусками данных, необходимость в эмпирических или эвристических предположениях, неизвестность модели и так далее.

Данная работа направлена не просто на создание аналитической платформы, в данной ситуации она выступает больше в качестве инструмента достижения цели, а на изменение подхода к обработке данных и визуализации результатов, получению данных из различных источников, работе с различными форматами и типами, использование кластерного анализа, для возможности проверять гипотезы, находить зависимости между этими данными.

Созданный программный продукт сможет функционировать не только в рамках отдельного предприятия, а сможет стать вполне самостоятельным коммерчески реализуемым проектом.

В настоящее время существует ряд проблем в области анализа данных. Так например существующие методы не рассматривают оценку интеллектуального ресурса пользователя при решении задач разной сложности в определенной предметной области.

В рамках исследования рассматривается предметная область – интеллектуальный анализ данных при оценке интеллектуального ресурса человека и сопоставление его с интеллектуальными запросами (задачами).

Необходимость измерения и сопоставления может возникнуть при приеме нового сотрудника, оценке знаний учащегося, визуализации данных и других

процессов, в которых необходимо измерить интеллектуальный ресурс человека для оценки его потенциала и возможностей решать определенные классы задач в предметной области.

Целью диссертации является исследование и моделирование процессов оценки интеллектуального ресурса пользователя и сопоставление его с интеллектуальными запросами (задачами). Объектом исследования является процесс оценки интеллектуального ресурса пользователя. Предметами исследования являютсяинтеллектуальные ресурсы пользователя при решении профессиональных задач.

Задачи,поставленные в рамках данной выпускной квалификационной работы:

– провести анализ существующих методов и средств оценки интеллектуальных ресурсов пользователей;

– изучить существующие средства для тестирования пользователей в области задач;

– разработать модели системы оценки интеллектуальных ресурсов;

– разработать интерфейс пользователя, который отвечает задаче оценки интеллектуальных ресурсов;

– разработать прототип программного средства для оценки интеллектуального ресурса пользователя в заданной предметной области.

Первая часть посвящена анализу предприятия, рассмотрению его структуры, программного обеспечения и оборудования. Затрагивает изучение существующих методов измерения интеллектуального ресурса человека, а также возможность их применения сто стороны задач в предметной области. Рассмотрены известные тесты структуры интеллекта Айзенка, прогрессивные матрицы Равена и другие методы. Также в теоретической части рассмотрена математическая модель метода измерения интеллектуального ресурса, ориентирующаяся на психофизические методы, а также математические

алгоритмы определяющие степени уверенности и глубину знаний человека в предметной области.

В аналитической части проведен анализ существующих программных решений используемых для тестирования пользователей. Так как была выбрана область рассмотрения навыков программиста, то были выделены решения в этой области, выявлены их минусы и плюсы. Во второй половине главы посвященной анализу, был обоснован выбор среды разработки, применяемой в рамках данной работы, а также обоснование выбранного инструментария.

Проектная часть затрагивает непосредственно разработку самого программного модуля. Он посвящен моделированию информационных процессов в среде BPWin, а также структуре модуля и его взаимодействию с системой «Познание», которая была разработана нами ранее. Такжев проектной частирассматриваются алгоритмы и структура модуля, результаты работы программы. В разделе программных модулей приведены отрывки кода, демонстрирующие алгоритмы и результаты выполнения программы.

1ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1 Программная и техническая архитектура ИС предметной области

Предприятие современной ИТ-отрасли обязательно имеет в своем распоряжении информационное обеспечение систем управления, включающее в себя базы данных и аппаратно-програмные комплексы для управления ими. Большую роль в принятии решений играет научно-техническая информация, содержащая актуальные научные знания, сведения об изобретениях, тенденций развития инновационной деятельности, а также конкурирующих фирм. Как правило это постоянно пополняемый общий фонд и потенциал знаний и технических решений, практическое и использование которого обеспечивает фирме высокий конкурентоспособный уровень на своем рынке[2].

Информация служит фундаментом для формирования различного рода аналитических отчетов, для разработки и принятия необходимых управленческих решений. Содержание каждой конкретной информации определяется потребностями аналитического отдела и вырабатываемых управленческих решений.

Информационный процесс, целью которого является получение аналитической, плановой, научно-техниче-ской, контрольной и учетной информации при организации информационной технологии, должен быть унифицирован для возможности использовать современные средства вычислительной техники.

Управленческая внутрифирменная информационная система представляет собой совокупность информационных процессов для удовлетворения потребности в информации разных уровней принятия решений.

В фирме формируются базы данных для организации этого процесса, которые наполнены данными по различным аспектам функционирования предприятия. Базы данных являются составной частью программного комплекса и формируются в процессе настройки программного обеспечения информационной технологии управления фирмой.

В целях защиты информации от несанкционированного доступа каждый пользователь компании, подключенный к локальной сети и информационной базе данных, имеет свободный доступ только к информации, необходимой ему для выполнения своих официальных функций, и, при необходимости, к информации, не имеющей прямого отношения к его функциям возможно только с разрешения высшего руководства компании.

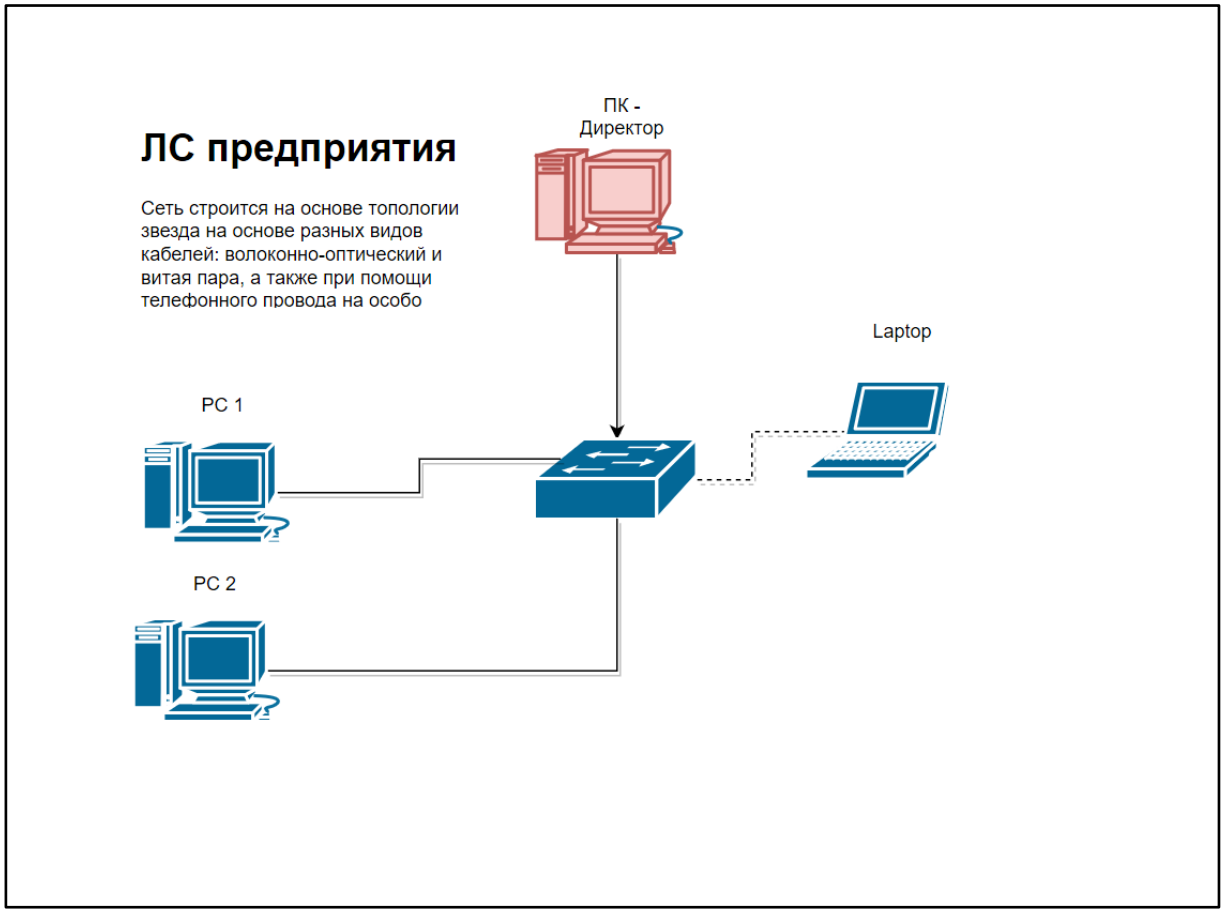


Рисунок 1.1. - Структура локальной сети предприятия

Современная информационная технология управляющей компании - это процесс, состоящий из четко регламентированных правил выполнения операций над информацией в информационной системе экономического объекта с целью принятия оптимального управленческого решения, связанного

с основной функцией любого предприятия - выпуском готовой продукции на заказ получить экономические результаты от продажи этих продуктов.

Движение потоков информации на предприятии осуществляется непосредственно через локальную вычислительную сеть и соединения с Интернетом.

Сеть строится на основе топологии звезда на основе разных видов кабелей: волоконно-оптический и витая пара, а также при помощи телефонного провода на особо длинных участках. Звезда - базовая топология компьютерной сети, в которой все компьютеры сети присоединены к центральному узлу, образуя физический сегмент сети. Локальная сеть предприятия отображена на рисунке 1.1.1.

К достоинствам данной топологии можно отнести:

– масштабируемость сети;

– выход из строя одной рабочей станции не отражается на работе всей сети в целом;

– лёгкий поиск неисправностей и обрывов в сети;

– высокая производительность сети (при условии правильного проектирования);

– безопасность.

ПКО имеет доступ к централизованной базе данных Dropbox. Данные, полученные с сервера БД, используются для имитации и построения трёхмерных моделей реальных объектов с планированием, проектировкой и расстановкой систем безопасности в программе для создания трёхмерных моделей GoogleSketchUp 8.

Данное предприятие имеет выход в Интернет, пользуясь услугой высокоскоростного интернета по ADSL- каналу с помощью ADSL модема.

Высокоскоростной интернет – это услуга высокоскоростного доступа к сети Интернет, предоставляемая провайдером ОАО «Ростелеком» через ADSL-подключение. А также имеется беспроводная сеть WiFi.

Локальная вычислительна сеть – это компьютерная сеть, которая как правило покрывает территорию одного или несколько зданий, находящихся не так удалённо друг от друга. Задача создания точки доступа состоит в том, чтобы обеспечить обмен данными между пользователями, использующими беспроводную сеть, а также обеспечить уровень доступа, при котором все клиенты будут иметь равные права в отношении доступа к среде передачи данных. Несколько компьютеров могут быть связаны друг с другом в локальной сети. В то же время эти компьютеры могут работать на совершенно разных операционных системах. Все они должны иметь одинаковый уровень доступа к сети.

Маршрутизаторы выполняют роль пограничных сетевых устройств. Они являются устройствами, которые устанавливаются и находятся на границе между двумя или более сетями, или же находятся между локальной сетью и сетью Интернет, выполняя роль сетевого шлюза. Маршрутизаторы с точки зрения конструкции должны иметь минимум два порта: одному будет подключаться сама локальная сеть, этот порт называется LAN-портом, а к другому будет подключенная внешняя сеть или же Интернет, такой порт называется WAN-портом. Те маршрутизаторы, которые используются в домашним условиях или в масштабах небольшого офиса (их называю SOHO-маршрутизаторами), имеют в наличие обычно несколько LAN-портов (от одного до четырёх) объединённых в коммутатор и один WAN-порт. Чаще всего WAN-порты имеют стандартный для себя интерфейс 10/100Base-TX. К такому интерфейсу может без каких-либо проблем подключиться xDSL-модем с соответствующим интерфейсом, либо же можно подключить Ethernet-кабель.

Та точка беспроводного доступа, которая интегрирована в маршрутизатор может организовывать сегмент сети с беспроводным доступом. Для маршрутизатора устройства, подключенные беспроводным способом и через LAN-порты ничем не отличаются друг от друга.

Такое использование маршрутизатора, который организует беспроводную точку доступа, выгодно, поскольку позволяет сэкономить на дополнительном

оборудовании, таком как дополнительные контроллеры Ethernet, мини-коммутаторы и т. Д. В то же время сам маршрутизатор предоставляет различные дополнительные средства для защиты сети и предотвращения несанкционированного взлома. В компании ИП Шевченко используется антивирус Касперского 2016[3].

Антивирусная программа или же антивирус — это программа в функции которой входит обнаружение компьютерных вирусов разных типов, различные подозрительных и вредоносных программ, работа и восстановление «заражённых» файлов, постоянный мониторинг всех поступающих извне данных и файлов, а также полноценное обеспечение безопасности операционной системы и компьютера.

«Лаборатория Касперского» это одна из самых популярных систем защиты персональных компьютеров в России, а так же одна из крупнейших в Европе. Лаборатория обеспечивает защиту от вирусов, спама, хакерских атак, вредоносного программного обеспечения и других нежелательных воздействий. Компания же в свою очередь производит такие программные решения в сфере информационной безопасности, которые позволяют ей входить в число ведущих мировых производителей в этой отрасли.

Все основные отделы предприятия используют следующее программное обеспечение:

- операционные системы: WindowsServer 2003;
- операционные системы: Windows 10 набазеAmazonAWS;
- офисный пакет приложений MicrosoftOffice 2007
- наборстандартныхпрограмм: Word, Excel, Access, PowerPoint, MS Outlook;
- графические редакторы: CorelDraw, Gimp;
- антивирусноеПО: Kaspersky Internet Security 2014;

1.2 Методы оценки интеллектуального ресурса

Решение различных задач (с помощью компьютера в том числе) требует от человека определенных знаний и умений, которые отражаются в его интеллектуальных способностях. С другой стороны – каждая задача, решаемая человеком, обладает определенными интеллектуальными ресурсами. Как измерить интеллектуальные ресурсы человека и сопоставить их с требуемыми интеллектуальными запросами (задачами)?

Подходы в этом направлении исследований существуют разные. В работе рассматривается один из них – основанный на понятии «осмысленной задачи». Будем считать, что задача понятна человеку, если любое ее решение он способен распознать как верное или неверное. Так, например, в информатике, осмысленной задачей будем считать ту, в которой наряду с исходными данными и заключением сформулирован некий критерий осмысленности, состоящий из алгоритма решения задачи, ограничений – позволяющих всегда получать решения и исключительных ситуаций – указывающих при каких условиях мы никогда не получим решение. Именно с такими задачами и стоит иметь дела при применении компьютеров в решении задач, поскольку при этом потребляются значительные ресурсы: человеческие, временные, стоимостные и др.

Интеллектуальный ресурс человека можно измерить лишь в том случае, когда он способен понять и решить задачу. Чем больше логических связей внутри задачи может правильно распознать и оценить человек, тем мощнее его интеллектуальный ресурс. Возникает вопрос, как определить, насколько сложные задачи может решить человек?

Попытки количественно оценивать интеллект начались в прошлом столетии. Для этой цели разработано так называемое понятие IQ – количественная оценка уровня интеллекта человека. На основании этого понятия созданы тесты для получения коэффициента интеллекта. Со временем единое число утратило свою однозначность из-за большого количества шкал

оценки, т.к. для каждой методики используется свой метод и подход к измерению.

Приведем некоторые известные методы измерения интеллектуальных способностей человека и так же анализ глубины интеллектуального ресурса предметной области, получаемые с помощью тестов Айзенка, Д. Векслера, Дж. Равена, Р. Амтхауэра.

Таблица 1.1

Сравнение систем оценивания интеллектуального ресурса

	Соответствие предметной области	Адаптируемость	Оценка чистого интеллекта	Оценка с точки зрения задач
Тест Д. Векслера	Обобщенно	Сложно адаптируема	Отсутствует	Отсутствует
Тест Айзенка	Общая осведомленность	Требует адаптации под социальную среду	Отсутствует	Отсутствует
Тест структуры интеллекта Амтхауэра	Выявление технического и математического типа мышления	Требует социальных навыков	Имеется	Отсутствует
Прогрессивные матрицы Рейвена	Отсутствует	Полная	Оценка чистого интеллекта	Отсутствует

В таблице 1.2 указаны ограничения рассмотренных систем тестирования. Тесты Д. Векслера и Айзенка делают упор исключительно на общую осведомленность человека, что не позволяет оценить интеллектуальный ресурс человека. Тесты Райвена и Амтхауэра позволяют наиболее полную картину о глубине мышления человека.

Тест Д. Векслера - определяет общее развитие на основе иерархической модели Д. Векслера. Тест определяет общую осведомленность тестируемого, его внимание, понятливость и т.д. Оригинальная версия сложно адаптируема.

Тест состоит из двух разделов, в которые включены 11 субтестов. Эти два раздела определяют вербальные и невербальные задания. По результатам тестирования определяется общая осведомленность.

Товар сначала подорожал на 10%, а потом подешевел на 10%. Какова его стоимость сейчас относительно первоначальной?	☐	0%
	☐	90%
	☐	99%
	☐	100%
	☐	101%
	☐	110%
	☐	111%
	☐	ни одно из вышеперечисленных
Сколько раз встречается цифра 4 в целых числах от 1 до 50?	☐	1
	☐	4
	☐	5
	☐	6
	☐	10
	☐	14
	☐	15
	☐	ни одно из вышеперечисленных

Рис 1.2. - Тест Д. Векслера

Результаты теста помогают судить о развитии человека, его способностях ориентироваться в социальной среде. Последнее можно отнести к ограничениям, так как это не позволяет получить чистый потенциал интеллектуальных способностей испытуемого, из-за связки с нормами и идеями общества. Тест не ориентирован на решение задач в информатике.

Тест Дж. Рейвена. Тест так же известен как «Прогрессивные матрицы Рейвена». В основе теста заложен поиск логических связей в наборе графических матриц. Каждая следующая задача сложнее предыдущей. Сложность задачи варьируется увеличением количества логических связей между элементами матрицы (рисунок 1.3).

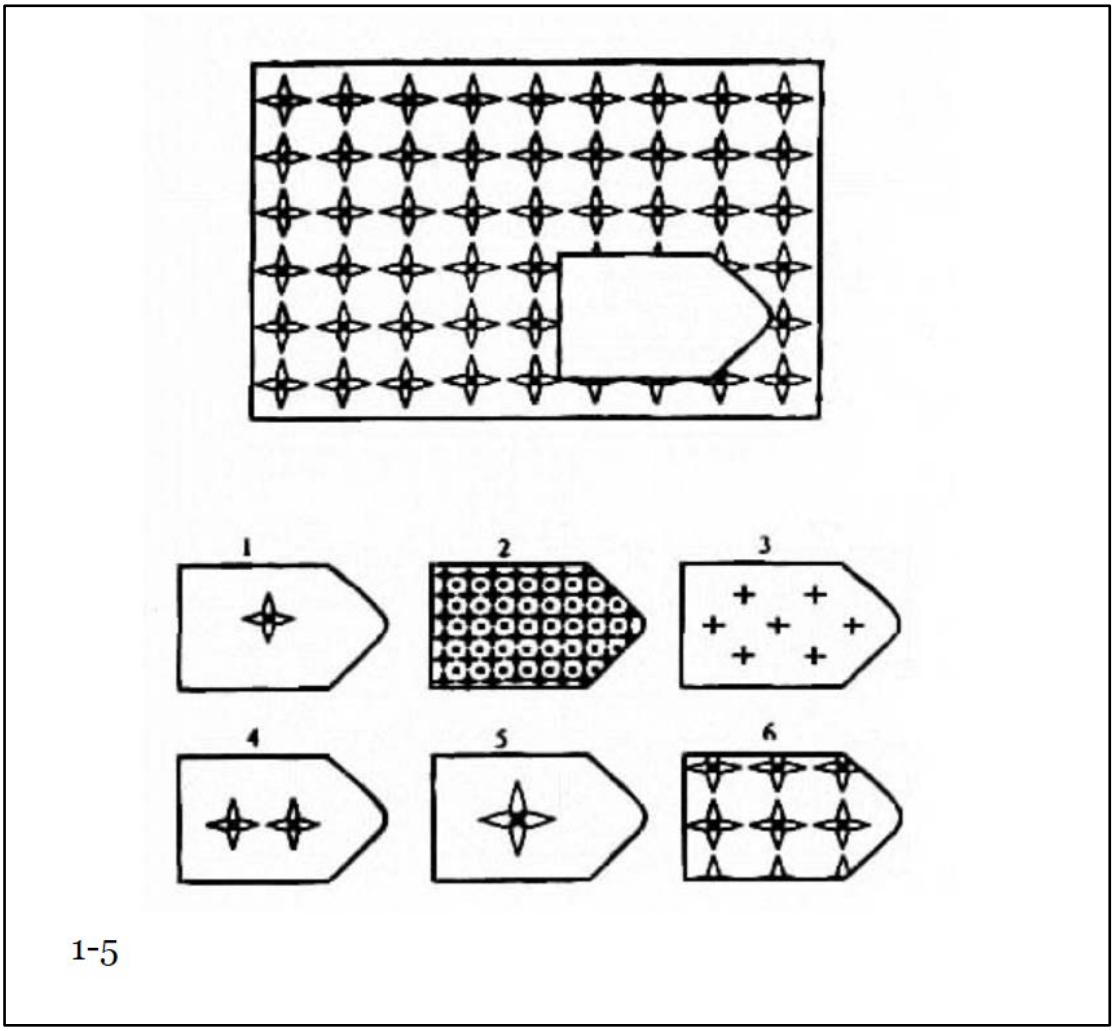


Рис 1.3 - Прогрессивные матрицы Дж. Рейвена, пример задания типа А

Распространены версии теста с рекомендациями к временному ограничению в 20 и 30 минут. В соответствии с официальным руководством [1] тестирование проводится без ограничения времени, для исключения дискриминации индивидов с «медленным» типом мышления. Вариант с временным ограничением рекомендуется выбирать в зависимости от задач,

стоящих перед тестируемым. Так, например, навык стоящий перед пользователем может требовать оперативного решения задач в практической деятельности.

Тест делится на пять групп заданий. В заданиях типа «А» заложен принцип дополнения изображения недостающим фрагментом. Выполнение задания требует анализа всего изображения и обнаружения особенностей каждого из фрагментов. После анализа требуется подставить объект в соответствии со структурой и паттерном исходного изображения.

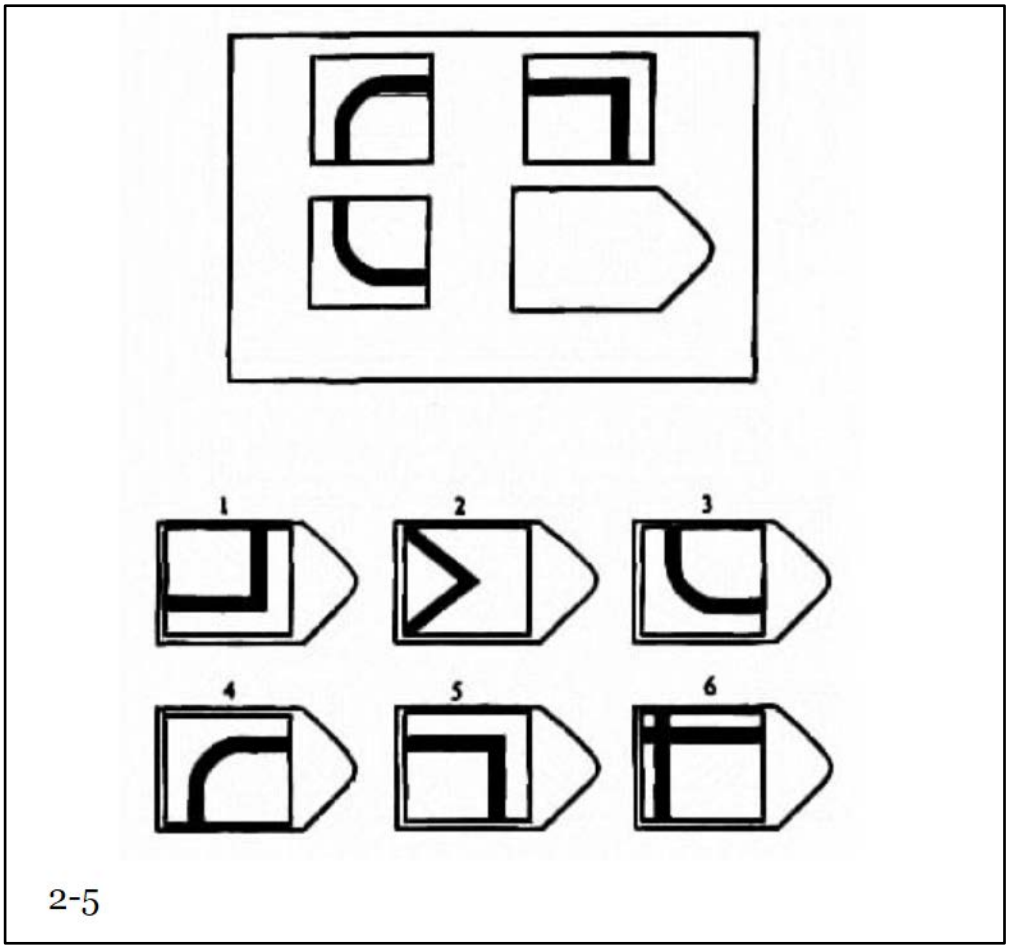


Рис 1.4 - Прогрессивные матрицы Дж. Рейвена, пример задания типа В

В задании типа «В», от испытуемого требуется найти взаимосвязь между группами фигур. Для успешного выполнения необходимо правильно определять оси симметрий.

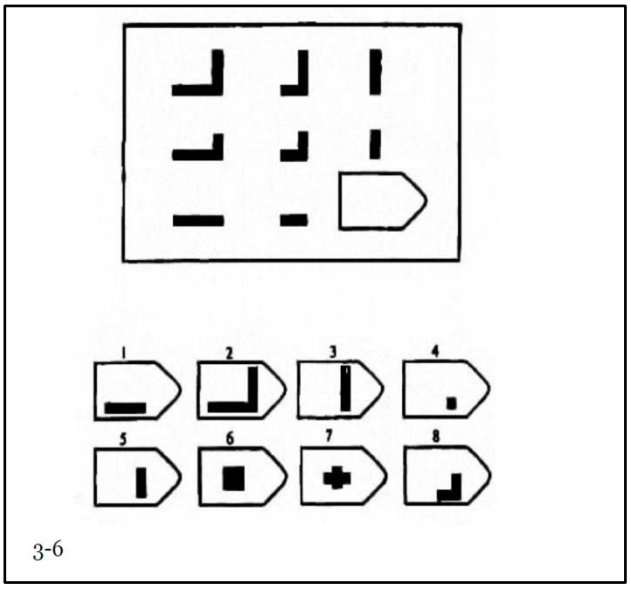


Рис 1.5 - Прогрессивные матрицы Дж. Рейвена, пример задания типа C

В задании типа «C» усложняется поиск фрагментов, за счет логического принципа преобразований фигур.

Тип «D» дополняется перегруппировкой объектов в матрице.

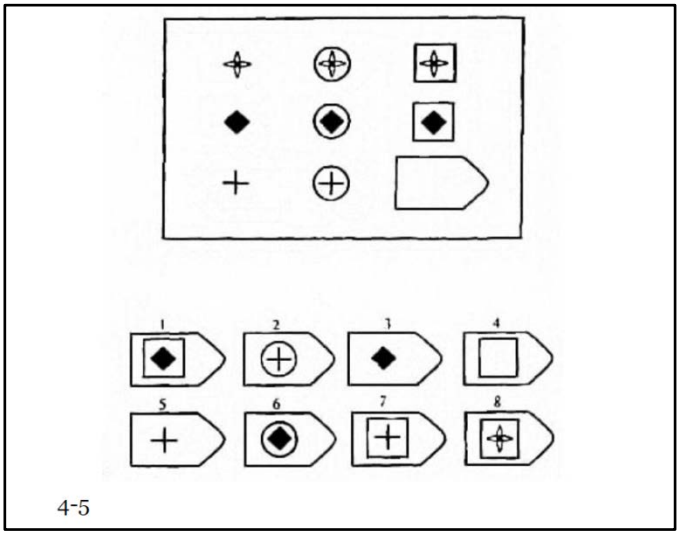


Рис 1.6 - Прогрессивные матрицы Дж. Рейвена, пример задания типа D

Тип заданий «E» дополняются модификацией форм и синтезом фигур. Задания данного типа требуют тщательного анализа и вычислений разных логических закономерностей.

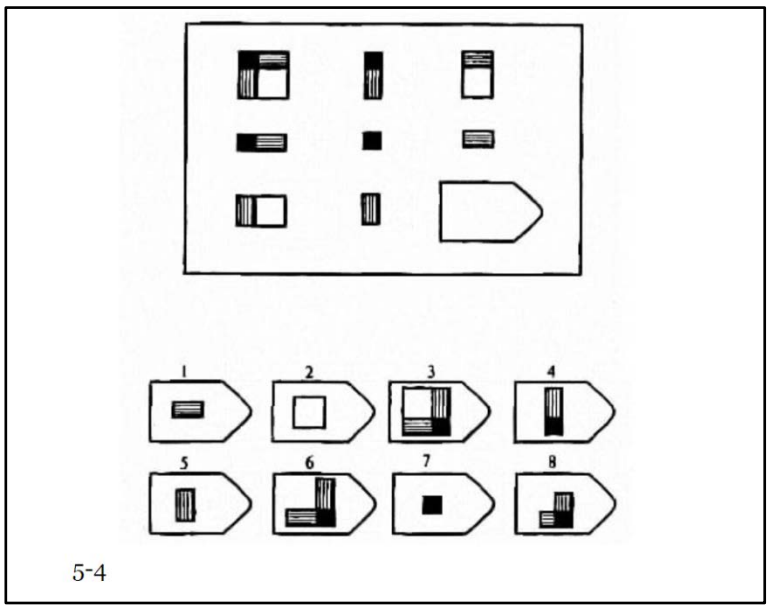


Рис 1.7 - Прогрессивные матрицы Дж. Рейвена, пример задания типа Е

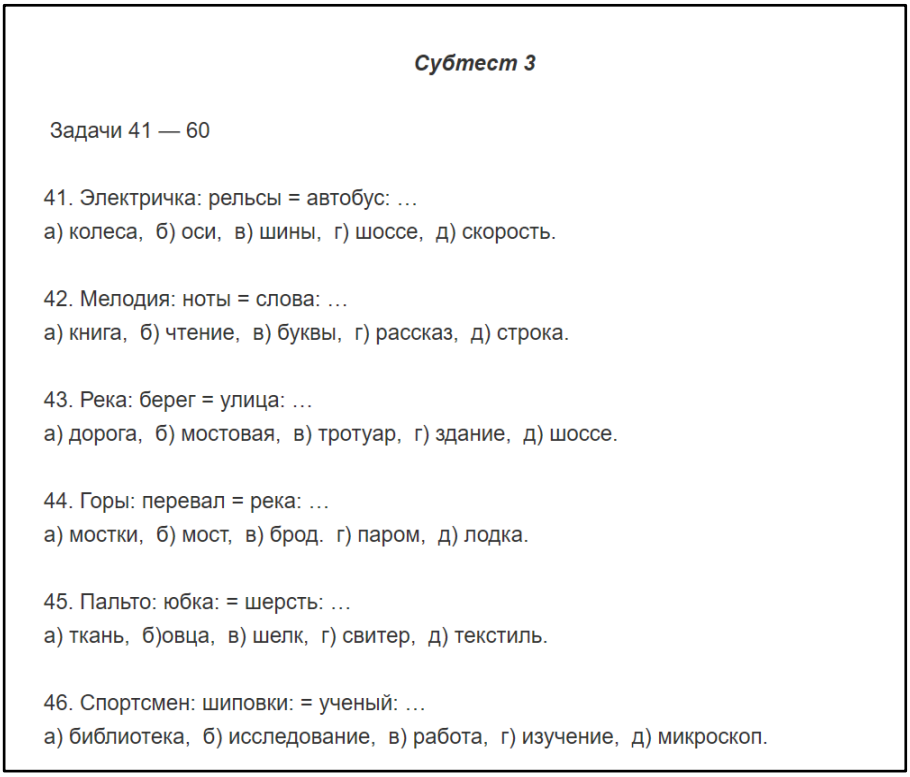
Для индивидов, набравших максимальные баллы предлагается тестирование повышенной сложности. Задания дополняются большим количеством логических связей, применение разных уровней абстракций, булевыми операциями, модификациями и метаморфозами объектов в матрице. Особенность матриц Райвена в их применимости к оценке испытуемых любого языкового состава, социокультурного фона и уровнем речевого развития.

Плюсом метода является оценка чистых интеллектуальных способностей испытуемого, не зависящих от влияния социо-культуры и уровня осведомленности. Тест может применяться при изучении зависимости и причин интеллектуальных различий, выявления индивидов со стратегическим мышлением.

Тест структуры интеллекта Амтхауэра - служит для оценки интеллектуального развития индивидов в возрасте от 13 до 61 года. Методика широко используется для выяснения профессиональной пригодности тестируемых. Тест охватывает навыки абстрагирования, классификаций, логического отбора, оценки математического мышления, комбинаторики, внимания и памяти.

Каждый из субтестов можно выделить в четыре группы:

1. Субтесты объединенные в вербальный комплекс (тесты 1-4). Каждый из них предполагает способности оперировать словами, как сигналами и символами.



Субтест 3

Задачи 41 — 60

41. Электричка: рельсы = автобус: …
а) колеса, б) оси, в) шины, г) шоссе, д) скорость.

42. Мелодия: ноты = слова: …
а) книга, б) чтение, в) буквы, г) рассказ, д) строка.

43. Река: берег = улица: …
а) дорога, б) мостовая, в) тротуар, г) здание, д) шоссе.

44. Горы: перевал = река: …
а) мостки, б) мост, в) брод. г) паром, д) лодка.

45. Пальто: юбка: = шерсть: …
а) ткань, б)овца, в) шелк, г) свитер, д) текстиль.

46. Спортсмен: шиповки: = ученый: …
а) библиотека, б) исследование, в) работа, г) изучение, д) микроскоп.

Рис 1.8 - Тест структуры

2. Субтесты 5-6 объединены математическим комплексом (5, 6). Эти тесты предполагаю оценку способностей в областипрограммирования и практической математики.

При преобладании сенсорных или наглядно-действенных форм общения, испытуемый во время теста будет сравнивать предметы по визуальным признакам, таким как величина, форма или принадлежность к определенной ситуации.

Наличие ориентирования на абстрактные или категориальные связи, говорит о том, что испытуемый способен преодолеть первичную визуальную или ситуативную информацию и вывести внутренние логические связи, способствующие прохождению данной категории заданий.

84. Два насоса выкачали из котлована 60 гектолитров воды. Первый при этом выкачал воды в 3 раза больше, чем второй. Сколько воды выкачал второй насос?

85. Банка с керосином весит 8 кг. Из нее вылили половину керосина, после чего она стала весить 4,5 кг. Определите вес банки.

Рис 1.9 - Тест структуры

3. Субтесты 7-8 являются конструктивным комплексом, предполагающим развитые пространственные способности практического и стратегического плана.

В субтесте включены задачи, в которых необходимо установить из каких частей можно сложить фигуру. В каждом задании изображены пять разных фигур.

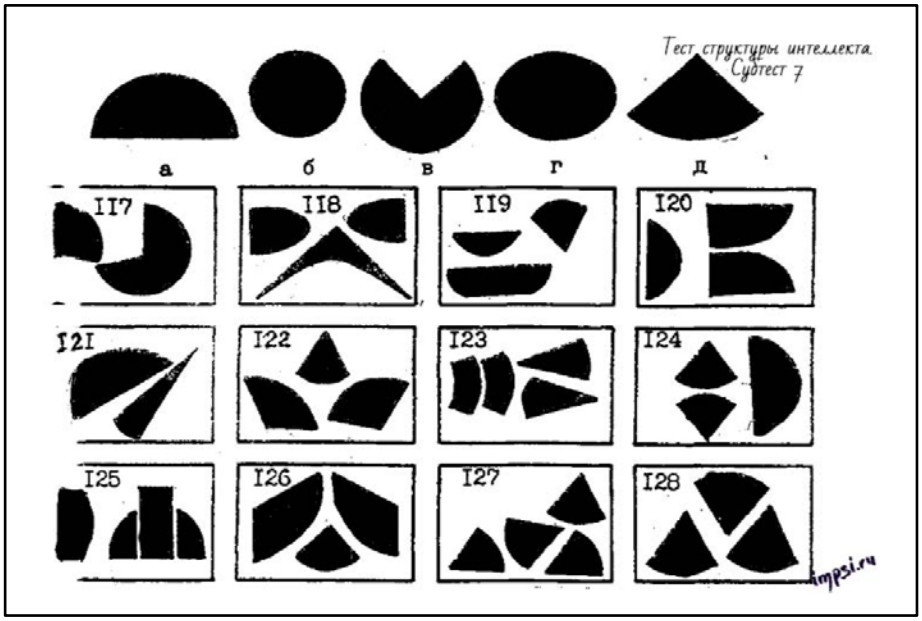


Рис 1.10 - Тест структуры

Этот комплекс субтестов позволяет оценить способность к оперированию человеком пространственными представлениями. Категория 8 отличается наличием пространственного положения объектов, не затрагивая структурной специфики расположения объектов.

4. Комплексы теоретических (2, 4) и практических планов способностей (1, 3). [2].

Для решения этой группы задач необходимо заучить представленный ряд слов в течении трех минут. После чего испытуемому выдаются задания связанные со способностью к оперирования с зафиксированной в памяти информацией.

Результаты данного субтеста позволяют выявить уровень кратковременной памяти испытуемого. Этот субтест никак не влияет на общую оценку памяти, в силу различных физико-психических способностей испытуемого.

Субтест 9

На этом листе находятся задания к словам, которые Вы только что заучили.

Пример 09:
Слово, начинающееся на букву «у» было в разделе …
а) цветы, б) инструменты, в) птицы, г) художественные произведения, д) животные.

Среди заученных Вами слов было только одно, которое начинается на букву «у», это слово «уж». Поскольку оно находилось в разделе «животные», то правильным ответом здесь является цифра «д». Поэтому в бланке ответов для примера 09 вычеркнута буква «д».

Рис 1.11 - Тест структуры

При длительной методике прохождения теста, 90 минут, и большого объема заданий, результаты дают надежную оценку профессиональной пригодности. В ограничения входят зависимости от социально-культурной среды развития тестируемых. Заведомое преимущество получают испытуемые с естественно-научной, технической и математической подготовкой, по сравнению с гуманитарной подготовкой[4].

Рис 1.12 - Пример задания теста Айзенка

Тест Айзенка предназначен для оценки интеллектуальных способностей для людей имеющих образование не ниже среднего в возрасте от 18 до 50 лет,. Тесты рассчитаны на оценку мыслительных способностей, а не уровня знаний (эрудированности). Приведенный коэффициент интеллекта это попытка оценки фактора общего интеллекта.

1.3 Метод оценки в области задач

В рамках данной работы мы можем предположить следующую систему оценки интеллектуального ресурса для данного человека p которую мы сумели зафиксировать путем специального опроса, аксиоматическую систему T, в рамках которой он собирается рассуждать как пользователь. Тогда ресурс p относительно T – тройка res(p,T)=(m1(p, T), m2(p, T), m3(p, T)) натуральных чисел m1(p, T), m2(p, T), m3(p, T), таких, что:

- m1(p,T) – наибольшая длина доказательств в T, все еще имеющих достаточно высокую (заранее фиксированную) балльную оценку степени убедительности для данного человека p;
- m2(p,T) – наибольшая длина последовательностей символов алфавита языка системы T, все еще имеющих достаточно высокую (заранее фиксированную) балльную оценку субъективной уверенности p в безошибочной распознаваемости их как формул (или не формул) языка системы;
- m3(p,T) – таких же, что и m2(p,T), но применительно к термам;

Исходя из вышесказанного проблему измерения ресурса пользователя p относительно T, можно свести к следующим задачам:

- исследовать вопрос о влиянии длины логических выводов в T на степень убедительности их для p;

- исследовать вопрос о влиянии длины последовательности символов на уверенность признания данным человеком p этой последовательности формулой (или не формулой) языка системы T;

- исследовать вопрос о влиянии длины последовательности символов на уверенность признания данным человеком p этой последовательности термом (или не термом) языка системы T;

Общая схема измерения интеллектуальных ресурсов пользователей заключается в следующем:

1. Строится «терм-калибровку», которая описывается как последовательность a1, a2, …, ai из i слов содержащихся в алфавите A

2. «Форм-калибровка» и «док-калибровка» строится по аналогичному методу.

3. Балльная шкалы TS, FS или PS устанавливаются, как характеризующая убедительность для испытуемого p распознавания им терма (формулы или доказательства) системы T.

4. Балл i фиксируется (при этом i<k) одной из шкал TS, FS или PS.

5. В качестве m3(p,T) и соответственно m2(p,T) и m1(p,T) берется натуральное число, на единицу меньшее, чем длина (i+1)-го «терм-калибра» соответствующей «терм-калибровки» . Аналогичные действия проводятся для m2 и m1.

6. Расчеты ведутся по формулам:

Для термов: $m_3(p,T) = |a_{i+1}| - 1$ или $m_3(p,T) = \sum \Delta(|a_j|)$.

Для формул: $m_2(p,T) = |b_{i+1}| - 1$ или $m_1(p,T) = \sum \Delta(|c_j|)$.

Для доказательств: $m_1(p,T) = |c_{i+1}| - 1$ или $\overline{m_1(p,T)} = \sum \Delta(|c_j|)$.

При измерении m1(p,T) необходимо учитывать следующее:

1. Слово «убедительность» в m1 играет ту же роль, что и слово «уверенность» в определении m2 и m3;

2. Алфавит A1 должен быть расширен на один символ-разделитель – символ пробела до алфавита.

Пусть L – язык (первого порядка с равенством) системы T; A1 – алфавит L; σ – сигнатура L. Предполагается, что сигнатура σ конечна. Пусть, далее, x, y, z, x1, y1, z1, … - переменные для слов (т. е. для конечных, включая пустую, последовательностей символов) алфавита A1; a, b, c, a1, b1, c1, … - константы для них же. Для любого слова x алфавита A1 пусть |x| обозначает длину этого слова.

При распознании пользователем конечной последовательности x символов алфавита A1 как терм или не терм сигнатуры σ, субъективное решение является всегда завершающим актом этого процесса, в общем случае сопровождающееся некоторыми сомнениями. Степенью уверенности распознавания x определяется их интенсивность, уверенность тем больше, чем меньше интенсивность сомнений. Но не смотря на это, максимальная уверенность акта распознавания x не может гарантировать безошибочность результата этого акта.

Возникает следующий вопрос: как можно измерить субъективную уверенность человека в правильности признания им слова x термом или не термом?

При распознавании двух произвольных слова a и b, человек p может установить путем самонаблюдения, что:

1) уверенность, связанная с распознаванием a, заметно превосходит уверенность, связанную с распознаванием b;

2) уверенность, связанная с распознаванием b, превосходит уверенность, связанную с распознаванием a;

3) не то и не другое.

Первый случай обозначается фразой «a терм-больше b для p», второй – фразой «a терм-меньше b для p», третий – фразой «a терм-равно b для p» (обозначение).

В таком случае искомое измерение субъективной уверенности схематически можно описать следующим образом.

Вначале отыскивается некая последовательность a1, a2, a3, …, am из m слов a1, a2, a3, …, am в алфавите A1, такая, что a1 терм-больше слова a2 для p, слово a2 терм-больше a3 для p, слово a3 терм-больше последующего и т. д. При некоторых предположениях это можно сделать так, чтобы всякое другое слово x оказалось либо терм-равным для p одновременно только двум смежным словам и из ряда a1, a2, a3, …, am, либо терм-меньшим для p любого слова из этого ряда. Такую последовательность слов будем называть терм-калибровкой для p и обозначать через $TC_{P\sigma}$: $TC_{P\sigma}$ = (a1, a2, a3, …, am). Каждое слово ai, i = 1..m из последовательности $TC_{P\sigma}$ будем называть i-м терм-калибром для p.

Когда терм-калибровка найдена, можно применить ее для измерения субъективной уверенности человека в правильности признания им слова x термом или не термом. Для этого нужно:

a) установить терм равенство (терм-равно) слова x одновременно с каким-нибудь другим смежным терм-калибрам и ?

b) если терм-неравно, то объявляется о том, что субъективная уверенность p в правильности признания им слова x термом или не термом меньше m баллов;

c) если терм-равно, то объявляется что субъективная уверенность p в правильности признания им слова x термом или не термом равна i баллам.

При попытке отыскать $TC_{P\sigma}$ сразу же возникает необходимость преодолеть одно специфическое затруднение, которое можно интерпретировать, как одну изверсией «парадокса кучи».

Пусть $T_a(x)$ означает: данным человеком p слово x распознается как терм или не терм сигнатуры σ ровно с такой же субъективной уверенностью, с какой

им распознается как терм или не терм сигнатуры σ слово a. Иначе говоря, пусть $T_a(x)$ означает: слово x является терм-равным для p слову a. При этом p может установить для себя истинность или ложность предиката $T_a(x)$ только путем самонаблюдения и самооценки своих состояний уверенности в процессе распознавания им слов; что же касается установления для p истинности или ложности предиката $T_a(x)$ внешним наблюдателем, то тут не обойтись без подобающего опроса испытуемого p. На первый взгляд может показаться, что повседневный лингвистический опыт свидетельствует, что свойство «быть равным фиксированному слову a» является размытым в следующем смысле:

Выполняется условие: $\forall xy(|x| \leq |y| \leq |x|+1) \rightarrow (T_a(x) \leftrightarrow T_a(y))$ (1), говорящее о том , что любые два слова x y алфавита A1, разнящиеся по длине не более, чем на один символ, терм-равны или не терм-равны данному слову a одновременно.

Однако это условие не является правильным обобщением экспериментальных данных.: из него следует, с точки зрения фактов, абсурдное заключение, что любое длинное слово алфавита A1 терм-равно данному слову a для данного человека p.

Такое заключение возникает из-за ошибочного принятия условия (1) за выражение смысла размытости предиката $T_a(x)$.

Рассмотрим пример. Задан спектр цветов от красного до фиолетового (рисунок 2).

В окне просмотра цветов требуется указанный цвет сравнить с цветом «красноватый»

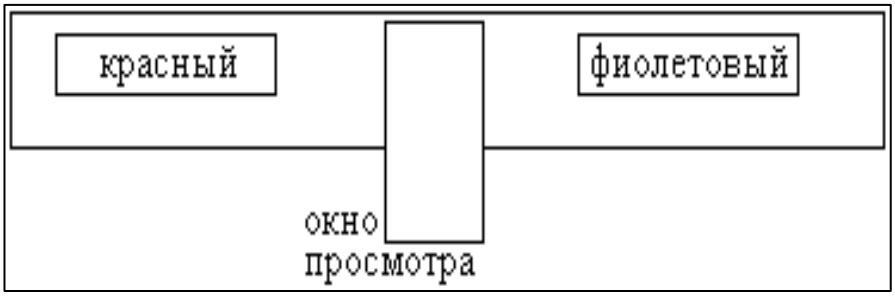


Рисунок 1.13 – Пример демонстрации размытости предиката

В зависимости от цвета выбранного как эталон сравнения будет определяться предикат. В этом и заключается свойство «размытости» предиката $T_a(x)$.

Для работы с предикатом $T_a(x)$ необходимо отталкиваться от следующего:

1) предикат $T_1(x, y)$ является исходным предикатом подлежащим непосредственному наблюдению;

2) предикат $T_1(x, y)$ не транзитивен;

3) предикат $T_1(x, y)$ рефлексивен, то есть результаты наблюдения $T_1(x, y)$ удовлетворяют условию $\forall x T_1(x, x)$

4) предикат $T_1(x, y)$ симметричен, т. е., что результаты наблюдения $T_1(x, y)$ удовлетворяют условию $\forall xy(T_1(x, y) \leftrightarrow T_1(y, x))$;

5) предполагается, что результаты наблюдения $T_1(x, y)$ удовлетворяют условию $\forall xy(|x| \leq |y| \leq |x|+1 \rightarrow T_1(x, y))$;

6) дано конкретное слово a;

7) предикат определяется соотношением $T_a(x) \leftrightarrow T_1(x, a)$;

8) для каждого слова x длины d1 найдется слово длины d2, d2>d1, такое, что $\neg T_1(x, y)$;

9) если x – произвольное слово длины d1, y – произвольное слово длины d2, d2>d1 и имеет место $T_1(x, y)$, то для любого отрезка z длины d, d1<d<d2, имеет место $T_1(x, z)$.

Далее рассмотрим процедуру отыскания терм-калибровки для пользователя p.

В этой связи уместно отметить так называемый «метод наименьших изменений», широко и успешно применяемый в психофизических исследованиях.

Метод наименьших изменений позволяет определить пороги раздражения. Выделят два вида порогов: это абсолютные и разностные.

Абсолютный порог раздражения – это та минимальная величина раздражения когда ощущение, вызванное действием, становится впервые заметным (когда получается «еле заметное» ощущение).

Разностным порогом называется минимальная величина раздражения, на которую нужно уменьшить или увеличить данное раздражение, чтобы впервые заметить хоть какое-то изменение первоначального ощущения (чтобы получить ощущение, едва заметное, отличное от данного).

Простейший способ определения разностного порога состоит в следующем (рисунок 1). Для данного нормального раздражения N создаем (или отыскиваем) переменное раздражение V, заметно большее, чем N, и уменьшаем его до тех пор, пока различие сделается впервые незаметным. Фиксируем полученное при этом значение переменного раздражения. Затем мы его снова увеличиваем, пока оно не покажется впервые снова заметно большим, чем N. Пусть этому соответствует значение . Величина h(N), которая лежит посередине между впервые заметным различием и впервые незаметным различием, есть верхний пункт равенства (в N):

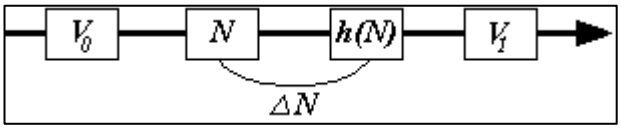


Рисунок 1.14 - Разностные пороги в психофизических измерениях

Величина ΔN, определяемая соотношением $\Delta N = h(N) - N$, называется верхним разностным порогом (в N).

Под стимулом раздражения (для испытуемого) мы будем понимать произвольное слово x (распознаваемое им как терм или не терм), а под величиной этого раздражения, измеряемой в абсолютной шкале, - длину |x| слова x. Под ощущением, вызываемым стимулом раздражения (словом x), понимается теперь субъективно воспринимаемое испытуемым человеком чувство некоторой (возможно, очень малой) уверенности, которой

сопровождается акт распознавания данным человеком слова x как терма или не терма. Если уверенность, связанная с распознаванием x, принимается за данное ощущение, то уверенность, связанную с распознаванием y, мы считаем заметно отличным ощущением от данного, если и только если имеет место ; и мы считаем ее едва заметно отличным ощущением от данного, если и только если имеет место , и для всякого слова z, длина |z| которого является промежуточной между длинами |x| и |y|, имеет место . В соответствии с этими замечаниями следует понимать и смысл заявления, что h(|x|) обозначает верхний пункт равенства в |x|, а обозначает верхний разностный порог (в |x|).

Тогда опишем простейшую реализацию процедуры отыскания терм-калибровки для пользователя p.

1. Берется пустое слово в качестве исходного терм-калибра a1 для p (слово нулевой длины).

2. Для a1 методом минимальных изменений определяем верхний пункт равенства h(|a1|) в |a1|; объявляем это слово вторым терм-калибром a2 для p.

3. Для слова a2 методом минимальных изменений определяем верхний пункт равенства h(h(|a1|)) в h(|a1|); берем произвольное слово в алфавите A1 длины h(h(|a1|)); объявляем это слово третьим терм-калибром a3 для p.

4. Для слова a3 методом минимальных изменений определяется … и т.д.

Осуществляется несколько таких этапов, зависящих от количества необходимых балльных оценок субъективной уверенности испытуемого p в правильности признания им распознаваемых слов термами или не термами.

Таким образом итого этого походя являетс терм-калибровка для p из m терм-калибров a1, a2, a3, …, am (для p).

Тот же принцип применяется для формул и выводов калибровки.Условимся, что уверенность, связанная с распознаванием x, равна i баллам, если и только если |x|≥|ai| и $Ta_i(x)$, и меньше i баллов, если и только если |x|>|ai| и $\neg Ta_i(x)$.

2 АНАЛИТИЧЕСКАЯ ЧАСТЬ

2.1 Постановка проблемы

Проблема – существующие методы не рассматривают оценку интеллектуального ресурса пользователя при решении задач разной сложности в определенной предметной области.

В рамках исследования рассматривается предметная область – интеллектуальный анализ данных при оценке интеллектуального ресурса человека и сопоставление его с интеллектуальными запросами (задачами).

Проблема состоит в том, как и какими методами измерять и сопоставлять интеллектуальные ресурсы человека и интеллектуальные потребности при решении задач.

Необходимость измерения и сопоставления может возникнуть при приеме нового сотрудника, оценке знаний учащегося, визуализации данных и других процессов, в которых необходимо измерить интеллектуальный ресурс человека для оценки его потенциала и возможностей решать определенные классы задач в предметной области.

Целью диссертации является исследование и моделирование процессов оценки интеллектуального ресурса пользователя и сопоставление его с интеллектуальными запросами (задачами). Объектами исследования является человек – пользователь, задача и связь.

Интеллектуальный ресурс пользователя и интеллектуальные запросы являются объектом исследования.

Предметом является оценка интеллектуального ресурса.

Проблемой является отсутствие эффективных методов для оценки и сопоставления интеллектуального ресурса с интеллектуальными запросами.

Целью исследования является оценка интеллектуальных ресурсов пользователей и их сопоставление с интеллектуальными запросами.

Задача: Разработка методов оценки интеллектуальных ресурсов и сопоставления их с интеллектуальными запросами при помощи логико-математического аппарата.

Задача в рамках данной работы – разработать прототип программного средства для оценки интеллектуального ресурса пользователя в заданной предметной области.

Гипотеза: Можно ли оценить интеллектуальный ресурс пользователя и сопоставить его с интеллектуальными запросами

2.2 Анализ программных средств в данной области (задачно-ориентированных)

В качестве области измерения интеллектуального ресурса была выбрана область программирования и разработки информационных систем. В рамках диссертации был проведен анализ существующих решений на рынке, а также рассмотрение их инструментария и возможность поиска решений в поставленной задаче.

Для иллюстрации проблемы в данной области возьмем такой пример: специалист HR в ИТ компании открывает вакансии программистов. В список кандидатов попадает Кандидат А со стажем работы в несколько лет, с наличием диплома крупного вуза. В списке так же имеется Кандидат Б, без опыта, из провинциального вуза. Преимущество дадут Кандидату А, несмотря на тот факт, что Кандидат А мог не получил должного опыта на предыдущем месте работы. Кандидатов Б мог получить необходимый ИТ-компании опыт работая на основе фриланса или получив этот опыт на практических занятиях вуза.

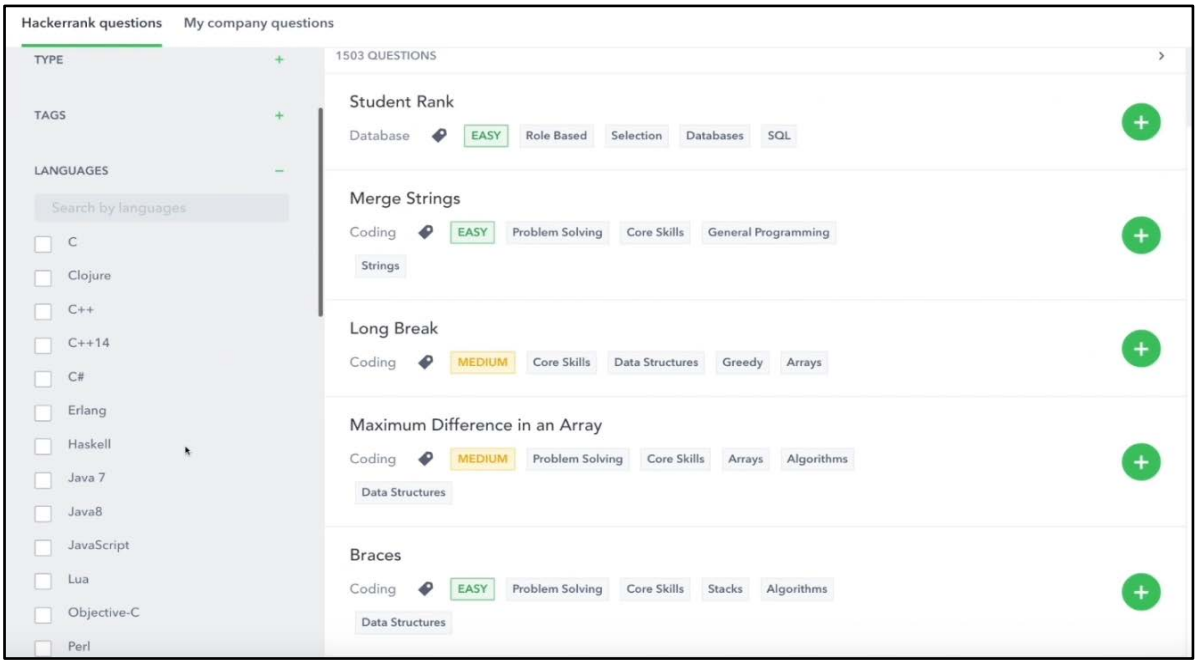


Рис 2.1 - Пример теста

Компания в таком случае выдаст тестовое задание, чтобы не упустить потенциального Кандидата Б. Но тестовое задание требует времени на выполнение и время на проверку, которого может не быть у начальника отдела и HR-специалиста.

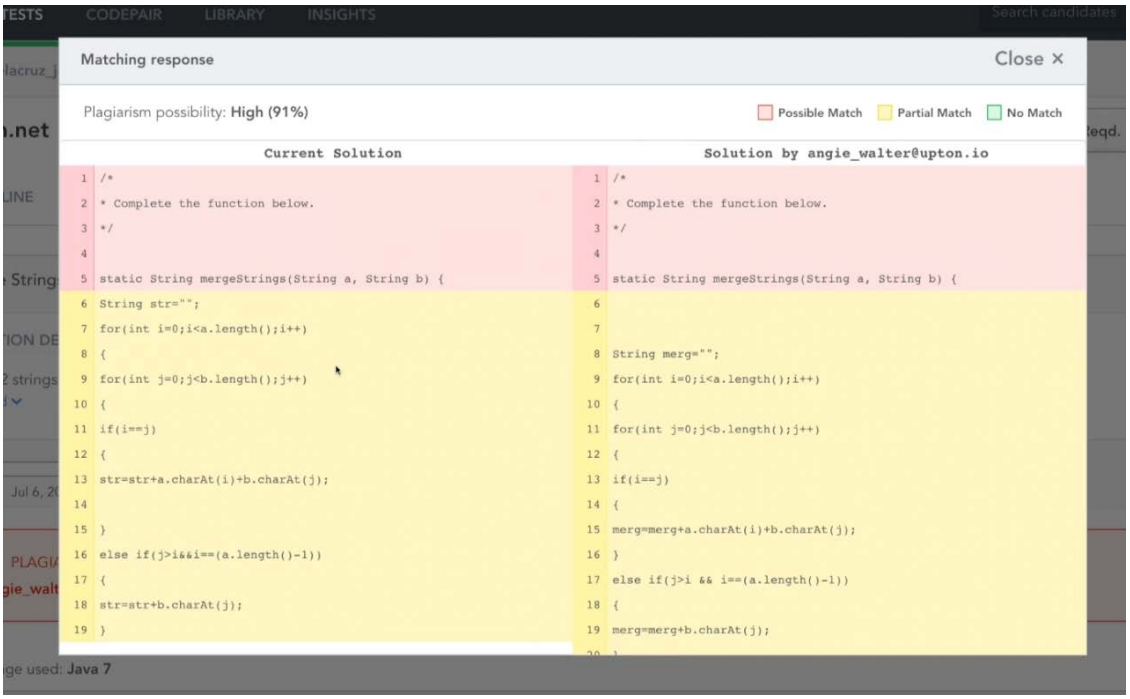


Рис 2.2 - Пример теста

Проблема может скалироваться если Кандидатов Б и Кандидатов А откликнулось несколько сотен человек. В таком случае отбор будет проводиться очень грубо.

HackerRank позволяет оценить технические навыки кандидатов с разным опытом работы, за счет базы с заранее подготовленными или нестандартными задачами. Система позволяет автоматизировать проверку кода, просматривать результаты в реальном времени.

Гораздо эффективнее компании или предприятию оценить интеллектуальный ресурс кандидатов и не упускать ценных кадров.

HackerRank это платформой для поиска и тестирования специалистов, которая является стандартом для оценки навыков разработчиков.

Платформа предоставляет возможность проверки кандидатов через среду кодирования HackerRank, в которой используется заготовленный синтаксис, и проверка успешности выполнения кода.

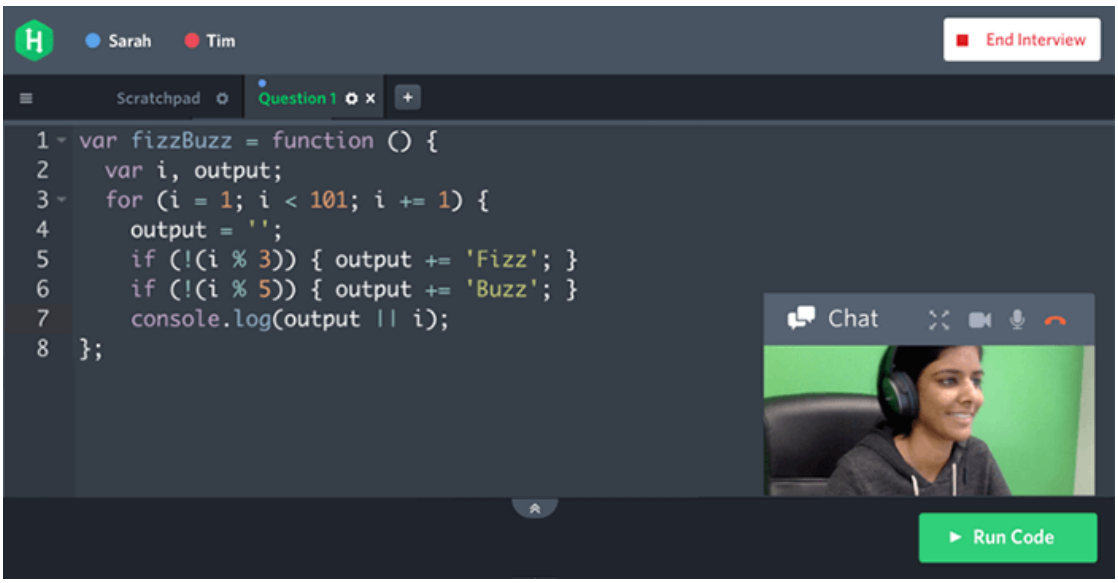


Рис 2.3 - Пример теста

Codility предоставляет онлайн-комнату для совместной работы и кодирования ваших кандидатов в технических интервью в режиме реального времени. Шаблонный режим CodeLive позволяет интервьюерам начать

техническое интервью с заранее выбранными задачами, готовыми к работе. Это улучшает стандартизацию собеседования и гарантирует, что кандидаты получают оценку «яблоки к яблоку», если сессия CodeLive заменяет оценку CodeCheck.

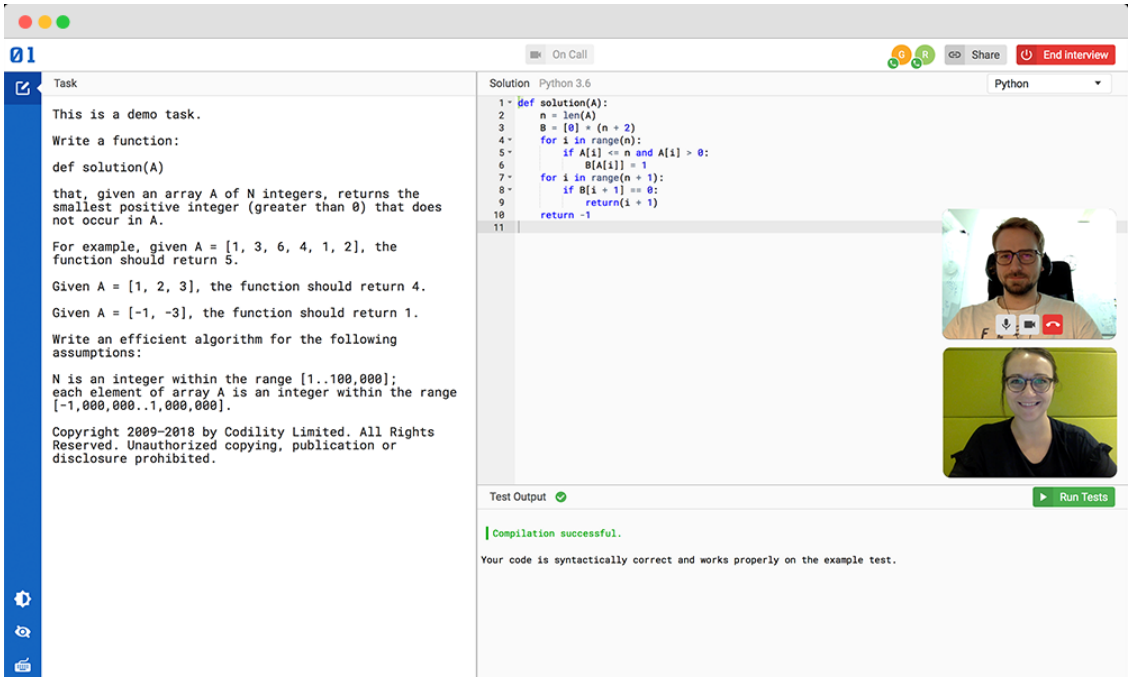


Рис 2.4 - Пример теста

Техническая платформа рекрутинга автоматически оценивает реальные и фундаментальные навыки разработки, поддерживает все популярные языки и структуры и охватывает любые технические роли, которые вы нанимаете. Создайте свои собственные вопросы или выберите из библиотеки Codility.

1. Готовая библиотека высококачественных задач
2. Простое создание пользовательских задач и вопросов
3. Автоматизированная оценка точности и масштабируемости
4. Воспроизведение кода, тестовые решения и оценка стиля

iTest - инструмент HR-менеджеров для аттестации, позволяющий оценить соответствие компетенций сотрудников и занимаемых ими должностей. При проведении опросов сервис iTest.ru гарантирует сотрудникам анонимность,

необходимую для откровенных и правдивых ответов, так как руководству компании доступна только сводная статистика ответов.

Тема теста	Оцениваемые знания
Навыки продавца-консультанта в торговом зале (B2C)	Поведение потребителей в зале, развитие контакта, работа с возражениями розничных клиентов
Навыки менеджера по продажам в отделе продаж (B2B)	Проведение презентаций, развитие (этапы) продажи, работа с возражениями корпоративных клиентов
Навыки переговоров по телефону	Телефонный этикет, работа с возражениями по телефону
Навыки оператора call-центра	Стрессоустойчивость, коммуникативность, телефонный этикет, работа с возражениями по телефону
Основы информационных технологий	Операционная система, форматы документов Microsoft Office, безопасность в Интернет, распространенное п/о для офиса
Работа в Microsoft Excel	Базовая функциональность Microsoft Excel
Работа в Microsoft Project	Базовая функциональность Microsoft Project
Основы тайм-менеджмента	Инвентаризация времени, приоритеты задач, самоконтроль
Основы управления проектами	Календарь проекта, управление ресурсами, контроль за выполнением
Деловая переписка и грамматика русского языка	Составление предложений при переписке общего характера с коллегами и партнерами
Основы безопасности и охраны труда	Основные правила охраны труда и безопасности на предприятии (без специфики производства)
Основы управления персоналом	Потребности и мотивация человека, делегирование задач.

Рис 2.5 - Пример теста

Сервис iTest включает в себя профильные тесты, с поддержкой банка вопросов, из которого случайно выбираются вопросы для тестирования, а так же распространенные личностные тесты, применяемые в практике диагностики персонала, 16PF Кэттелла, или Айзенка.

Таблица 2.1

Сравнительная таблица

	HackerRank	Codility	iTest	Система из ВКР
Тестирование базовых навыков в области ИТ	+	+	+	-
Тестирование навыков испытуемого в области разработки ПО	+	+	-	+
Оценка интеллектуального ресурса	-	-	+	+
Поддержка русского языка	-	-	+	+
Система для наблюдения за кандидатом в реальном времени	+	+	-	-
Система для компиляции и проверки решений в области программирования	+	+	-	Не требуется
Оценка испытуемого в области решения задач	+	-	-	+

Вопрос представления данных в системах DateMining. Пользователей условно можно разделить на три группы исполнителей, разработчиков, стратегов. Каждый решает свои задачи, с разным уровнем сложности. Возникает вопрос, в каком виде предоставлять информацию? Может пользователю необходима логическая структура анализа, а может ему хватит обычной столбиковой диаграммы? Система сможет подстраиваться под интеллектуальные запросы пользователей.

2.3 Используемые системы кодирования и классификаторы

Для классификаторов экономической информации отводится важное место в составе информационного обеспечения среди рассматриваемого комплекса задач. Обеспечение сжатия признаковых признаков, а следовательно, и объема информации, необходимой для решения задач, облегчения обработки информации позволяет классифицировать и кодировать информацию. Процесс присвоения объектам кодовых обозначений называется кодированием. Основной целью кодирования является однозначное обозначение объектов, а также обеспечение достоверной кодируемой информации. При проектировании кодов предъявляется ряд требований:

1. Охват всех объектов, подлежащих кодированию, а также их однозначное обозначение.

2. Возможность расширения объектов кодирования без изменения правил их обозначения.

3. Максимальная информативность кода при минимальном его значении.

Классифицированние - это процесс распределения объектов данного набора в подмножества. Классифицированние является результатом упорядоченного распределения объектов данного набора [5].

Различают иерархическую и многоаспектную системы классификации. Система иерархической классификации включает в себя разделение исходного набора на подмножества, между которыми устанавливаются иерархические

отношения. В зависимости от количества классификационных признаков может быть несколько уровней классификации. Уровень классификации - это набор групп классификации, расположенных на одинаковых уровнях классификации. В многомерных системах классификации несколько независимых признаков используются параллельно в качестве классификации, то есть исходный набор рассматривается одновременно в разных аспектах. Системы кодирования делятся на регистрацию и классификацию. Система регистрационного кодирования используется для идентификации объектов, которые не требуют предварительной классификации и не зависят от характера решаемых задач. Существуют порядковые и серийно-порядковые системы кодирования.

Порядковая система кодирования заключается в последовательном порядке регистрации объектов. Отсутствуют признаки классификации, что впоследствии не позволит получить промежуточные результаты.

В соответствии с приведенными требованиями к кодировке в проекте используется серийная система кодирования, позволяющая кодировать установившиеся несвязанные множества объектов, что дает возможность расширения кодирующего числа и распространения по одному признаку классификации. Для того чтобы обнаруживать ошибки, повторяющиеся вычисления были проверены. Поэтому они обнаруживаются с помощью корректирующего кода.

Классификационные коды. При проведении предварительной классификации объектов применяются методы классификационного кодирования. Можно выделить следующие методы классификации, такие как последовательное кодирование, берущее в основу иерархическую систему классификации и параллельное кодирование которое основывается на фасетной системе классификации.

Идентификационные коды. Если объекты не классифицированы, методы идентификационного кодирования используются для однозначной идентификации объектов. Для идентификации объектов используется порядковая или серийно-порядковая нумерация объектов.

Смешанные коды. Для некоторой номенклатуры объектов используются коды, которые содержат как классификационные, так и идентификационные части, которые не зависят друг от друга. В других случаях коды используются для идентификации объектов в классификационных группах.[7].

Следует также отметить, что кодирование, как правило, приводит к значительному сокращению объема используемых данных. В этом случае особая роль отводится методам классификации экономической информации. Это объясняется тем, что разнообразие форм и ценностей, которые могут приобретать различные экономические показатели, используемые в системах управления национальными экономическими объектами на разных уровнях, обусловливает необходимость использования определенных принципов систематизации этой информации с целью обеспечения ее хранения. Поиск и использование в процессе подготовки управленческих решений.

2.4 Обоснование проектных решений по видам обеспечения

Проектные решения должны быть обоснованы по трём видам:

- по техническому обеспечению;
- по информационному обеспечению;
- по программному обеспечению.

Ряд характеристик определяет правильный выбор электронного компьютера (компьютера). И именно этими характеристиками нужно руководствоваться, полагаясь на них при выборе необходимого компьютера. Стоимость затрат, простота использования, надежность, производительность и др., Именно на этих характеристиках следует основываться. Способность работать с созданным программным обеспечением и прямой успех создания системы будут зависеть от этих параметров.

Сейчас на рынке представлено несколько классов компьютеров: персональные компьютеры (ПК) и рабочие станции, серверы, мэйнфреймы и кластерные архитектуры.

Класс персональных компьютеров очень уверенно и быстро занял хорошую позицию на рынке компьютеров и компьютеров благодаря своей низкой стоимости, что в конечном итоге привело к созданию новых программных средств, ориентированных на пользователя. Именно отсюда исчезли все «дружественные пользовательские интерфейсы», различные инструменты, упрощающие разработку программ, и в то же время проблемные среды.

Использование компьютеров класса мэйнфреймов и кластерной архитектуры может быть исключено. Это слишком дорогие устройства для решения задач в ходе диссертации. Их возможности также используются не полностью. Все это перевесит положительный эффект от их использования.

Поскольку в наше время компьютеры широко распространены и дешевы, выбор был сделан в пользу персональных компьютеров, они также являются IBM PC и совместимы с ними. Чтобы решить эту проблему, этот конкретный класс компьютеров подходит, потому что у него есть все возможности для этого.

К достоинствам таких эвм относятся:

- небольшая стоимость персональных компьютеров относительно других классов ЭВМ;
- диалоговое взаимодействие с пользователем, что обеспечивает простоту использованию и быстрое освоение;
- высокие возможности касательно переработки информации и работе с данными;
- интегрированные компоненты обеспечивают простой и недорогой ремонт, а также высокую надёжность;
- огромный выбор периферийных устройств и программного обеспечения;

По техническому обеспечению компьютер должен соответствовать минимальным требованиям для установки на него операционной системы Windows 7:

- центральный процессор с частотой 1ГГц и выше;

- 2GB RAM;
- 20 ГБ места на HDD/SDD.

Программное обеспечение, разрабатываемое в ходе диссертации, работает на версии операционной системы Windows 7 и выше. Эта обусловленность будет более подробно описана в разделе программного обеспечения.

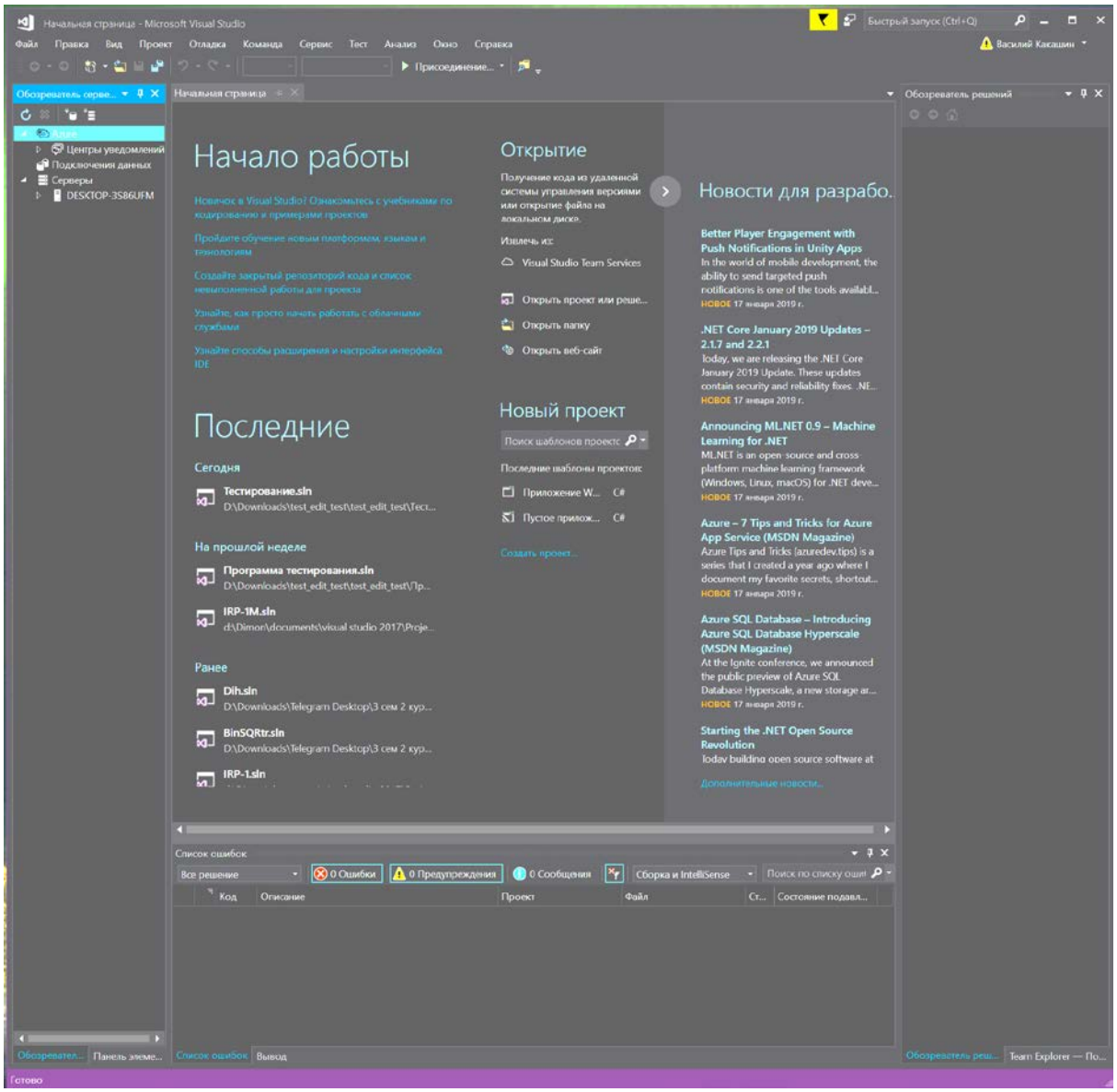


Рисунок 2.6 - Главноеокно Microsoft Visual Studio 2017

В прототипе системы реализованы модули импорта файлов таких форматов, как: .doc, .xls, .accdb, .mdb, .txt. Список импортируемых форматов будет расширен. Разработан модуль кластерного анализа k-means, основанной

на мере близости объектов. В будущей разработке планируется использовать логическую систему, основанную на предикатах первого порядка, для выделения таксонов (кластеров) в массивах данных.

В качестве проектного решением в области языка разработки был выбран C#. Средой разработки будет являться MicrosoftVisualStudio 2017.

MicrosoftVisualStudio — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств.

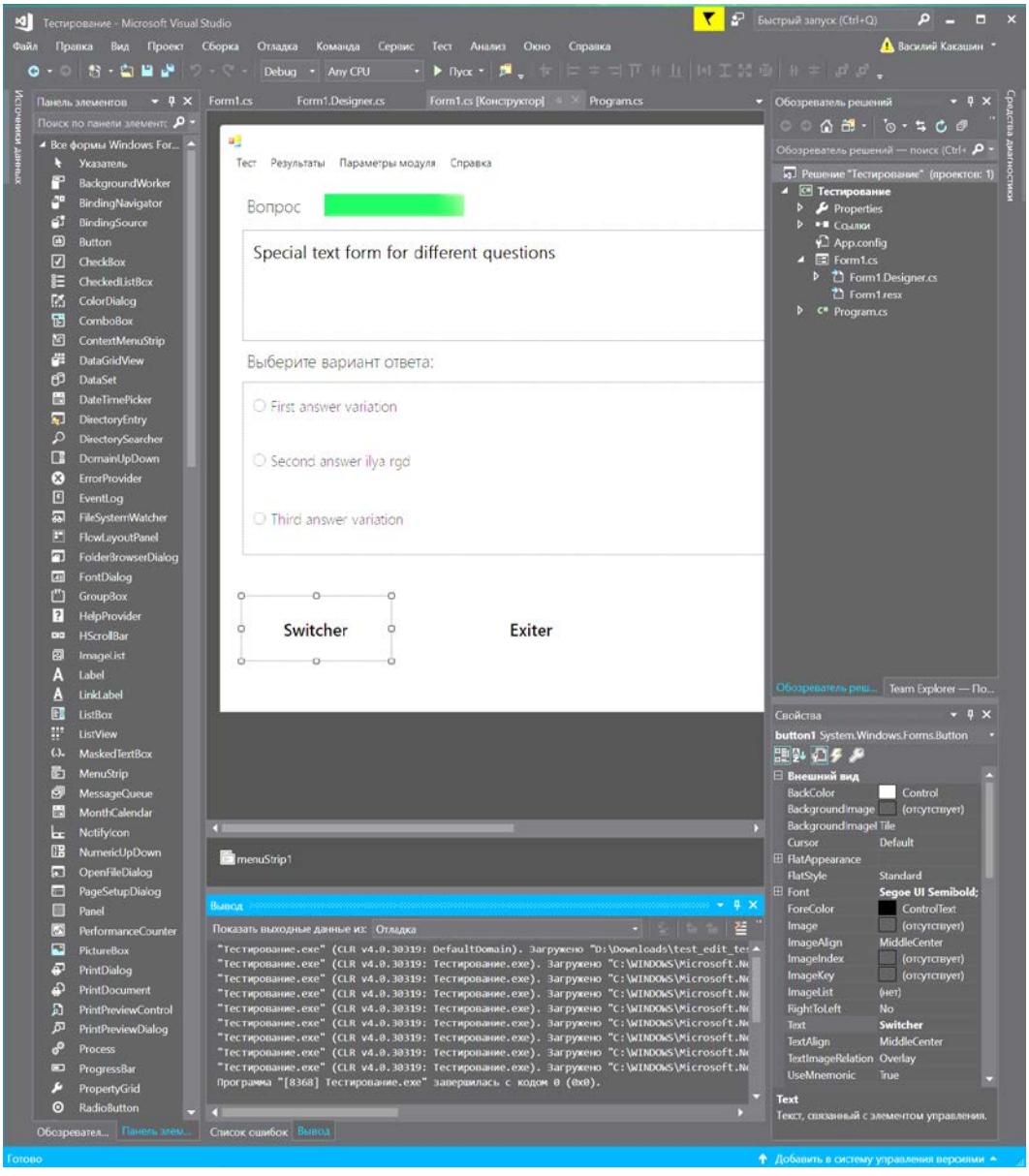


Рисунок 2.7 - Окно визуального редактора среды MicrosoftVisualStudio 2017

Данныепродуктыпозволяютразрабатыватькакконсольныеприложения, такиприложениясграфическиминтерфейсом, втомчислесподдержкойтехнологииWindowsForms, атакжевеб-сайтыдлявсехплатформ, поддерживаемыхWindows, WindowsMobile, WindowsCE, .NETFramework, Xbox, WindowsPhone .NETCompactFrameworkиSilverlight.

VisualStudio включает в себя редактор кода, поддерживающий IntelliSense (компонент завершения кода), а также рефакторинг кода.

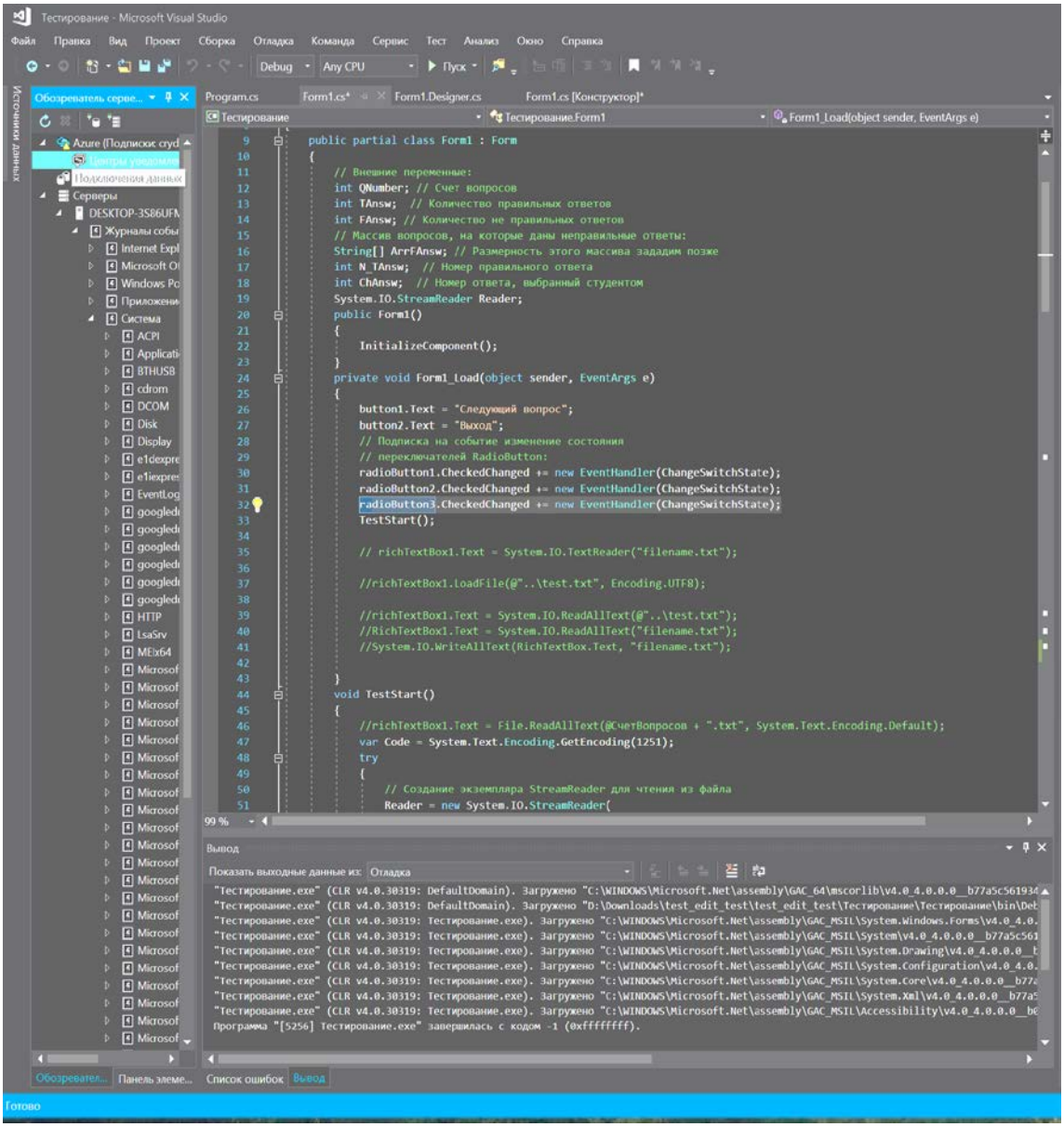


Рисунок 2.8 - Окно редактора кода MicrosoftVisualStudio 2017

Интегрированный отладчик работает как отладчик уровня источника и отладчик уровня машины.

Другие встроенные инструменты включают профилировщик кода, конструктор форм для создания приложений с графическим интерфейсом, веб-дизайнер, дизайнер классов и конструктор схем базы данных.

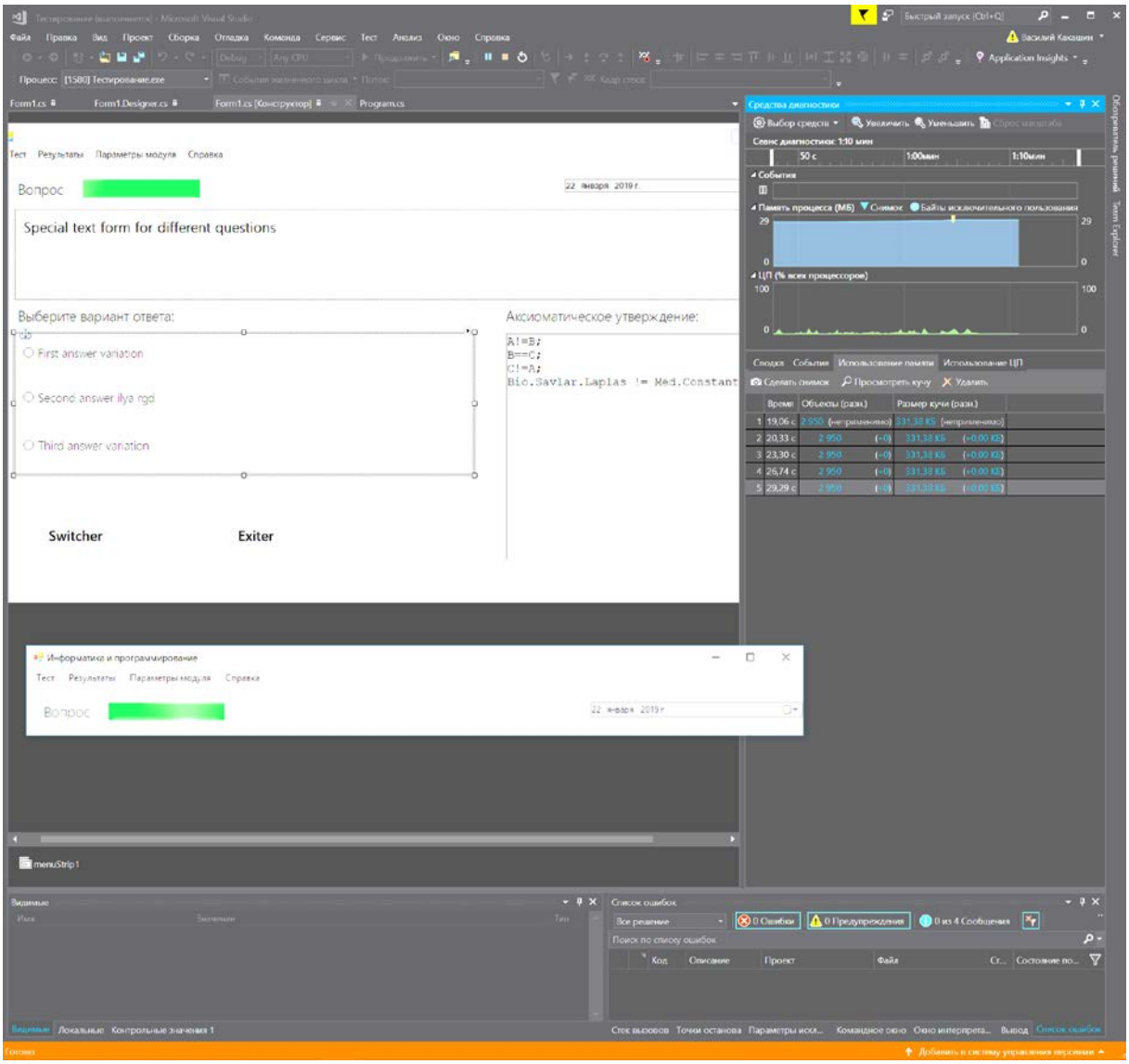


Рисунок 2.9 - Окно диагностики и отладки MicrosoftVisualStudio 2017

Он принимает плагины, которые расширяют функциональные возможности практически на каждом уровне, включая добавление поддержки систем контроля версий (таких как Subversion и Git) и добавление новых

наборов инструментов, таких как редакторы и визуальные дизайнеры для языков, специфичных для предметной области, или наборов инструментов для других аспектов разработки программного обеспечения. жизненный цикл (например, клиент TeamFoundationServer: TeamExplorer).

.NET Framework - программная платформа, выпущенная компанией Microsoft в 2002 году. Основой платформы является общеязыковая среда исполнения CommonLanguageRuntime (CLR), которая подходит для разных языков программирования. Функциональные возможности CLR доступны в любых языках программирования, используя эту среду.

Несмотря на то, что .NET является патентованной технологической корпорацией, Microsoft и официально рассчитанные на работу в операционных семьях MicrosoftWindows предоставляют независимые проекты (в первую очередь, Mono и Portable.NET), позволяющие запускать программы .NET на некоторых других бесплатных программах. В настоящее время .NET Framework получает развитие в виде .NET Core, изначально предполагающей кроссплатформенную разработку и эксплуатацию.

Программа для .NET Framework, написанная на любом совместимом языке программирования, сначала переводит компилятор в единый для .NET промежуточный байт-код CommonIntermediateLanguage (CIL) (ранее назывался MicrosoftIntermediateLanguage, MSIL). Общедоступная среда выполнения (CLR) либо транслируется утилитой NGen.exe в исполняемый код для целевого процессора. Виртуальные машины являются обязательными. В случае использования виртуальной машины CLR встроенный в неё JIT-компилятор «на лету» (точно вовремя) преобразует промежуточный байт-код в машинные коды нужного процессора. Современная технология динамической компиляции позволяет достичь высокого уровня быстродействия. Виртуальная машина CLR также заботится о безопасности, управлении памятью и системе исключений.

Достоинствами выбранного языка является:

1. полностью объектно-ориентированный;

2. мощный язык с возможностью наследования и универсализации;
3. огромное количество полезных и удобных библиотек;
4. удобство отладки;
5. удобное взаимодействие со всеми продуктами, выпускаемыми компанией microsoft;
6. активное развитие платформы;
7. удобство сборки;
8. защищенность и контроль версий подключаемых алгоритмов;
9. увеличение надёжности по сравнению с C++ и C .

Одной из главных особенностей, диктующих выбор языка C# является наличие большого количества библиотек, как встроенных в среду разработки, так и подключаемых из сети. Эта особенность делает процесс разработки мобильным, продуктивным и менее ресурсозатратным, что очень актуально на сегодняшний день.

Выводы по разделу:

В разделе была проанализирована деятельность и структура ИП Шевченко, а так же взаимодействие подразделений предприятия, определены задачи аналитика. Были изучены организационная и техническая структура предприятия. Изучено используемое программное обеспечение и технические средства, используемые для решения конкретных задач. Работа аналитика на данном промежутке времени работы предприятия занимает длительное время, большинство расчетов проводится вручную. В связи с этим было предложено решение этой проблемы. В ходе анализа существующего программного обеспечения, был сделан вывод, что большая часть существующих предложений не позволяют эффективно решать задачи и зачастую имеют высокую стоимость лицензии. Поэтому было принято решение разработки системы «Познание», сопоставляющее интеллектуальные запросы пользователя и его интеллектуальный ресурс.

3 ПРОЕКТНАЯ ЧАСТЬ

3.1 Информационная модель и ее описание

Информационная модель – это взаимосвязь входных, промежуточных и результатных информационных потоков, и функций предметной области (структурно-функциональной диаграмма или диаграмма потоков данных). В описании информационной модели необходимо объяснить, какие входные документы берутся в основу и какие нормативно-справочные информационные базы используются при выполнение функций обработки данных.

Цель информационного моделирования — создать точное и полное отображение реального мира, используемое в дальнейшем в качестве источника информации для построения БД.

3.2 Характеристика нормативно-справочной и входной и выходной оперативной информации

Под входной информацией понимается вся информация, необходимая для решения задачи и расположенная на различных носителях: первичных документах, машинных носителях, в памяти персонального компьютера.

Эффективность и результативность управления зависят от рациональной организации ввода информации предприятия, методов сбора, записи, передачи, хранения и обработки информации, ее состава и своевременного получения.

Данные импортируются из различных доступных источников, что позволяет расширить количество типов файлов, с которыми может работать программа. Таким образом, используя форму импорта данных, пользователь может получить необходимые данные для дальнейшего анализа, выбрав файл нужного ему формата.

Полученная информация будет обработана данными, визуализированными в форме, подходящей для решения проблемы. Пользователь получает подтверждение или опровержение своих гипотез, что позволит принимать управленческие решения на его основе и устанавливать зависимость одних признаков от других.

Пользователь может оценить, насколько результаты связаны друг с другом и насколько они зависят друг от друга, что уменьшит пространство функций в больших базах данных и примет различные решения на основе зависимостей, где количество функций относительно невелико.

3.3 Моделирование информационных процессов

Целостная и достаточно подробная модель может быть получена с использованием специальных методологий структурного анализа, таких как IDEF. Согласно синтаксису, модель IDEF0 представляет собой набор иерархически выровненных диаграмм, каждая из которых представляет собой описание процесса (действия).

Практика у ИП Шевченко дала возможность понять специфику рабочего процесса на предприятии. Для описания работы предприятия был использован инструмент CASE, который автоматизирует построение структурных моделей - Bpwin.

IDEF0 - это функциональная модель, предназначенная для описания бизнес-процессов на предприятии, которая позволяет понять, какие объекты или информация являются сырьем для процессов, какие результаты приводят к работе, каковы управляющие факторы и какие ресурсы для этого необходимы.

Диаграммы потоков данных (DFD) используются для описания рабочего процесса и информации о процессе. Как и IDEF0, DFD представляет модель системы как сеть взаимосвязанных действий. Их можно использовать как дополнение к модели IDEF0 для более наглядного отображения текущих операций рабочего процесса в корпоративных системах обработки

информации. DFD описывают функции обработки информации, документы, средства и сотрудников или отделы, которые участвуют в обработке информации.

Элемент	Описание	Нотация Йордона-Де Марко	Нотация Гейна-Сарсона
Функция	Работа.	Имя функции Номер	Имя функции Номер
Поток данных	Объект, над которым выполняется работа. Может быть логическим или управляющим. (Управляющие потоки обозначаются пунктирной линией со стрелкой).	Имя объекта	Имя объекта (Понятие управляющего потока отсутствует)
Хранилище данных	Структура для хранения информационных объектов.	Имя объекта	Имя объекта
Внешняя сущность	Внешний по отношению к системе объект, обменивающийся с нею потоками.	Имя внешнего объекта	Имя внешнего объекта

Рисунок 3.1 - Синтаксис DFD

Синтаксис DFD включает, помимо задач и стрелок, дополнительные элементы: внешний объект, который служит для представления объектов, внешних по отношению к проектируемой системе (например, клиент, бухгалтерия, каталоги), и хранилище данных.

Важной задачей решаемой аналитиками, является задача кластеризации данных, учитывающая работу в разных шкалах. Работа аналитика требует согласования интеллектуальных запросов пользователя с его интеллектуальными возможностями.

Для решения этой задачи необходимо смоделировать деятельность предприятия «КАК ЕСТЬ», провести анализ построенной модели и провести реорганизацию деятельности (если это необходимо). Для этой цели

воспользуемся технологией SADT. Чтобы увидеть все входные и выходные воздействия, а также механизмы управления и прочее.

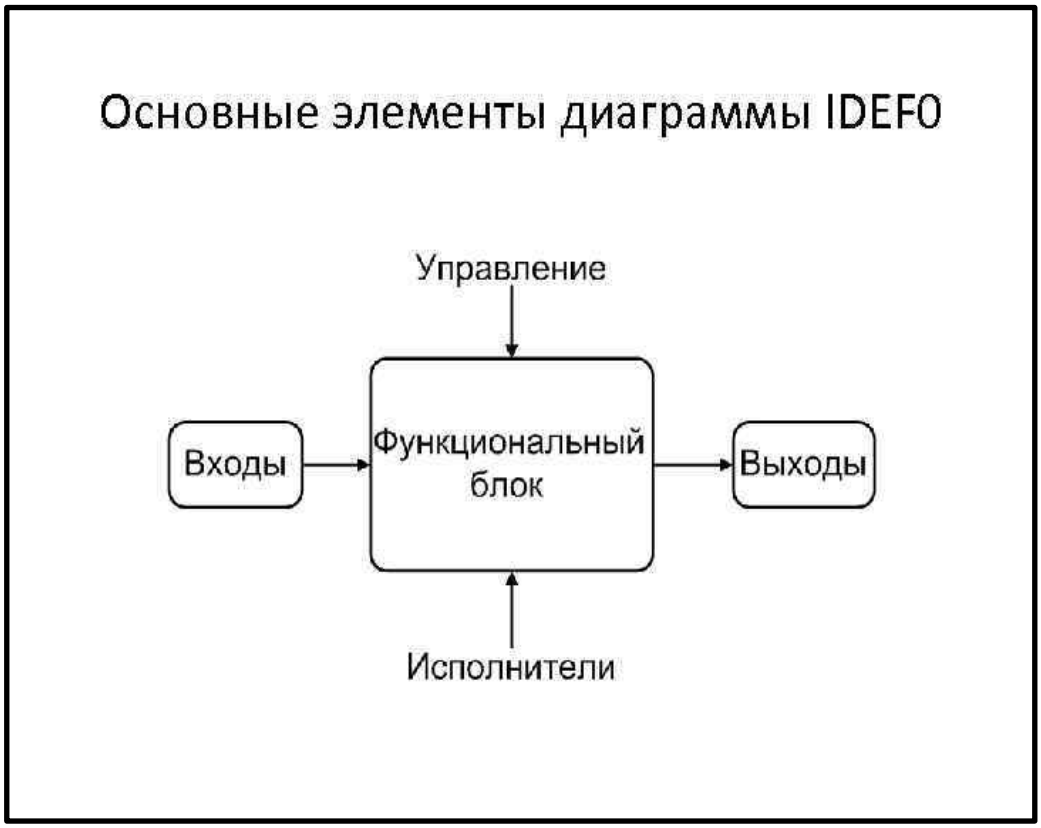


Рисунок 3.2 - Структура графической нотации IDEF0

Сначала построим контекстную диаграмму со звездообразной топологией, в центре которой находится так называемый главный процесс, соединенный с приемниками и источниками информации, посредством которых с системой взаимодействуют пользователи и другие внешние системы.

Для моделирования процессов оценки интеллектуального ресурса пользователя и сопоставление его с интеллектуальными запросами была выбрана графическая нотация IDEF0 и DFD.

Диаграммы процессов были смоделированы в BPWin. Главная диаграмма выполнена в нотации IDEF0. Верхним уровнем абстракции графической модели выбирается сопоставление интеллектуального ресурса пользователя с интеллектуальным запросом.

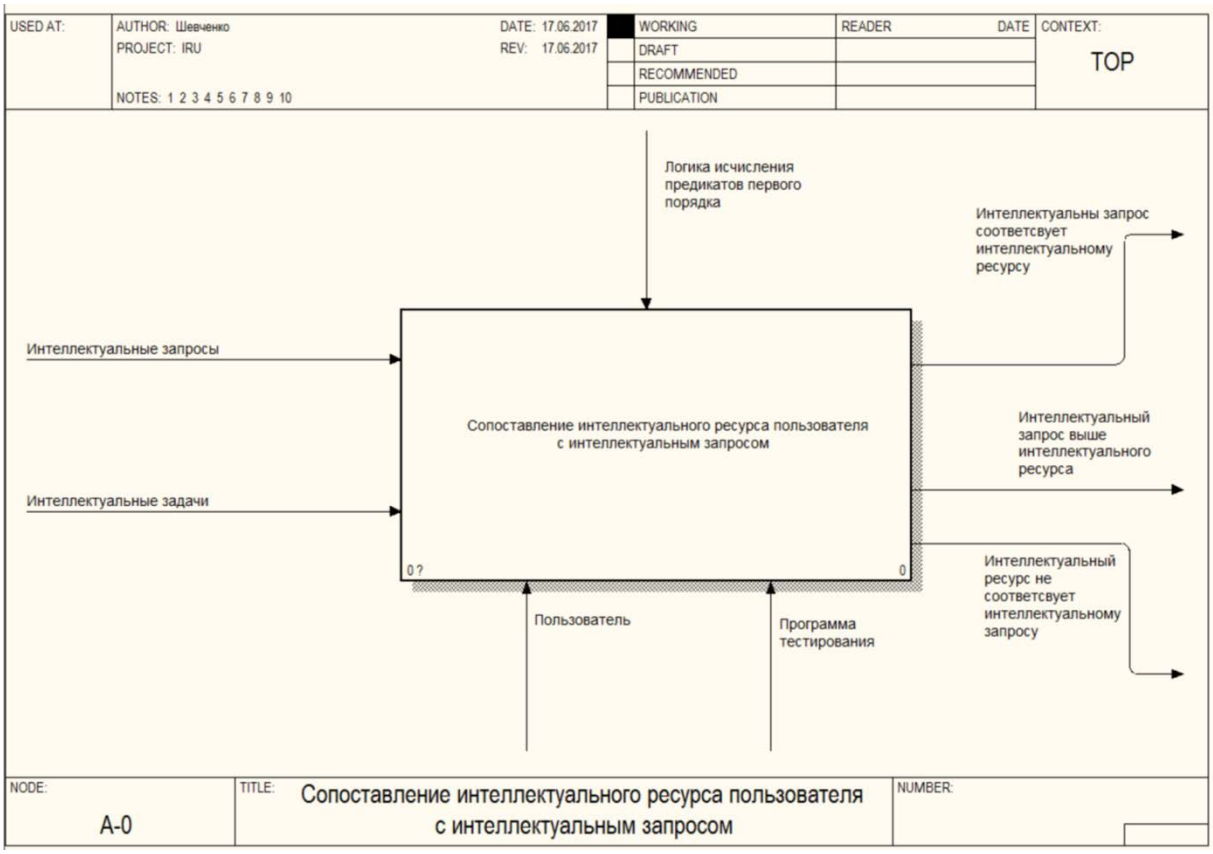


Рисунок 3.3 – Главная диаграмма процесса

В процесс входят интеллектуальные запросы,основанные на осмысленных, интеллектуальных задачах. Весь процесс контролирует логика исчисления предикатов первого порядка, необходимая для создания термов, связывания их в формулы и получения выводов. Терм-формулы с выводом необходимы для проверки глубины интеллектуального ресурса пользователя, что будет рассмотрено в декомпозиции процесса.

На выходе выдаются результаты сопоставления интеллектуальных ресурсов с интеллектуальными запросами. Это соответствие запросам, их несоответствие в большую или меньшую сторону. Результаты необходимы для классификации пользователя в одну из групп, сформированных на основе глубины интеллектуального ресурса.

Главная диаграмма имеет декомпозицию, включающую в себя три подпроцесса. Это «Выбор предметной области», «Загрузка базы знаний» и «Оценка интеллектуального ресурса» (Рис. 3.3.4):

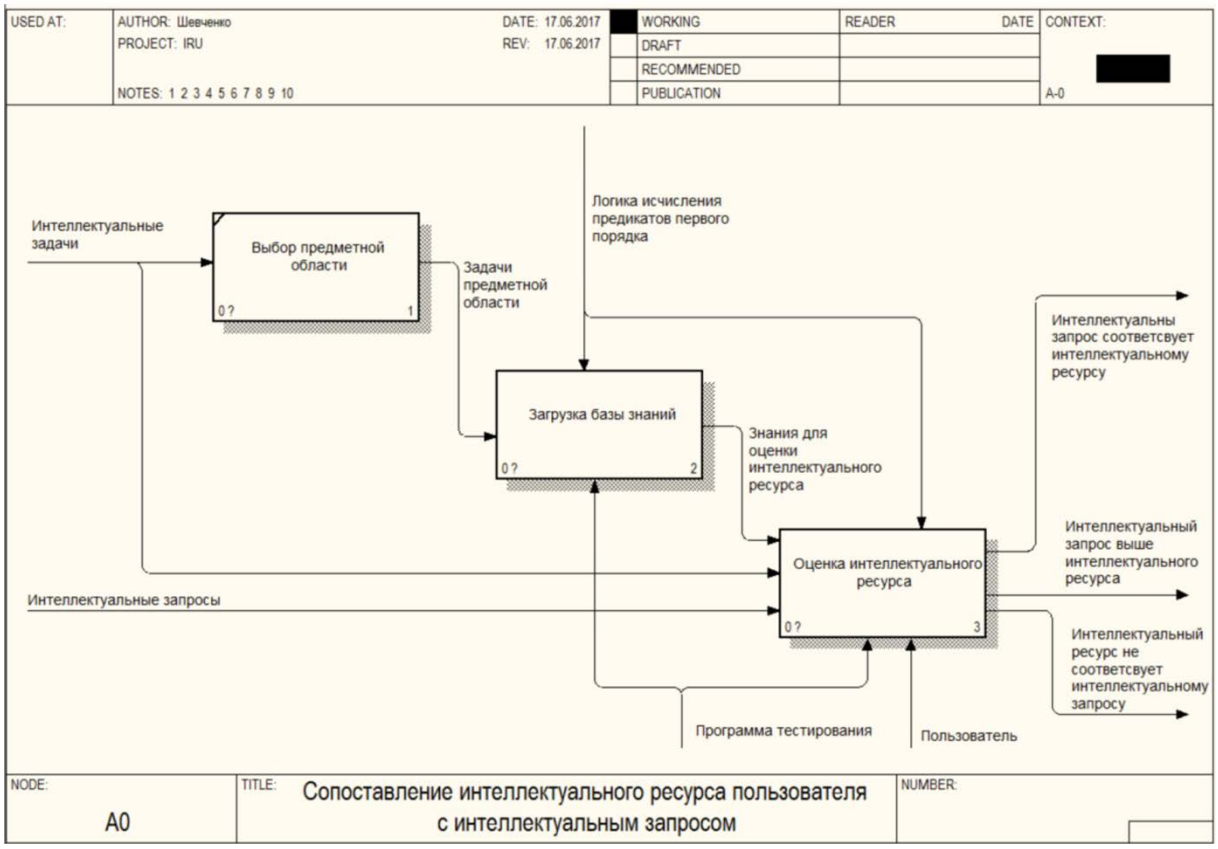


Рисунок 3.4 – Декомпозиция главной диаграммы процесса

Процесс «Выбор предметной области» определяет круг решаемых задач, на основе которых будут построены знания.Полученные задачи в рамках предметной области обрабатываются внутри процесса «Загрузка базы знаний». Этот процесс необходим для формирования базы знаний, на основе которой будут построены логические формулы и предикаты необходимые для измерения интеллектуального ресурса. Контролирует процесс логика исчисления предикатов первого порядка.

Блок оценки интеллектуального ресурса использует интеллектуальные запросы для их сопоставления, знания из предыдущего процесса из базы знаний. На выходе получаются выводы, оценивающие уровень интеллектуального ресурса пользователя.

Процессы загрузки базы знаний и оценки интеллектуального ресурса проходят при использовании программы тестирования.

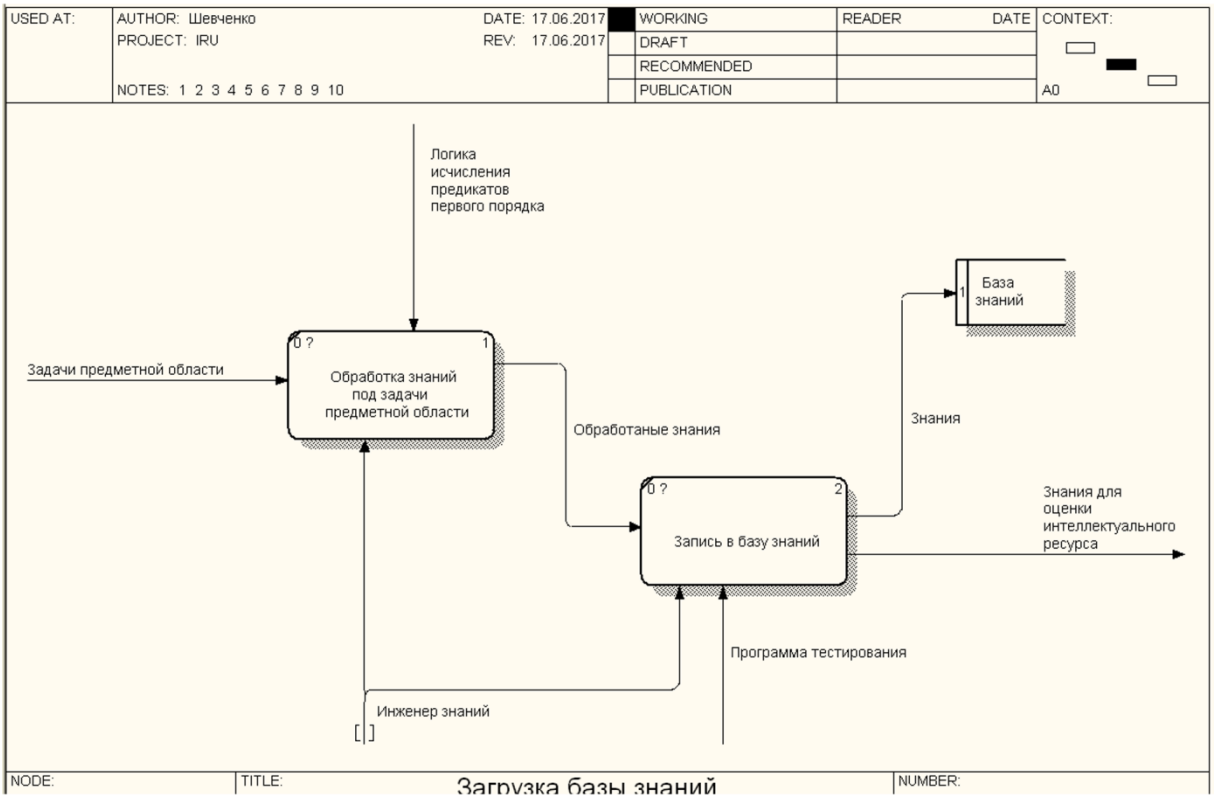


Рисунок 3.5 – Процесс загрузки базы знаний

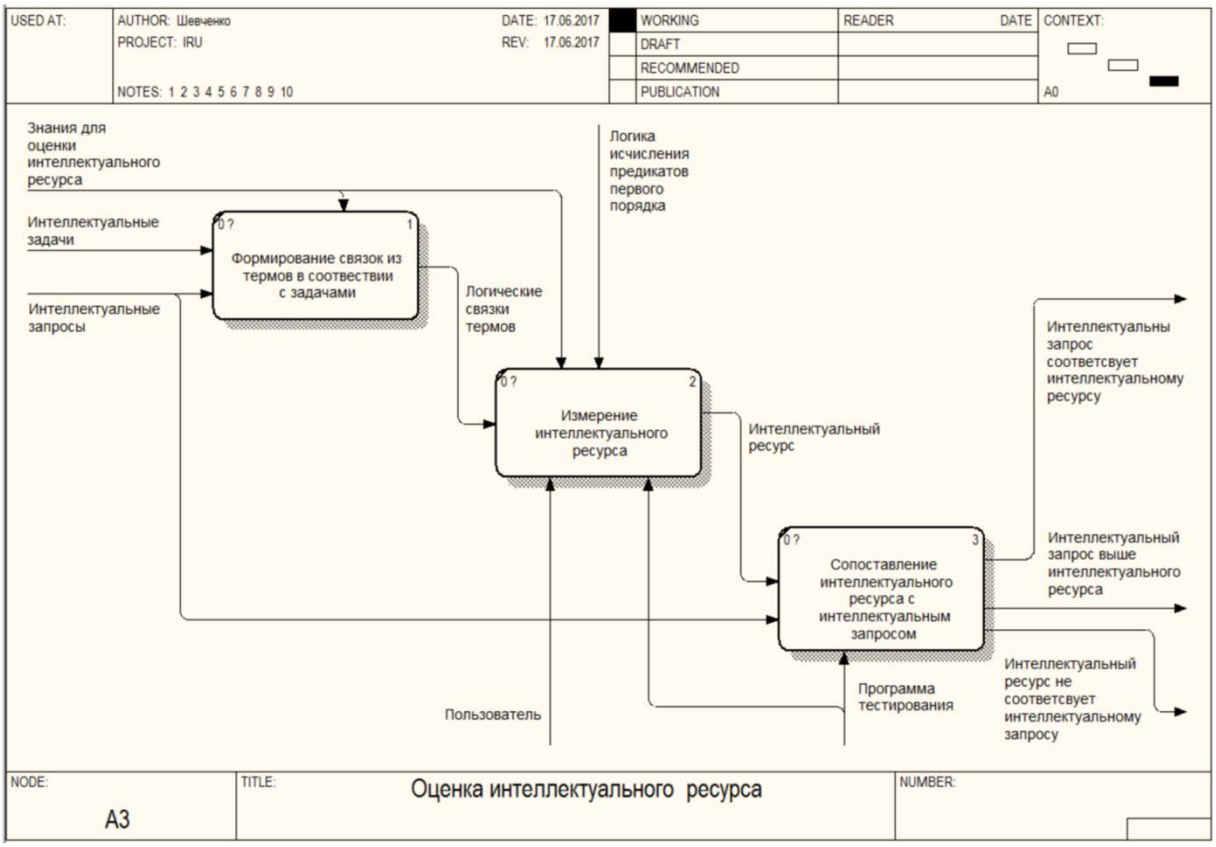


Рисунок 3.6 – Декомпозиция процесса оценки интеллектуального ресурса

Для описания процесса загрузки базы знаний используется нотация DFD. В процессе принимает участие инженер знаний для формирования и наполнения базы знаний.

Первым процессом является обработка знаний под задачи предметной области. Процесс контролируется логикой исчисления предикатов первого порядка. На выходе получаются обработанные знания, которые записываются в базу знаний. Знания отправляются в процесс тестирования и оценки интеллектуального ресурса пользователя.

Процесс оценки интеллектуального ресурса выполнен в нотации DFD. Процесс состоит из трех блоков:

1. Формирование связок из термов в соответствии с задачами,
2. Измерение интеллектуального ресурса,
3. Сопоставление интеллектуального ресурса с интеллектуальными запросами.

На основе полученных знаний формируются связки термов. Из этих логических связок строятся однозначные выводы.

Программа тестирования использует полученный интеллектуальный ресурс и сопоставляет его с интеллектуальными запросами. На основании этого сравнения строятся результаты.

3.5 Пользовательский интерфейс

Интерфейс пользователя является связывающим звеном между компьютером и самим пользователем. Пользовательский интерфейс - это набор инструментов и методов, с помощью которых человек (пользователь) может взаимодействовать с набором сложных элементов и устройств.

Пользовательский интерфейс имеет не только общий дизайн и внешний вид. Пользователь видит всю программу целиком и взаимодействует через нее.

Потому что концепция пользовательского интерфейса исключительно как внешний вид программы слишком узка.

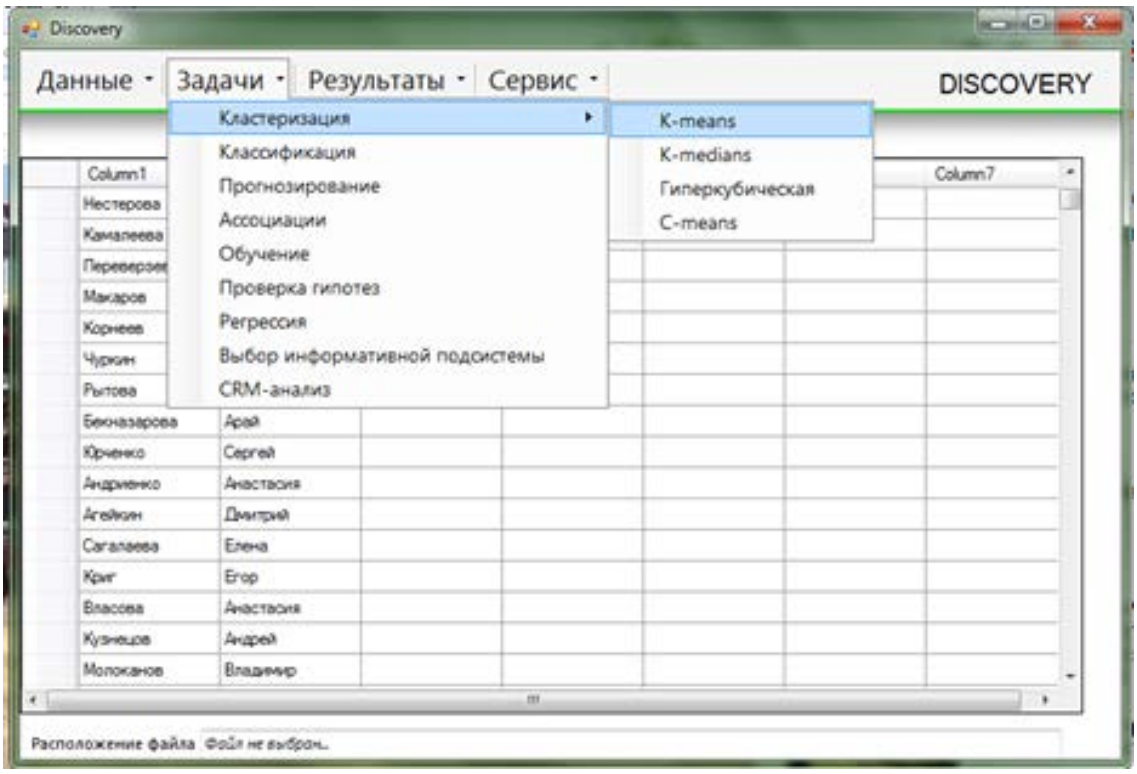


Рисунок 3.7 – Главное окно приложения «Познание»

Ключевым показателем любого программного обеспечения является его удобство использования. Эта концепция включает в себя такие характеристики, как понятность пользовательского интерфейса, простота изучения, пользовательский опыт работы с программным обеспечением, сложность решения проблем с ним, а также частота ошибок и сбоев. Для разработки удобных программ необходимо учитывать контекст их использования, готовность пользователя и его глубину знаний в предметной области, в которой работает программа. Но самый важный фактор - помогает ли данная программа решать действительно важные для пользователей задачи.

Целью создания эргономичного интерфейса является отображение информации наиболее эффективным для человеческого восприятия способом и структурирование отображения интерфейса на экране монитора таким образом,

чтобы привлечь внимание к наиболее важным частям информации. Основная цель - минимизировать общую информацию на экране и представлять только то, что нужно пользователю.

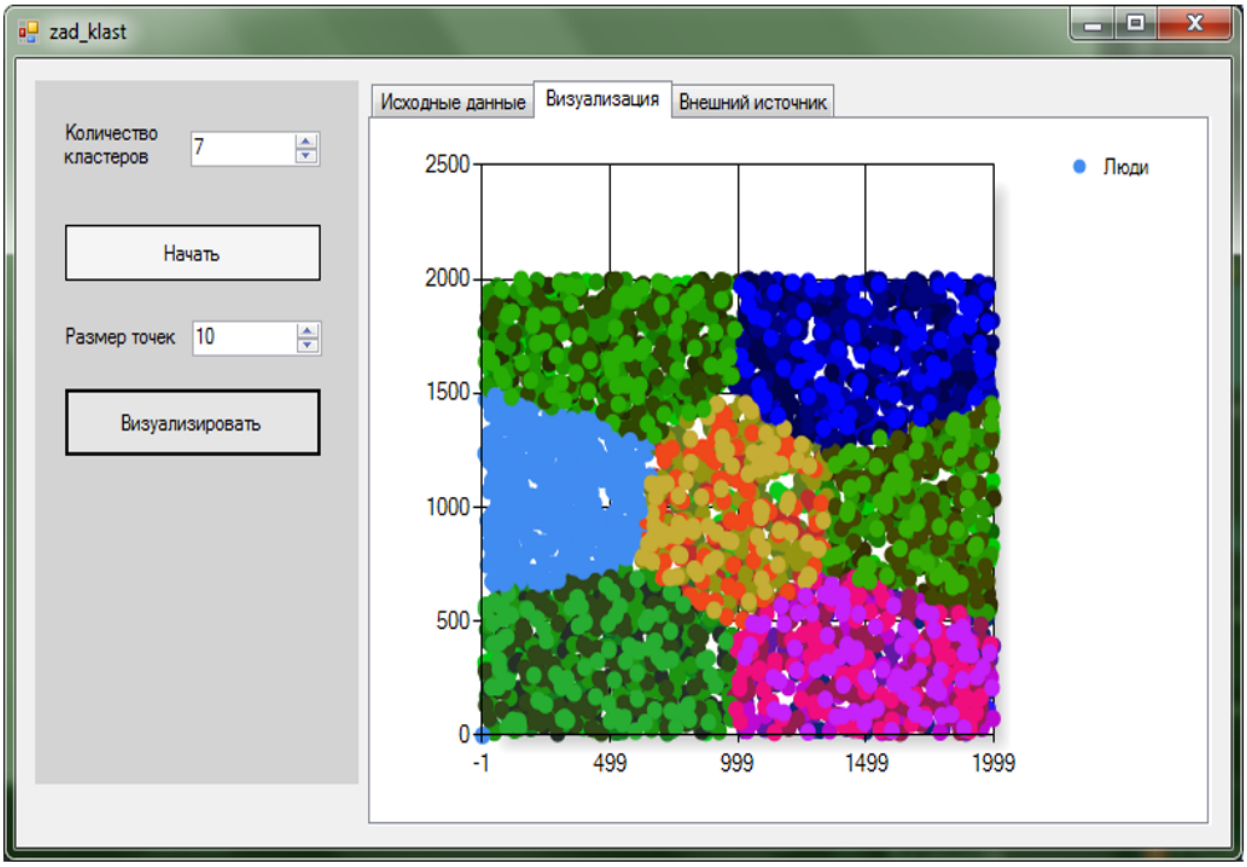


Рисунок 3.8 – Визуализация результатов кластерного анализа

Формы - основной элемент интерфейса. Назначение форм - удобный ввод и просмотр данных, статуса, сообщений, автоматизированная система. Форма предназначена для более удобного, более понятного и быстрого достижения решения проблемы.

Главное окно, показанное на рисунке 3.4.3 состоит из раскрывающегося меню в верхней части. Также в зависимости от выбранных задач раскрывается меню инструментов необходимых для аналитики.

Размещение информационных блоков в пространстве формы должно соответствовать логике его будущего использования: оно зависит от необходимой последовательности доступа к информационным блокам, а также

от относительной важности элементов. Важно использовать пустое пространство для создания баланса и симметрии между информационными элементами формы, чтобы сосредоточить внимание пользователя в правильном направлении.

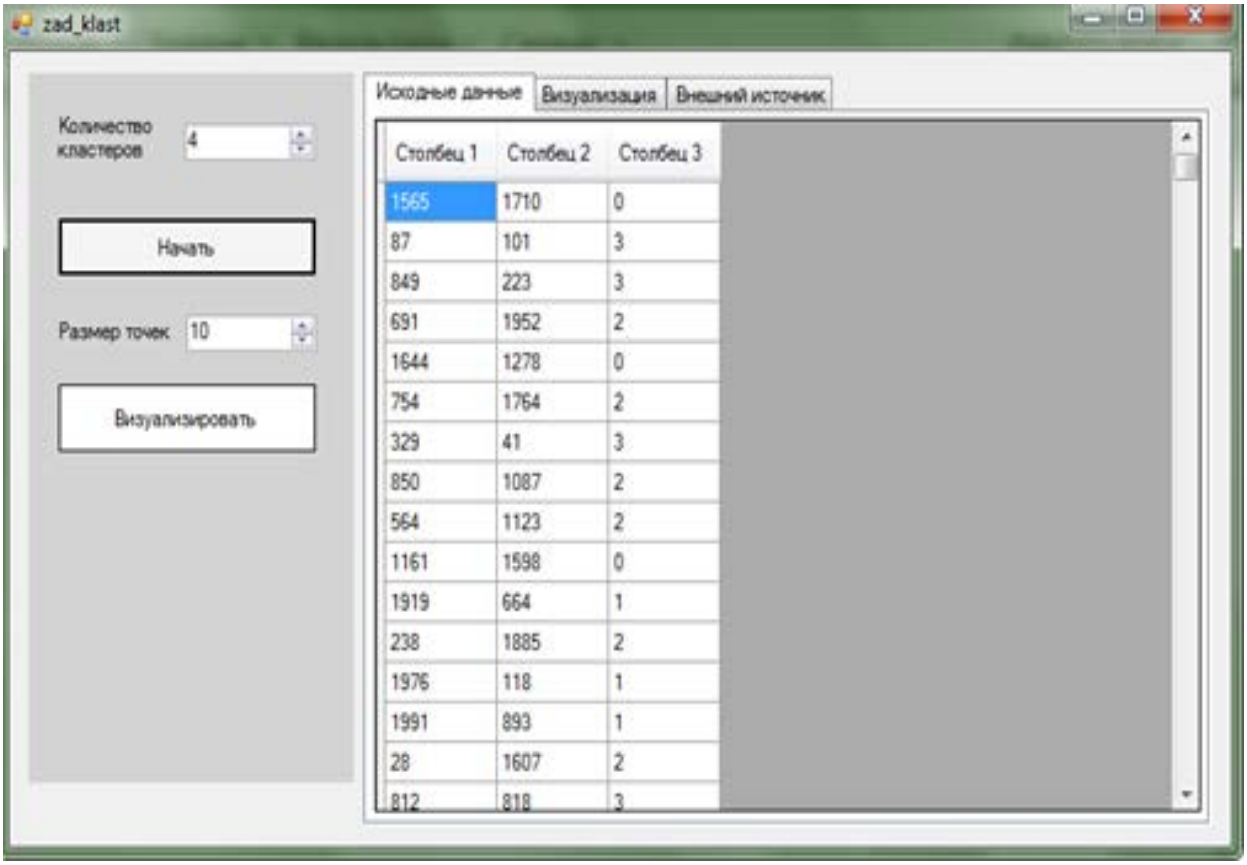


Рисунок 3.9 – Окно работы с исходными данными

В качестве примера эргономики интерфейса можно привести окно проведения операции визуализации, которое открывается на отдельной вкладке. В нем можно перейти в режим отображения графической информации результатов анализа.

В данном режиме можно настраивать параметры отображения визуализированных результатов, а также подготовить вывод во внешний файл для экспорта результатов обработки.

Логические группы элементов должны быть разделены пробелами, линиями, цветом или другими визуальными средствами. Взаимозависимые или

связанные элементы должны отображаться в одной форме.

Интерфейс программного средства располагается различных формах соответствующих каждой из задач, задуманных в системе.Цель разработки интерфейса заключалась в использовании задачного подхода.

3.5 Интерфейс модуля тестирования

Для прохождения теста пользователю предлагается выбрать предметную область задач, в рамках которых будет проводиться тестирование. Окно выбора изображено на рисунке 3.5.1

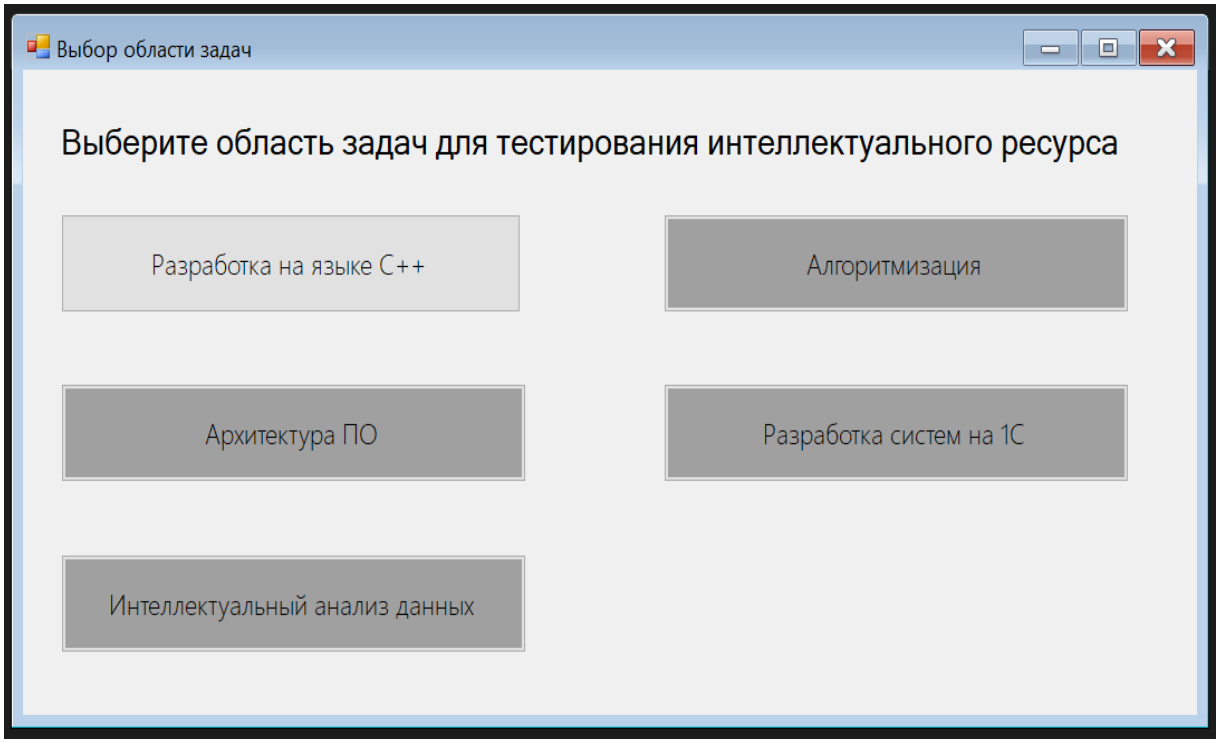


Рисунок 3.10 - Окно выбора предметной области

После выбора предметной области в которой будет проходить тестирование, открывается главное окно модуля. Все окно делится на несколько функциональных блоков:

- Главное меню модуля
- Шкала прогресса

- Область вывода вопроса
- Варианты ответов
- Окно аксиом и правил

Тесты различных этапов отличаются длинами конструкций, относительно которых тестируемый должен принять решение.

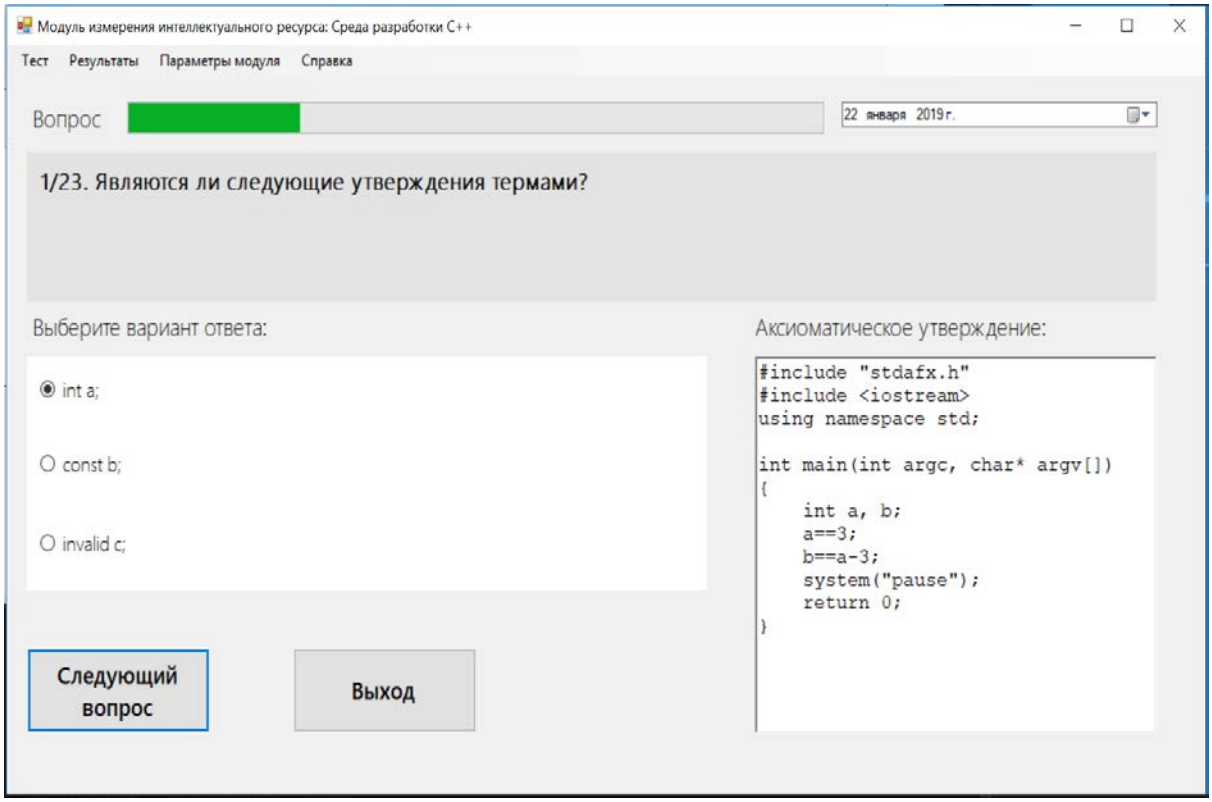


Рисунок 3.11 – Окно тестирования интеллектуального ресурса

После прохождения тестирования интеллектуального ресурса, пользователю предлагается выбор самого оптимального для него вида отображения информации (учитывается также контекст ранее поставленной пользователем задачи).

При этом пользователь имеет возможность ознакомиться с другими вариантами визуализации результатов анализа (см. рисунок 3.5.2).

Таким образом пользовательский интерфейс не становится сам задачей, не вводя пользователя в заблуждение и мешая его изысканиям.

3.6 Описание программных модулей

Разрабатываемая система, как и любая другая состоит из различных программных модулей, каждый из которых обладает своими функциональными возможностями. Общая структура пакета модульная, имеющая единую платформу. В качестве платформы используется центральный модуль, внутри которого происходит отображение информации, имеется центральный интерфейс пользователя (при помощи которого можно получить доступ к любому модулю системы).

Ядро системы имеет децентрализацию, относительно каждой из решаемых задач. За счет центрального модуля каждое ядро имеет связь и может передавать данные и обмениваться процессами через это ядро.

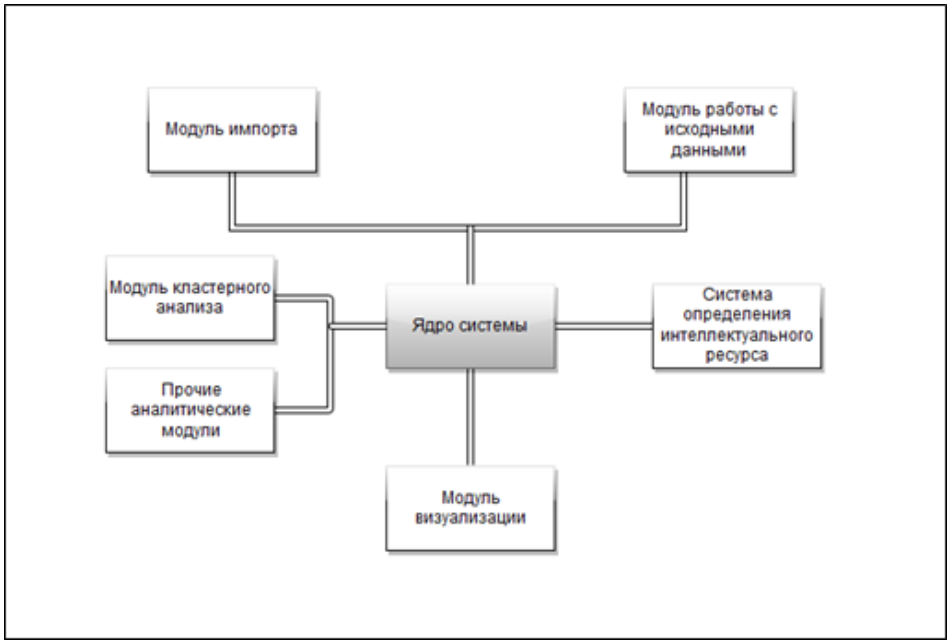


Рисунок 3.12 – Структура программных модулей системы

Каждый плагин системы имеет общий протокол обмены данными с другими модулями системы, при этом в пассивном состоянии не занимают процессорную и оперативную память (вплоть до момента вызова).

Удобство системы заключается к возможности дополнения модулями - плагинами (от plugin «подключать»), независимо компилируемые модули

системы. Это дает возможность программировать и проводить процесс отладки, рефакторинга кода не затрагивая центральный модуль и все прочие ядра системы.

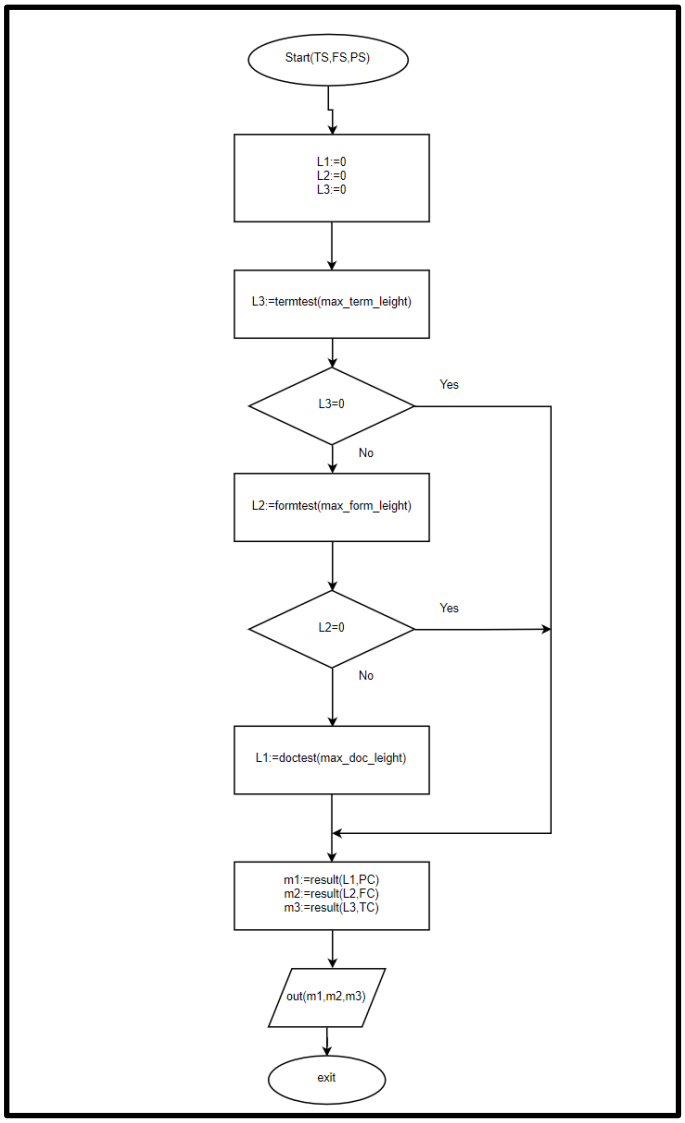


Рисунок 3.13 - Алгоритм работы программы

Данная архитектура системы позволяет не только разрабатывать независимо дополняемые модули системы, но так же и предоставляет сторонним разработчикам создавать свои плагины.

Алгоритм работы программы представлен на рисунке 3.14.

TS, FS, PS – балльные шкалы убедительности для испытуемого p распознавания им терма, формулы, доказательства системы T.

Ниже представлен алгоритм терм калибровки:

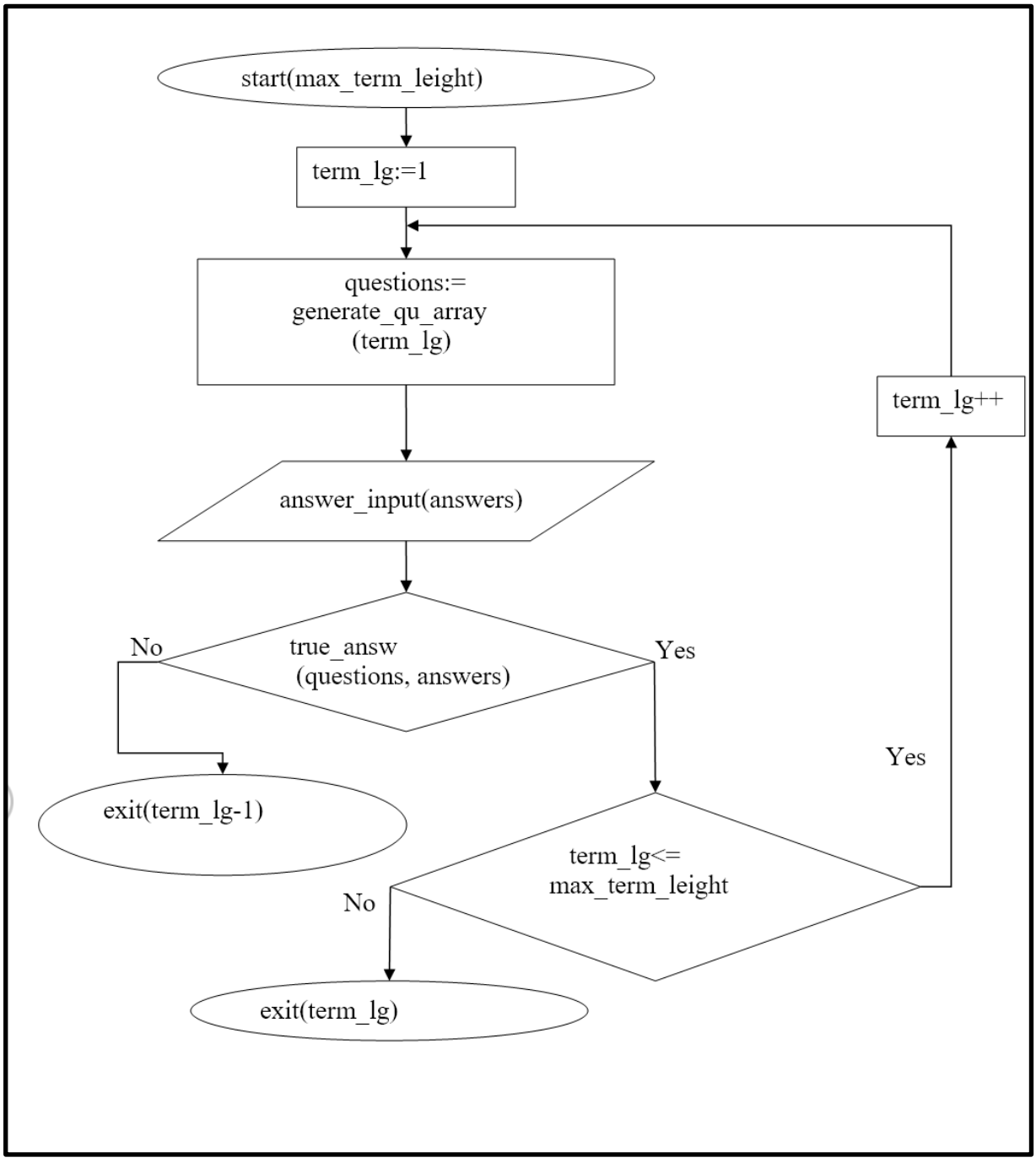


Рисунок 3.14 - Алгоритм терм калибровки

Суть алгоритма сводится к следующему: изначально терм-калибру присваивается значение 1, что означает распознавание тестируемым одной минимальной логической единицы - терма, в предметной области. После

выполнения этого программного блока генерируется массив вопросов, который подгружается из внешней базы данных.

Каждый блок данных вопроса в базе данных содержит в себе, помимо самих вопросов, данные для подгрузки в функциональные окна программы, такие как номер вопроса, но также и варианты правильных ответов.

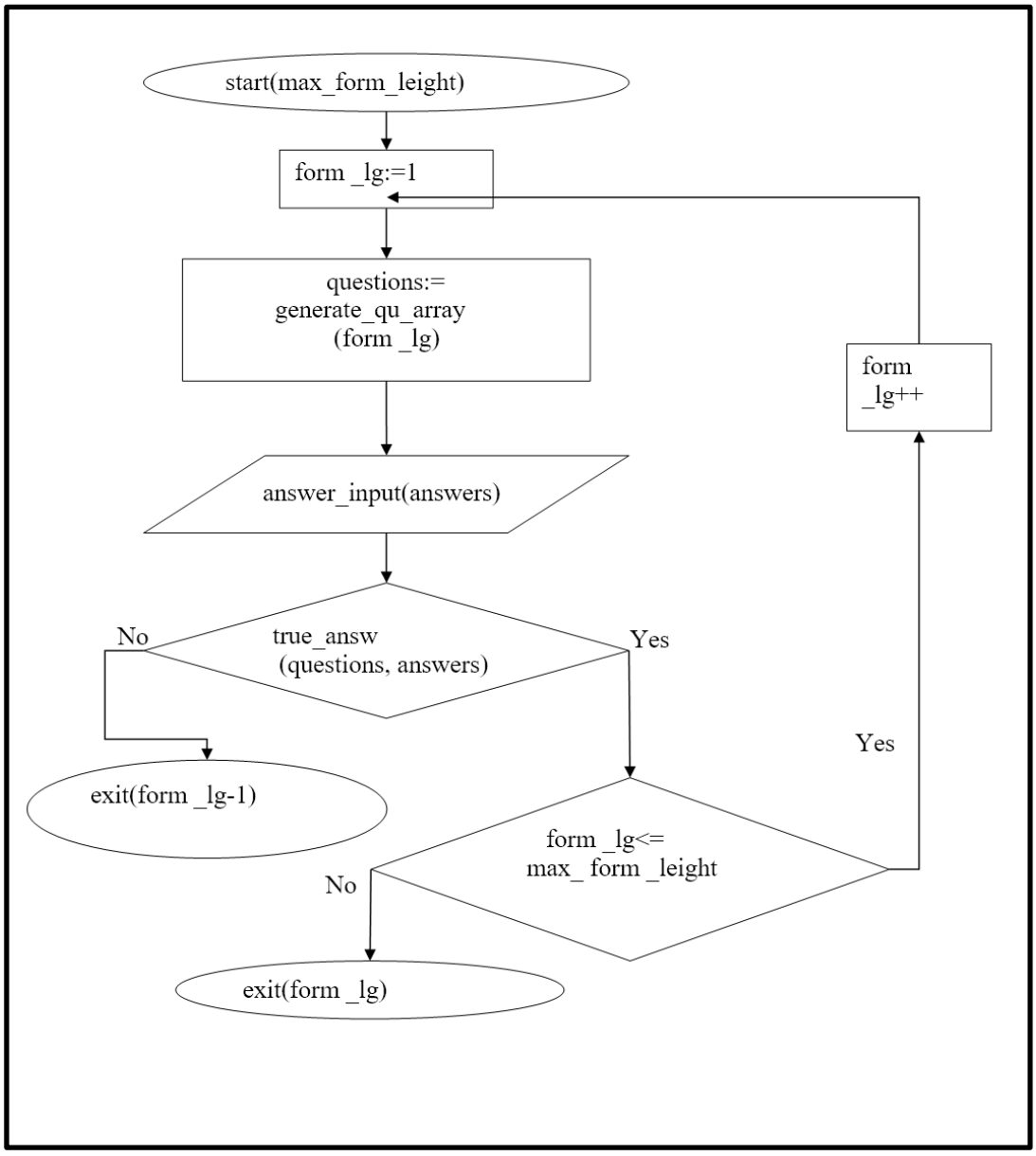


Рисунок 3.15 - Алгоритм форм калибровки

После ввода ответа пользователем в диалоговом окне, алгоритм программы проверяет соответствие ответа. При положительном результате

программа увеличивает переменную отвечающую за показатель терм-калибра на единицу и возвращается к блоку подгрузки вопросов. При несоответствии выдается текущее значение терм-калибра уменьшенное на единицу.

Итогом тестирования терм-калибра пользователя является балл, который означает количество термов распознаваемых им в данной предметной области.

Конструкция форм-калибровки представляет собой усложнение терм-калибровки, за счет увеличения логических связей. Пользователю предлагается определить связку термов объединенных логической конструкцией, то есть формулой. Этот тест требует понимания взаимодействия тех или иных высказываний внутри области задач тестирования.

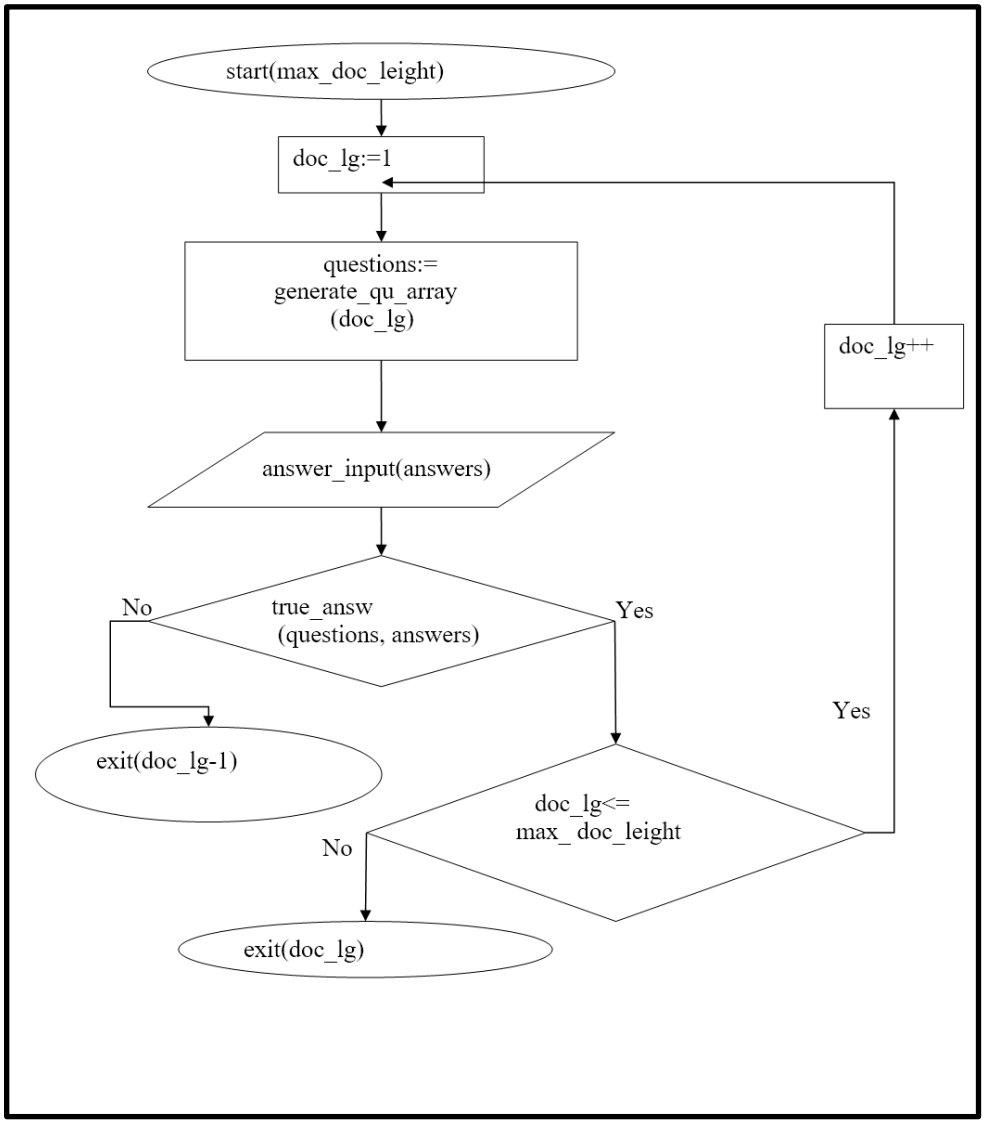


Рисунок 3.16 - Алгоритм док калибровки

Алгоритм форм-калибровки в конструкции схож с алгоритмом терм-калибровки и повторяет все вышеуказанные итерации. Алгоритм форм-калибровки показан на рисунке 3.6.3.

Переменная form_lg отвечает за длину формулы, то есть количественный показатель связок термов. Переменная max_form_leight хранит значение максимальной формульной длинны.

Док калибровка имеет также схожую структуру и выдает аналогичным образом числовой балл.

На рисунке 3.6.5 показан алгоритм док калибровки. Переменная doc_lg отвечает за длину доказательств.

При окончании тестирования выводятся три числовых показателя, каждый из которых отвечает за калибр каждого из типов: терм-калибра док-калибра и форм-калибра (TC, FC, и DC)

На основании численных показателей можно сделать выводы об интеллектуальном ресурсе конкретного пользователя и понять, какие классы задач он способен решать в данной предметной области.

3.7 Организация технологии сбора, передачи, обработки и выдачи информации

Технологический процесс машинной обработки экономической информации представляет собой совокупность операций, выполняемых в строго определенной последовательности от начального момента до окончательного получения указанных результатов (рисунок 3.7.1).

Его можно разделить на четыре интегрированных этапа: основной, подготовительный, основной и заключительный. На начальном этапе исходные данные собираются, регистрируются и передаются для ввода в компьютер.

Подготовительный этап охватывает операции получения, управления и записи входной информации и ее переноса на машиночитаемый носитель.

Рисунок 3.17 - Технологический процесс обработки информации

Основной этап предусматривает непосредственную обработку информации на компьютере. На последнем этапе осуществляется контроль, выпуск и передача полученной информации потребителю.

После этого для дальнейшей работы пользователю необходимо использовать функцию импорта данных и импортировать данные из файла выбранного формата. Схематичный процесс изображен на рисунке 3.7.2.

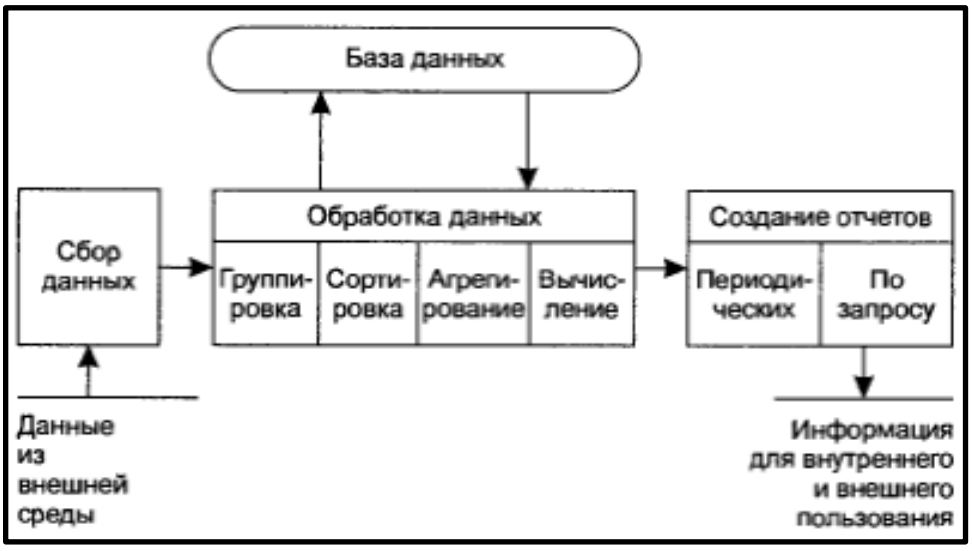


Рисунок 3.18 - Процесс импорта и обработки данных из БД

Данные будут помещены в таблицу и предстанут перед пользователем. В зависимости от выбранного типа могут появиться дополнительные настройки, используя которые пользователь может выбрать необходимые ему данные.

В дальнейшем интерфейс программы будет развиваться, а основной упор будет сделан на работу с интеллектуальным ресурсом пользователя. То есть, в зависимости от возможностей пользователя, он будет получать версию программы, которая, собственно, и соответствует этим возможностям. Что позволит любому типу пользователей решать свои задачи, не нагружая их при этом не нужными для них функциями.

В конечном итоге, информация о результатах будет выведена ниже таблицы данных и пользователь сможет увидеть, какие признаки информативны.

3.8 Схема технологического процесса сбора, передачи, обработки и выдачи информации

На рисунке 3.8.1 отображен процесс технологического сбора информации, его подготовки и обработки внутри системы.

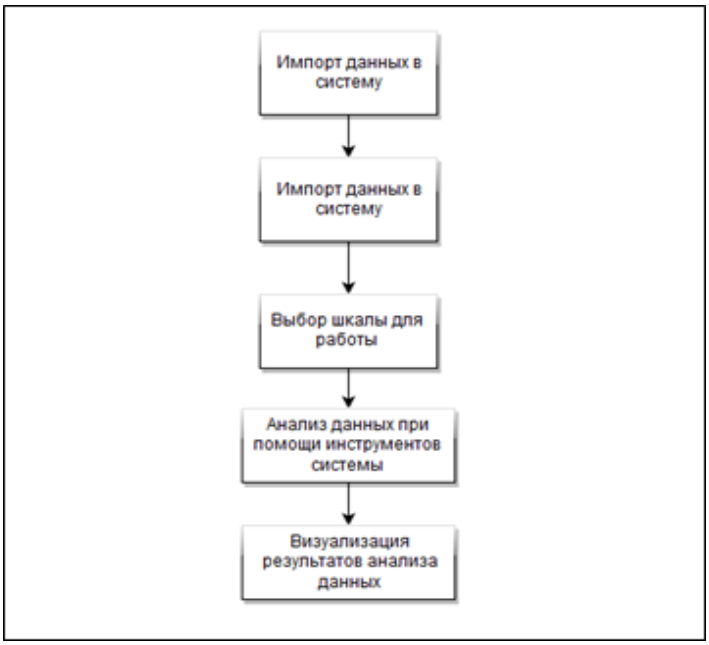


Рисунок 3.19 - Схема технологического процесса сбора, передачи, обработки и выдачи информации

На заключительных этапах процесса результаты выдаются в визуализированном представлении.

Выводы к разделу:

Во время разработки модуля для системы «Познание» была решена главная задача – измерение интеллектуального ресурса пользователя в области задач.

Для этого были реализованы следующие функции:

- формирование гипотез произвольного вида при помощи предикатов первого порядка;
- решение задачи уменьшения пространства признаков уменьшение и пространства признаков при помощи логико-статистических методов вычислений;
- учёт интеллектуального ресурса пользователя;
- импорт файлов из различных источников.

В результате реализации проекта, в соответствии с требованиями руководства ИП Шевченко к разрабатываемой системе автоматизации, система удовлетворяет требованиям аналитика, а именно:

- позволяет измерять интеллектуальный ресурс пользователя;
- позволяет формулировать гипотезы свободного вида;
- позволяет принимать управленческие решения.

ЗАКЛЮЧЕНИЕ

В ходе диссертации была исследована предметная область деятельности проектно-конструкторского отдела ИП Шевченко. Были выявлены основные проблемы в аналитической деятельности предприятия. В ходе анализа деятельности аналитического отдела возникла необходимость в методах, которые решающих интеллектуальные задачи и сопоставляющие их с интеллектуальным ресурсом пользователя. Разработаны методы и средства на основе изученных дисциплин, обеспечивающие решение этих проблем.

В процессе разработки были разработаны методы и средства измерения интеллектуального ресурсов пользователя, а так же была проанализирована деятельность и структура ИП Шевченко, а так же взаимодействие подразделений предприятия, определены задачи аналитика. Были изучены организационная и техническая структура предприятия. Изучено используемое программное обеспечение и технические средства, используемые для решения конкретных задач. Работа аналитика на данном промежутке времени работы предприятия занимает длительное время, большинство расчетов проводится вручную. В связи с этим было предложено решение этой проблемы.

В ходе анализа существующего программного обеспечения, был сделан вывод, что большая часть существующих предложений не позволяют эффективно решать задачи и зачастую имеют высокую стоимость лицензии. Поэтому было принято решение разработки системы «Познание», сопоставляющее интеллектуальные запросы пользователя и его интеллектуальный ресурс.

Задачи, выполненные в рамках данной выпускной квалификационной работы:

– проведен анализ существующих методов и средств оценки интеллектуальных ресурсов пользователей;

– изучены существующие средства для тестирования пользователей в области задач;

– разработана модель системы оценки интеллектуальных ресурсов;

– разработан интерфейс пользователя, который отвечает задаче оценки интеллектуальных ресурсов;

– разработан прототип программного средства для оценки интеллектуального ресурса пользователя в заданной предметной области.

Для этого были реализованы следующие функции:

- формирование гипотез произвольного вида при помощи предикатов первого порядка;
- решение задачи уменьшения пространства признаков уменьшение и пространства признаков при помощи логико-статистических методов вычислений;
- учёт интеллектуального ресурса пользователя;
- импорт файлов из различных источников.

В результате реализации проекта, в соответствии с требованиями руководства ИП Шевченко к разрабатываемой системе автоматизации, система удовлетворяет требованиям аналитика, а именно:

- позволяет измерять интеллектуальный ресурс пользователя;
- позволяет формулировать гипотезы свободного вида;
- позволяет принимать управленческие решения.

Во время разработки модуля для системы «Познание» была решена главная задача – измерение интеллектуального ресурса пользователя в области задач.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Анфилатов В. С., Емельянов А. А., Кукушкин А. А. Системный анализ в управлении. — М. Финансы и статистика, 2012.

2. Базы данных. Интеллектуальный анализ данных. Автор Нестеров А.С. год выпуска 2011, издательство СПб.: Изд-во Политехн. ун-та;

3. Гуц А.К. Математическая логика и теория алгоритмов. — Наследие, Диалог-Сибирь, 2013.

4. Дюк В., Самойленко А. «DataMining: учебный курс», 2010;

5. Ершов Ю.Л., Палютин Е.А. Математическая логика. — М.: Наука, Физматлит, 2007.

6. Журавлёв Ю.И., Рязанов В.В., Сенько О.В. РАСПОЗНАВАНИЕ. Математические методы. Программная система. Практические применения. — М.: Изд. «Фазис», 2011

7. Зиновьев А. Ю. Визуализация многомерных данных. — Красноярск: Изд. Красноярского государственного технического университета, 2010.

8. Игошин В.И. Математическая логика и теория алгоритмов. — Academia, 2008.

9. Ильясов Ф. Н. Шкалы и специфика социологического измерения. Мониторинг общественного мнения: экономические и социальные перемены. 2014.

10. Клини С.К. Математическая логика. — М.:Мир, 2013.

11. Мендельсон Э. Введение в математическую логику. — М. Наука, 2012.

12. Новиков П.С. Элементы математической логики. — М.:Наука, 2013.

13. Паклин Н. Б., Орешков В. И. Бизнес-аналитика: от данных к знаниям— СПб.: Изд. Питер, 200

14. Перегудов Ф. И., Тарасевич Ф. П. Введение в системный анализ. — М.: Высшая школа, 2009.

15. Суппес П., Зиннес Д. Основы теории измерений. Психологические измерения.

16. Чубукова И. А. DataMining: учебное пособие. — М.: Интернет-университет информационных технологий: БИНОМ: Лаборатория знаний, 2012.

17. Ian H. Witten, Eibe Frank and Mark A. Hall. Data Mining: Practical Machine Learning Tools and Techniques. — 3rd Edition. — Morgan Kaufmann, 2011.

18. MSDN – сеть разработчиков Microsoft [Электронный ресурс]. – Режим доступа. – URL: http://www.msdn.microsoft.com/ (дата обращения 01.12.2018)

19. TechNet - ресурсы по администрированию, виртуализации, облачным вычислениям [Электронный ресурс]. – Режим доступа. – URL: https://technet.microsoft.com/ (дата обращения 01.12.2018)

20. BusinessIntelligence - Википедия [Электронный ресурс]. – Режим доступа. – URL: https://ru.wikipedia.org/wiki/Business_Intelligence/ (дата обращения 01.12.2018)

21. Гручко А.А., Тимонина Н.Н. Теоретические основы защиты информации. – М.: издательство Агентство «Яхтсмен». 2011 г. – 333с.

22. Базы данных. Интеллектуальный анализ данных. Автор Нестеров А.С. год выпуска 2011, издательство СПб.: Изд-во Политехн. ун-та. – 442с.

23. Дюк В., Самойленко А. «DataMining: учебный курс», 2010; – 237с.

24. Паклин Н. Б., Орешков В. И. Бизнес-аналитика: от данных к знаниям— СПб.: Изд. Питер. – 200с.

25. Журавлёв Ю.И., Рязанов В.В., Сенько О.В. РАСПОЗНАВАНИЕ. Математические методы. Программная система. Практические применения. — М.: Изд. «Фазис», 2011– 367с.

26. Зиновьев А. Ю. Визуализация многомерных данных. — Красноярск: Изд. Красноярского государственного технического университета, 2010. – 422с.

27. Чубукова И. А. DataMining: учебное пособие. — М.: Интернет-университет информационных технологий: БИНОМ: Лаборатория знаний, 2012. – 286с.

28. Ian H. Witten, Eibe Frank and Mark A. Hall. Data Mining: Practical Machine Learning Tools and Techniques. — 3rd Edition. — Morgan Kaufmann, 2011. – 321с.

29. Гуц А.К. Математическая логика и теория алгоритмов. — Наследие, Диалог-Сибирь, 2013.– 400с.

30. Ершов Ю.Л., Палютин Е.А. Математическая логика. — М.: Наука, Физматлит, 2007.– 420с.

31. Игошин В.И. Математическая логика и теория алгоритмов. — Academia, 2008. – 353с.

32. Клини С.К. Математическая логика. — М.:Мир, 2013.– 239с.

33. Мендельсон Э. Введение в математическую логику. — М. Наука, 2012. – 420с.

34. Новиков П.С. Элементы математической логики. — М.:Наука, 2013.– 323с.

35. Анфилатов В. С., Емельянов А. А., Кукушкин А. А. Системный анализ в управлении. — М. Финансы и статистика, 2012. – 290с.

ПриложениеА

Листинги кода программы:

```
namespaceТестирование
{
	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()
	{
	this.button1 = new System.Windows.Forms.Button();
	this.button2 = new System.Windows.Forms.Button();
	this.label1 = new System.Windows.Forms.Label();
	this.radioButton1 = new System.Windows.Forms.RadioButton();
	this.radioButton2 = new System.Windows.Forms.RadioButton();
	this.radioButton3 = new System.Windows.Forms.RadioButton();
```

```
this.richTextBox1 = new System.Windows.Forms.RichTextBox();
this.SuspendLayout();
//
// button1
//
this.button1.Location = new System.Drawing.Point(20, 515);
this.button1.Margin = new System.Windows.Forms.Padding(3, 4, 3, 4);
this.button1.Name = "button1";
this.button1.Size = new System.Drawing.Size(246, 102);
this.button1.TabIndex = 0;
this.button1.Text = "button1";
this.button1.UseVisualStyleBackColor = true;
this.button1.Click += new System.EventHandler(this.button1_Click);
//
// button2
//
this.button2.Location = new System.Drawing.Point(395, 515);
this.button2.Margin = new System.Windows.Forms.Padding(3, 4, 3, 4);
this.button2.Name = "button2";
this.button2.Size = new System.Drawing.Size(246, 102);
this.button2.TabIndex = 1;
this.button2.Text = "button2";
this.button2.UseVisualStyleBackColor = true;
this.button2.Click += new System.EventHandler(this.button2_Click);
//
// label1
//
this.label1.AutoSize = true;
this.label1.Location = new System.Drawing.Point(32, 105);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(51, 20);
this.label1.TabIndex = 2;
this.label1.Text = "label1";
//
// radioButton1
//
this.radioButton1.AutoSize = true;
this.radioButton1.Location = new System.Drawing.Point(36, 196);
this.radioButton1.Margin = new System.Windows.Forms.Padding(3, 4, 3, 4);
this.radioButton1.Name = "radioButton1";
this.radioButton1.Size = new System.Drawing.Size(126, 24);
this.radioButton1.TabIndex = 3;
```

```
        this.radioButton1.TabStop = true;
        this.radioButton1.Text = "radioButton1";
        this.radioButton1.UseVisualStyleBackColor = true;
        //
        // radioButton2
        //
        this.radioButton2.AutoSize = true;
        this.radioButton2.Location = new System.Drawing.Point(36, 292);
        this.radioButton2.Margin = new System.Windows.Forms.Padding(3, 4, 3, 4);
        this.radioButton2.Name = "radioButton2";
        this.radioButton2.Size = new System.Drawing.Size(126, 24);
        this.radioButton2.TabIndex = 4;
        this.radioButton2.TabStop = true;
        this.radioButton2.Text = "radioButton2";
        this.radioButton2.UseVisualStyleBackColor = true;
        //
        // radioButton3
        //
        this.radioButton3.AutoSize = true;
        this.radioButton3.Location = new System.Drawing.Point(36, 402);
        this.radioButton3.Margin = new System.Windows.Forms.Padding(3, 4, 3, 4);
        this.radioButton3.Name = "radioButton3";
        this.radioButton3.Size = new System.Drawing.Size(126, 24);
        this.radioButton3.TabIndex = 5;
        this.radioButton3.TabStop = true;
        this.radioButton3.Text = "radioButton3";
        this.radioButton3.UseVisualStyleBackColor = true;
        this.radioButton3.CheckedChanged += new
System.EventHandler(this.radioButton3_CheckedChanged);
        //
        // richTextBox1
        //
        this.richTextBox1.Location = new System.Drawing.Point(764, 102);
        this.richTextBox1.Name = "richTextBox1";
        this.richTextBox1.Size = new System.Drawing.Size(540, 515);
        this.richTextBox1.TabIndex = 6;
        this.richTextBox1.Text = "";
        //
        // Form1
        //
        this.AutoScaleDimensions = new System.Drawing.SizeF(9F, 20F);
        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
```

```
        this.ClientSize = new System.Drawing.Size(1333, 642);
        this.Controls.Add(this.richTextBox1);
        this.Controls.Add(this.radioButton3);
        this.Controls.Add(this.radioButton2);
        this.Controls.Add(this.radioButton1);
        this.Controls.Add(this.label1);
        this.Controls.Add(this.button2);
        this.Controls.Add(this.button1);
        this.Margin = new System.Windows.Forms.Padding(3, 4, 3, 4);
        this.Name = "Form1";
        this.Text = "Form1";
        this.Load += new System.EventHandler(this.Form1_Load);
        this.ResumeLayout(false);
        this.PerformLayout();

        }

        #endregion

        private System.Windows.Forms.Button button1;
        private System.Windows.Forms.Button button2;
        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.RadioButton radioButton1;
        private System.Windows.Forms.RadioButton radioButton2;
        private System.Windows.Forms.RadioButton radioButton3;
        private System.Windows.Forms.RichTextBox richTextBox1;
        }
}
using System;
using System.Windows.Forms;
usingSystem.IO;
// Другие директивы using удалены, поскольку они не используются в данной
// программе
namespace Тестирование
{
        public partial class Form1 : Form
        {
        // Внешние переменные:
        intQNumber; // Счет вопросов
        intTAnsw;  // Количество правильных ответов
        intFAnsw; // Количество не правильных ответов
        // Массив вопросов, на которые даны неправильные ответы:
```

```
        String[] ArrFAnsw; // Размерность этого массива зададим позже
        intN_TAnsw;  // Номер правильного ответа
        intChAnsw;  // Номер ответа, выбранный студентом
        System.IO.StreamReader Reader;
        public Form1()
        {
        InitializeComponent();
        }
        private void Form1_Load(object sender, EventArgs e)
        {
        button1.Text = "Следующий вопрос";
        button2.Text = "Выход";
        // Подписка на событие изменение состояния
        // переключателей RadioButton:
        radioButton1.CheckedChanged += new EventHandler(ChangeSwitchState);
        radioButton2.CheckedChanged += new EventHandler(ChangeSwitchState);
        radioButton3.CheckedChanged += new EventHandler(ChangeSwitchState);
        TestStart();

        // richTextBox1.Text = System.IO.TextReader("filename.txt");

        //richTextBox1.LoadFile(@"..\test.txt", Encoding.UTF8);

        //richTextBox1.Text = System.IO.ReadAllText(@"..\test.txt");
        //RichTextBox1.Text = System.IO.ReadAllText("filename.txt");
        //System.IO.WriteAllText(RichTextBox.Text, "filename.txt");

        }
        void TestStart()
        {
        //richTextBox1.Text = File.ReadAllText(@СчетВопросов + ".txt",
System.Text.Encoding.Default);
        var Code = System.Text.Encoding.GetEncoding(1251);
        try
        {
        // Создание экземпляра StreamReader для чтения из файла
        Reader = new System.IO.StreamReader(
        System.IO.Directory.GetCurrentDirectory() +
        @"\test.txt", Code);
        this.Text = Reader.ReadLine(); // Название предмета
        // Обнуление всех счетчиков:
        QNumber = 0; TAnsw = 0; FAnsw = 0;
```

```
        // Задаем размер массива для НеПравилОтветы:
        ArrFAnsw = new String[100];
        }
        catch (Exception Situation)
        {   // Отчетовсехошибках:
        MessageBox.Show(Situation.Message, "Ошибка",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        }
        ReadNextQ();
        }
        void ReadNextQ()
        {
        richTextBox1.Text = File.ReadAllText(@"quest\"+ QNumber + ".txt",
System.Text.Encoding.Default);

        label1.Text = Reader.ReadLine();
        // Считывание вариантов ответа:
        radioButton1.Text = Reader.ReadLine();
        radioButton2.Text = Reader.ReadLine();
        radioButton3.Text = Reader.ReadLine();
        // Выясняем, какойответ - правильный:
        N_TAnsw = int.Parse(Reader.ReadLine());
        // Переводим все переключатели в состояние "выключено":
        radioButton1.Checked = false;
        radioButton2.Checked = false;
        radioButton3.Checked = false;
        // Первую кнопку задаем не активной, пока студент не выберет
        // вариантответа
        button1.Enabled = false;
        QNumber = QNumber + 1;
        // Проверка, конецлифайла:
        if (Reader.EndOfStream == true) button1.Text = "Завершить";
        }
        void ChangeSwitchState(Object sender, EventArgs e)
        {
        // Кнопка "Следующий вопрос" становится активной, и ей
        // передаемфокус:
        button1.Enabled = true; button1.Focus();
        RadioButton Switch = (RadioButton)sender;
        vartmp = Switch.Name;
        // Выясняем номер ответа, выбранный студентом:
        ChAnsw = int.Parse(tmp.Substring(11));
```

```
}
private void button1_Click(object sender, EventArgs e)
{
// Щелчок на кнопке
// "Следующий вопрос/Завершить/Начать тестирование снач"
// Счет правильных ответов:
if (ChAnsw == N_TAnsw) TAnsw =
TAnsw + 1;
if (ChAnsw != N_TAnsw)
{
// Счет неправильных ответов:
FAnsw = FAnsw + 1;
// Запоминаем вопросы с неправильными ответами:
ArrFAnsw[FAnsw] = label1.Text;
}
if (button1.Text == "Начать тестирование сначала")
{
button1.Text = "Следующий вопрос";
// Переключатели становятся видимыми, доступными для выбора:
radioButton1.Visible = true;
radioButton2.Visible = true;
radioButton3.Visible = true;
// Переходкначалуфайла:
TestStart(); return;
}
if (button1.Text == "Завершить")
{
// Закрываем текстовый файл:
Reader.Close();
// Переключатели делаем невидимыми:
radioButton1.Visible = false;
radioButton2.Visible = false;
radioButton3.Visible = false;
// Формируемоценкузатест:
label1.Text = String.Format("Тестированиезавершено.\n" +
"Правильных ответов: {0} из {1}.\n" +
"Оценка в пятибальной системе: {2:F2}.", TAnsw,
QNumber, (TAnsw * 5.0F) / QNumber);
// 5F - это максимальная оценка
button1.Text = "Начать тестирование сначала";
// Вывод вопросов, на которые "Вы дали неправильный ответ":
varStr = "СПИСОК ВОПРОСОВ, НА КОТОРЫЕ ВЫ ДАЛИ " +
```

```
        "НЕПРАВИЛЬНЫЙОТВЕТ:\n\n";
        for (int i = 1; i <= FAnsw; i++)
        Str = Str + ArrFAnsw[i] + "\n";

        // Если есть неправильные ответы, то вывести через
        // MessageBox список соответствующих вопросов:
        if (FAnsw != 0) MessageBox.Show(
        Str, "Тестирование завершено");
        } // Конец условия if (button1.Text == "Завершить")
        if (button1.Text == "Следующийвопрос") ReadNextQ();
        }
        private void button2_Click(object sender, EventArgs e)
        {
        // Щелчок на кнопке "Выход"
        this.Close();
        }

        private void radioButton3_CheckedChanged(object sender, EventArgs e)
        {

        }
        }
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;
using Microsoft.Office.Interop.Excel;
using System.Data.SqlClient;
using WordObj = Microsoft.Office.Interop.Word;
using System.IO;

namespace WindowsFormsApplication1
{
        public partial class Form1 : Form
        {
```

```
        public Form1()
        {
        InitializeComponent();
        /* toolStrip1.Items[0].MouseMove += delegate
        {
        toolStrip1.Items[0].BackColor = Color.Red;
        };

        for (int i = 0; i < toolStrip1.Items.Count; i++)
        toolStrip1.Items[i].MouseLeave += delegate
        {
        toolStrip1.Items[0].BackColor = Color.WhiteSmoke;
        };*/

        }




        private void Form1_Load(object sender, EventArgs e)
        {

        //this.reportViewer1.RefreshReport();
        }

        string FileName = null;

        private void accessToolStripMenuItem1_Click(object sender, EventArgs e)
        {
        OpenFileDialogfd = new OpenFileDialog();

        if (fd.ShowDialog() == DialogResult.OK)
        {
        FileName = fd.FileName;
        textBox1.Text = FileName;

        treeView1.Nodes.Clear();
        OleDbConnection con = new
OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0; Data Source=" + FileName);
        con.Open();

        System.Data.DataTabletableNames = con.GetSchema("Tables");
```

```
        System.Data.DataTablecolumnNames = con.GetSchema("Columns");

        int p = 0;
        foreach (DataRowdr in tableNames.Rows)
        {
        string tableName = (string)dr["TABLE_NAME"];
        if (!tableName.Contains("MSys"))
        {
        treeView1.Nodes.Add(tableName);
        foreach (DataRow dr1 in columnNames.Rows)
        {
        if (tableName == (string)dr1["TABLE_NAME"]) ;
        {
        treeView1.Nodes[p].Nodes.Add((string)dr1["COLUMN_NAME"]);
        }
        }
        p++;
        }
        }
        con.Close();
        // pictureBox1.Visible = false;
        treeView1.Visible = true;
        dataGridView1.Visible = true;
        dataGridView2.Visible = false;
        dataGridView3.Visible = false;
        dataGridView4.Visible = false;
        comboBox1.Visible = false;
        textBox1.Visible = true;
        label1.Visible=true;
        }
        }

        private void treeView1_AfterSelect(object sender, TreeViewEventArgs e)
        {
        OleDbConnection con = new
OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0; Data Source=" + FileName);
        con.Open();

        if (treeView1.SelectedNode.Level == 0)
        {
        OleDbDataAdapter da = new OleDbDataAdapter("select * from " +
treeView1.SelectedNode.Text, con);
```

```
      OleDbCommandBuilder cd = new OleDbCommandBuilder(da);

      DataSet ds = new DataSet();

      da.Fill(ds, treeView1.SelectedNode.Text);
      dataGridView1.DataSource = ds.Tables[0];
      }
      else
      {
      OleDbDataAdapter da = new OleDbDataAdapter("select * from " +
treeView1.SelectedNode.Parent.Text, con);
      OleDbCommandBuildercb = new OleDbCommandBuilder(da);

      DataSet ds = new DataSet();
      da.Fill(ds, treeView1.SelectedNode.Parent.Text);
      dataGridView1.DataSource = ds.Tables[0];
      }

      }

      private void excelToolStripMenuItem_Click(object sender, EventArgs e)
      {
      string str;
      intrCnt;
      intcCnt;

      OpenFileDialogopf = new OpenFileDialog();
      opf.Filter = "Файл Excel|*.XLSX;*.XLS";
      opf.ShowDialog();
      System.Data.DataTabletb = new System.Data.DataTable();
      string filename = opf.FileName;

      Microsoft.Office.Interop.Excel.ApplicationExcelApp = new
Microsoft.Office.Interop.Excel.Application();
      Microsoft.Office.Interop.Excel._WorkbookExcelWorkBook;
      Microsoft.Office.Interop.Excel.WorksheetExcelWorkSheet;
      Microsoft.Office.Interop.Excel.RangeExcelRange;

      ExcelWorkBook = ExcelApp.Workbooks.Open(filename, 0, true, 5, "", "",
true, Microsoft.Office.Interop.Excel.XlPlatform.xlWindows, "\t", false,
      false, 0, true, 1, 0);
      ExcelWorkSheet =
```

```
(Microsoft.Office.Interop.Excel.Worksheet)ExcelWorkBook.Worksheets.get_Item(1);

      ExcelRange = ExcelWorkSheet.UsedRange;
      for (rCnt = 1; rCnt<= ExcelRange.Rows.Count; rCnt++)
      {
      dataGridView2.Rows.Add(1);
      for (cCnt = 1; cCnt<= 2; cCnt++)
      {
      str = (string)(ExcelRange.Cells[rCnt, cCnt] as
Microsoft.Office.Interop.Excel.Range).Value2;
      dataGridView2.Rows[rCnt - 1].Cells[cCnt - 1].Value = str;
      }
      }
      ExcelWorkBook.Close(true, null, null);
      ExcelApp.Quit();

      releaseObject(ExcelWorkSheet);
      releaseObject(ExcelWorkBook);
      releaseObject(ExcelApp);

      //pictureBox1.Visible = false;
      treeView1.Visible = false;
      dataGridView1.Visible = false;
      dataGridView2.Visible = true;
      dataGridView3.Visible = false;
      dataGridView4.Visible = false;
      comboBox1.Visible = false;
      textBox1.Visible = true;
      label1.Visible = true;
      }


      private void releaseObject(object obj)
      {
      try
      {
      System.Runtime.InteropServices.Marshal.ReleaseComObject(obj);
      obj = null;
      }
      catch (Exception ex)
      {
      obj = null;
```

```
MessageBox.Show("Unable to release the object " + ex.ToString());
}
finally
{
GC.Collect();
}
}

private void wordToolStripMenuItem_Click(object sender, EventArgs e)
{

comboBox1.Items.Clear();
OpenFileDialog dialog = new OpenFileDialog();
dialog.Filter = "MS Word 2003 (*.doc)|*.doc|MS Word 2007 (*.docx)|*.docx";
dialog.Title = "Выберите документ для загрузки данных";
if (dialog.ShowDialog() == System.Windows.Forms.DialogResult.OK)
{
textBox1.Text = dialog.FileName;
application = new WordObj.Application();
Object FileName = dialog.FileName;
application.Documents.Open(ref FileName);
document = application.ActiveDocument;
for (int i = 1; i <= document.Tables.Count; i++)
{
comboBox1.Items.Add("Таблица №" + i);
}
comboBox1.SelectedIndex = 0;
LoadData(comboBox1.SelectedIndex + 1);

//pictureBox1.Visible = false;
treeView1.Visible = false;
dataGridView1.Visible = false;
dataGridView2.Visible = false;
dataGridView3.Visible = true;
dataGridView4.Visible = false;
comboBox1.Visible = true;
textBox1.Visible = true;
label1.Visible = true;
}

}
```

```
WordObj.Application application;
WordObj.Document document;
public void LoadData(intTableNum)
{
WordObj.Table table = document.Tables[TableNum];
if (table.Rows.Count> 0 &&table.Columns.Count> 0)
{
List<string> headers = new List<string>();
System.Data.DataTabledataTable = new System.Data.DataTable();
for (int i = 0; i <table.Columns.Count; i++)
{
dataTable.Columns.Add();
headers.Add(table.Cell(1, i + 1).Range.Text.Trim('\a', '\r', '\n', '\t'));
}
for (int i = 0; i <table.Rows.Count - 1; i++)
{
string[] row = new string[table.Columns.Count];
for (int j = 0; j <table.Columns.Count; j++)
row[j] = table.Cell(i + 2, j + 1).Range.Text.Trim('\a', '\r', '\n', '\t');
dataTable.Rows.Add(row);
}
dataGridView3.DataSource = dataTable;
for (int i = 0; i <headers.Count; i++)
dataGridView3.Columns[i].HeaderText = headers[i];
}
}

private void Form1_FormClosed(object sender, FormClosedEventArgs e)
{
}

private void textToolStripMenuItem_Click(object sender, EventArgs e)
{

// OpenFileDialog openFileDialog1 = new OpenFileDialog();
/*openFileDialog1.ShowDialog();


string fname = openFileDialog1.FileName;
string[] lines = File.ReadAllLines(fname);
string[] inpstr;
char[] delim = new char[] { '\t' };
```

```
      for (int i = 0; i <lines.Length; i++)
      {
      if (lines[i] != null || lines[i] != "")
      {
      inpstr = lines[i].Split(delim);
      dataGridView4.Rows.Add(inpstr);
      }
      }
      */
      //TXT2
      if (openFileDialog1.ShowDialog() != DialogResult.Cancel)
      {

      String sLine = "";

      try
      {
      System.IO.StreamReaderFileStream = new
System.IO.StreamReader(openFileDialog1.FileName);
      dataGridView4.AllowUserToAddRows = false;
      sLine = FileStream.ReadLine();
      string[] s = sLine.Split(';');
      for (int i = 0; i <= s.Count() - 1; i++)
      {
      DataGridViewColumncolHold = new DataGridViewTextBoxColumn();
      colHold.Name = "col" + System.Convert.ToString(i);
      colHold.HeaderText = s[i].ToString();
      dataGridView4.Columns.Add(colHold);
      }
      sLine = FileStream.ReadLine();
      while (sLine != null)
      {
      dataGridView4.Rows.Add();
      for (int i = 0; i <= s.Count() - 1; i++)
      {
      s = sLine.Split(';');
      dataGridView4.Rows[dataGridView4.Rows.Count - 1].Cells[i].Value =
s[i].ToString();
      }
      sLine = FileStream.ReadLine();
      }
```

```
      FileStream.Close();
      }
      catch (Exception err)
      {
      //Display any errors in a Message Box.
      //System.Windows.Forms.MessageBox.Show("Error+ err.Message, "Program
Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
      }
      }

      //pictureBox1.Visible = false;
      treeView1.Visible = false;
      dataGridView1.Visible = false;
      dataGridView2.Visible = false;
      dataGridView3.Visible = false;
      dataGridView4.Visible = true;
      comboBox1.Visible = false;
      textBox1.Visible = true;
      label1.Visible = true;

      }



      private void toolStripDropDownButton2_Click(object sender, EventArgs e)
      {

      }

      private void kmeansToolStripMenuItem_Click(object sender, EventArgs e)
      {
      Form klast = new zad_klast();
      klast.Show();
      }

      private void наборИнструментовToolStripMenuItem_Click(object sender,
EventArgs e)
      {

      }

      private void
```

```
проверкаИнтеллектуальногоРесурсаToolStripMenuItem_Click(object sender, EventArgs
e)
      {
      Form dia = new dia();
      dia.Show();
      }

      /*          private void ыузфкфещкToolStripMenuItem_Click(object sender,
EventArgs e)
      {

      }

      private void toolStripMenuItem2_Click(object sender, EventArgs e)
      {

      }*/



      }
}

namespace WindowsFormsApplication1
{
      partial class dia
      {
      /// <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()
        {
        this.label1 = new System.Windows.Forms.Label();
        this.label2 = new System.Windows.Forms.Label();
        this.button1 = new System.Windows.Forms.Button();
        this.pictureBox3 = new System.Windows.Forms.PictureBox();
        this.pictureBox1 = new System.Windows.Forms.PictureBox();
        this.pictureBox2 = new System.Windows.Forms.PictureBox();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox3)).BeginInit()
;
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginInit()
;
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox2)).BeginInit()
;
        this.SuspendLayout();
        //
        // label1
        //
        this.label1.AutoSize = true;
        this.label1.Font = new System.Drawing.Font("Segoe UI", 15F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
((byte)(204)));
        this.label1.Location = new System.Drawing.Point(155, 213);
        this.label1.Name = "label1";
        this.label1.Size = new System.Drawing.Size(399, 28);
        this.label1.TabIndex = 2;
        this.label1.Text = "Тестированиеинтеллектуальногоресурса";
        //
        // label2
        //
        this.label2.AutoSize = true;
        this.label2.Font = new System.Drawing.Font("Segoe UI Light", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
```

```
((byte)(204)));
      this.label2.ForeColor = System.Drawing.SystemColors.WindowFrame;
      this.label2.Location = new System.Drawing.Point(162, 244);
      this.label2.Name = "label2";
      this.label2.Size = new System.Drawing.Size(389, 63);
      this.label2.TabIndex = 3;
      this.label2.Text = "В процессе тестирования вам будут заданы вопросы,
\r\nпо результатам ответов вам бу" +
      "дет предложен \r\noптимальный вариант визуализации";
      this.label2.TextAlign = System.Drawing.ContentAlignment.MiddleCenter;
      //
      // button1
      //
      this.button1.BackColor =
System.Drawing.Color.FromArgb(((int)(((byte)(224)))), ((int)(((byte)(224)))),
((int)(((byte)(224)))));
      this.button1.FlatStyle = System.Windows.Forms.FlatStyle.Flat;
      this.button1.Font = new System.Drawing.Font("Segoe UI Light", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
((byte)(204)));
      this.button1.Location = new System.Drawing.Point(244, 326);
      this.button1.Name = "button1";
      this.button1.Size = new System.Drawing.Size(224, 46);
      this.button1.TabIndex = 4;
      this.button1.Text = "Продолжить...";
      this.button1.UseVisualStyleBackColor = false;
      this.button1.Click += new System.EventHandler(this.button1_Click);
      //
      // pictureBox3
      //
      this.pictureBox3.Image =
global::WindowsFormsApplication1.Properties.Resources.diagr1;
      this.pictureBox3.Location = new System.Drawing.Point(3, 1);
      this.pictureBox3.Name = "pictureBox3";
      this.pictureBox3.Size = new System.Drawing.Size(704, 438);
      this.pictureBox3.TabIndex = 5;
      this.pictureBox3.TabStop = false;
      this.pictureBox3.Visible = false;
      //
      // pictureBox1
      //
      this.pictureBox1.BackColor = System.Drawing.Color.PaleTurquoise;
```

```
      this.pictureBox1.BackgroundImageLayout =
System.Windows.Forms.ImageLayout.Zoom;
      this.pictureBox1.Image =
global::WindowsFormsApplication1.Properties.Resources.description;
      this.pictureBox1.Location = new System.Drawing.Point(307, 68);
      this.pictureBox1.Name = "pictureBox1";
      this.pictureBox1.Size = new System.Drawing.Size(94, 102);
      this.pictureBox1.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.CenterImage;
      this.pictureBox1.TabIndex = 0;
      this.pictureBox1.TabStop = false;
      //
      // pictureBox2
      //
      this.pictureBox2.BackColor = System.Drawing.Color.WhiteSmoke;
      this.pictureBox2.Image =
global::WindowsFormsApplication1.Properties.Resources.circle;
      this.pictureBox2.Location = new System.Drawing.Point(260, 28);
      this.pictureBox2.Name = "pictureBox2";
      this.pictureBox2.Size = new System.Drawing.Size(182, 182);
      this.pictureBox2.SizeMode = System.Windows.Forms.PictureBoxSizeMode.Zoom;
      this.pictureBox2.TabIndex = 1;
      this.pictureBox2.TabStop = false;
      //
      // dia
      //
      this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
      this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
      this.BackColor = System.Drawing.Color.WhiteSmoke;
      this.ClientSize = new System.Drawing.Size(708, 444);
      this.Controls.Add(this.pictureBox3);
      this.Controls.Add(this.button1);
      this.Controls.Add(this.label2);
      this.Controls.Add(this.label1);
      this.Controls.Add(this.pictureBox1);
      this.Controls.Add(this.pictureBox2);
      this.FormBorderStyle =
System.Windows.Forms.FormBorderStyle.FixedToolWindow;
      this.Name = "dia";
      this.Text = "Тестирование интеллектуального ресурса";
      ((System.ComponentModel.ISupportInitialize)(this.pictureBox3)).EndInit();
      ((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndInit();
```

```
      ((System.ComponentModel.ISupportInitialize)(this.pictureBox2)).EndInit();
      this.ResumeLayout(false);
      this.PerformLayout();

      }

      #endregion

      private System.Windows.Forms.PictureBox pictureBox1;
      private System.Windows.Forms.PictureBox pictureBox2;
      private System.Windows.Forms.Label label1;
      private System.Windows.Forms.Label label2;
      private System.Windows.Forms.Button button1;
      private System.Windows.Forms.PictureBox pictureBox3;
      }
}

private void accessToolStripMenuItem1_Click(object sender, EventArgs e)
      {
      OpenFileDialogfd = new
OpenFileDialog();//инициализацияэлементаоткрытияфайлов

      if (fd.ShowDialog() ==
DialogResult.OK)//проверкавключенияокнадиалогаоткрытияфайла
      {
      FileName = fd.FileName
      textBox1.Text = FileName; ;//заданиепеременнойименифайла

      treeView1.Nodes.Clear();

      OleDbConnection con = new
OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0; Data Source=" +
FileName);//подключениепровайдера
      con.Open();

      System.Data.DataTabletableNames = con.GetSchema("Tables");
      System.Data.DataTablecolumnNames =
con.GetSchema("Columns");//получениевпамятьиментаблициколонок

      int p = 0;

      foreach (DataRowdr in tableNames.Rows)//циклвыгрузкиданныхизфайлабазданных
```

```
        {
        string tableName = (string)dr["TABLE_NAME"];
        if (!tableName.Contains("MSys"))
        {
        treeView1.Nodes.Add(tableName);
        foreach (DataRow dr1 in columnNames.Rows)
        {
        if (tableName == (string)dr1["TABLE_NAME"]) ;
        {
        treeView1.Nodes[p].Nodes.Add((string)dr1["COLUMN_NAME"]);
//процесс отображения структуры базы данных
        }
        }
        p++;
        }
        }
        con.Close();

        treeView1.Visible = true;//инициализация элементов
        dataGridView1.Visible = true;
        dataGridView2.Visible = false;
        dataGridView3.Visible = false;
        dataGridView4.Visible = false;
        comboBox1.Visible = false;
        textBox1.Visible = true;
        label1.Visible=true;
        }
        }
        private void button1_Click(object sender, EventArgs e)
        {
        webBrowser1.Url = new Uri((Application.StartupPath + "\\canvas.html"));

        KMeansoKmeas = new KMeans(Convert.ToInt16(numericUpDown1.Value));

        hasilCluster = oKmeas.Compute(data);
        for (int i = 0; i <hasilCluster.Length; i++)
        {
        dataGridView1[2, i].Value = hasilCluster[i].ToString();
        }
        }
private void excelToolStripMenuItem_Click(object sender, EventArgs e)
        {
```

```
      string str;
      intrCnt;
      intcCnt;

      OpenFileDialogopf = new OpenFileDialog();
      opf.Filter = "Файл Excel|*.XLSX;*.XLS";//подключениефайлов excel
      opf.ShowDialog();
      System.Data.DataTabletb = new System.Data.DataTable();
      string filename = opf.FileName;

      Microsoft.Office.Interop.Excel.ApplicationExcelApp = new
Microsoft.Office.Interop.Excel.Application();

      Microsoft.Office.Interop.Excel._WorkbookExcelWorkBook;
      Microsoft.Office.Interop.Excel.WorksheetExcelWorkSheet;

      Microsoft.Office.Interop.Excel.RangeExcelRange;
      ExcelWorkBook = ExcelApp.Workbooks.Open(filename, 0, true, 5, "", "",
true, Microsoft.Office.Interop.Excel.XlPlatform.xlWindows, "\t", false,
      false, 0, true, 1, 0);
      ExcelWorkSheet =
(Microsoft.Office.Interop.Excel.Worksheet)ExcelWorkBook.Worksheets.get_Item(1);/
/работасмодулями Microsoft office

      ExcelRange = ExcelWorkSheet.UsedRange;
      for (rCnt = 1; rCnt<= ExcelRange.Rows.Count; rCnt++)
      {
      dataGridView2.Rows.Add(1);//заполнение dataGridView2
      for (cCnt = 1; cCnt<= 2; cCnt++)
      {
      str = (string)(ExcelRange.Cells[rCnt, cCnt] as
Microsoft.Office.Interop.Excel.Range).Value2;
      dataGridView2.Rows[rCnt - 1].Cells[cCnt - 1].Value = str;
      }
      }
      ExcelWorkBook.Close(true, null, null);
      ExcelApp.Quit();
releaseObject(ExcelWorkSheet);
      releaseObject(ExcelWorkBook);
      releaseObject(ExcelApp);

      treeView1.Visible = false;//скрытиеиотображениеэлементов
```

```
dataGridView1.Visible = false;
dataGridView2.Visible = true;
dataGridView3.Visible = false;

dataGridView4.Visible = false;
comboBox1.Visible = false;
textBox1.Visible = true;
label1.Visible = true;
}
```

Приложение Б

Таблица тип шкал и величин:

Тип шкалы (т.е. класс C_1) и название допустимого преобразования из C_1	Название типа шкалы	Примеры величин, измеряемых в шакалах данного типа
$\{\varphi \mid \varphi: f(A)\to B, \varphi(x) = x\}$, тождественное преобразование	Абсолютный	Результат счета
$\{\varphi \mid \varphi: f(A)\to B, \varphi(x) = \alpha x, \alpha>0\}$, преобразование подобия	Отношений	Масса, температура по Кельвину, время (интервалы), длина, коэффициент интеллектуальности и т. д.
$\{\varphi \mid \varphi: f(A)\to B, \varphi(x) = \alpha x+\beta, \alpha>0\}$, позитивное преобразование	Интервалов	Температура по Фаренгейту, Цельсию и т. д. время (календарь)
$\{\varphi \mid \varphi: f(A)\to B, \varphi(x) > \varphi(y) \leftrightarrow x>y\}$, (строго) монотонное возрастающее преобразование	Порядковый	Предпочтение, твердость по Моосу, степень умения и т. д.
$\{\varphi \mid \varphi: f(A)\to B, x \neq y \leftrightarrow f(x) \neq f(y)\}$, взаимно однозначное преобразование	Номинальный	Коды, названия профессий и т. д.