

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

09.03.03 Прикладная информатика

(код и наименование направления подготовки, специальности)

«Бизнес-информатика»

(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Информационная система поддержки сетевого обучения (на примере АО «ИВК»)

Обучающийся

Д. М. Деревянко

(Инициалы Фамилия)

(личная подпись)

Руководитель

Н.Н. Казаченок

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2022

Аннотация

Тема бакалаврской работы – «Информационная система поддержки сетевого обучения (на примере АО «ИВК»)».

Целью бакалаврской работы является разработка информационной системы для поддержки сетевого обучения в учебном центре компании АО «ИВК».

Работа выполнена в объёме 92 страниц, содержит 68 рисунков и 5 таблиц.

Бакалаврская работа состоит из введения, трёх глав, заключения и списка использованной литературы.

Во введении к бакалаврской работе подробно описывается актуальность выбранной темы, определяется цель работы и задачи по её достижению, формулируются объект и предмет исследования.

Первая глава работы описывает функциональное моделирование проектируемой информационной системы. В ней проводится анализ предметной области, выделяются интересующие нас бизнес-процессы, проводится их функциональная декомпозиция и формируются требования к информационной системе.

Во второй главе проводится концептуальное и логическое проектирование информационной системы, выстраивается её архитектура, разрабатывается диаграмма вариантов использования и логические схемы данных информационной системы.

Третья глава бакалаврской работы посвящена практической реализации функций информационной системы. В ней проводится выбор средств реализации ИС, а затем на выбранном движке реализуются ранее спроектированные функциональные возможности.

В заключении подводятся итоги работы, перечисляются выполненные задачи и мероприятия, описывается практическая ценность бакалаврской работы.

Оглавление

Введение.....	5
Глава 1 Функциональное моделирование системы сетевого обучения АО ИВК	8
1.1 Техничко-экономическая характеристика системы сетевого обучения АО ИВК.....	8
1.2 Описание уровня информационных технологий предприятия	11
1.3 Концептуальное моделирование системы сетевого обучения АО ИВК.....	14
1.3.1 Разработка и анализ бизнес-процессов «Как есть»	14
1.3.2 Обоснование необходимости автоматизации	17
1.4 Анализ существующих разработок	18
1.5 Постановка задачи и требования к разработке проекта АИС в АО ИВК.....	25
Глава 2 Логическое проектирование системы сетевого обучения АО ИВК ..	30
2.1 Логическая модель системы сетевого обучения АО ИВК и ее описание.....	30
2.1.1 Разработка модели прикладных объектов конфигурации в виде диаграммы классов.....	31
2.2 Информационное обеспечение системы сетевого обучения АО ИВК	33
2.2.1 Используемые классификаторы и системы кодирования	33
2.2.2 Характеристика нормативно-справочной и входной оперативной информации.....	36
2.2.3 Характеристика выходной информации	37
2.3 Проектирование базы данных системы сетевого обучения.....	38
2.3.1 Выбор технологии проектирования базы данных системы сетевого обучения.....	38

2.3.2	Разработка концептуальной базы данных системы сетевого обучения	40
2.3.3	Разработка логической модели данных системы сетевого обучения	42
2.4	Требования к аппаратно-программному обеспечению системы сетевого обучения	43
Глава 3	Физическое проектирование АИС	45
3.1	Выбор технологии и платформы разработки программного обеспечения АИС	45
3.2	Описание выбранной платформы разработки	49
3.3	Разработка физической модели данных АИС	51
3.4	Разработка программного обеспечения	56
3.4.1	Схема взаимосвязи модулей приложения ССО и их описание	56
3.4.2	Разработка прикладных объектов	59
3.4.3	Разработка запросов и отчетов	69
3.5	Тестирование и описание функциональности системы сетевого обучения	72
3.5.1	Выбор методики и критериев тестирования	72
3.5.2	Тестирование системы сетевого обучения	73
	Заключение	87
	Список используемой литературы и используемых источников	89
	Приложение А Текст программных модулей справочников	93
	Приложение Б Текст общих программных модулей	95
	Приложение В Текст программных модулей документов и регистров	102

Введение

Темой работы является разработка информационной системы поддержки сетевого обучения сотрудников АО ИВК. На предприятии существует учебный центр, в котором все сотрудники предприятия периодически проходят обучение с целью повышения уровня своей квалификации. Разрабатываемая система сетевого обучения (ССО) позволит организовать процесс доведения до сотрудников теоретического материала по изучаемым курсам повышения квалификации, а также позволит объективно оценить их знания и навыки по различным компетенциям при помощи практических занятий и тестов.

Интерактивные средства представления учебного материала давно используются в преподавании курсов повышения квалификации. Состав их очень широк – от простых текстовых файлов, до видео и интерактивных презентаций. Разрабатываемая ССО позволит хранить в себе разрозненные теоретические материалы, созданные при помощи различных программных инструментов, в едином хранилище.

Но преподнесение теоретического материала – только часть процесса повышения квалификации. Контроль усвоения материала, умение выполнять практические задания по изученному теоретическому материалу являются второй важнейшей составляющей образовательного процесса. Современные информационные технологии также успешно применяются в процессе контроля знаний. В основном это выражается в системах тестирования знаний, когда сотруднику, проходящему курсы повышения квалификации, предлагается выбрать правильный ответ на вопрос из списка возможных ответов. Но современные системы контроля знаний имеют различные формы от простого тестирования до интерактивного взаимодействия с виртуальным преподавателем.

Разработка такой системы и является целью данной работы. При этом разрабатываемая система должна стать не просто системой тестирования, а

комплексной системой автоматизации процесса обучения по всем преподаваемым курсам повышения квалификации в учебном центре предприятия. Отличительной ее особенностью станет возможность обратной связи с преподавателями центра.

Актуальность работы состоит в автоматизации некоторой рутинной части образовательного процесса, которую возможно выполнить без личного общения с преподавателем. Тем самым высвобождается время для творческой разработки содержания теоретического материала, интересных и сложных практических заданий.

Немаловажным преимуществом разрабатываемой системы должна стать возможность ее удаленного применения как обучающимися, так и преподавателями. Это позволяет иметь доступ к образовательному процессу вне стен учебного центра, планировать прохождение материала индивидуально, разносить изучение материала во времени. При этом важнейшей особенностью системы будет обратная связь с преподавателем, что даст возможность заниматься с каждым обучающимся индивидуально, что практически невозможно при обычных аудиторных занятиях.

Цель исследования – разработка информационной системы поддержки процессов сетевого обучения сотрудников АО ИВК.

Объектом исследования является процесс организации сетевого обучения сотрудников.

Предмет исследования – автоматизация процесса преподнесения теоретического материала обучающимся сотрудникам и прохождения ими практических занятий.

Задачи исследования:

- обоснование необходимости разработки сетевой системы обучения, обеспечивающей сотрудников, проходящих обучение в центре, как теоретическими материалами для подготовки к занятиям, так и позволяющей выполнять практические задания;

- формализация задачи, постановка задачи на разработку программного продукта;
- выбор платформы разработки;
- разработка структуры системы;
- разработка программного обеспечения задачи;
- тестирование разработанного программного продукта, устранение ошибок и недостатков.

Для реализации функциональных возможностей системы использована платформа «1С:Предприятие 8.3». Ее широкие возможности позволили реализовать все необходимые функции системы [10].

Теоретический материал, практические задания и тесты должны составляться квалифицированными специалистами учебного центра, полностью владеющими предметом и способными подать материал наиболее емко и полезно.

Результатом работы будет являться готовая система, которую можно использовать для обеспечения сотрудников, проходящих обучение в центре, теоретическими материалами для подготовки к занятиям, а также проверки их практической подготовки. Проверка практической подготовки осуществляется путем выставления сотрудникам практических заданий, которые они должны выполнить или тестов, которые они должны пройти.

Практическая значимость работы состоит в возможности применения разработанной системы сетевого обучения в образовательном процессе учебного центра.

Глава 1 Функциональное моделирование системы сетевого обучения АО ИВК

1.1 Технико-экономическая характеристика системы сетевого обучения АО ИВК

Система сетевого обучения (ССО) разрабатывается для сотрудников АО ИВК. АО ИВК – предприятие, занимающееся разработкой программного обеспечения. Сфера деятельности определяют и структуру фирмы, представленную на рисунке 1.

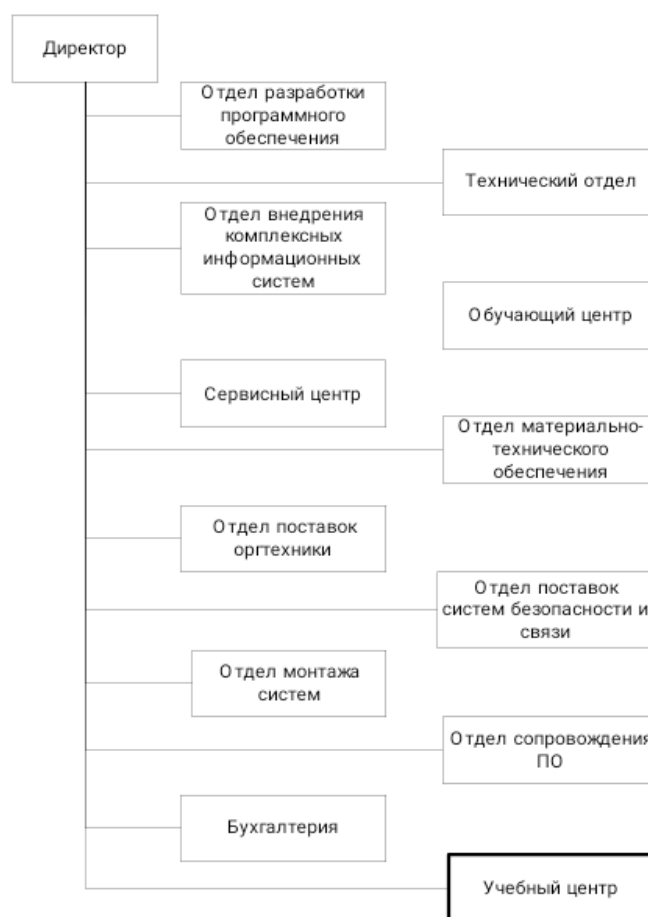


Рисунок 1 – Структура предприятия

Наибольший интерес в работе представляет учебный центр. Сотрудникам предприятия требуется постоянно повышать свою

квалификацию, чтобы быть в тренде современных технологий и соответствовать высоким требованиям. Для этого на предприятии существует учебный центр, задачей которого является повышение уровня теоретических знаний сотрудников и улучшение их профессиональных навыков. Сотрудники предприятия периодически проходят курсы повышения квалификации по различным направлениям в учебном центре. В основном это происходит без отрыва от производства, в том числе удаленно. Многие материалы сотрудникам, обучающимся в центре, приходится изучать самостоятельно.

Ядро команды учебного центра составляют профессионалы, обладающие уникальным опытом и многолетней образовательной практикой. Отработанное за много лет сотрудничества командное взаимодействие, понимание каждым преподавателем центра своей роли в образовательном процессе, профессионализм и нацеленность на единый конечный результат, позволяют центру держать высокую планку качества в деле повышения квалификации сотрудников предприятия.

Преподавание сложных курсов повышения квалификации требует от преподавателей учебного центра определенных усилий. Слушателей курсов нужно обеспечить учебными, методическими, обучающими материалами. Обеспечение материалами должно проводиться в рамках направлений, в разрезе которых проводятся занятия, а в рамках каждого направления – в разрезе занятий.

Кроме обеспечения учебными материалами важно проверить усвоение этих материалов, убедиться, что слушатели курсов повышения квалификации усвоили эти материалы, смогут применять полученные знания и навыки в своей профессиональной деятельности в будущем. Это можно сделать только путем выполнения слушателями курсов практических заданий по изучаемому занятию в рамках выбранного направления курса повышения квалификации. Как правило, эти задания носят практический характер, результатом их выполнения является какая-то разработка, в виде файлов.

Проверку освоенности материала можно организовать и путем прохождения тестов. Тесты позволяют быстро и достаточно объективно составить представление о степени освоенности слушателем курсов материала и оценить его готовность к практической работе в будущем.

Улучшить уровень подготовки сотрудников, проходящих обучение в центре, можно, если определены конкретные образовательные цели и стандарты их достижения, то есть если определен конечный результат обучения. Под конечным результатом понимается совокупность знаний, умений, опыта и компетенций, которыми должен обладать сотрудник к концу этапа обучения по одному из направлений курсов повышения квалификации в учебном центре.

Система оценки качества прохождения курсов повышения квалификации и усвоения преподаваемого материала должна давать возможность получения информации об уровне компетенций, которые получили сотрудники в результате прохождения курсов. Система должна давать возможность использовать различные варианты оценки полученных сотрудниками знаний. Обычный подсчет баллов, на основе которого формируется оценка за зачет или экзамен, ставит под сомнение оценку эффективности обучения только на основе автоматически подсчитанных баллов. На успешность сдачи экзамена влияет большое количество случайных факторов, не относящихся к усвоению материала и получению практических навыков. Систематически собираемые данные о результатах обучения гораздо эффективнее, так как оценка становится «интегральной» составляющей образовательного процесса.

Необходимо помнить о том, что качество тестов, их справедливость должны контролироваться на всем их жизненном цикле, начиная от их создания до прохождения слушателями курсов. Для этого преподаватели центра должны быть вовлечены в процесс разработки и реализации системы сетевого обучения.

1.2 Описание уровня информационных технологий предприятия

С уверенностью можно сказать о том, что процедура автоматизации образовательного процесса в рамках исследуемого предприятия – важная и целесообразная процедура. Дело в том, что у предприятия на данный момент имеется мощная программная, техническая инфраструктура. Заново создавать ничего не придется; достаточно только адаптировать некоторые решения под создаваемую ССО. Представим эту архитектуру.

В качестве базы ИТ-архитектуры предприятия выступает сервер «Dell R440». Это техническое, современное решение, которое позволяет обеспечивать доступ пользователей к принтерам, общим данным, файлам и иным сервисам. В частности, «Dell R440» имеет такие параметры:

- процессор: 2 × Intel Xeon Silver 4208 (8C 11M Cache 2.10 GHz);
- оперативная память: 4x DDR4 RDIMM 2133MHz REF 32 Гб, всего 128 Гб;
- контроллер RAID: Dell H330 (ZM);
- модуль удаленного управления iDRAC 9 Enterprise;
- интегрированная сетевая карта Dell 2port 1Gb;
- жесткий диск: Dell 1.2TB SAS 10k 2.5» G14;
- блок питания: 2x Dell 550w Hot Plug.

В ООО «ПродТорг» указанный сервер исполняет функции контролера домена, сервера почты, контроля над доступом в Интернет. В некотором плане, это сервер приложений 1С и сервер БД. Работа его выстраивается на основе Windows Server Standard 2014 OEM. В качестве ИБП выступает «APC Smart-UPS RT 1000VA 230V».

У каждого сотрудника на рабочем месте стоит ПК с архитектурой «x86» или «x86-64». Примерный ПК имеет следующие характеристики:

- процессор – «Intel I3»;
- оперативная память: 4 Gb;
- жесткий диск 500 Гб.

Далее на рисунке 2 будет представлена техническая архитектура предприятия.

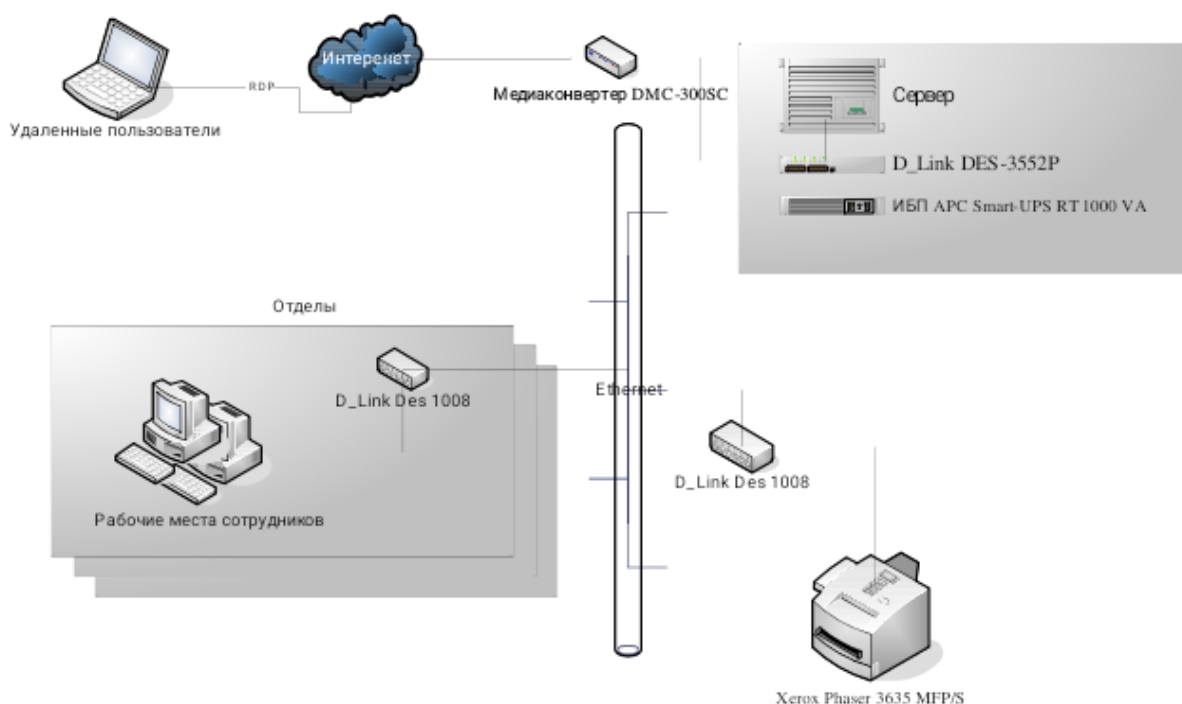


Рисунок 2 – Техническая архитектура ИС предприятия

Кроме того, стоит добавить, что в компании установлены и успешно работают программные системы. Как и прочие компании, данное предприятие ведет бухгалтерский учет, формирует и передает налоговую, финансовую отчетность. Также осуществляется процедура учета работников компании, в том числе – учет зарплаты. Специально для этого используется ПП под названием «1С: Бухгалтерия 8, редакция 3.0», на основе «1С: Предприятие 8.3» [16]. На самом деле, у представленной системы есть множество возможностей, которые позволяют вести бухгалтерский учет, а также учет сотрудников.

«1С: Бухгалтерия 8, редакция 3.0» применяется на основе клиент-серверного варианта. Ранее было сказано о том, что в качестве физического сервера применяется «Dell R440», на нем работает сервер приложений 1С. Он осуществляет отсылку на СУБД «Microsoft SQL Server 2014», который

используется для хранения данных. «Microsoft SQL Server 2014» развернут на том же физическом сервере «Dell R440». У сотрудников на рабочих местах стоит клиентская часть программы «1С: Бухгалтерия 8». На ПК установлена операционная система «Windows 7» или «Windows 10» [1]. Именно она дает возможность подключаться к сети, применять ее всевозможные ресурсы. На рисунке 3 далее будет представлена программная архитектура компании.

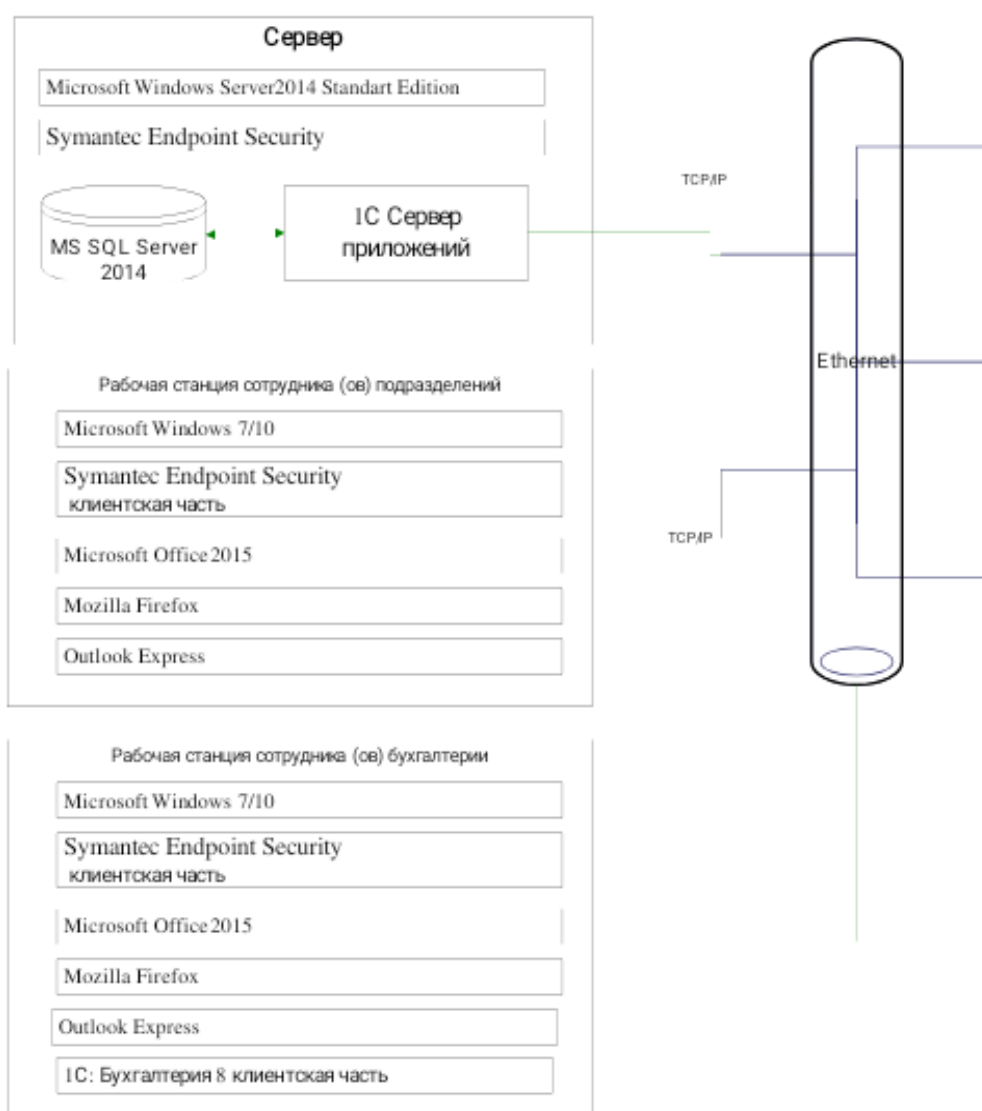


Рисунок 3 – Программная архитектура ИС предприятия

Известно о том, что файл-сервер компании выступает в качестве хранилища информации по учету. Посредством папки под названием «Торговый учет» ведется учет операций [3]. В целом, это файлы в форматах «word» и «excel». Каждый участник процесса на основе шаблонов вносит в шапку «Торговый учет» на файловом сервере собственные данные, связанные с торговыми и складскими операциями. Тот же сервер «Dell R440» применяется в качестве файл-сервера.

1.3 Концептуальное моделирование системы сетевого обучения АО ИВК

1.3.1 Разработка и анализ бизнес-процессов «Как есть»

Приведем описание процесса обучения и контроля знаний при помощи IDEF - диаграмм. На рисунке 4 изображена диаграмма нулевого уровня.

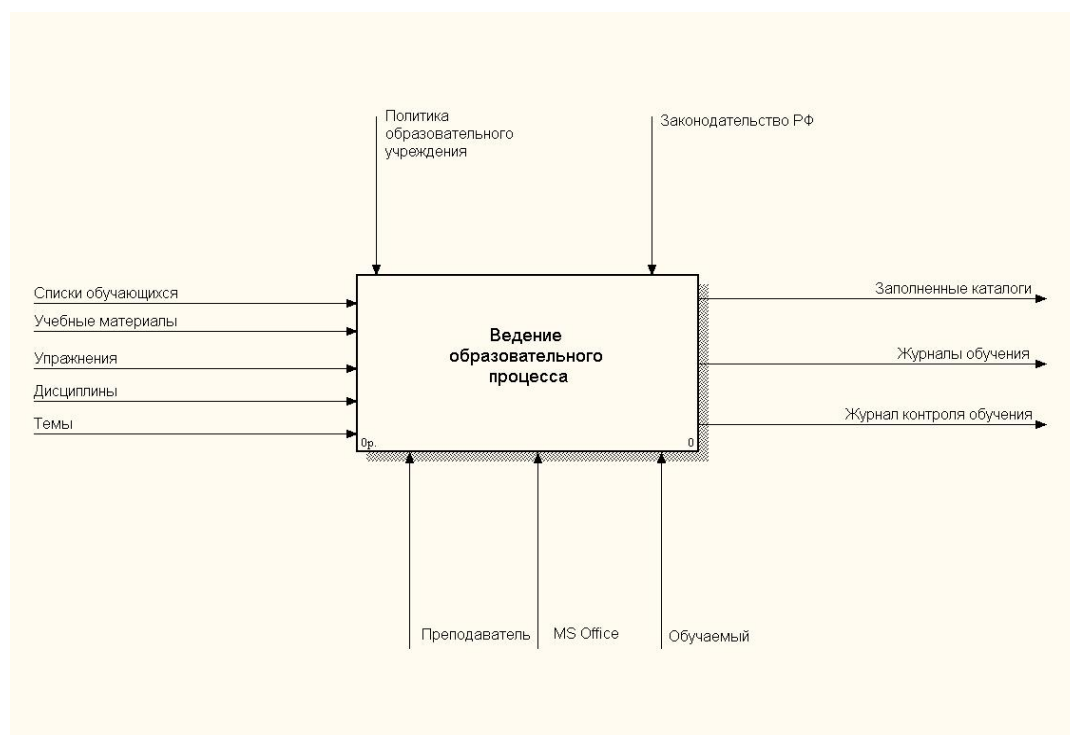


Рисунок 4 – Диаграмма нулевого уровня процесса обучения и контроля знаний

Далее проведена декомпозиция процесса на подпроцессы, которая приведена на рисунке 5.

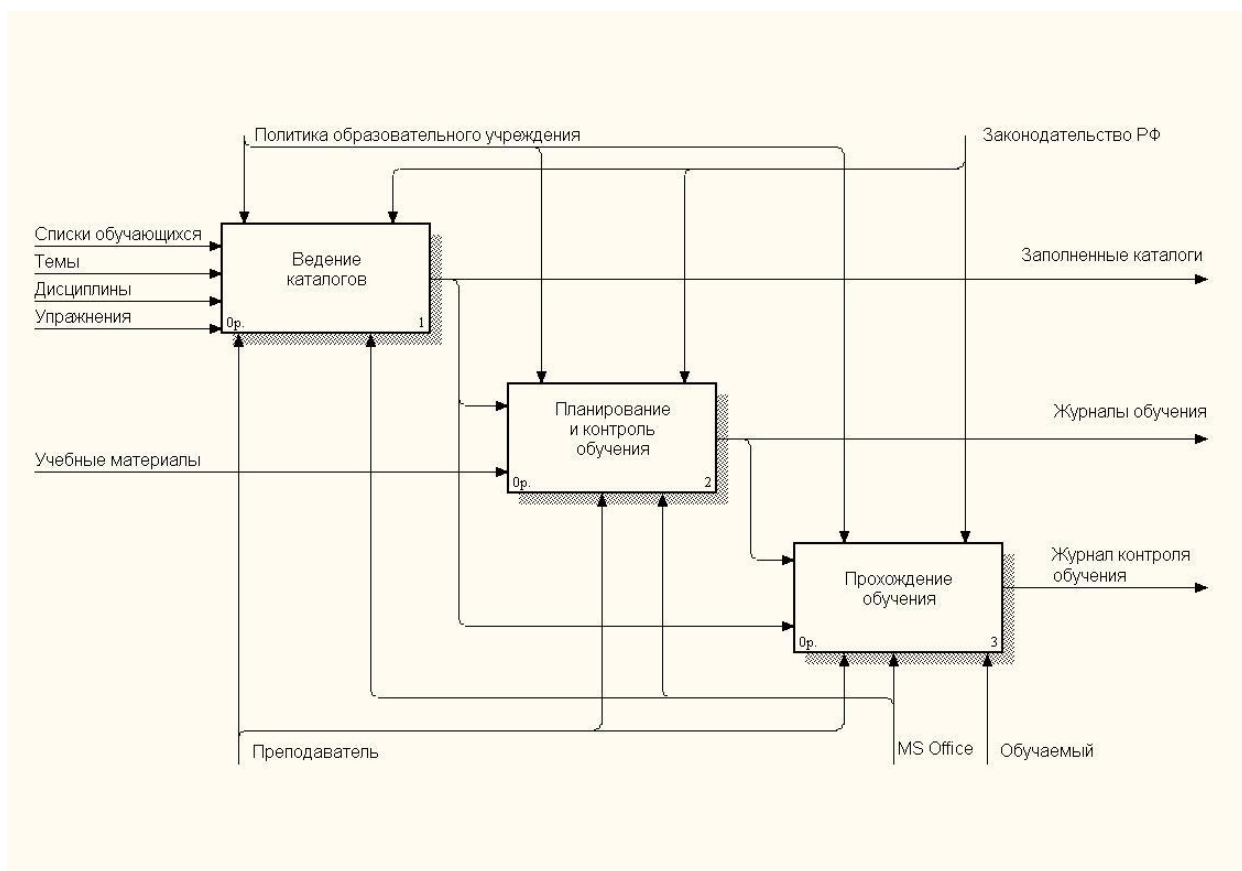


Рисунок 5 – Декомпозиция процесса обучения и контроля знаний

Основной проблемой существующей системы обучения в центре является разрозненность хранения учебных и методических материалов по отдельным папкам на сервере учебного центра, а также хранение части данных в бумажном виде. Это приводит к сложности поиска нужных файлов, возможности их потери, ошибках в оценке знаний, большой объем ручного ввода данных, невозможности автоматического получения отчетов.

Далее проведена декомпозиция каждого из подпроцессов. На рисунке 6 приведена декомпозиция процесса «Ведение каталогов». На рисунке 7 приведена декомпозиция процесса «Планирование и контроль обучения», на рисунке 8 приведена декомпозиция процесса «Прохождение обучения».

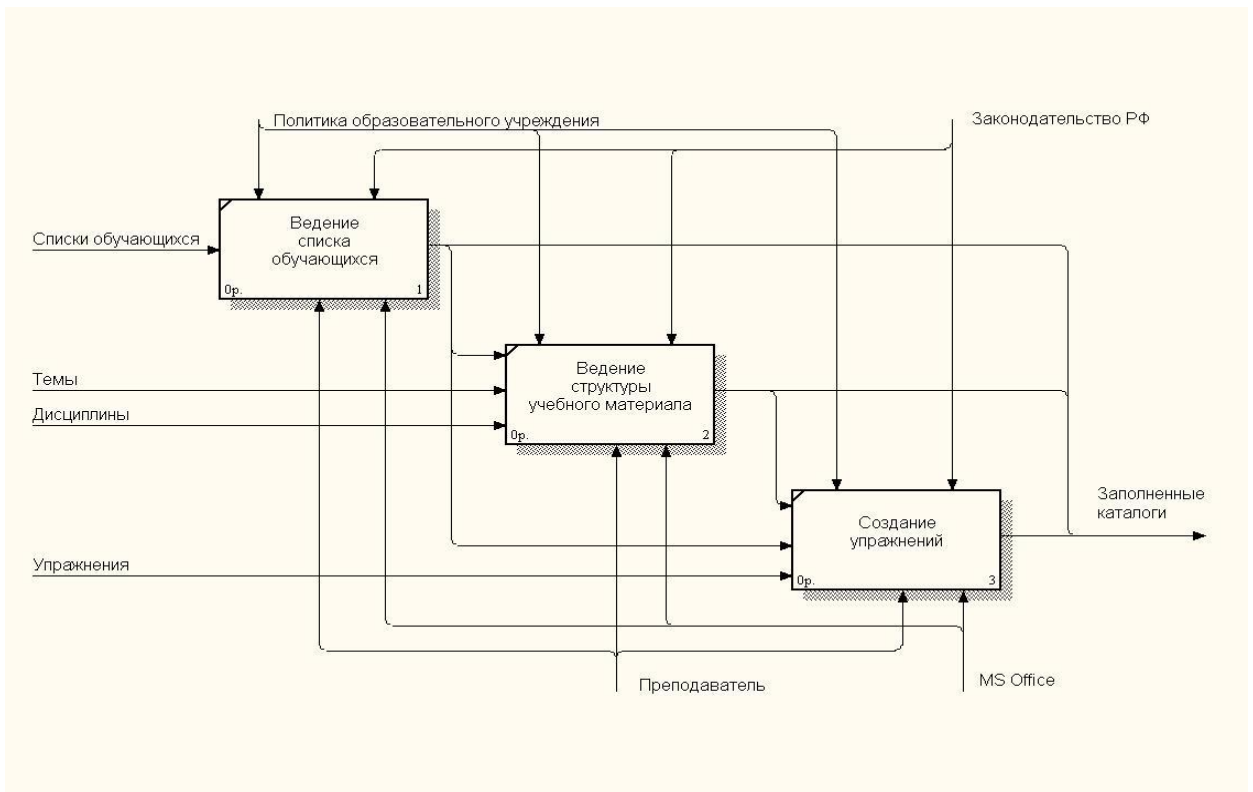


Рисунок 6 – Декомпозиция процесса «Ведение каталогов»

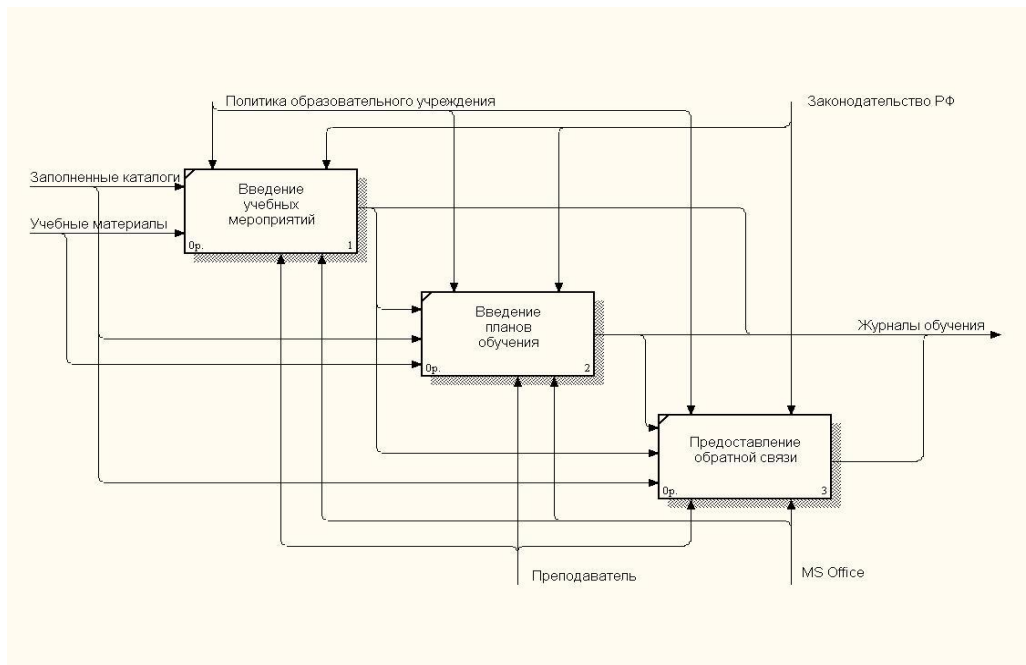


Рисунок 7 – Декомпозиция процесса «Планирование и контроль обучения»

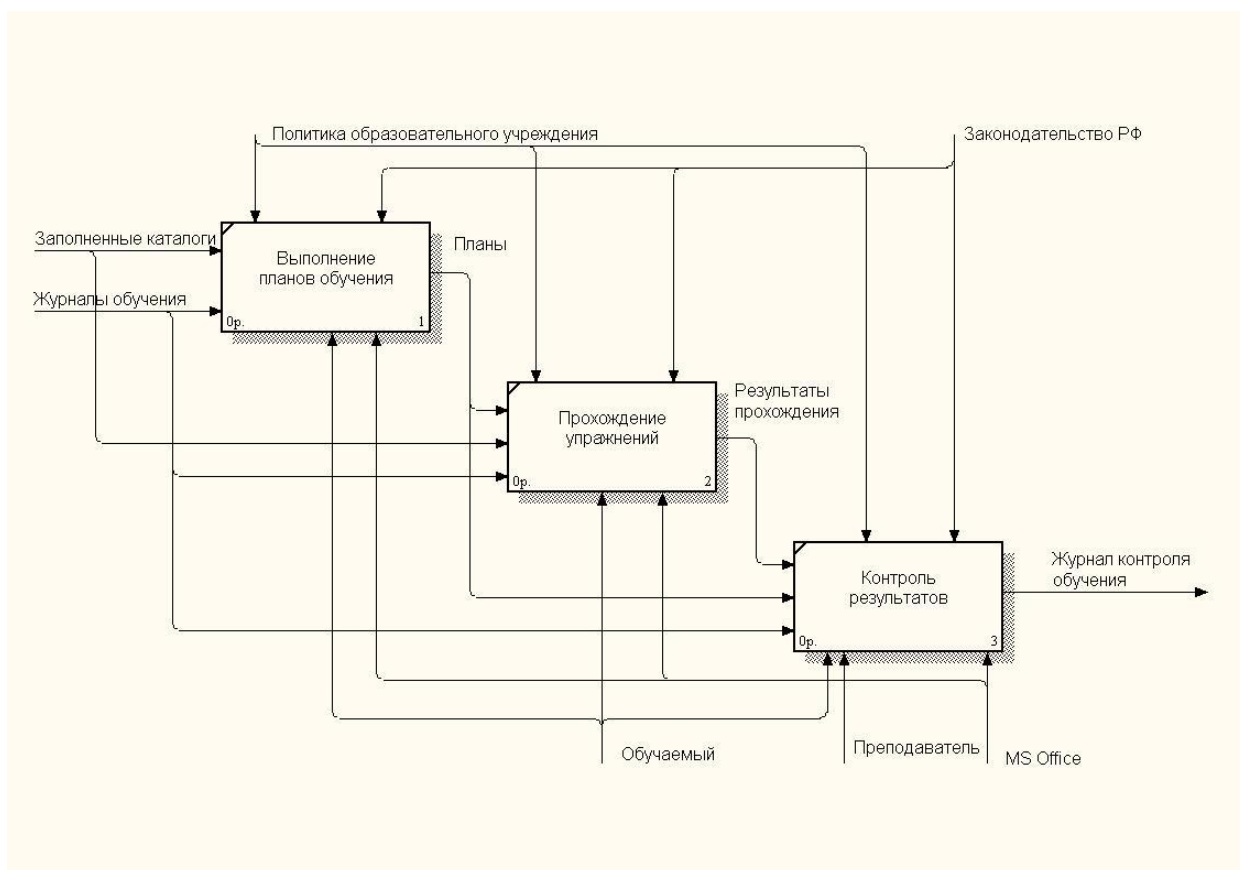


Рисунок 8 – Декомпозиция процесса «Прохождение обучения»

Из анализа процессов видно, что в принципе процессы построены верно. Неверно выбраны инструменты, при помощи которых эти процессы реализуются: разрозненные файлы формата Word и Excel, хранящиеся на файл-сервере предприятия. Заменить эти инструменты на ССО и есть главная цель работы.

1.3.2 Обоснование необходимости автоматизации

В работе решается задача автоматизации процедуры подготовки сотрудников, проходящих обучение по различным курсам повышения квалификации, преподаваемым в учебном центре, и контроля степени подготовки сотрудников [4]. Актуальность задачи обусловлена тем, что объем данных процесса подготовки учебных материалов и контроля знаний таков, что его нормальное функционирование без использования средств вычислительной техники и программного обеспечения невозможно.

Основной проблемой существующей системы обучения в центре является разрозненность хранения учебных и методических материалов по отдельным папкам на сервере учебного центра, а также хранение части данных в бумажном виде. Это приводит к сложности поиска нужных файлов, возможности их потери, ошибках в оценке знаний, большой объем ручного ввода данных, невозможности автоматического получения отчетов [25].

Разработка ССО, позволяющей централизованно и удаленно снабдить всех участников образовательного процесса материалами и в тоже время позволяющей составить для каждого слушателя курсов индивидуальный план подготовки, становится очень актуальной задачей. Неотъемлемой частью системы должна стать возможность выкладывать не только теоретический, но и практический материал и проверять выполнение слушателями курсов практических заданий [8].

Таким образом, разработка и последующее внедрение использование системы сетевого обучения, выполняющей предоставление теоретического материала для подготовки к занятиям и проверку (в том числе тестирование) знаний слушателей курсов, поднимает качество образовательного процесса на новый уровень, что дает неоспоримый эффект.

1.4 Анализ существующих разработок

Существует большое количество программных продуктов, направленных на обучение и тестирование знаний. Рассмотрим некоторые из них.

В сфере обучения наблюдается бум использования платформ для управления и администрирования онлайн-курсов, известных как системы управления обучением или LMS.

Платформы LMS обладают способностью объединять людей, процессы и оборудование в виртуальные классы для онлайн-обучения. В основном, платформа LMS содержит шаблон для предоставления новостей, контента,

организации курса, мероприятий, дискуссионных форумов, ссылок на документы в различных форматах [12].

Эти платформы не являются открытыми, они требуют разрешений, участник должен использовать «имя пользователя» и «пароль», чтобы иметь возможность войти в курс. На главной странице есть несколько опций, представляющих интерес для пользователя, например:

- общая информация об учебном заведении, спонсирующем курсы;
- список курсов и предлагаемых курсов;
- информация о процессах зачисления и личные данные;
- контактная информация для представителей учреждения;
- доступ к интересующим сайтам (электронные библиотеки);
- ориентация студентов;
- подробная информация о процессе поступления.

Разработкой курсов обычно занимаются организации, которые стремятся к созданию университетов или учебных центров.

В настоящее время на рынке существуют сотни платформ, поэтому прежде, чем выбрать одну из них для разработки курса, важно проанализировать их, чтобы оценить, отвечают ли они потребностям и целям учебного заведения, характеристикам целевой аудитории и образовательным проектам.

Когда мы ссылаемся на понятия, относящиеся к технологическим аспектам или к информационной сфере, мы склонны путать некоторые элементы, используемые в среде, из-за их концептуальной близости; например, LMS часто путают с виртуальным классом или электронным обучением, но это не так.

С помощью LMS можно разработать специализированные курсы с конкретными целями, например: системы управления обучением, специально разработанные для учебных заведений, компаний или рабочих областей, целью которых является виртуальное образование, или которым необходимо обучать определенную группу населения, которая не может посещать очные

занятия. Этот тип платформ подходит для групп населения, которые находятся на больших расстояниях или которым, в силу различных ситуаций с расписанием, требуется гибкость и индивидуальность в процессе обучения. Поэтому эта среда развивается в открытых пространствах, где не существует термина «лицом к лицу», а скорее «репрезентативность»; поскольку важно не физическое присутствие людей, а их репрезентативность в различных видах деятельности. LMS может быть синхронной: когда они происходят в одно и то же время, примеры: электронные разговоры и видеоконференции. В них требуется, чтобы участники были согласованы во времени, но не обязательно в месте; возможно, чтобы каждый из них был подключен одновременно из любого места связи. Или может быть асинхронной: когда каждый участник может вмешиваться или выполнять свою работу в разное время, без необходимости совпадения по времени. Примерами асинхронной деятельности являются: дискуссионные форумы, групповые форумы, отправка и получение документов, отправка сообщений электронной почты и другое.

Как отмечено выше, платформы электронного обучения отличаются своей природой, поскольку они полагаются на гибкость управления временем для разработки различных задач в разное время.

В современной методологии обучения, в связи с быстрым развитием информационных и коммуникационных технологий, концепция электронного обучения (дистанционного обучения) становится все более актуальной. Для программного обеспечения электронного обучения часто используется термин LMS, который является аббревиатурой от «системы управления обучением» [11].

Существует два типа учебных платформ: коммерческие и свободно распространяемые. Коммерческие платформы приобретаются; Интернет предоставляет нам информацию о каждой из них. Иногда можно протестировать программное обеспечение, получить доступ к ограниченным версиям для оценки программы. Если после оценки принято решение

приобрести платформу, необходимо произвести соответствующий платеж производителю. Примерами таких платформ являются следующие: WebSt Campus Edition и BlackBoard, которые являются одними из наиболее широко используемых на рынке.

Бесплатные или открытые программные LMS – это те, которые находятся в Интернете, являются некоммерческими и могут быть загружены бесплатно. Для их использования необходимо прочитать руководства пользователя и, в некоторых случаях, зарегистрироваться в качестве пользователя. Примерами таких платформ являются: Moodle, dotLRN, Claroline, Dokeos; среди прочих.

В книге Баженовой И. Ю система Moodle описывается так: «Moodle (модульная объектно-ориентированная динамическая учебная среда) – это свободная система управления обучением, ориентированная прежде всего на организацию взаимодействия между преподавателем и учениками, хотя подходит и для организации традиционных дистанционных курсов, а также поддержки очного обучения. Используя Moodle преподаватель может создавать курсы, наполняя их содержимым в виде текстов, вспомогательных файлов, презентаций, опросников и тому подобное. Для использования Moodle достаточно иметь любой веб-браузер, что делает использование этой учебной среды удобной как для преподавателя, так и для обучаемых. По результатам выполнения учениками заданий, преподаватель может выставлять оценки и давать комментарии. Таким образом, Moodle является и центром создания учебного материала и обеспечения взаимодействия между участниками учебного процесса. Moodle создается множеством разработчиков и переведена на десятки языков, в том числе и русский.» [7]

Moodle является веб-ориентированной средой. Для его работы требуется веб-сервер с поддержкой PHP (например Apache2) и сервер баз данных (по умолчанию используется MySQL).

Moodle – одно из самых популярных решений для электронного обучения на рынке. Moodle в первую очередь предназначен для учебных

заведений, хотя его можно использовать и вне этого, условно говоря, «институционального» контекста. Он разработан таким образом, что не представляет собой замену классическому, прямому способу обучения, скорее, можно сказать, что это дополнение к стандартной системе обучения, которое открывает целый ряд новых возможностей.

Если попытаться определить одним предложением, что такое Moodle – это унифицированный набор инструментов, который позволяет получить доступ к общим источникам знаний, обмениваться учебными материалами, а также использовать дополнительные модули, обеспечивающие расширение функциональности.

Первая версия Moodle была выпущена в середине 2002 года и с тех пор постоянно развивается. По состоянию на март этого года актуальна версия 2.8.5 Moodle написан на языке программирования PHP как решение с открытым исходным кодом, опубликованное под лицензией GNU GPL версии 3. Он отличается простотой использования.

Аббревиатура Moodle называется «модульная объектно-ориентированная динамическая среда обучения», она достаточно ярко говорит о ее назначении и архитектуре. Со временем программное обеспечение приобрело большую пользовательскую базу, и явным показателем его качества является тот факт, что сайт bestelearningplatforms.com Moodle занимает первое место в категории платформ, инструментов и программного обеспечения для электронного обучения для академических учреждений. Не следует упускать из виду тот факт, что Moodle – единственное бесплатное решение в списке, которое конкурирует с 4 другими коммерческими программными решениями.

Некоторые из функций, которые активируются с помощью дополнительных модулей, – это мониторинг работы студентов, тестирование и оценивание, взаимное взаимодействие и общение между участниками учебного процесса, то есть студенты и профессора, а также другие практические вещи.

Хотя Moodle разработан на платформе Linux, он является полностью мультиплатформенным решением. Программное обеспечение, необходимое для установки, включает веб-сервер с поддержкой языка программирования PHP и доступ к локальному или удаленному серверу базы данных, при этом следует отметить, что поддерживаются следующие реализации сервера базы данных: PostgreSQL, MariaDB, MySQL, MSSQL и Oracle (не полностью).

Сложность оборудования чрезвычайно относительно и зависит от нескольких факторов, но рекомендуется как минимум двухъядерный процессор с тактовой частотой 2 ГГц, 1 ГБ ОЗУ, 160 МБ дискового пространства для базовой установки плюс пара дополнительных гигабайт для хранения учебных материалов. Конечно, требования к оборудованию во многом зависят от требований конкурирующих пользователей, поэтому применяется правило – чем больше количество пользователей, тем мощнее оборудование, которое необходимо предоставить, с приблизительным расчетом, что на каждые 10–20 конкурирующих подключений должен быть 1 ГБ ОЗУ.

В дополнение к бесспорным качествам в виде простоты использования, масштабируемости, гибкости, безопасности, бесплатного использования и многочисленных сообществ, которые сформировались с течением времени, потенциальные недостатки включают сложную отчетность, управление пользователями и более низкую производительность при большом количестве конкурирующих соединений [9].

Moodle – это продукт электронного обучения, который постоянно развивается и за эти годы приобрел завидную популярность. Мы подтверждаем это утверждение данными с сайта <https://moodle.org> – Moodle локализован более чем на 100 языков и присутствует более чем в 200 странах [29].

Во всем мире зарегистрировано более 50 000 сайтов Moodle, а в России Moodle в настоящее время используется 198 университетами, школами и другими образовательными учреждениями. В разработку этой LMS было

вложено более 10 лет, и для каждой новой версии характерно множество нововведений и улучшений по сравнению с предыдущей, поэтому версия 2.8 содержала более 280 обновленных и новых функций по сравнению с предыдущей [30].

«MyTest X» - удобная АИС для создания и проведения компьютерного тестирования, сбора и анализа результатов тестирования.

С помощью «MyTest X» возможна организация и проведение тестирования, зачетов и экзаменов в образовательных учреждениях любого формата (вузы, колледжи, школы) как с целью контроля знаний по различным дисциплинам, так и с обучающими целями. Организации могут осуществлять аттестацию и сертификацию своих сотрудников.

Программа «MyTest X» работает с девятью типами заданий: одиночный выбор, множественный выбор, установление соответствия, установление порядка следования, ручной ввод числа, указание истинности или ложности утверждений, ручной ввод текста, перестановка букв, выбор места на изображении. В тесте можно задействовать любое количество этих типов заданий. В заданиях с выбором ответа (одиночный, множественный выбор, указание истинности, указание порядка) можно использовать до 10 (включительно) вариантов ответа.

Программа «MyTest X» состоит из трех модулей: «Модуль тестирования» (MyTestStudent), «Редактор тестов» (MyTestEditor) и «Журнал тестирования» (MyTestServer). Она предусматривает несколько независимых друг от друга режимов работы: обучающий, свободный, штрафной и монопольный.

Рассмотрев несколько из наиболее популярных и мощных готовых решений в области тестирования знаний, тем не менее, мы делаем вывод о необходимости разработки собственного программного продукта. Такая установка была сделана исходя из того, что в собственной разработке всегда можно реализовать специфические, необходимые только в данном учебном заведении возможности. Не исключено, что требования задействовать такие

возможности появятся уже после внедрения ССО, на этапе работы, если у заказчика появятся новые взгляды на необходимый функционал ССО. Реализовать такие требования возможно, лишь имея системы собственной разработки, так как готовые решения поставляются, как правило, без исходных кодов программы и не могут быть модифицированы или дополнены.

1.5 Постановка задачи и требования к разработке проекта АИС в АО ИВК

Система должна выполнять функции, представленные на рисунке 9.



Рисунок 9 – Состав функций ССО

Для более подробной демонстрации функций ССО были разработаны диаграммы «как будет» процесса обучения. Как уже говорилось ранее, в целом бизнес-процессы обучения построены верно. Неверно выбран инструмент их реализации – пакет MS Office. В новом варианте этот инструмент заменен на новый – информационную систему. На рисунке 10 приведена диаграмма нулевого уровня.

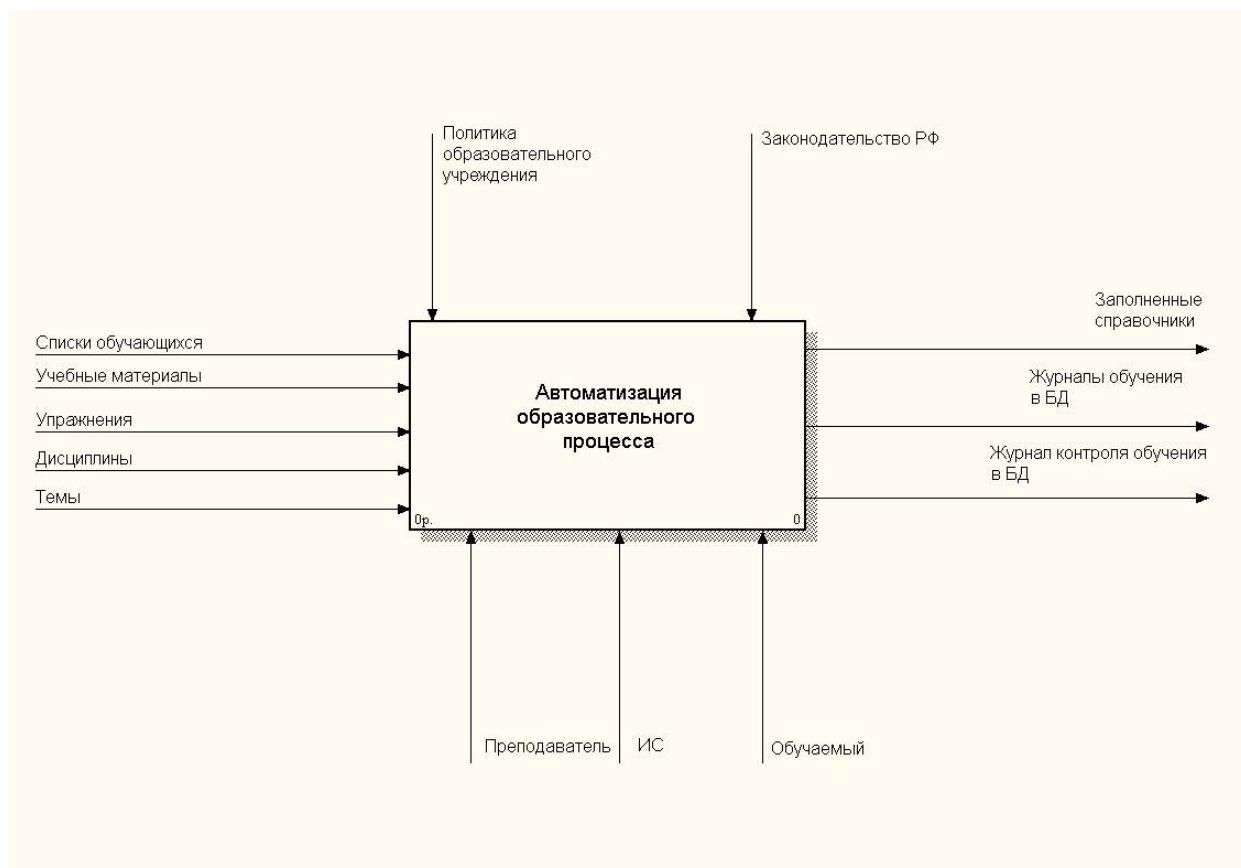


Рисунок 10 – Диаграмма нулевого уровня «Как будет»

Далее проведена декомпозиция основного процесса, представленная на рисунке 11 и декомпозиция каждого из подпроцессов. На рисунке 12 приведена декомпозиция процесса «Ведение справочников». На рисунке 13 приведена декомпозиция процесса «Планирование и контроль обучения», на рисунке 14 приведена декомпозиция процесса «Прохождение обучения».

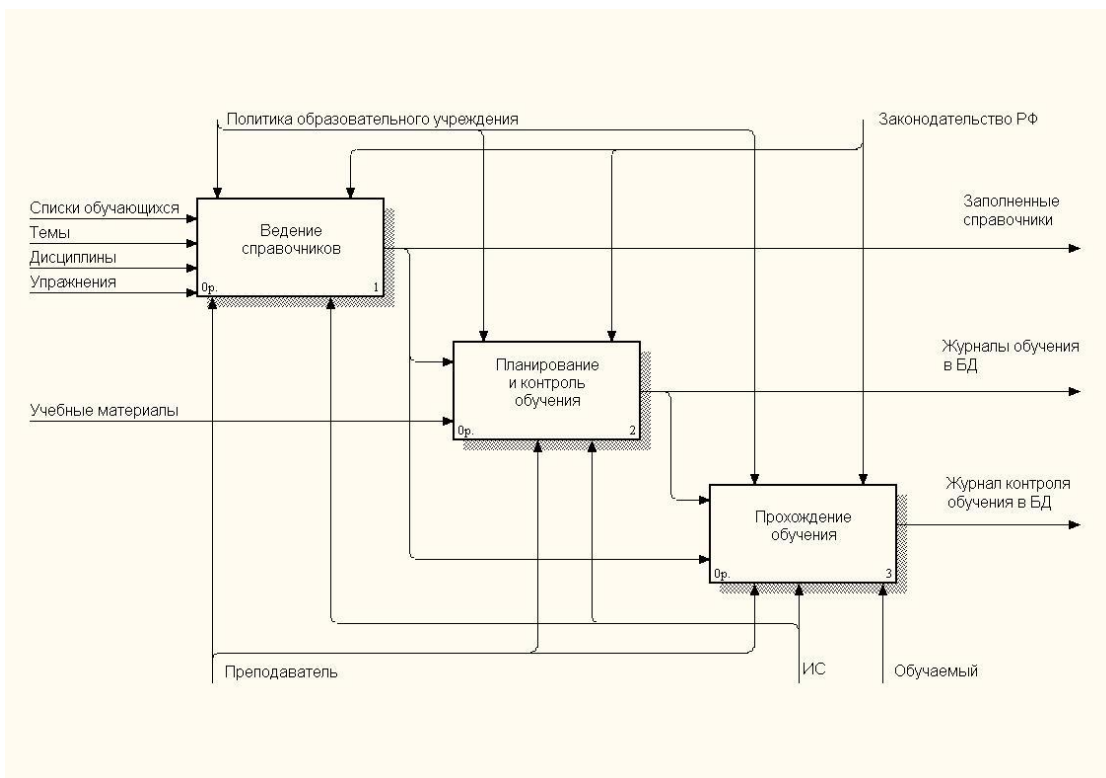


Рисунок 11 – Декомпозиция процесса обучения и контроля знаний с использованием ИС

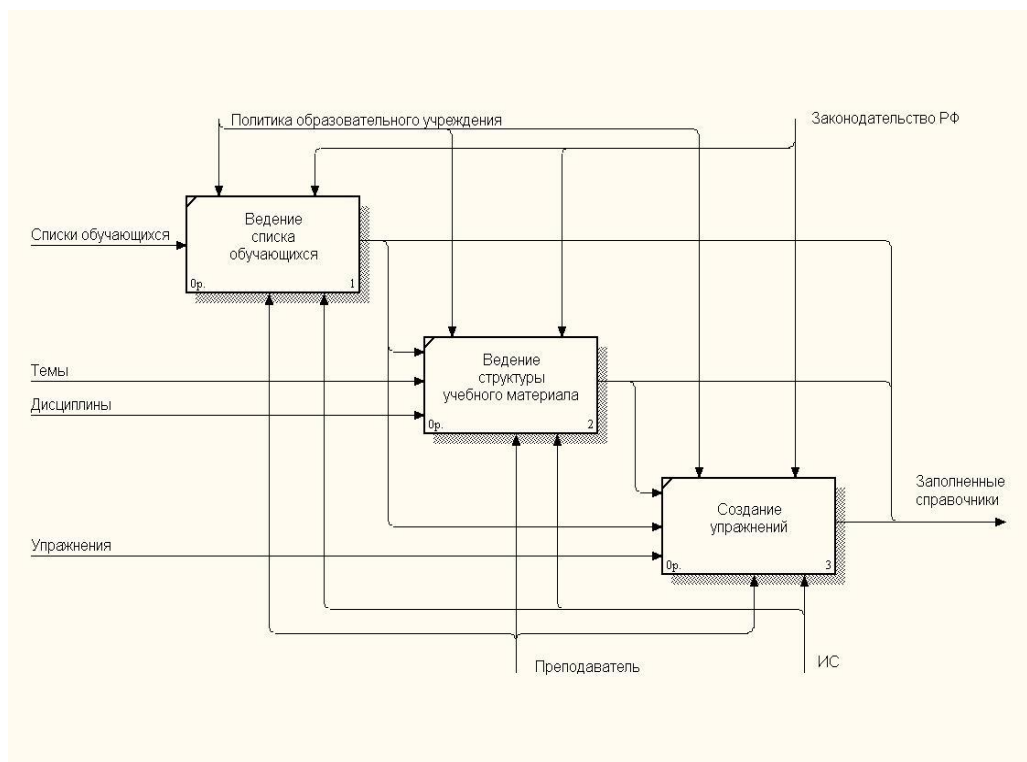


Рисунок 12 – Декомпозиция процесса «Ведение справочников»

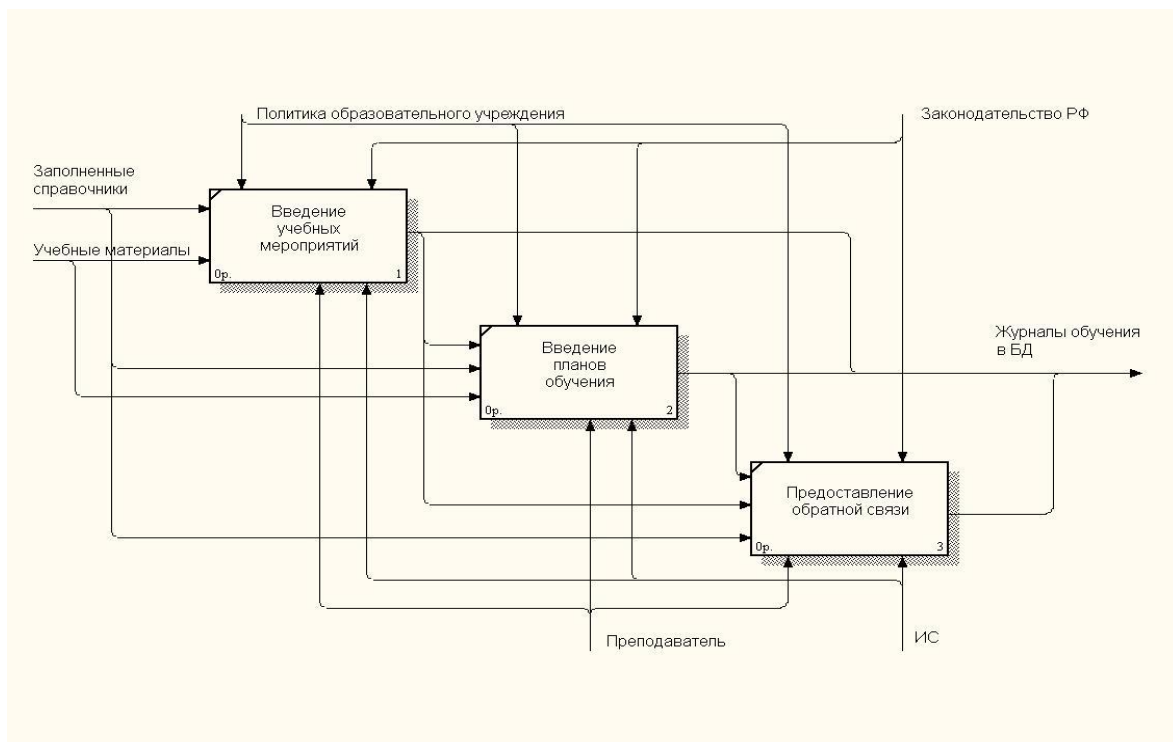


Рисунок 13 – Декомпозиция процесса «Планирование и контроль обучения»

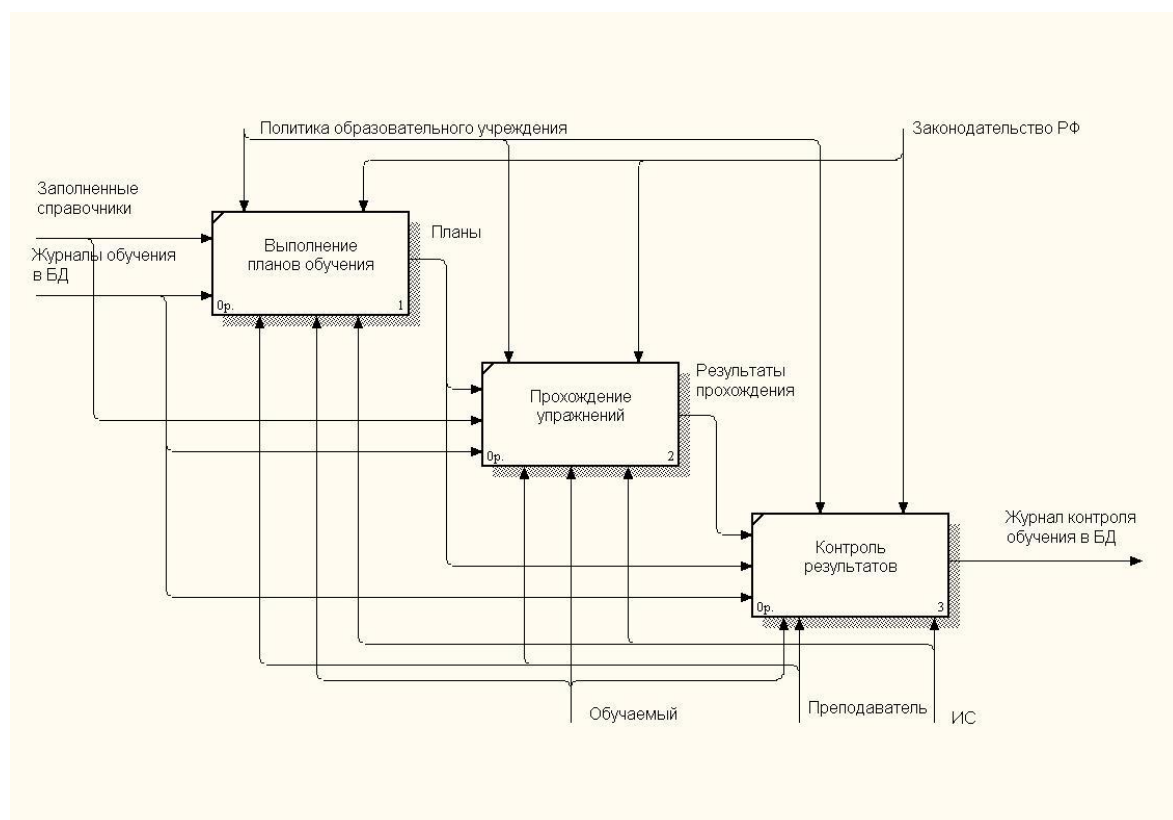


Рисунок 14 – Декомпозиция процесса «Прохождение обучения»

Далее были сформулированы требования, которым должна соответствовать будущая ССО:

- система должна автоматизировать образовательный процесс в АО ИВК и построить его как самостоятельное изучение обучающимися сотрудниками учебных и методических материалов;
- система должна вести список упражнений (теоретического материала и практических заданий) в рамках учебных тем;
- система должна быть разработана на базе платформы «1С: Предприятие 8.3» по технологии «Управляемое приложение»;
- система должна предоставлять обучающемуся сотруднику возможность выполнения заданий и тестов;
- система должна позволять преподавателю в качестве практических вопросов задавать не только тесты, но и практические задания, выполнять которые обучающиеся сотрудники должны вне рамок разрабатываемой системы, а результаты выполнения прикреплять к выполненному упражнению в виде файлов;
- система должна предоставлять средства контроля и оценки качества образовательного процесса;
- система должна предоставлять обратную связь между обучающимися и преподавателями;

На основании представленных требований переходим к логическому и физическому проектированию системы.

Выводы по первой главе

В первой главе доказано преимущество использования автоматизированного варианта преподнесения слушателям курсов теоретического материала, а также контроля усвоения этого материала. Проанализированы существующие автоматизированные системы контроля знаний и сделан вывод о том, что они нам не подходят. Сформулированы требования к разрабатываемой системе сетевого обучения.

Глава 2 Логическое проектирование системы сетевого обучения АО ИВК

2.1 Логическая модель системы сетевого обучения АО ИВК и ее описание

Все пользователи разрабатываемой ССО разделяются на три категории: преподаватели, обучающиеся и администраторы.

Для более наглядного представления функций каждого из пользователей была разработана диаграмма вариантов использования, представленная на рисунке 15.

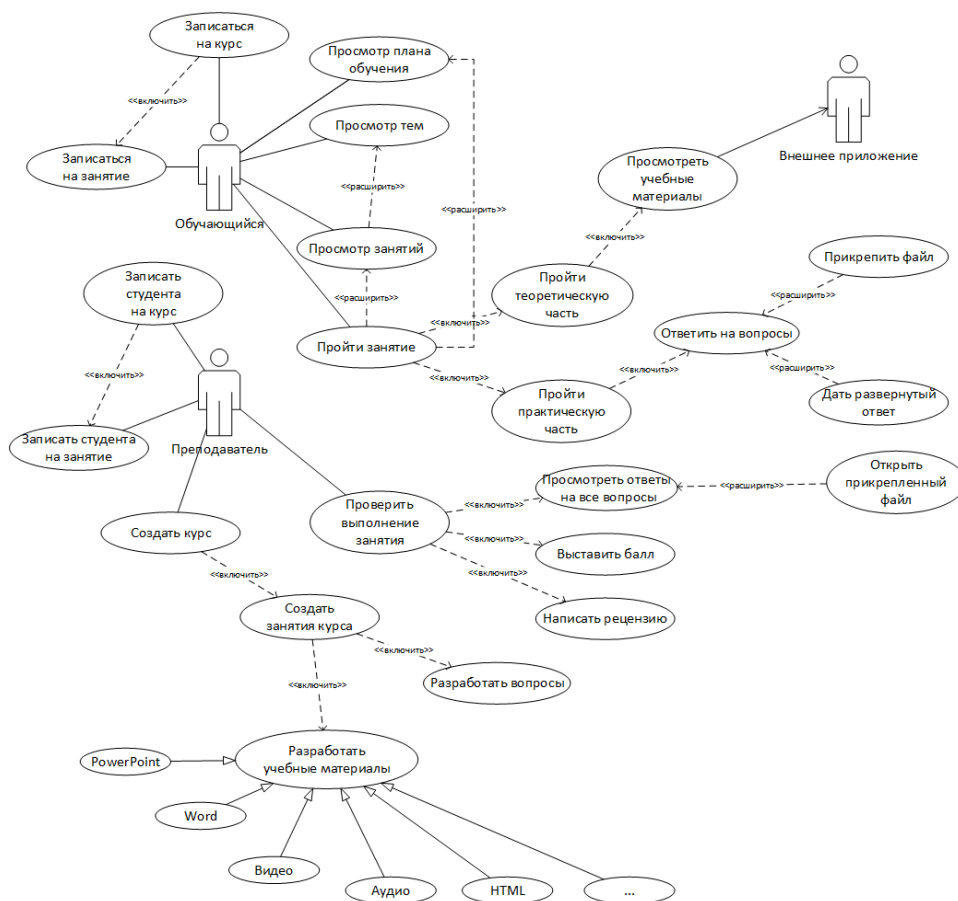


Рисунок 15 – Диаграмма вариантов использования

Разработанная диаграмма вариантов показывает следующие данные о системе.

В системе присутствуют следующие актеры:

Преподаватель – записывает студента на курс, создает курсы и занятия, проверяет занятия и курсы.

Обучающийся – записывается на курсы и просматривает занятия, проходит тесты.

Внешнее приложение – с его помощью просматриваются учебные материалы в системе.

2.1.1 Разработка модели прикладных объектов конфигурации в виде диаграммы классов

Взаимодействие объектов разрабатываемой системы представлено на диаграмме классов на рисунке 16.

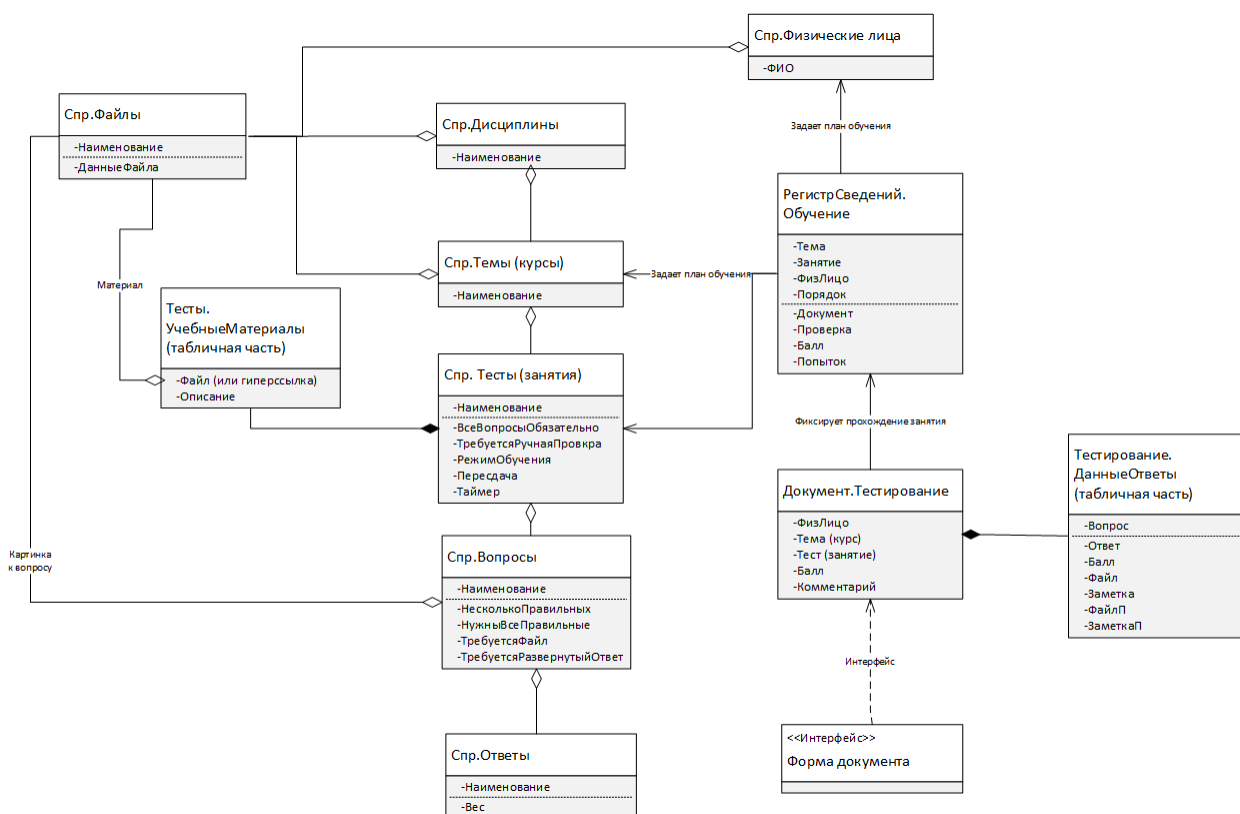


Рисунок 16 – Диаграмма классов

Общий процесс прохождения задания представлен на диаграмме активности на рисунке 17.

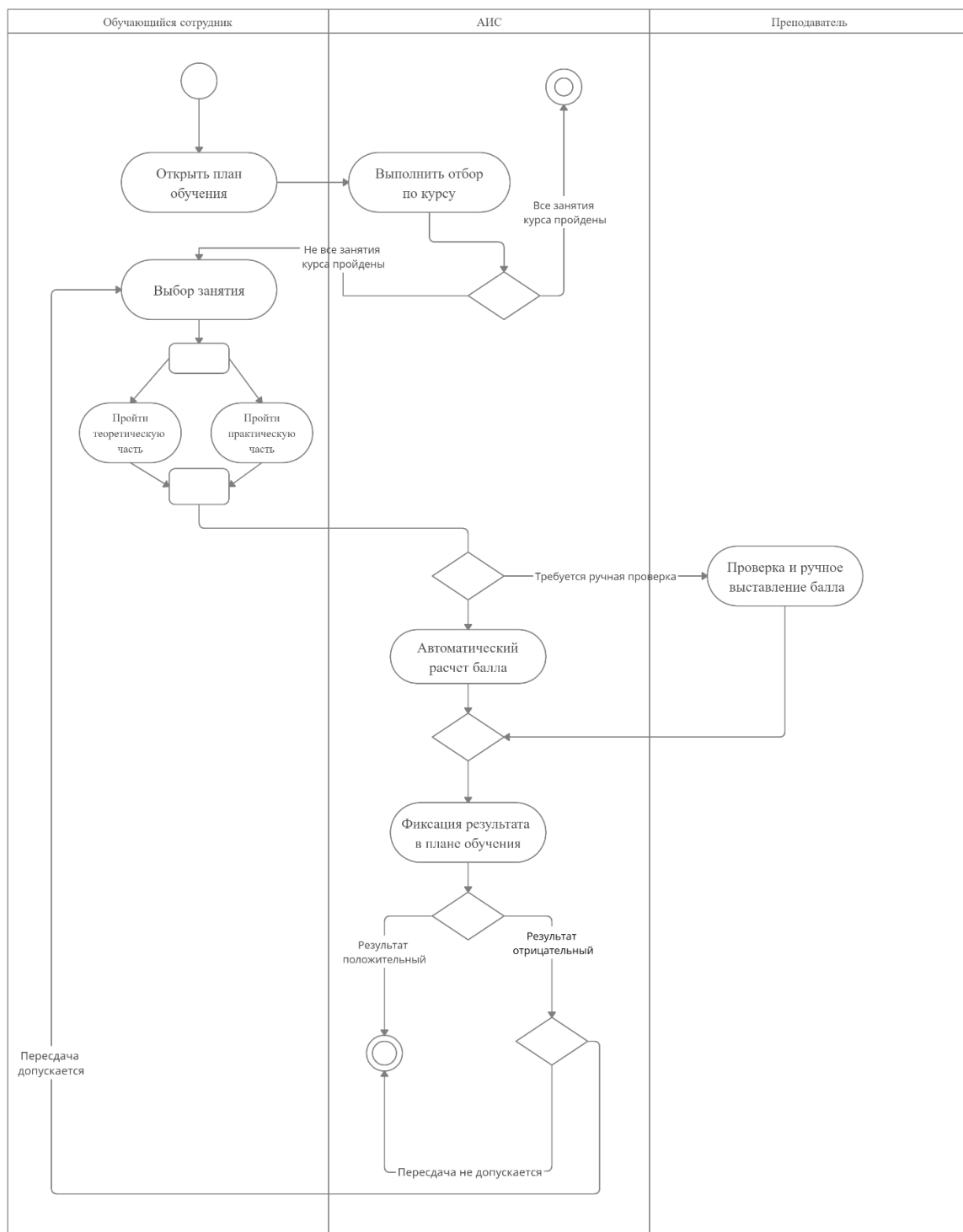


Рисунок 17 – Диаграмма активности

Разработанные диаграммы стали основой для построения структуры базы данных и модулей разрабатываемого приложения.

2.2 Информационное обеспечение системы сетевого обучения АО ИВК

2.2.1 Используемые классификаторы и системы кодирования

Для эффективного поиска, обработки и передачи информации необходимо выполнить классификацию и осуществить кодирование элементов информации [14].

Ажеронок В. А. дает следующее определение классификации: «Под классификацией подразумевается логическая операция, заключающаяся в распределении элементов рассматриваемого множества предметов и явлений по подмножествам (классам) на основании некоторого общего признака или группы признаков [15]. Упорядоченное расположение классифицируемых элементов на основе установленных связей и зависимостей между их признаками составляет систему классификации. Эта система определяет комплекс классификаторов. Классификатор представляет собой документ, с помощью которого осуществляется формализованное описание экономической информации в ССО [31]. Он содержит наименования объектов, наименования классификационных группировок и их кодовые обозначения.» [2]

Для решения поставленной задачи используется порядковая система кодирования.

Разработка системы на базе платформы «1С:Предприятие 8.3» вносит свои особенности в формирование кодов объектов. Так во всех классификаторах может быть включена возможность автоматического формирования кода классификатора. Также может быть установлено условие обязательного контроля уникальности кода, если код доступен для редактирования пользователем [34]. Для справочников автоматическое

формирование и контроль уникальности производятся в пределах серий кодов, которые могут формироваться:

- во всем справочнике;
- в пределах подчинения;
- в пределах подчинения владельцу.

Тип кода классификатора может быть числовым или строковым. В последнем случае код может нести в себе более подробную информацию об объекте кодирования.

Задание параметров формирования кода справочников приведено на примере справочника «Физические лица» на рисунке 18.

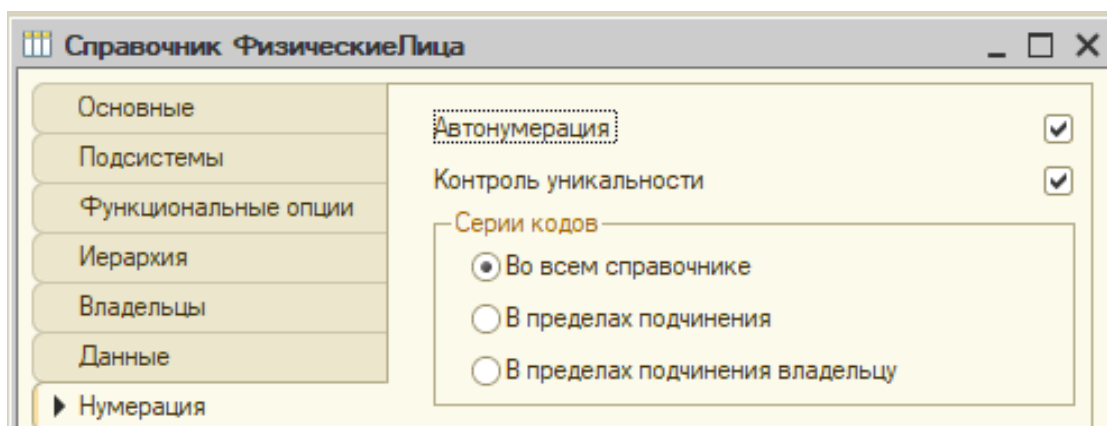


Рисунок 18 – Задание параметров формирования кодов справочника «Физические лица»

Для документов код является основным представлением объекта в интерфейсе системы. Для справочников код хоть и является идентификатором объекта, но, как правило, не является основным представлением объекта. В качестве основного представления справочников обычно выступает наименование.

Для документов предусмотрен дополнительный параметр кода – периодичность, который позволяет формировать уникальный код документа только в пределах одного периода:

- года,

- квартала,
- месяца,
- дня.

По истечении периода коды документов могут вновь повторяться, но это не считается нарушением уникальности.

Для документов платформа «1С:Предприятие 8.3» предусматривает возможность использования нумераторов для автоматического формирования кодов документов [17]. Нумератор обладает теми же параметрами: длина кода, тип, периодичность и контроль уникальности. Один и тот же нумератор может быть назначен нескольким документам, в этом случае эти документы будут иметь сквозную нумерацию.

Задание параметров формирования кода документов приведено на примере документа «Тестирование» на рисунке 19.

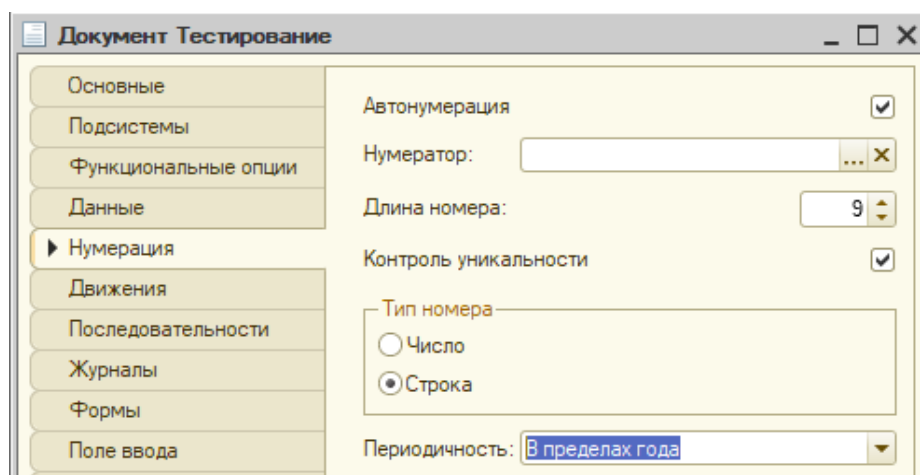


Рисунок 19 – Задание параметров формирования кода документа «Тестирование»

В таблице 1 приведены сводные данные по используемым в разрабатываемой системе классификаторам.

Таблица 1 – Классификаторы системы

Кодируемое множество объектов	Тип кода	Длина кода	Автономность	Контроль уникальности	Серии кодов	Вид классификатора
Справочники любого вида	Строка	9	Да	Да	Во всем справочнике	Локальный
ДокументТестирование	Строка	9	Да	Да	В пределах года	Локальный

Разработанная система классификации позволит правильно и безошибочно построить нумерацию объектов разрабатываемой конфигурации.

2.2.2 Характеристика нормативно-справочной и входной оперативной информации

Входными данными разрабатываемой системы являются, во-первых, данные физических лиц – участников процесса обучения, во-вторых, – данные по учебным материалам, вносимым преподавателями в систему: теоретическими и практическим материалам по занятиям.

Первая группа входных данных представляет из себя учетные данные физических лиц – участников процесса обучения: обучающихся и преподавателей. Как правило, эти данные представлены унифицированными документами – паспорт или иной документ, удостоверяющий личность, документы об образовании и так далее. Для целей разрабатываемой системы из всего многообразия этих данных важны лишь некоторые: Фамилия, дата рождения, пол, учебная группа (для обучающихся), контактные данные. Различные дополнительные данные об обучающемся могут быть прикреплены к записи базы данных в виде файлов любого формата, что неограниченно расширяет возможности хранения данных об обучающихся.

Вторая группа входных данных – учебные материалы – очень разнообразна. Это могут быть электронные файлы любого формата,

содержащие любые данные [30]. Файлы прикрепляются к учебным занятиям, созданным в системе, и могут быть открыты для просмотра в процессе изучения теоретического материала занятия тем приложением, которое связано с этим типом файла.

Возможность прикрепления файла любого типа неограниченно расширяет информационные возможности разрабатываемой системы.

2.2.3 Характеристика выходной информации

Выходными данными системы являются пройденные задания из практической части занятий, составляющих тот или иной курс обучения. В общем случае выполненное задание имеет только электронное представление в базе данных разрабатываемой системы в виде документа «Тестирование». Однако если выполняемое задание требует прикрепления каких-либо файлов, то обучающемуся необходимо это сделать в ходе выполнения задания. Прикрепляемый файл может быть любого формата, его форма определяется сутью задания.

Также выходными данными является заполненный план обучения по каждому обучающемуся.

В плане обучения задается необходимость прохождения обучающимся тех или иных занятий по темам изучаемого курса. Отметкой о прохождении занятия является ссылка на документ «Тестирование», которым «закрывается» ячейка плана обучения.

План обучения не имеет строго заданной печатной формы и ведется только в электронном виде в системе.

Экранные формы всех объектов разрабатываемой системы будут построены по технологии «Управляемые формы», предоставляемой разработчику платформой «1С:Предприятие 8.3».

Отличительной особенностью этой технологии является декларативное, а не «по пикселям» размещение элементов управления на форме и особенный, узнаваемый интерфейс.

2.3 Проектирование базы данных системы сетевого обучения

2.3.1 Выбор технологии проектирования базы данных системы сетевого обучения

С. Блетин и К. Паркер в своей статье приводят следующее определение: «Integration DEFinition for information modeling (IDEF1X) – это язык моделирования данных для разработки семантических моделей данных. IDEF1X используется для создания графической информационной модели, которая представляет структуру и семантику информации в среде или системе. IDEF1X позволяет строить семантические модели данных, которые могут служить для поддержки управления данными как ресурсом, интеграции информационных систем и создания компьютерных баз данных. Этот стандарт является частью семейства языков моделирования IDEF в области программной инженерии» [32].

Метод моделирования данных используется для моделирования данных стандартным, последовательным и предсказуемым образом с целью управления ими как ресурсом [18]. Она может быть использована в проектах, требующих стандартных средств определения и анализа ресурсов данных в организации. Такие проекты включают в себя включение методики моделирования данных в методологию, управление данными как ресурсом, интеграцию информационных систем или проектирование компьютерных баз данных. Основными целями стандарта IDEF1X являются:[32]

- средства для полного понимания и анализа ресурсов данных организации;
- общие средства представления и передачи информации о сложности данных;
- техника для представления общего представления данных, необходимых для работы предприятия;

- средства для определения независимого от приложений представления данных, которое может быть проверено пользователями и преобразовано в физический дизайн базы данных;
- метод получения интегрированного определения данных из существующих ресурсов данных.

Далее С. Блетин и К. Паркер пишут, что «основной целью IDEF1X является поддержка интеграции. Подход к интеграции фокусируется на получении, управлении и использовании единого семантического определения ресурса данных, называемого «концептуальной схемой». «Концептуальная схема» обеспечивает единое интегрированное определение данных на предприятии, которое не ориентировано на какое-либо одно применение данных и не зависит от способа физического хранения данных или доступа к ним. Основной целью концептуальной схемы является обеспечение последовательного определения значений и взаимосвязей между данными, которые могут быть использованы для интеграции, обмена и управления целостностью данных. Концептуальная схема должна обладать тремя важными характеристиками:

- согласованность с инфраструктурой предприятия и истинность во всех областях применения;
- расширяемость, то есть возможность определения новых данных без изменения ранее определенных данных;
- возможность преобразования как в требуемые пользовательские представления, так и в различные структуры хранения и доступа к данным.» [32]

IDEF1X является удобным и функциональным стандартом разработки, который позволяет выполнять моделирование для большинства современных баз данных, при этом большинство основанных на нем CASE средств поддерживают реверс-инжиниринг, и прямую генерацию кода.

2.3.2 Разработка концептуальной базы данных системы сетевого обучения

Проведем концептуальное моделирование базы данных информационной системы. Для этого словесно опишем основные сущности, входящие в базу данных.

Физические лица, или обучающиеся – проходят тестирование в системе.

Каждое физическое лицо имеет определенный пол, относится к определенной группе, к определенному типу, может содержать контактные данные.

Каждый тест в системе содержит вопросы, по нему проводится тестирование.

Физическое лицо дает ответы на тест, к ответам могут прикрепляться файлы. Каждый файл может относиться к определенной категории.

По всем ответам сохраняются данные о конкретном ответе.

Все данные по тестированию сохраняются в реестре сведений «Обучение».

Таким образом, сущности, которые должны быть представлены в системе – следующие:

- Физические лица,
- Пол,
- Учебные Группы,
- Физические Лица.Контакты,
- Типы ФизЛиц,
- Тесты,
- Тестирование,
- Тестирование.Ответы,
- Тестирование.ДанныеОтветы,
- Вопросы,
- Хранимые Файлы,

- КатегорииФайлов,
- Темы.

Концептуальная модель базы данных приведена на рисунке 20.

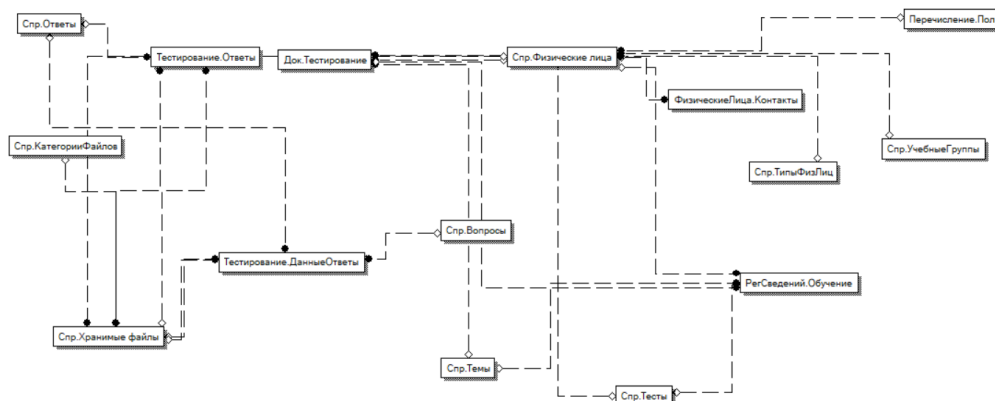


Рисунок 20 – Концептуальная модель данных

В результате проектирования была разработана информационная модель системы, которая представляет собой модель организации работы системы и схематично поясняет, на основании каких входных документов, а также нормативно-справочной информации происходит функционирование системы и получение выходных данных, то есть процесс преобразования данных в информационной системе. Схема представлена на рисунке 21.

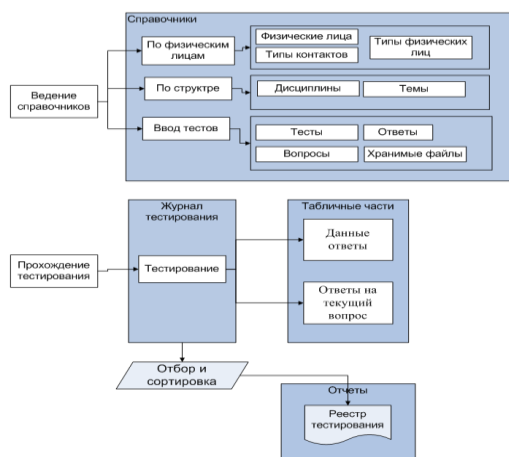


Рисунок 21 – Информационная модель системы

После проектирования концептуальной модели данных можно поступить к проектированию логической модели.

2.3.3 Разработка логической модели данных системы сетевого обучения

Логическая модель разрабатывается в нотации IDEF1X, описанной ранее. Для разработки логической модели использовалось средство CASE моделирования ErWin Data Modeler. Логическая модель разработанной базы данных приведена на рисунке 22.

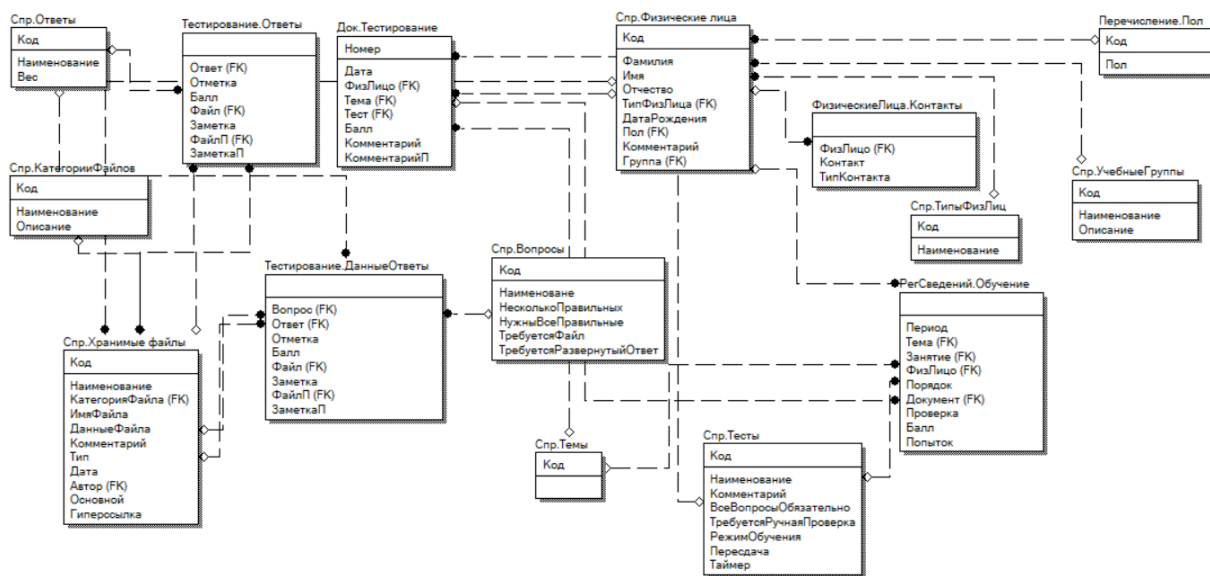


Рисунок 22 – Логическая модель данных

На разработанной модели данных приведена схема разрабатываемой базы данных.

Согласно разработанной модели – система содержит несколько справочников, несколько основных документов, а также регистр сведений «Обучение», в который заносятся данные о пройденных пользователями тестах.

Далее данная логическая модель была преобразована в физическую модель данных, которая приведена в третьей главе.

2.4 Требования к аппаратно-программному обеспечению системы сетевого обучения

Далее была описана конфигурация аппаратно-программного обеспечения, которое предоставляет стабильную работу проектируемой системы сетевого обучения. Для успешной оптимизации бизнес-процесса поддержки сетевого обучения в компании, необходимо проводить разработку ССО на базе программного продукта «1С:Предприятие» с версией не ниже 8.3.

Возможны два способа сетевого взаимодействия с системой: через проведенную в учебном центре компании локальную сеть или посредством виртуального веб-сервера. Для первого варианта система должна функционировать на ПК на базе архитектуры не ниже x64, x86, которые связаны в локальной сети, для второго необходим ПК с доступом к веб-серверу и веб-браузером.

Требования к программно-аппаратным средствам для системы стандартные для системы Windows 10, на которой функционирует система 1С предприятие, а также браузер.

Это следующие требования:

- процессор с архитектурой x86-64 (Intel с поддержкой EM64T, AMD с поддержкой AMD64);
- оперативная память 2048 Мб и выше;
- жесткий диск 40Гб и выше;
- устройство чтения компакт-дисков;
- USB-порт;
- SVGA-видеокарта.

Для работы по локальной сети на компьютере пользователя должна быть установлена система 1С: Предприятие, с подключенной сетевой или локальной версией разработанной базы данных [20].

Работа через интернет может проводиться с использованием любого современного браузера - Google Chrome, MS Edge, Opera и так далее.

Сервер в локальной сети или с выходом в интернет, на котором размещается база данных информационной системы – должен обладать достаточной конфигурацией для установки серверной версии 1С: Предприятие [22]. Это следующие требования:

- процессор с архитектурой x86-64 (Intel с поддержкой EM64T, AMD с поддержкой AMD64);
- оперативная память 2048 Мб и выше;
- жесткий диск 40Гб и выше;
- устройство чтения компакт-дисков;
- USB-порт;
- SVGA-видеокарта.

Описанная конфигурация аппаратно-программного обеспечения, предоставляет стабильную работу проектируемой системы сетевого обучения.

Выводы по второй главе

Во второй главе были созданы концептуальная и логическая модели проектируемой системы. Таким образом была описана специфика деятельности учебного центра в АО ИВК.

Разработаны UML-диаграммы: диаграмма вариантов использования, диаграмма классов и диаграмма активности. Приведено описание входных и выходных данных. Разработанная логическая модель системы и модель данных стали основой для создания конфигурации.

Глава 3 Физическое проектирование АИС

3.1 Выбор технологии и платформы разработки программного обеспечения АИС

Согласно книге Б. Бабэ: «Одним из известнейших языков программирования (ЯП), применяющихся для разработки приложений является C++. В нем объекты реализуются «классами». «Класс» определяет, каким образом ведут себя объекты этого типа, как они создаются и удаляются. При написании приложения с помощью классов текст программы становится понятным и структурированным. Встроенный аппарат объектно-ориентированного программирования (ООП) дает возможность описывать различные предметные области и создавать настраиваемые для нужд пользователей программные пакеты системного и прикладного применения» [6].

В статье Т. Китченса написано следующее: «Наследником языка C++ является язык программирования C#. C# – это язык программирования, который спроектирован для создания приложений, работающих в среде «.NET Framework». Язык C# прост, типобезопасен и объектно-ориентирован. Благодаря множеству новшеств C# дает возможность быстрой разработки приложения» [31].

Согласно статье О. Дурковича и Л. Раковича: «Самой популярной средой разработки на языке C# является Microsoft Visual Studio. Visual C# – это реализация языка C# корпорацией «Майкрософт». Поддержка Visual C# в Visual Studio выполняется при помощи редактора кода, компилятора, шаблонов проектов, конструкторов, мастеров кода, отладчика и так далее. Библиотека классов «.NET Framework» дает доступ к большинству служб ОС, а также к другим полезным классам, что существенно ускоряет время разработки» [33].

В учебнике Баженовой И. Ю. приводится следующая информация: «Среду разработки Microsoft Visual Studio можно использовать и как платформу для разработки приложений на языке C++. Однако разработчики предпочитают специализированные среды разработки, например, RAD C++ Builder» [7].

Ашарина И. В. приводит следующее описание: «Дальнейшее развитие идеи кросс-платформенного программирования привело к появлению языка Java. Приложения Java транслируются в специальный байт-код, который затем выполняется на любой компьютерной архитектуре с помощью специальной виртуальной Java-машины (JVM). Это основная часть системы Java, так называемой Java Runtime Environment (JRE). JVM может также использоваться для выполнения программ, написанных на других языках программирования.

Java является языком с Си-подобным синтаксисом и близок в этом отношении к C/C++ и C#.

Так как JVM доступны для почти всех аппаратных и программных платформ, Java может рассматриваться и как связующее программное обеспечение, и как самостоятельная платформа. Использование одного и того же байт-кода для многих платформ позволяет охарактеризовать Java как «скомпилировано однажды, запускается везде» – мечта кросс-платформенного программирования.

Достоинством использования технологии JRE является полная независимость кода от операционной системы и оборудования, что позволяет выполнять Java-приложения на любом устройстве, для которого разработана соответствующая виртуальная машина. Другой важнейшей особенностью языка Java, выгодно отличающей его от языка C/C++, является гибкая система безопасности, в рамках которой исполнение программы полностью контролируется JVM. Любые операции, которые превышают установленные полномочия программы (например, попытка несанкционированного доступа к данным или соединения с другим компьютером), вызывают немедленное

прерывание. Наиболее популярной платформой для написания программ на Java является Eclipse» [5].

Написание приложений на рассмотренных языках и платформах программирования является так называемым «классическим» программированием [23]. Однако на сегодняшний день становятся все более популярными другие методологии программирования, отличные от «классических».

Одна из популярных на сегодняшний день методологий разработки информационных систем – использование метаданных. Метаданные – это данные о другой информации, или данные, относящиеся к дополнительной информации о содержимом данных или объекте.

Существует много платформ разработки, основанных на метаданных. Самой популярной из них является платформа «1С:Предприятие».

В «1С:Предприятие» метаданные представляют собой совокупность проблемно-ориентированных объектов, поддерживаемых на уровне самой технологической платформы. Состав объектов, которые поддерживаются технологической платформой, является результатом длительного анализа предметных областей, в которых применяется платформа, и выделения используемых в этих областях бизнес-сущностей [24]. Задача разработчика заключается в том, чтобы собрать из этих объектов необходимую структуру прикладного решения и затем описать специфические алгоритмы функционирования и взаимодействия этих объектов, отличающиеся от их типового поведения.

Основным отличием системы «1С: Предприятие 8.3» как и других ERP-систем является предоставление системой способа управления данными при помощи метаданных и специального языка работы с данными. Метаданные описывают объекты предметной области, а модули на специальном языке реализуют необходимую функциональность этих объектов. Совокупность представлений объектов и программных модулей образует прикладные решения – конфигурации [21].

Опишем основные преимущества использования платформы «1С:Предприятие 8.3».

Предметная ориентированность – программирование ведется в терминах предметной области. Программист сосредотачивается на реализации функционала системы, а не на внутренних проблемах языка и платформы разработки.

Быстрая разработка приложений – приложение строится из прикладных объектов как конструктор. В общем случае в приложении может вообще не быть программного кода, при этом оно будет выполнять возложенную на него задачу [26]. Если же возникает необходимость написания программного кода, то в распоряжении программиста есть большое количество конструкторов и шаблонов.

Большое количество видов прикладных объектов, при помощи которых моделируются практически любые объекты любых предметных областей.

Огромное количество сервисных функций, которые предоставляет сама платформа. Даже самое простое приложение при наличии такого количества дополнительных сервисных функций превращается в мощное решение.

Поэтому платформа «1С:Предприятие» является самым популярным инструментом разработки экономических приложений в России и странах СНГ.

Мокий М. С. отмечает, что «платформа «1С:Предприятие» подходит не для всех задач. Ее есть смысл применять для «учетных» задач, ориентированных на автоматизацию бизнес-процессов учета деятельности хозяйствующих субъектов. В то время как классические языки программирования позволяют решать любые задачи» [19]. Но задача автоматизации является именно учетной задачей и использование платформы «1С:Предприятие 8.3» в данном случае более чем оправдано.

3.2 Описание выбранной платформы разработки

Система разработана на платформе «1С:Предприятие 8.3». Данная платформа, выпущенная фирмой «1С», за последние десятилетия стала в России самой популярной платформой для быстрой разработки приложений для автоматизации учета деятельности практически в любой сфере [27]. Ее наиболее сильные качества проявляются при автоматизации учетных задач, а задача торгового учета является как раз такой задачей.

В книге Габец А. П. так написано о причинах популярности платформы: «Причин популярности у потенциальных пользователей платформы «1С:Предприятие 8.3» действительно много. Первая состоит в том, что фирмой-разработчиком на базе собственной платформы создано много готовых решений для самых разных отраслей. На рисунке 23 представлен перечень этих разработок. Любая организация, в какой бы сфере она ни работала, найдет подходящий продукт в этом перечне» [10].

«Кроме самой фирмы «1С» разработкой типовых продуктов на базе платформы «1С:Предприятие 8.3» занимаются партнеры-франчайзи фирмы «1С». Ими созданы сотни типовых продуктов. Любое предприятие может выбрать среди них подходящее под свою специфику» [10].

Вторым главнейшим фактором выбора решений от фирмы «1С» является беспрецедентная поддержка фирмой «1С» своих партнеров.

Платформа «1С:Предприятие» является платформой разработки прикладных приложений. Можно смело утверждать, что данная платформа стоит в одном ряду с такой популярной платформой разработки как MS Visual Studio [28].

Для разработчика программных приложений самой главной особенностью платформы «1С:Предприятие 8.3» является строгое разделение между разрабатываемым приложением и самой платформой.

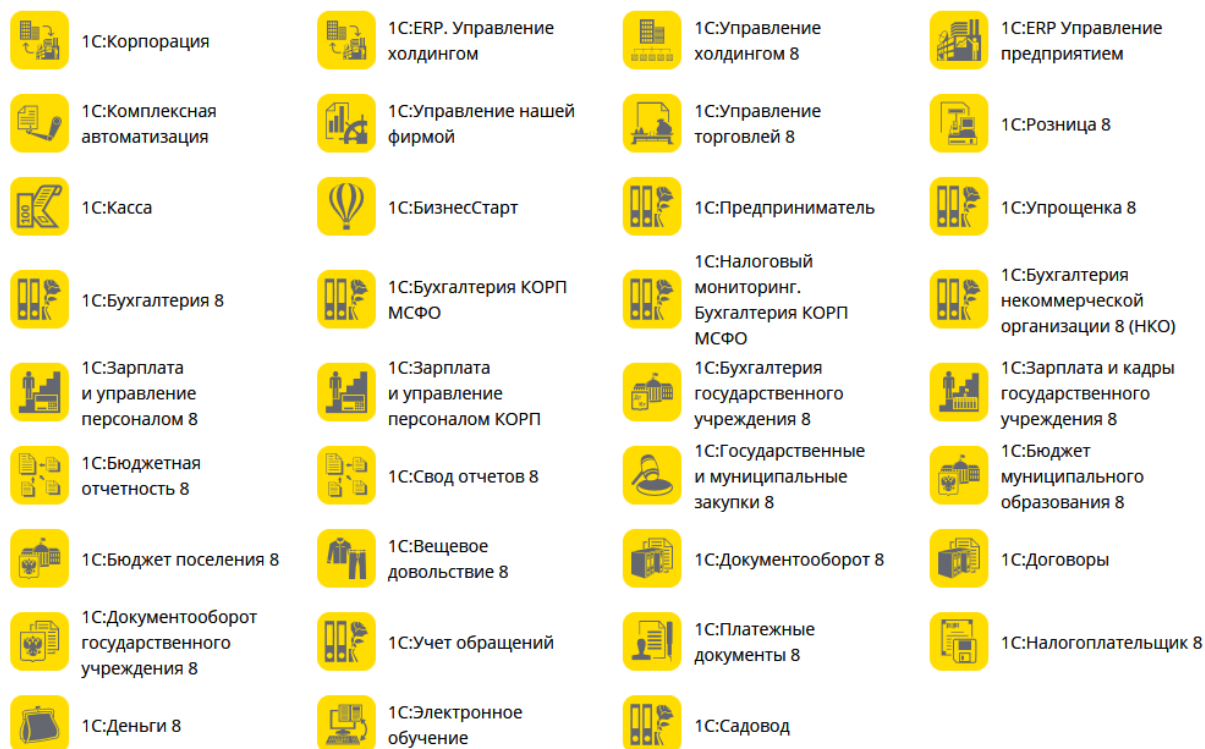


Рисунок 23 – Перечень типовых разработок «1С»

Платформа «1С:Предприятие 8.3» обладает большим количеством технологических возможностей, которые представлены на рисунке 24. Любой из пунктов на этом рисунке – это сложный, продуманный и детальный механизм, которому стоит посвящать отдельную тему.

Отдельно Габец А. П. отмечает, что «рамки платформы «1С:Предприятие 8.3» четко ограничены – это среда для разработки приложений автоматизации бизнес-задач. Но именно в этой сфере платформа проявляет себя лучше всего. Строгое определение заданного круга проблем позволяет точно выбирать методики и технологии программирования, что делает платформу лучшей в решении именно этих задач по сравнению с традиционными системами разработки. Главное преимущество платформы «1С:Предприятие 8.3» – простота и скорость разработки» [10].

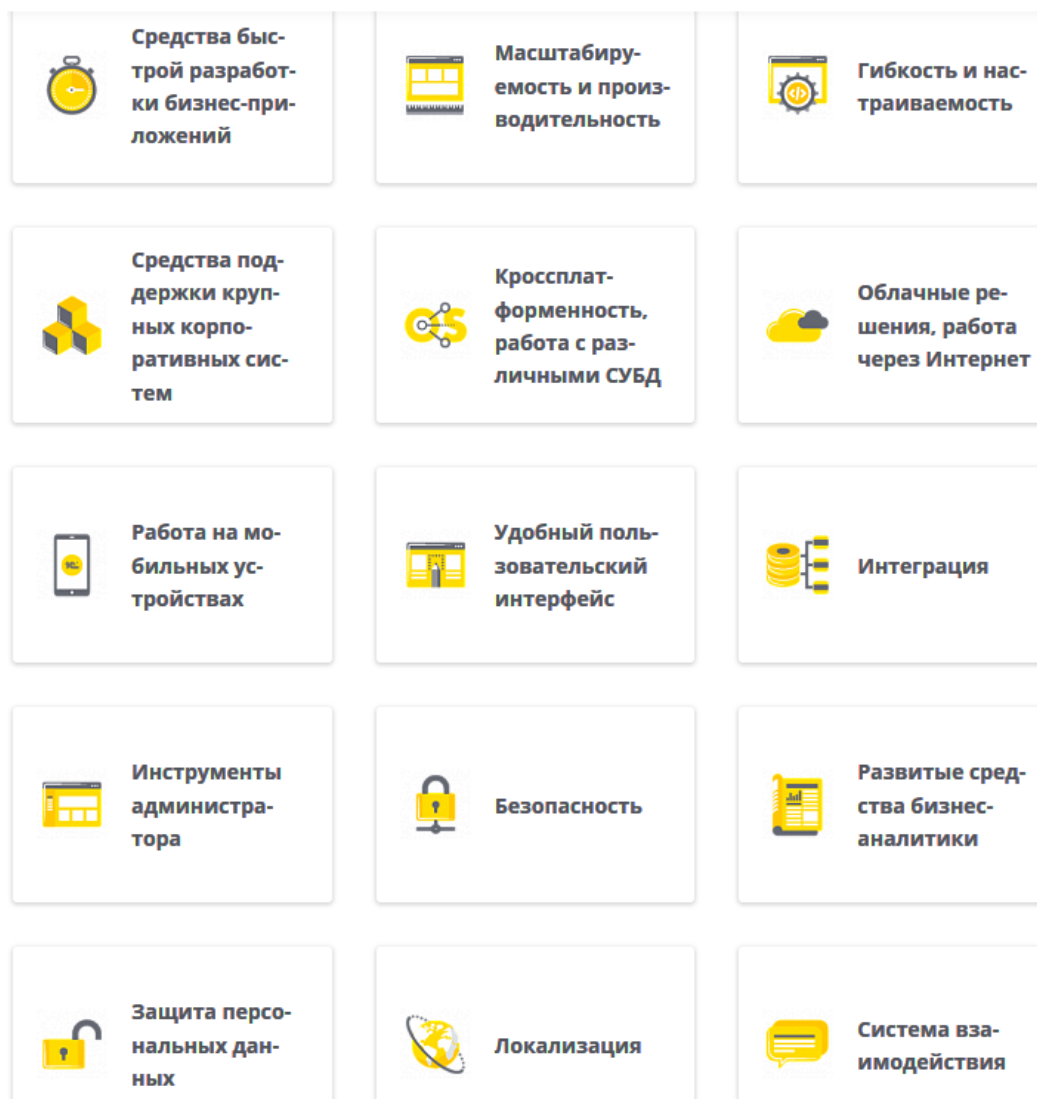


Рисунок 24 – Возможностей платформы «1С:Предприятие 8.3»

Таким образом все разработанные решения, и сама платформа превращается в мощную среду разработки, при помощи которой можно как дорабатывать типовые решения, так и создавать новые «с нуля».

3.3 Разработка физической модели данных АИС

Опишем структуру прикладных объектов конфигурации разрабатываемой системы в виде схем данных [13]. Из названия реквизитов объектов понятно назначение этих реквизитов. Типы данных реквизитов указаны в таблице для каждого объекта. Связи между объектами указаны на

рисунках стрелками. Для каждого объекта приводится взаимосвязь с другими объектами. Структура справочников приведена на рисунке 25.

На рисунке 26 представлена структура документа «Тестирование» и регистра сведений «Обучение».

Так как документ «Тестирование» является центральным объектом разрабатываемой системы, приведем подробно его структуру и структуру его табличных частей в таблицах 2-4.

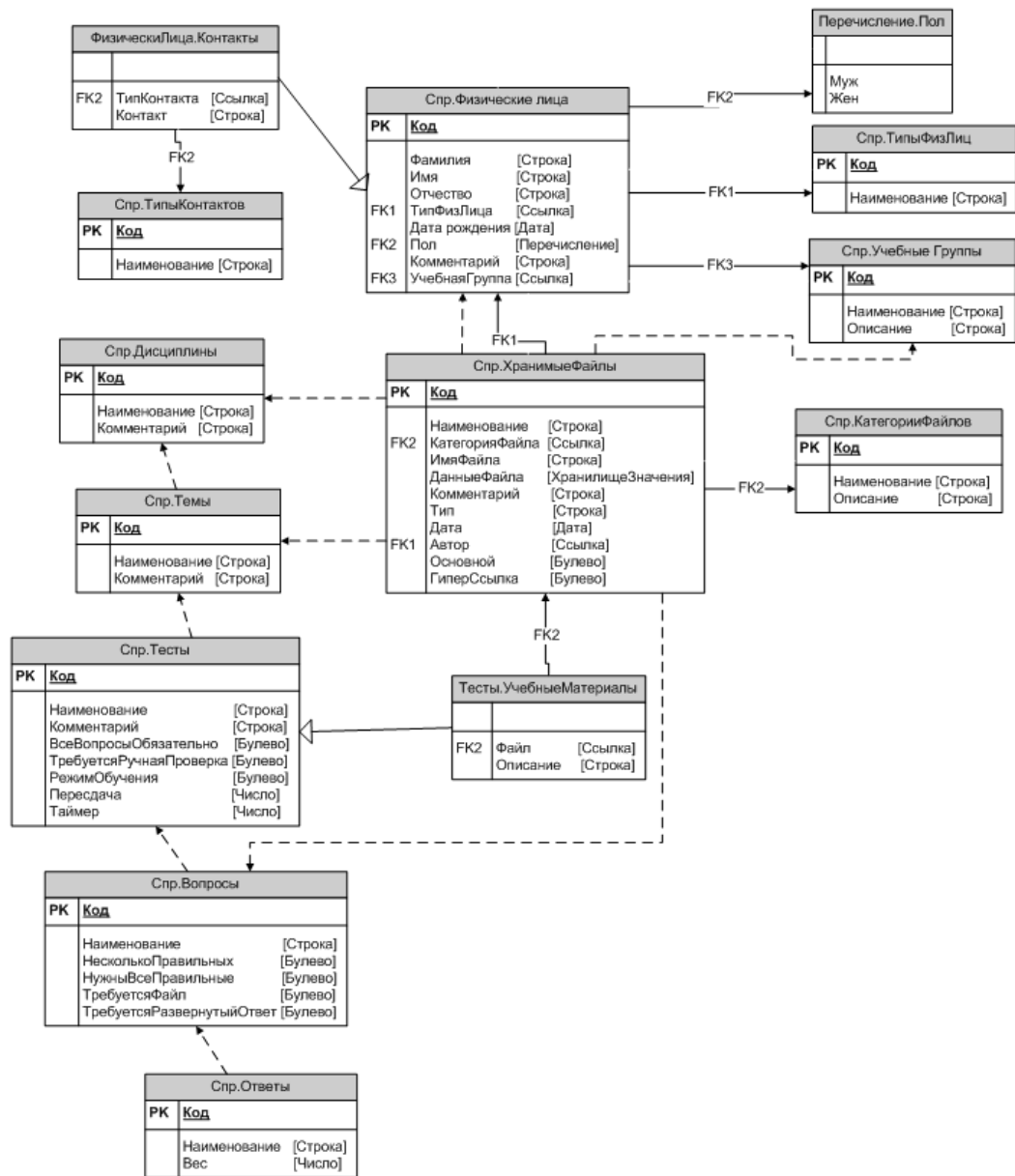


Рисунок 25 – Структура справочников

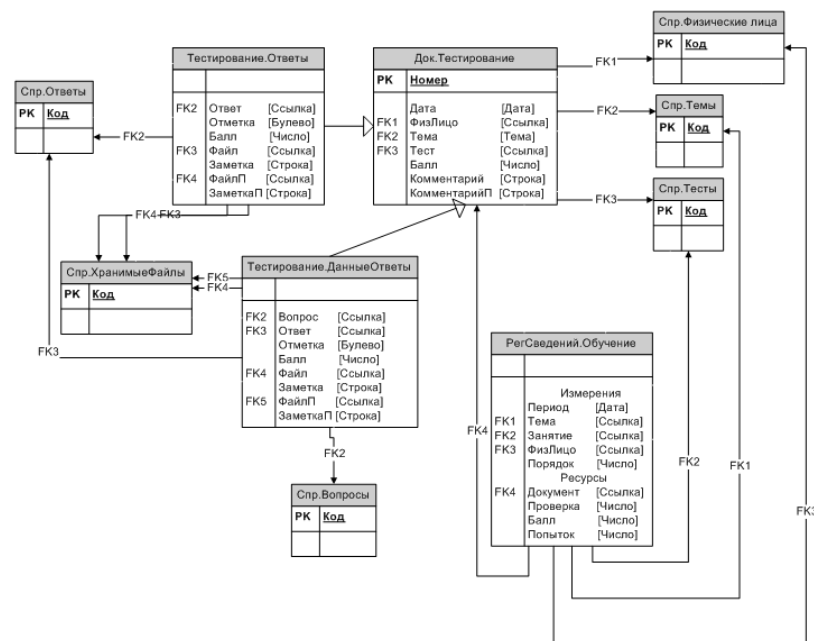


Рисунок 26 – Структура документа «Тестирование» и регистра сведений «Обучение»

Таблица 2 – Состав полей документа «Тестирование»

Наименование поля	Идентификатор поля	Тип поля
Физическое лицо – обучающийся, проходящий занятие	ФизЛицо	СправочникСсылка.ФизическиеЛица
Курс, к которому относится занятие	Тема	СправочникСсылка.Темы
Занятие	Тест	СправочникСсылка.Тесты
Рассчитанный или выставленный преподавателем балл	Балл	Число (3,2)
Комментарий обучающегося к выполнению занятия	Комментарий	Строка (0)
Комментарий преподавателя после проверки выполнения занятия	КомментарийП	Строка (0)

Табличная часть «Ответы» документа «Тестирование» предназначена для хранения вариантов ответов на текущий вопрос, который в данный момент решает обучающийся из списка вопросов практической части

занятия, ссылка на которое содержится в поле «Тест» шапки документа. Табличная часть «Данные Ответы» документа «Тестирование» предназначена для хранения данных пользователем ответов на вопросы занятия. В форме документа происходит визуальное «наложение» этих табличных частей для отображения данных пользователем ответов на вопросы при навигации между вопросами занятия.

Таблица 3 – Состав полей табличной части «Ответы» документа «Тестирование»

Наименование поля	Идентификатор поля	Тип поля
Вариант ответа на текущий вопрос	Ответ	СправочникСсылка.Ответы
Выбор варианта ответа	Отметка	Булево
Балл, рассчитанный за ответ	Балл	Число (3,2)
Прикрепленный к ответу файл с решенным заданием	Файл	СправочникСсылка.ХранимыеФайлы
Развернутый ответ на текущий вопрос	Заметка	Строка (0)
Файл, прикрепленный к вопросу преподавателем при проверке выполнения занятия	ФайлП	СправочникСсылка.ХранимыеФайлы
Развернутый ответ преподавателя на вопрос, при проверке выполнения занятия	ЗаметкаП	Строка (0)

Таблица 4 – Состав полей табличной части «Данные Ответы» документа «Тестирование»

Наименование поля	Идентификатор поля	Тип поля
Вопрос из списка вопросов практической части занятия	Вопрос	СправочникСсылка.Вопросы
Вариант ответа, выбранный пользователем	Ответ	СправочникСсылка.Ответы
Балл, рассчитанный за ответ	Балл	Число (3,2)
Прикрепленный к ответу файл с решенным заданием	Файл	СправочникСсылка.ХранимыеФайлы
Развернутый ответ на текущий вопрос	Заметка	Строка (0)

Продолжение таблицы 4

Наименование поля	Идентификатор поля	Тип поля
Файл, прикрепленный к вопросу преподавателем при проверке выполнения занятия	ФайлП	СправочникСсылка.ХранимыеФайлы
Развернутый ответ преподавателя на вопрос, при проверке выполнения занятия	ЗаметкаП	Строка (0)

План обучения, с отметками о прохождении обучающимися тех или иных занятий также относится к выходным данным системы. Отметка о прохождении занятия – это ссылка на документ «Тест», которым это занятие и «закрывается» в плане обучения. Структура регистра сведений «Обучение», которым реализуется план обучения, представлена в таблице 5.

Таблица 5 – Структура регистра сведений «Обучение»

Наименование поля	Идентификатор поля	Тип поля
Измерения		
Учебный курс	Тема	СправочникСсылка.Темы
Занятие, назначенное обучающемуся в план обучения	Занятие	СправочникСсылка.Тесты
Обучающийся, которому назначается план обучения	ФизЛицо	СправочникСсылка.ФизЛица
Порядковый номер занятия в учебном курсе	Порядок	Число (10)
Ресурсы		
Ссылка на последний документ, которым пройдено занятие	Документ	ДокументСсылка.Тест
Признак необходимости или проведения проверки	Проверка	Число (1) (Три состояния)
Полученный балл	Балл	Число (10,3)
Количество попыток прохождения занятий	Попыток	Число (2)

Описанная структура данных объектов разрабатываемой конфигурации позволит эффективно хранить данные о процессе обучения и даст возможность их дальнейшего анализа.

3.4 Разработка программного обеспечения

3.4.1 Схема взаимосвязи модулей приложения ССО и их описание

Общая функциональная структура разрабатываемой ССО представлена на рисунке 27.



Рисунок 27 – Функциональная структура системы

Подсистема ведения справочников предназначена для ведения данных списочного характера, которые используются на всех последующих этапах работы. Состав и назначение справочников разрабатываемой ССО представлены на рисунке 28.

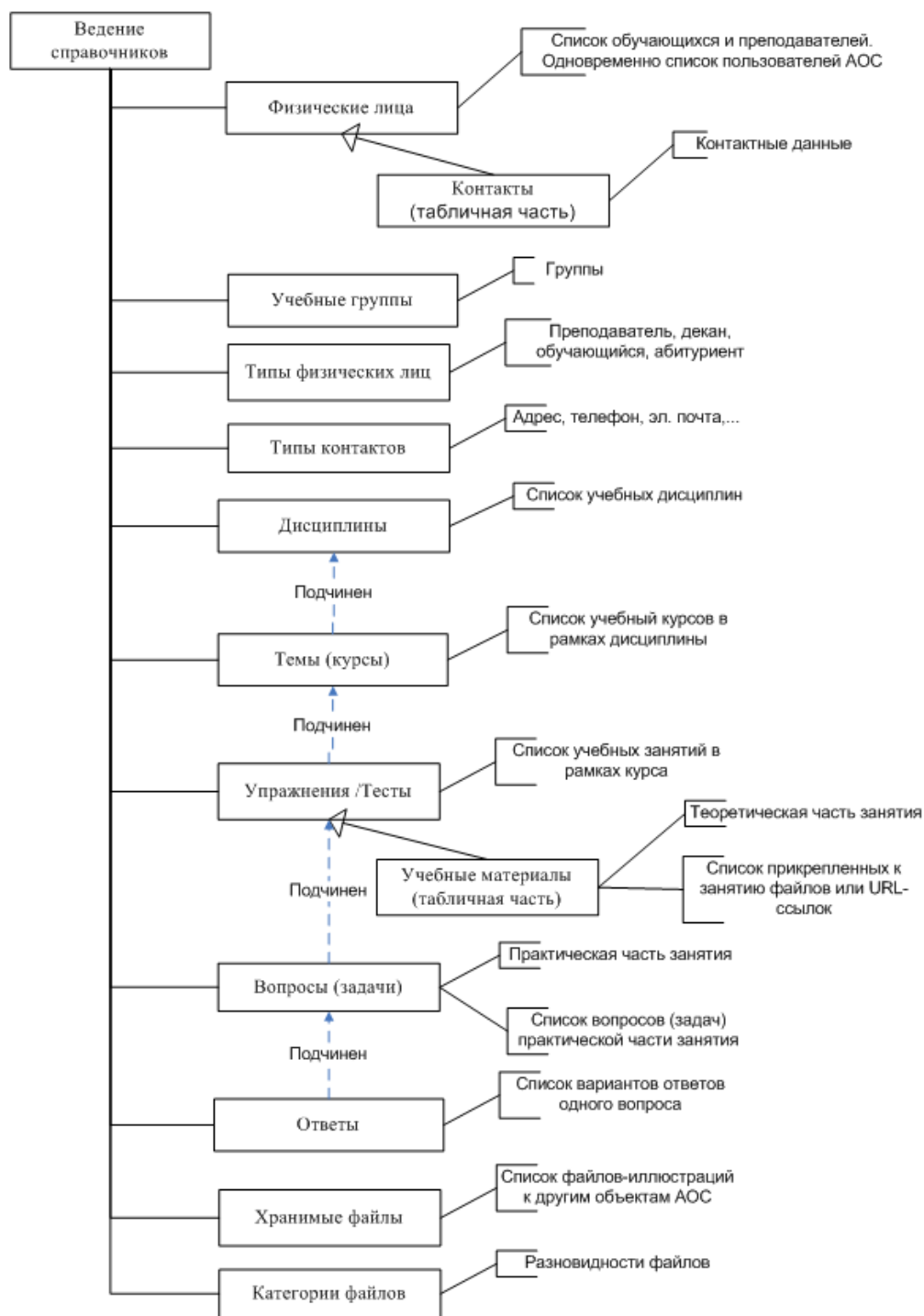


Рисунок 28 – Состав и назначение справочников

Подсистема редактирования упражнений (конструктор упражнений) физически представляет собой часть справочников разрабатываемой АИС, а организационно предназначена для ввода в систему структуры учебных курсов, упражнений и их наполнения.

Центральным элементом разрабатываемой АИС является упражнение. Упражнение – это структурная единица учебного курса, логически законченная его часть, подробно раскрывающая ту или иную тему.

Любое упражнение состоит из двух частей: теоретической и практической. Теоретическая часть представляет из себя набор учебно-методических материалов, разработанных преподавателем по теме упражнения. Практическая часть занятия представляет из себя список вопросов по теме упражнения, на которые обучающийся должен дать ответ. При помощи настроек преподаватель, разрабатывающий упражнение, может сконструировать его практическую часть как обыкновенный тест или как некое практическое задание. Структура вопроса (задачи) представлена на рисунке 29.



Рисунок 29 – Структура вопроса/задачи

Подсистема прохождения упражнений предназначена для предоставления обучающимся интерфейса для выбора упражнения, изучения теоретического материала, выполнения практического задания или прохождения теста, подсчета результатов выполнения задания или теста, сохранения факта процесса прохождения и результатов в базе данных.

Подсистема планирования обучения заключается в задании плана прохождения обучающимся (группы обучающихся) тех или иных упражнений различных учебных курсов. План составляется преподавателем путем назначения обучающихся на упражнение, что фиксируется в регистре сведений разрабатываемой АИС. В отдельных случаях обучающийся может сам записаться на упражнение или все упражнения курса. Интерфейсно план обучения представлен в виде списка упражнений, которые обучающийся должен пройти. Контроль выполнения плана обучения осуществляется в том же списке, в котором автоматически отмечается последний документ, фиксирующий выполнение практического задания/теста упражнения, количество попыток, набранный балл, факт ручной проверки выполненного задания преподавателем (если она требуется).

Ведение списка пользователей заключается в создании пользователей и отнесения их к одной из категорий: администраторы, преподаватели, учащиеся.

3.4.2 Разработка прикладных объектов

Состав справочников разрабатываемой системы, как он выглядит в модуле «Конфигуратор», приведен на рисунке 30.

Для каждого справочника необходимо задать структуру – то есть состав реквизитов справочника и состав его табличных частей, для каждой из которых необходимо создать свою структуру реквизитов. В реквизитах целесообразно хранить свойства объектов информации. На рисунке 31 представлены структура реквизитов и табличных частей справочника «ФизическиеЛица».

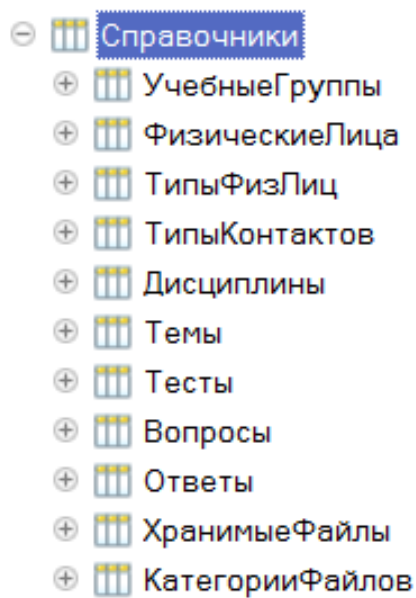


Рисунок 30 – Список справочников системы

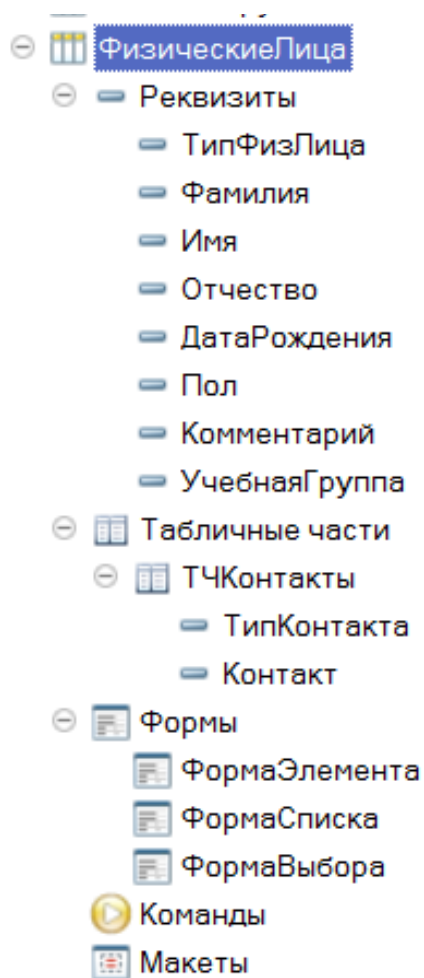


Рисунок 31 – Структура справочника «Физические Лица»

Большое значение имеет подчиненность справочников «Дисциплины», «Темы», «Упражнения (тесты)», «Вопросы» и «Ответы» друг другу. Такая подчиненность задает, во-первых, однозначное соответствие данных, а во-вторых, позволяет организовать интерфейс просмотра данных в соответствии с иерархией подчиненности, приведенной на рисунке 32.



Рисунок 32 – Структура подчиненности данных

Текст программных модулей справочников приведен в приложении А.

Центральным объектом разрабатываемой системы является регистр сведений «Обучение». Он хранит план обучения по каждому обучающемуся в разрезе упражнений (тестов), то есть хранит список упражнений, который обучающийся должен пройти. В этом же регистре хранятся и результаты

прохождения занятий. Структура регистра сведений «Обучение» представлена на рисунке 33.

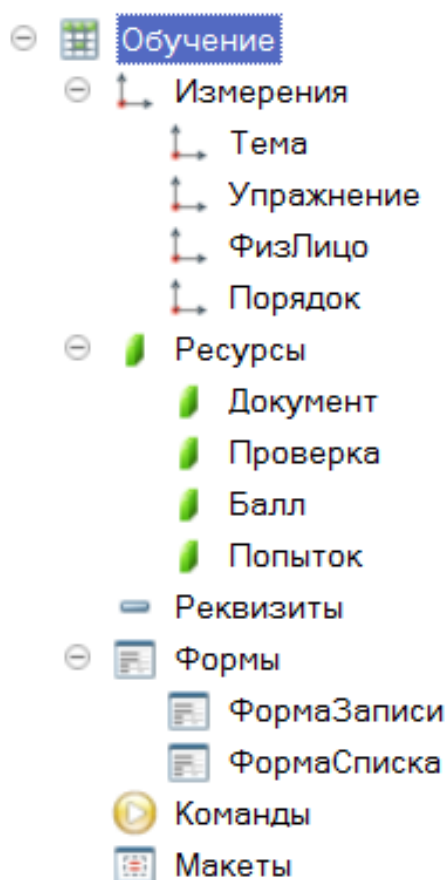


Рисунок 33 – Структура регистра сведений «Обучение»

Наличие записи в регистре «Обучение» говорит о том, что обучающийся «ФизЛицо» записан на прохождение упражнения «Упражнение», которое относится к теме «Тема», то есть в плане обучения обучающемуся назначена необходимость пройти это упражнение. Период (дата) записи представляет собой дед-лайн – крайнюю дату прохождения упражнения. Если у этой записи, кроме того, заполнены поля-ресурсы, то значит обучающийся данное занятие прошел, а результаты его прохождения представлены значениями полей-ресурсов. Назначение полей регистра сведений «Обучение» представлено на рисунке 34.

РегСведений.Обучение		
Измерения		
Период	[Дата]	Последняя дата прохождения упражнения)
Тема	[Ссылка]	Тема (курс), к котрому относится упражнение
Упражнение	[Ссылка]	Упражнение, котрое необходимо пройти
ФизЛицо	[Ссылка]	Обучающийся
Порядок	[Число]	Последовательность прохождения упражнений в теме (курсе)
Ресурсы		
Документ	[Ссылка]	Документ "Занятие", котрым зафиксировано прохождение практической части упражнения
Проверка	[Число]	Необходимость ручной проверки(0 - не нужна, 1- нужна, но не пройдена, 2, нужна и пройдена),
Балл	[Число]	Балл по результату прохождения Документа
Попыток	[Число]	Количество попыток (количество Документов)

Рисунок 34 – Назначение полей регистра «Обучение»

Заполнение одной записи регистра сведений «Обучение» происходит поэтапно. Сначала заполняются только измерения, тем самым выставляется план обучения, затем при проведении документа «Тестирование» заполняются ресурсы этой записи, тем самым фиксируются результаты выполнения практической части упражнения. При необходимости ручной проверки, преподаватель открывает документ «Занятие», корректирует рассчитанные автоматически баллы, прикрепляет свои файлы и дает рецензии к ответам, после чего препроводит документ – запись в регистре «Обучение» корректируется. На рисунке 35 представлен внешний вид формы списка регистра.

Тема	Упражнение	Учебная г.	Учащийся	Занятие	К.	Балл	Поп...
Массивы данных	Программирование. Массивы. Основн...	AAA-3	Милютин Андрей Борисович	Занятие_000000043 от 17.07.2020 16...		1	5
Массивы данных	Массивы. Тест№1	AAA-3	Милютин Андрей Борисович			-	-
Массивы данных	Программирование. Массивы. Практиче...	AAA-3	Милютин Андрей Борисович			-	-
Массивы данных	Программирование. Массивы. Практиче...	AAA-3	Милютин Андрей Борисович	Занятие_000000045 от 21.07.2020 16...	✓	0,63	2
Процедуры и ф...	Программирование. Вызов Функций	Гр А-4	Аксенов Сергей Васильевич			-	-
Процедуры и ф...	Программирование. Процедуры	Гр А-4	Аксенов Сергей Васильевич			-	-
Массивы данных	Программирование. Массивы. Основн...	Гр А-4	Аксенов Сергей Васильевич			-	-
Массивы данных	Массивы. Тест№1	Гр А-4	Аксенов Сергей Васильевич			-	-
Массивы данных	Программирование. Массивы. Практиче...	Гр А-4	Аксенов Сергей Васильевич			-	-
Массивы данных	Программирование. Массивы. Практиче...	Гр А-4	Аксенов Сергей Васильевич			-	-
Динамические ...	Программирование. Списки. Обход спи...	Гр А-4	Аксенов Сергей Васильевич			-	-

Рисунок 35 – Регистр сведений «Обучение» – план обучения и результат выполнения этого плана

Добавление записей в регистр сведений «Обучение» производится при помощи общей команды «В План», доступной из контекста форм всех

объектов, ссылки на которые являются измерениями регистра. Например, на рисунке 36 представлен процесс добавления в командную панель формы списка справочника «Темы».

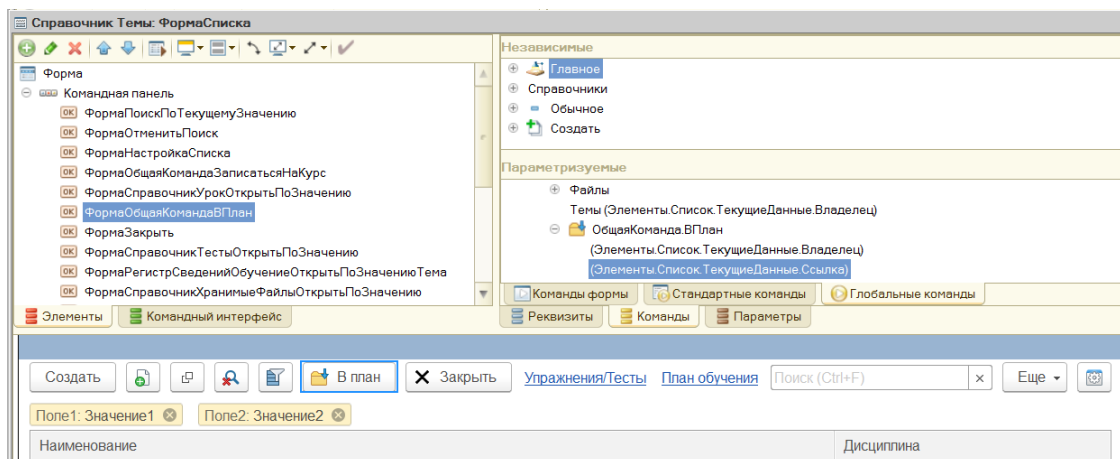


Рисунок 36 – Добавление общей команды «В План» в форму списка справочника «Темы»

Аналогично команда «В план» присутствует практически во всех формах объектов, имеющих отношение к регистру «Обучение». Например, эта команда присутствует в формах справочника «Учебные группы». В этом случае выполнение этой команды предполагает добавление в регистр сведений «Обучение» записей по всем учащимся выбранной учебной группы.

Сам текст общей команды «В план» достаточно простой и заключается в вызове общей формы «В план», это показано на рисунке 37.

```

&НаКлиенте
[ Процедура ОбработкаКоманды (ПараметрКоманды, ПараметрыВыполненияКоманды)
    Если МожетЗаписыватьДругих () Тогда // записывает других
        ПараметрыФормы = Новый Структура ("ЧтоЗаписать", ПараметрКоманды );
        ОткрытьФорму ("ОбщаяФорма.ВПлан", ПараметрыФормы, ПараметрыВыполненияКоманды.Источник,
            Иначе // записывается сам
                ОбщийСрв.НазначитьВПлан (ПараметрКоманды, ОбщийСрв.ПолучитьГлПользователя ());
                Оповестить ("Планобучения"); // переоткрыть форму регистра сведений, если произошли из
            КонецЕсли;
    КонецПроцедуры

&НаСервере
[ функция МожетЗаписыватьДругих ()
    Возврат РольДоступна ("Преподаватель") Или РольДоступна ("Администратор");
    Конецфункции

```

Рисунок 37 – Текст команды «В план»

Внешний вид общей формы «В план» представлен на рисунке 38.

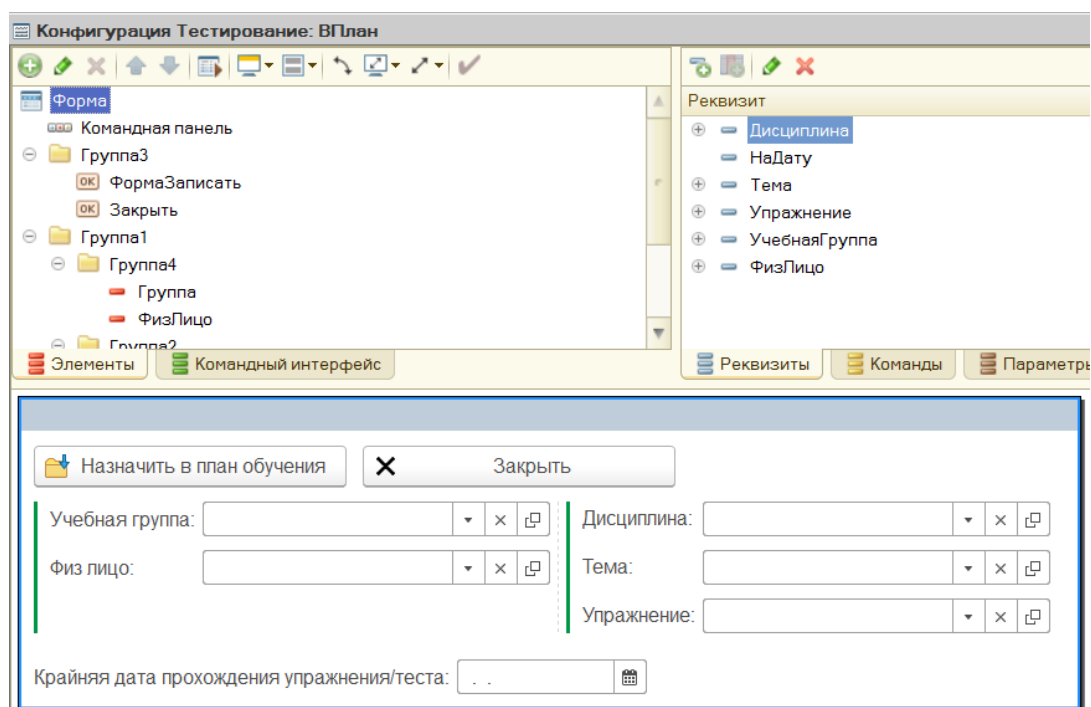


Рисунок 38 – Общая форма «В план»

Форма предоставляет пользователю возможность выбора обучающегося, запись для которого формируется в регистре сведений «Обучение» и упражнения, которое записывается в эту запись. Форма предполагает «групповое» добавление записей, например, для всех учащихся определенной группы, на все упражнения определенной темы. При этом тот объект, из контекста которого вызвана данная форма, будет заполнен автоматически. Форма используется преподавателями для записи обучающихся на определенные курсы (темы) или самими обучающимися для самостоятельной записи на упражнения или темы. Функциональность данных команд реализована в общих программных модулях, приведенных в приложении Б.

Фиксация прохождения практической части упражнения выполняется документом «Тестирование» (синоним «Занятие»). Документ «Тестирование» – это формальное представление реального выполнения

практического задания или теста, составляющих практическую часть упражнения. Документ организует процесс прохождения упражнения (теста) и выступает хранилищем результатов. Открыв ранее проведенный документ «Тестирование», можно просмотреть данные обучающимся ответы на вопросы задания/теста, прикрепленные им файлы и данные развернутые ответы, для этого предназначена табличная часть «ДанныеОтветы». Варианты ответов на текущий вопрос представлены в табличной части «Ответы», которая перезаполняется каждый раз при смене текущего вопроса. Структура документа «Тестирование» приведена на рисунке 39.

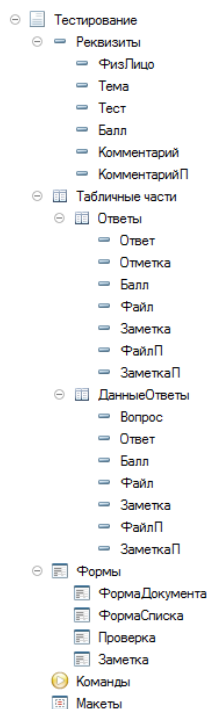


Рисунок 39 – Структура документа «Тестирование»

Практическая часть упражнения может представлять собой простой тест, а может являться практическим заданием. В последнем случае вопросы являются задачами (составными частями) задания, а ответы на вопросы требуют прикрепления файлов с результатами выполнения задач и/или дачи развернутых письменных ответов. Пример практической части упражнения,

представляющего из себя классический тест, приведен на рисунке 40. А на рисунке 41 приведено практическое задание.

← → ☆ Занятие 000000041 от 18.03.2021 0:00:00

Провести и закрыть Записать Провести ✕ Закрыть

Номер: 000000041 Дата: 18.03.2021 0:00:00 Балл: 0,14

Физ лицо: Милютин Андрей Борисович

Тема: Массивы данных

Занятие: Программирование. Массивы. Основные понятия

Вопрос 1/7

Сколько элементов в массиве A[10]

← Предыдущий Следующий →

<input type="checkbox"/>	0.0	9
<input checked="" type="checkbox"/>	1	10
<input type="checkbox"/>	0.0	11
<input type="checkbox"/>	0.0	Все ответы неверные

Рисунок 40 – Документ «Тестирование» в варианте теста

← → ☆ Занятие 000000046 от 30.03.2021 0:00:00

Провести и закрыть Записать Провести ✕ Закрыть

Номер: 000000046 Дата: 30.03.2021 0:00:00 Балл: 1,00

Физ лицо: Милютин Андрей Борисович

Тема: Массивы данных

Занятие: Программирование. Массивы. Практическое занятие 1

Вопрос 1/4

Написать модуль обхода массива на языке C#

← Предыдущий Следующий → Проверка

<input checked="" type="checkbox"/>	Прикрепите сюда файл с текстом модуля	Выполненное задание
<input checked="" type="checkbox"/>	Прикрепите сюда скриншот с результатами работы модуля	Выполненное задание

Рисунок 41 – Документ «Тестирование» в варианте практического задания

Сам алгоритм прохождения задания реализуется в форме документа «Тестирование» и представлен на рисунке 42.

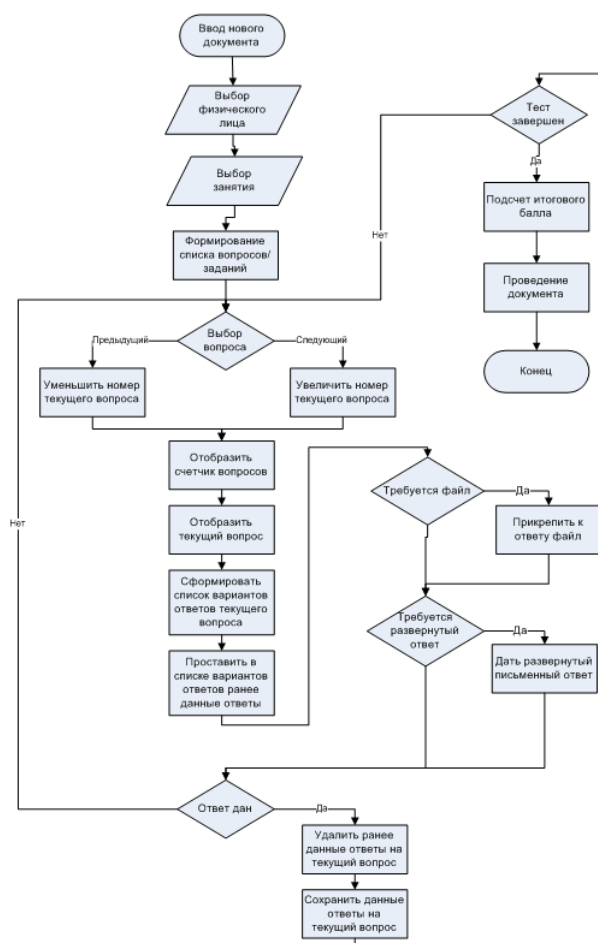


Рисунок 42 – Алгоритм прохождения практической части занятия

Проведение документа «Тестирование» означает фиксацию результатов. При этом рассчитывается итоговый балл прохождения теста и запрещается любое дальнейшее редактирование документа пользователями с правами «Обучающийся».

Функциональность документа «Тестирование» реализована в его программных модулях, приведенных в приложении В.

Если практическая часть упражнения построена в виде практического задания, то предполагается обратная связь (ручная проверка) документа преподавателем. Преподаватель должен открыть документ, просмотреть данные на вопросы (задачи) ответы, открыть и проверить прикрепленные файлы, прочитать развернутые ответы. На основе этого преподаватель выставляет балл за каждый вопрос-задачу, прикрепляет свои файлы и дает

развернутый комментарий-рецензию на данные обучающимся ответы. После этого обучающийся может открыть отрецензированный документ и ознакомиться с результатами ручной проверки, в том числе с файлами преподавателя и его комментариями к данным ответам.

Совокупность документов тестирования представляет из себя журнал тестирования, реализуемый формой списка документа «Тестирование». Главным моментом в форме является отбор по реквизитам документа и их сочетаниям, реализуемый при помощи сервисных механизмов платформы.

3.4.3 Разработка запросов и отчетов

Все отчеты в конфигурации созданы при помощи механизма «система компоновки данных», представленного на рисунке 43.

Схема создается при помощи конструктора, который позволяет полностью описать данные будущего отчета [29]. В частности, при помощи настроек можно задать варианты отчета и определить структуру выводимых данных (группировок), выбранные поля, порядок сортировки и так далее.

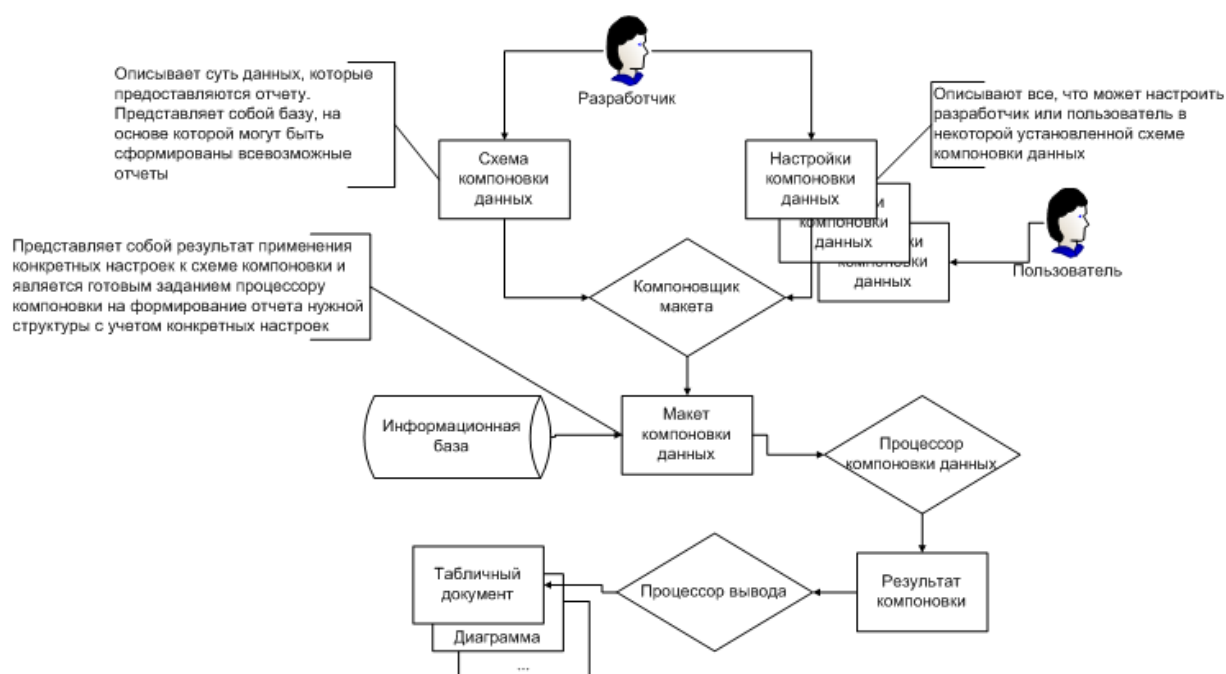


Рисунок 43 – Система компоновки данных

Для получения усредненных значений баллов по пройденным учащимися упражнениям в различных разрезах служит отчет «Успеваемость». Отчет построен при помощи системы компоновки данных. Основой таких отчетов является запрос. В данном отчете запрос строится к регистру сведений «Обучение», являющимся источником данных для отчета. Текст запроса отчета «Успеваемость» представлен на рисунке 44.

```

ВЫБРАТЬ
    Обучение . Тема . Владелец      КАК Дисциплина ,
    Обучение . Тема . Ссылка        КАК Тема ,
    Обучение . Упражнение . Ссылка  КАК Упражнение ,
    Обучение . Упражнение . Автор   КАК Преподаватель ,
    Обучение . физЛицо . Ссылка     КАК Учащийся ,
    Обучение . Балл                 КАК Балл ,
    Обучение . Попыток              КАК Попыток ,
    Обучение . Документ             КАК Занятие ,
    Обучение . Период               КАК Дата

ИЗ
    РегистрСведений . Обучение . СрезПоследних  КАК Обучение
  
```

Рисунок 44 – Текст запроса отчета «Успеваемость»

Цель отчета – получение среднего значения поля регистра «Балл» в различных разрезах. Для этой цели в отчете введен ресурс «Балл», вычисляемый как агрегатная функция «Среднее» по полю «Балл» запроса. Ресурс «Балл» отчета «Успеваемость» представлен на рисунке 45.

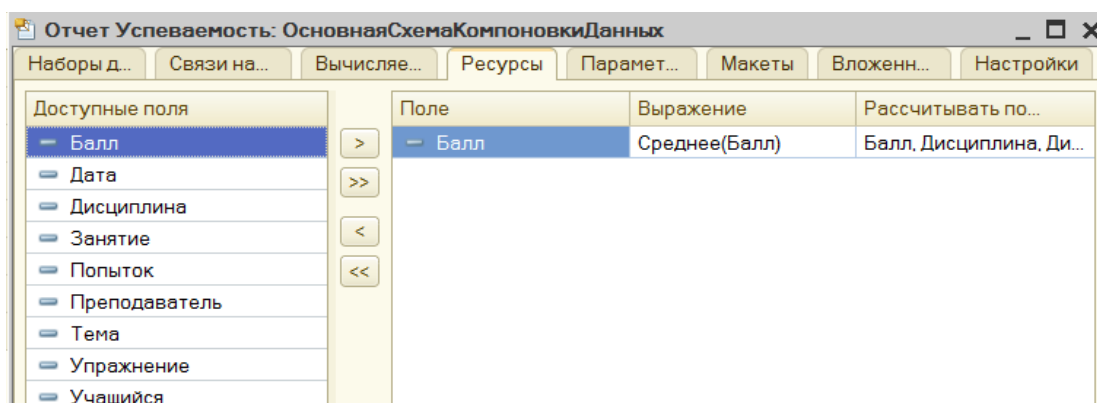


Рисунок 45 – Ресурс «Балл» отчета «Успеваемость»

Далее этот ресурс выводится во всех группировках отчета, вариантов которых благодаря использованию механизма СКД, может быть сколько угодно много. На рисунке 46 представлен вариант группировок отчета «По дисциплинам». В этом варианте оцениваются средний балл, показываемый всеми учащимися, проходящими занятия по определенной теме или дисциплине, то есть как бы оценивается успешность работы преподавателей.

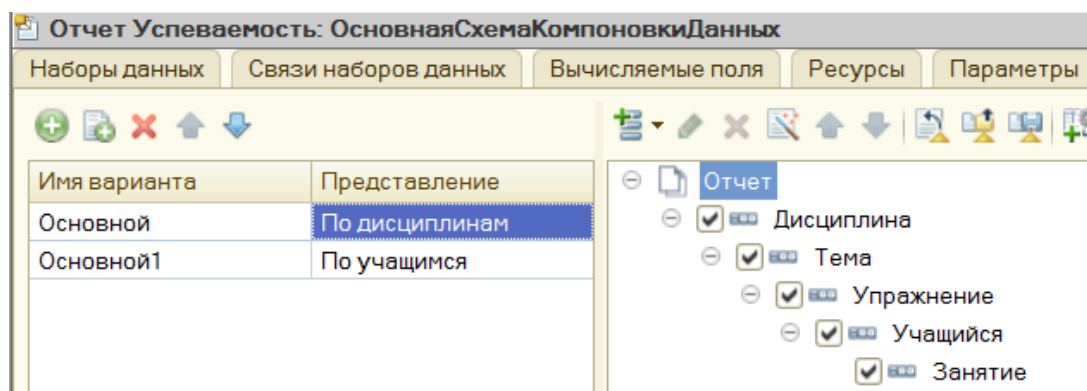


Рисунок 46 – Вариант «По дисциплинам» отчета «Успеваемость»

На рисунке 47 представлен вариант этого же отчета «По учащимся», показывающий среднюю успеваемость одного учащегося в разрезе дисциплин, тем или упражнений.

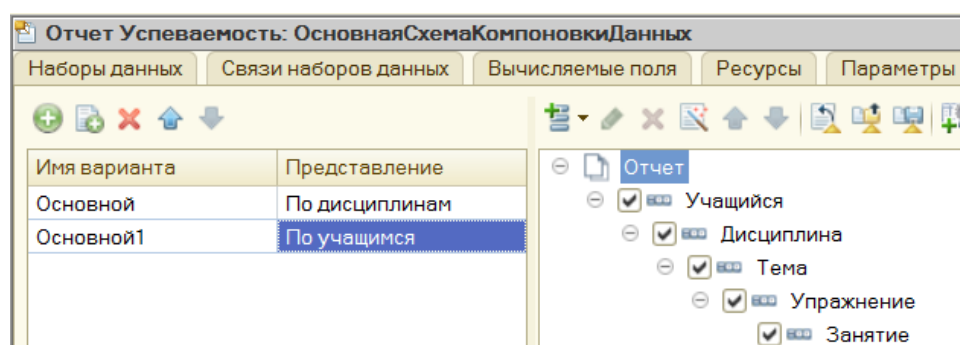


Рисунок 47 – Вариант «По учащимся» отчета «Успеваемость»

Возможны и другие варианты настроек отчета, позволяющие получить «много отчетов в одном». А использование широких возможностей отбора, сортировки, условного оформления и других, предоставляемых платформой, делает его использование удобным и практичным в различных вариантах.

3.5 Тестирование и описание функциональности системы сетевого обучения

3.5.1 Выбор методики и критериев тестирования

Разработанная система обладает большим количеством функций. При тестировании разработанной системы критически важно протестировать следующий набор функций:

- ввод в систему упражнений (тестов);
- прикрепление к объектам структуры обучения файлов;
- создание практической части упражнения: список вопросов и ответов;
- настройка логики автоматической оценки правильности данного ответа в варианте теста;
- настройка ответов на вопросы, представляющих собой практическое задание;
- составление плана обучения для учащихся;
- прохождение теста с точки зрения обучающегося;
- выполнение упражнения, представляющего собой практическое задание с точки зрения обучающегося;
- наличие обратной связи с преподавателем;
- работа отчета «Успеваемость».

Тестирование будет проводиться при помощи полного прогона функций разработанной системы на платформе «1С:Предприятие 8.3» в режиме «Предприятие».

3.5.2 Тестирование системы сетевого обучения

С точки зрения преподавателей главной функцией подсистемы работы со справочниками является ввод в систему упражнений (тестов). Упражнения (тесты) целесообразно вводить в последовательности Дисциплины-Темы-Упражнения. Командный интерфейс позволяет для текущей дисциплины в списке дисциплин открыть список ее тем, а для текущей темы – список упражнений (тестов). Последовательность Дисциплины-Темы-Упражнения представлена на рисунке 48.

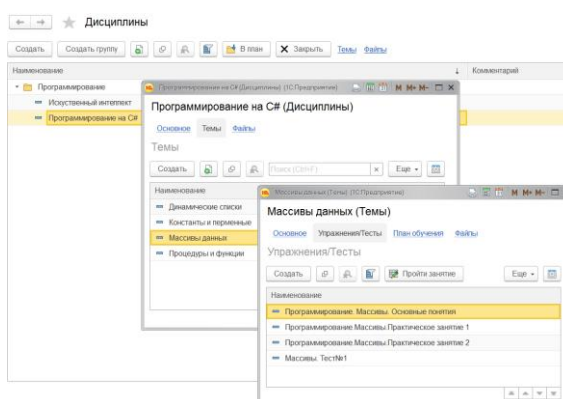


Рисунок 48 – Последовательность Дисциплины-Темы-Упражнения

К любой дисциплине или теме можно прикрепить любое количество файлов, иллюстрирующих эту дисциплину или тему. Так для дисциплины это может быть программа обучения в word-файле, а для темы – учебные материалы по этой теме в pdf-файле. Открывается список файлов при выборе пункта «Файлы» в командном интерфейсе. Список файлов, иллюстрирующих объект учебных материалов, представлен на рисунке 49.

Файлы, прикрепленные к упражнению (тесту), являются учебными материалами этого упражнения. Они задаются несколько по-другому – не через пункт «Файлы» командного интерфейса, а на закладке «Теория» формы упражнения, что показано на рисунке 50. Отличительной особенностью файлов, прикрепленных к упражнению, является то, что прикреплять можно не только «личные» файлы данного упражнения, но и любые файлы из

библиотеки файлов, возможно даже относящиеся к другим темам или дисциплинам. Это логично – нет смысла создавать еще один файл, содержащий базовые знания по теме к конкретному упражнению, если он уже прикреплен к самой теме или дисциплине, достаточно дать на него ссылку.

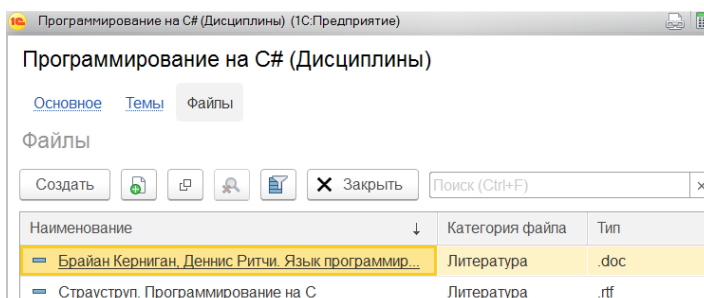


Рисунок 49 – Список файлов, иллюстрирующих объект учебных материалов

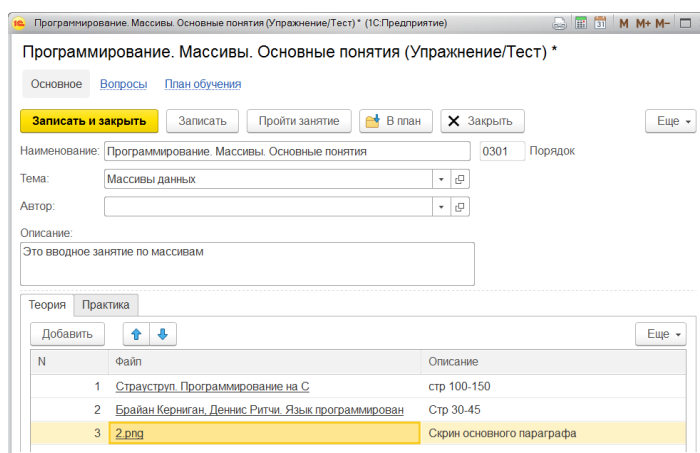


Рисунок 50 – Учебные материалы упражнения (теста)

Сами файлы хранятся в базе данных разработанной системы, а не в файловой системе сервера, на котором хранится информационная база. По команде «Открыть файл» в списке файлов или в форме самого файла запускается приложение, ассоциированное с типом выбранного файла, в котором и открывается нужный файл.

После создания упражнения (теста) и наполнения его учебными материалами можно переходить к созданию практической части упражнения. Это самая важная часть работы преподавателя – от такого насколько грамотные и разнообразные вопросы и задания будет включать в себя практическая часть упражнения, зависит качество образовательного процесса.

Разработка практической части упражнения (теста) заключается во вводе списка вопросов упражнения (теста), вводе вариантов ответов для каждого вопроса, заданиям им весов. Редактирование тестов можно проводить в последовательности Упражнение-Вопросы-Ответы, что показано на рисунке 51. А можно каждую категорию данных открывать в отдельном окне, воспользовавшись командами формы списка соответствующего окна.

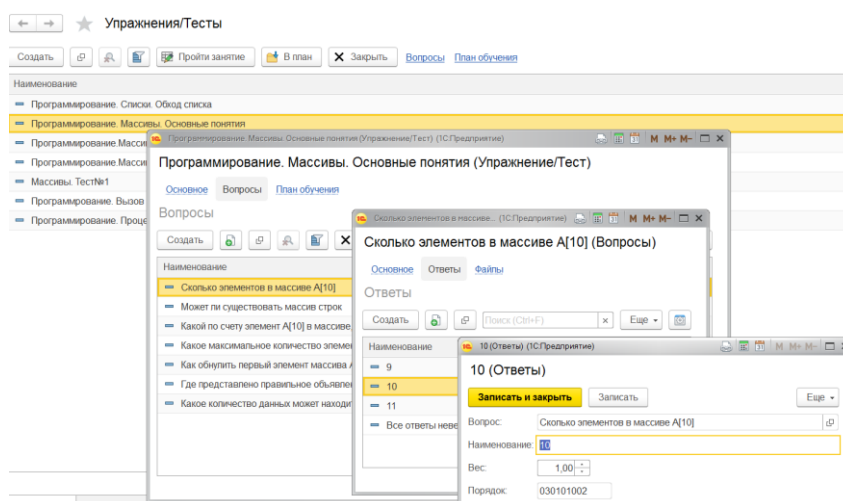


Рисунок 51 – Последовательность Упражнение-Вопросы-Ответы

Разрабатываемая система предоставляет возможности гибкой настройки практической части упражнений (тестов). Если практическая часть упражнения представляет собой просто тест, то, как правило, он состоит из нескольких вопросов, на каждый из которых учащийся должен выбрать один или несколько правильных, по его мнению, ответов.

При редактировании вопроса к нему также можно присоединить файлы, его иллюстрирующие. Если файл имеет признак «Основной» и графический формат, то данный файл будет выведен в качестве картинки данного вопроса при прохождении упражнения.

Варианты ответов вводит преподаватель, при этом каждому ответу задается вес, принимающий значение «0» для неправильного ответа и «1» для правильного. Однако веса могут принимать не только значения «0» и «1», они могут быть дробными для «не совсем правильных» ответов, принимать значения больше единицы для очень сложных вопросов, или, наоборот, отрицательные значения для «очень неправильных» ответов.

На некоторые вопросы пользователь может дать не один, а несколько вариантов ответов. Для таких вопросов устанавливается свойство «Несколько ответов». При этом правильным ответам следует задать дробные значения весов, так чтобы в сумме они дали значение «1». Ответы с дробными весами показаны на рисунке 52.

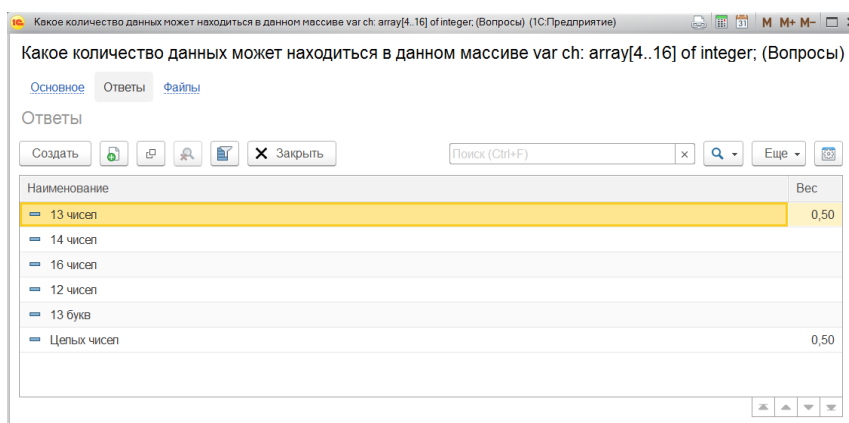


Рисунок 52 – Ответы с дробными весами

Если по логике вопроса нужно, чтобы балл за такой вопрос начислялся, только если выбраны все правильные варианты ответов, то вопросу следует установить признак «Нужны все правильные», такая настройка показана на рисунке 53.

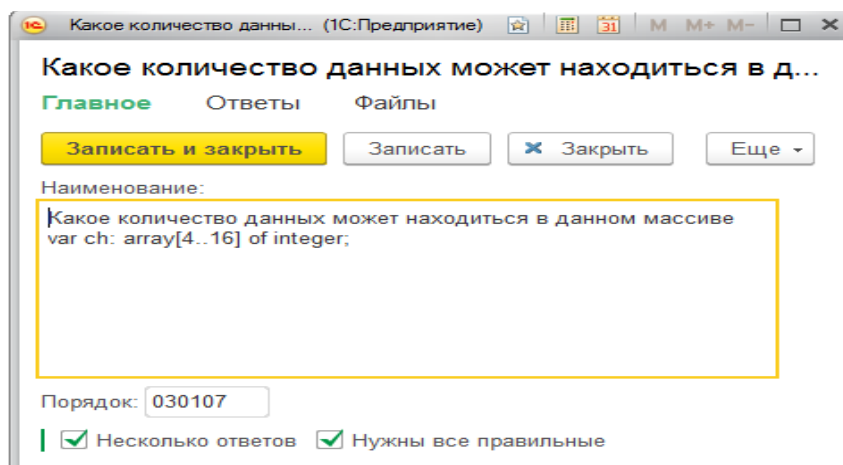


Рисунок 53 – Настройка «Нужны все правильные ответы» в вопросе с несколькими ответами

Если практическая часть упражнения представляет собой набор практических заданий, то вопросы, как правило, представляют из себя текст (возможно иллюстрированный) этих заданий, а ответы – приглашения прикрепить файлы с результатами выполнения задания или дать развернутый ответ. Варианты ответов на вопросы практического задания показаны на рисунке 54. Веса в этом случае назначаются вариантам ответов формально, лишь для того чтобы отобразить в автоматически рассчитываемом балле факт выполнения всех заданий (прикрепления всех файлов). Итоговый балл выставляется преподавателем вручную.

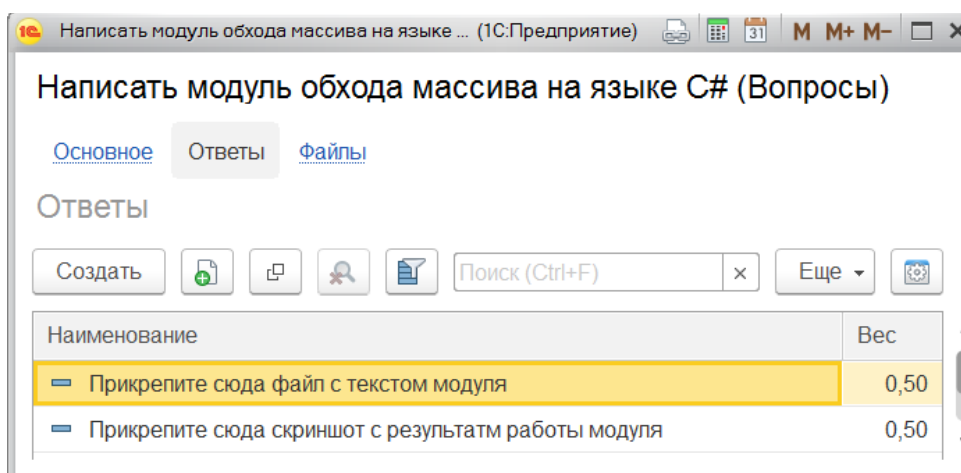


Рисунок 54 – Варианты ответов на вопросы практического задания

Однако гибкость системы позволяет совместить тесты и практические задания. На рисунке 55 представлен вопрос (задание), ответом на который должен являться файл, но прикрепить этот файл нужно именно к правильному варианту ответа, что показано на рисунке 56.

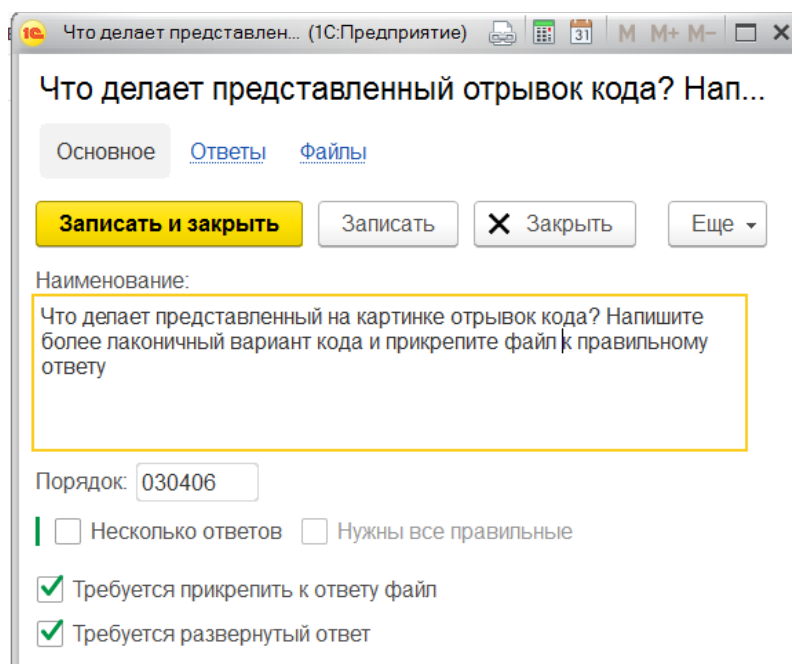


Рисунок 55 – Вариант совмещения теста и практического задания

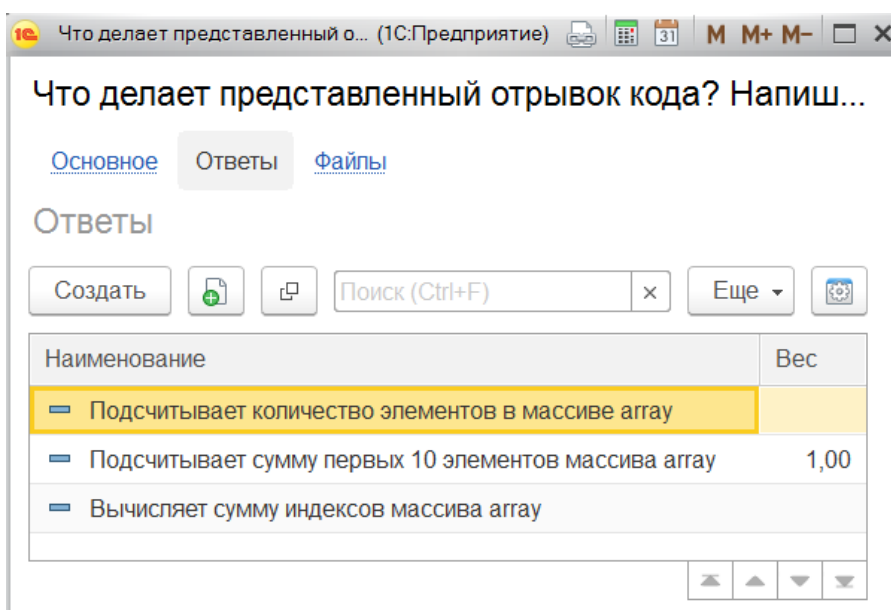


Рисунок 56 – Варианты ответов на вопрос-задание

После ввода в систему учебных материалов (упражнений по темам и дисциплинам) необходимо составить план обучения по каждому учащемуся. С точки зрения преподавателя это логично делать по группам учащихся (иерархия справочника «Физические лица») или по направлениям обучения (учебные группы), так как в большинстве случаев планы обучения у учащихся одних и тех же групп одинаковые. Учащийся может записать сам себя на заинтересовавшее его упражнение или тему, если самостоятельная запись предусмотрена учебным центром и программой обучения. Запись производится путем выполнения команды (нажатия кнопки) «В План», показанной на рисунке 57, которая доступна в контексте почти любого объекта системы.

Откроется окно «В план», представленное на рисунке 58, в котором автоматически будут заполнены объекты, из которых вызвана команда, их можно изменить и дополнить. После нажатия кнопки «Назначить в план обучения» в регистре сведений «Обучение» будут созданы соответствующие записи.

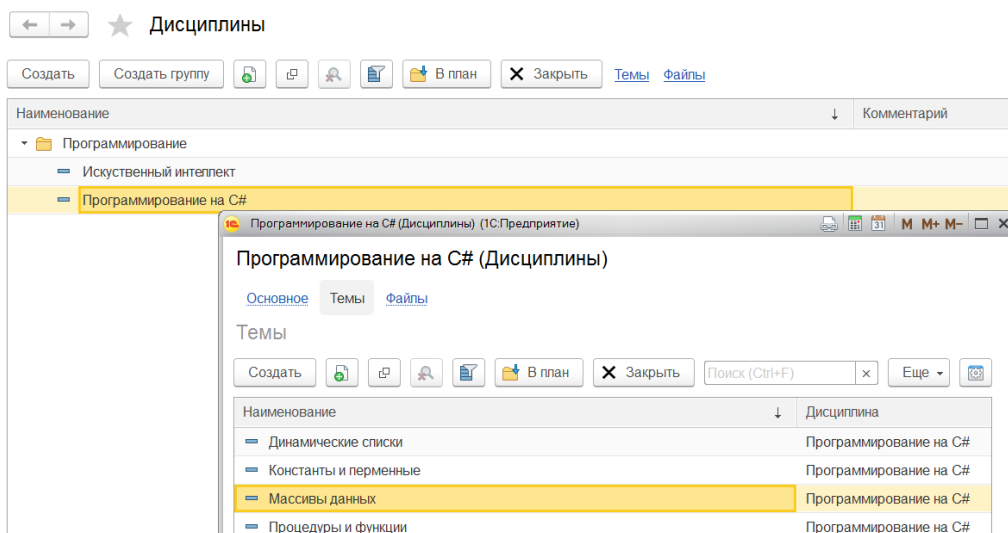


Рисунок 57 – Команда «В план» в форме справочников «Дисциплины» и «Темы»

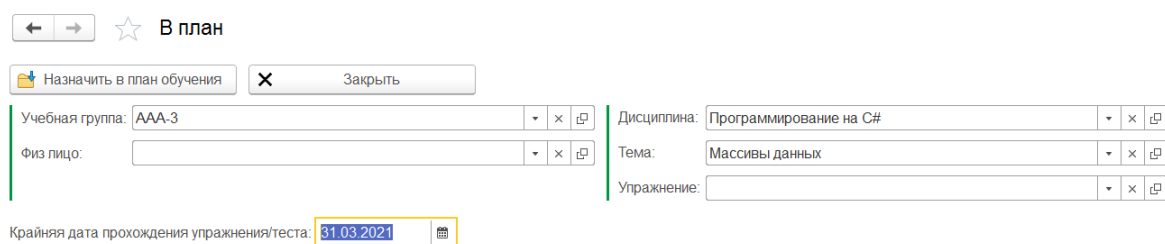


Рисунок 58 – Окно «В план»

Как уже говорилось выше план обучения становится основным инструментом планирования и контроля обучения. Огромные и гибкие настройки параметров отбора, сортировки, группировок, условного оформления и другие позволяют сделать его удобным инструментом для решения любых задач. На рисунке 59 представлен пример, в котором преподаватель «Володин» сделал отбор записей плана обучения, требующих его внимания, а именно занятий, пройденных учащимися и ждущих его ручной проверки.

Дата	Дисциплина	Тема	Упражнение	Учащийся	Занятие	К...	Балл	Поп...
01.02.2022	Программирование на C#	Массивы данных	Программирование.Массивы.П...	Милютин Андр...	Занятие 00000004...	✓	0,63	2
08.02.2022	Программирование на C#	Массивы данных	Программирование.Массивы.П...	Аксенов Серг...			-	

Рисунок 59 – Отбор плана обучения по преподавателю

Для учащегося план обучения – это руководство к действию. Он открывается в начальной странице конфигурации и наглядно демонстрирует, какие из назначенных упражнений уже пройдены и каковы их результаты, а какие еще предстоит пройти. План обучения с точки зрения учащегося показан на рисунке 60.

Дата	Дисциплина	Тема	Упражнение	Учащийся	Занятие	К..	Балл	Поп..
01.02.2022	Программирование ...	Массивы данных	Программирование. ...	Милютин Андрей Бо...	Занятие 000000042 от 10...		-	4
01.02.2022	Программирование ...	Массивы данных	Массивы. Тест№1	Милютин Андрей Бо...			-	
01.02.2022	Программирование ...	Массивы данных	Программирование...	Милютин Андрей Бо...			-	
01.02.2022	Программирование ...	Массивы данных	Программирование...	Милютин Андрей Бо...	Занятие 000000045 от 05...	✓	0,63	2
01.02.2022	Программирование ...	Динамические списки	Программирование. ...	Милютин Андрей Бо...	Занятие 000000044 от 07...			1

Рисунок 60 – План обучения с точки зрения учащегося

Само прохождение упражнения (теста) заключается в создании документа «Занятие». Создать новое «Занятие» можно из журнала «Журнал», который открывается по команде «Журнал» основного раздела; выбрав команду «Создать-Занятие» в основном разделе, или ввести документ на основании выбранного упражнения /теста в списке упражнений, или из плана обучения (нажав кнопку «Пройти занятие»), процесс показан на рисунке 61.

Упражнения/Тесты

Пройти занятие В план Закреть План обучения

Наименование
Программирование. Списки. Обход списка
Программирование. Массивы. Основные понятия
Программирование. Массивы. Практическое занятие 1
Программирование. Массивы. Практическое занятие 2
Массивы. Тест№1
Программирование. Вызов Функций
Программирование. Процедуры

Рисунок 61 – Создание занятия на основании упражнения/теста

В любом случае откроется новый документ «Занятие». Система автоматически заполнит поле «ФизЛицо», ассоциировав его с авторизовавшимся пользователем, а также тему и упражнение, если документ создан из контекста списка упражнений или плана обучения.

Прохождение занятия заключается в проставлении ответов на вопросы практической части упражнения, по которому проводится занятие. Учащемуся показывается текст вопроса, картинка вопроса (если она задана автором теста), варианты ответов на текущий вопрос. Пользователь может выбрать вариант ответа, который он считает правильным, или несколько вариантов, если вопрос предполагает выбор нескольких вариантов ответов. При переходе к следующему или предыдущему вопросу ответ, данный пользователем, сохраняется и вновь отображается при возврате к этому вопросу. Пример прохождения занятия, созданный на основе упражнения-теста, показан на рисунке 62.

The screenshot shows a user interface for a learning management system. At the top, there is a navigation bar with a star icon and the text "Занятие 000000040 от 03.02.2022 0:00:00". Below this is a toolbar with buttons: "Провести и закрыть", "Записать", "Провести", and "Закрыть". There is also a dropdown menu labeled "Еще".

Below the toolbar, there are fields for "Номер: 000000040", "Дата: 03.02.2022 0:00:00", and "Балл: 0,14". There is also a field for "Физ лицо: Милютин Андрей Борисович".

The main content area shows the question: "Вопрос 6/7" and "Где представлено правильное объявление двумерного массива?". To the right of the question, there are three code snippets in dark boxes:

```
int a[2]={ int b[3];
           int c[5];
           };

char a[2][3]={
             {'a', 'r', 'd'},
             {'0', ' ', ' '}};

double a[2][3]={
              {'1', '2'}
              {'5', '2.3'}
              {'4', '-1'}};
```

Below the question, there are two buttons: "Предыдущий" and "Следующий".

At the bottom, there is a table with three rows:

<input checked="" type="checkbox"/>	1	A
<input type="checkbox"/>	1	B
<input type="checkbox"/>	0.0	B

Рисунок 62 – Прохождение упражнения-теста

Если упражнение представляет собой практическое задание, то ответ на вопрос, как правило, требует прикрепления файла или дачи развернутого ответа или комментария, что показано на рисунке 63.

← → ☆ Занятие 000000045 от 05.02.2022 0:00:00

Провести и закрыть | Записать | Провести | ✕ Закрыть

✓ Номер: 000000045 | Дата: 05.02.2022 0:00:00 | Балл: 0,63

Физ лицо: Милютин Андрей Борисович

Тема: Массивы данных

Занятие: Программирование.Массивы.Практическое занятие 1

Вопрос 1/4

Написать модуль обхода массива на языке C#

← Предыдущий | Следующий → | Проверка

✓	Прикрепите сюда файл с текстом модуля	Выполненное задание	Это текст
✓	Прикрепите сюда скриншот с результатами работы модуля	Выполненное задание	Это результат

Рисунок 63 – Выполнение практического задания

При этом тесты могут быть совмещены с практическим заданием. На рисунке 64 представлен пример вопроса, в котором сначала нужно выбрать правильный вариант ответа, а потом выполнить для этого варианта практическое задание (прикрепить файл) и дать комментарий.

← → ☆ Занятие 000000045 от 05.02.2022 0:00:00

Провести и закрыть | Записать | Провести | ✕ Закрыть

✓ Номер: 000000045 | Дата: 05.02.2022 0:00:00 | Балл: 0,63

Физ лицо: Милютин Андрей Борисович

Тема: Массивы данных

Занятие: Программирование.Массивы.Практическое занятие 1

Вопрос 3/4

Что делает представленный на картинке отрывок кода? Напишите более лаконичный вариант кода и прикрепите файл к правильному ответу

← Предыдущий | Следующий → | Проверка

```
int b=0;
for ( int count = 0; count < 10; count++)
    b += array[count];
```

	Подсчитывает количество элементов в массиве arrau		
✓	Подсчитывает сумму первых 10 элементов массива arrau	Выполненное задание	Вот правильный р...
	Вычисляет сумму индексов массива arrau		

Рисунок 64 – Выполнение практического задания, совмещенного с тестом

Занятие завершается проведением документа, при этом рассчитывается итоговый балл теста и документ закрывается для редактирования пользователем категории «Обучающийся» и «Преподаватель». При этом его по-прежнему можно просматривать точно так же, как и при прохождении

занятия: можно просматривать не только итоговый балл, но и ответы, данные пользователем на вопросы теста.

В случае если упражнение представляет собой практическое задание, то предполагается обратная связь с преподавателем, а рассчитанный автоматически балл представляет из себя лишь формальную оценку выполнения всех заданий занятия (прикрепления всех необходимых файлов и/или дачи развернутых ответов). Обратная связь заключается в ручной проверке преподавателем таких занятий. Преподаватель может зайти в такие занятия, просмотреть данные ответы, прикрепленные файлы, дать комментарий и прикрепить свой файл к каждому ответу и занятию в целом, а также вручную изменить автоматически рассчитанный балл по каждому ответу. Ручная проверка занятия преподавателем показана на рисунке 65. Но при этом редактировать данные учащимся ответы преподаватель не может. Учащийся в дальнейшем может по каждому занятию увидеть реакцию преподавателя.

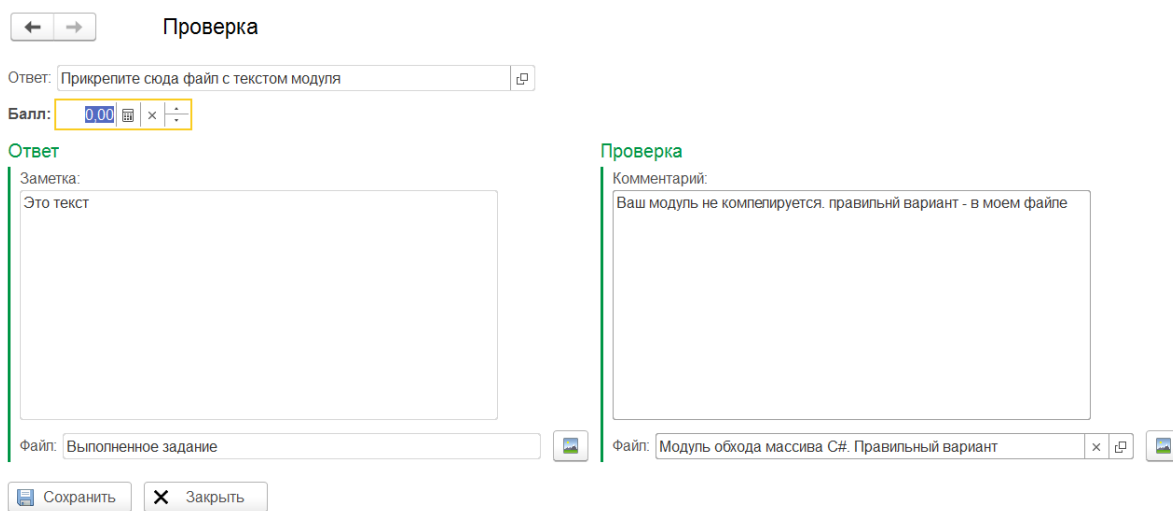


Рисунок 65 – Ручная проверка занятия преподавателем

Преподаватель использует журнал занятий и план обучения для оценки пройденных обучающимися занятий. Для этого он может использовать огромные возможности настройки списка документов, предоставляемые

платформой, среди которых важнейшей является отбор. Однако и журнал занятий и план обучения не подходят для анализа качества образовательного процесса и для получения средних оценок. Для этого предназначен отчет «Успеваемость», который позволяет получать средние оценки, показанные учащимися при прохождении ими занятий в различных разрезах, с наложением различных условий и группировок. При этом гибкость настроек позволяет использовать отчет для различных целей. Задав группировки так, как показано на рисунке 66, мы получаем отчет, в котором видна средняя успеваемость группы «ААА-3» по одной из дисциплин в разрезе тем и упражнений этой дисциплины, что представлено на рисунке 67.

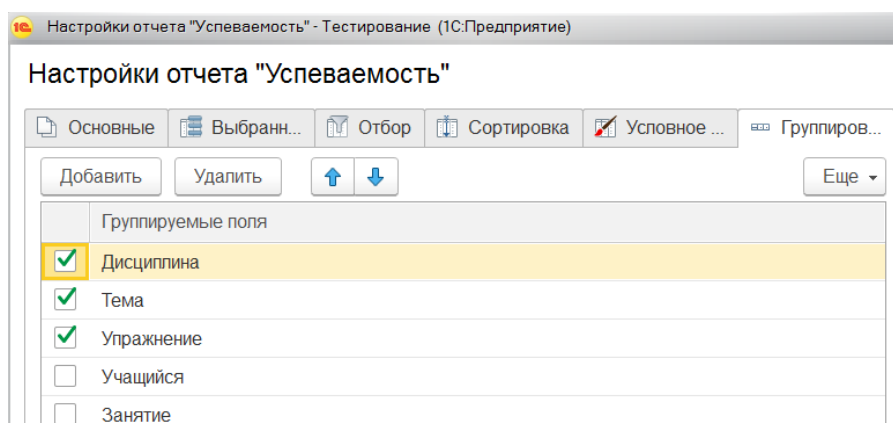


Рисунок 66 – Первый вариант группировок отчета

Дисциплина	Балл
Программирование на С#	0,53
Динамические списки	1,00
Программирование. Списки. Обход списка	1,00
Массивы данных	0,41
Массивы. Тест№1	-
Программирование. Массивы. Основные понятия	1,00
Программирование. Массивы. Практическое занятие 1	0,63
Программирование. Массивы. Практическое занятие 2	-
Итого	0,53

Рисунок 67 – Средняя успеваемость по дисциплине

Этот же отчет, изменив его настройки, можно использовать для получения картины успеваемости одного учащегося, показанной на рисунке 68, в том числе с расшифровкой для понимания, откуда взялись средние цифры.

← → ☆ Успеваемость (По учащимся)

Сформировать Выбрать вариант... Изменить вариант... Сохранить вариант...

Учащийся: Милютин Андрей Борисович

Отбор: Учащийся Равно "Милютин Андрей Борисович"

Учащийся	Балл
Дисциплина	
Тема	
Упражнение	
Занятие	
Милютин Андрей Борисович	0,33
Программирование на C#	0,33
Динамические списки	1,00
Программирование. Списки. Обход списка	1,00
Занятие 000000044 от 07.02.2022 0:00:00	1,00
Массивы данных	0,16
Массивы. Тест№1	-
Программирование. Массивы. Основные понятия	-
Занятие 000000042 от 10.02.2022 0:00:00	-
Программирование.Массивы.Практическое занятие 1	0,63
Занятие 000000045 от 05.02.2022 0:00:00	0,63
Программирование.Массивы.Практическое занятие 2	-
Итого	0,33

Рисунок 68 – Успеваемость по учащемуся

Занятия, в результате которых образовались средние значения, можно открывать прямо из отчета, как и остальные объекты.

Выводы по третьей главе

В третьей главе описывается процесс разработки 1С-конфигураци на основе ранее описанного проекта. Подробно описывается процесс разработки объектов конфигурации, их структуры, программных модулей. В конце главы приводится описание процесса тестирования системы, которое фактически является руководством для пользователя.

Заключение

В результате выполнения работы были изучены особенности процесса обучения и контроля знаний, полученных обучающимися сотрудниками в ходе образовательного процесса. Был сделан вывод о возможности применения ССО как одного из лучших инструментов системы контроля знаний.

Сделан вывод о необходимости разработки автоматизированной информационной системы, позволяющей вводить теоретический материал, практические задания и тесты, а также изучать этот материал, выполнять задания и проходить тесты.

Были сформулированы требования к разрабатываемой системе, определены ее задачи и цели.

Система сетевого обучения позволит организовать процедуру обучения на порядок лучше, чем традиционный вариант.

В качестве среды разработки системы выбрана платформа «1С:Предприятие 8.3», как наиболее подходящая для целей системы. Рассмотрены основные особенности архитектуры этой системы, ее функциональность и механизмы, при помощи которых строятся прикладные решения. Найдены способы использования основных механизмов системы «1С:Предприятие 8.3» для целей проекта.

В среде разработки «1С:Предприятие 8.3» была реализована конфигурация, выполняющая поставленную задачу. Были созданы объекты конфигурации, определена их структура, реквизиты, экранные формы, написаны программные модули, реализующие заданную функциональность этих объектов. Особое внимание уделено разработке структуры объектов информационной базы: справочников, перечислений, документов.

Проведенное тестирование разработанной конфигурации показывает, что она полностью удовлетворяет поставленной задаче, работает стабильно и без ошибок, обладает большой функциональностью и удобным интерфейсом.

Разработанную конфигурацию можно внедрять в учебном центре для организации процесса обучения.

Разработанная система может быть использована как для дистанционного, так и для смешанного обучения. В случае полностью дистанционного варианта обучения система выступает как платформа общения преподавателей и обучающихся через сеть Интернет.

В случае смешанного варианта обучения предполагается, что обучающиеся сотрудники посещают очные занятия и прослушивают изложение теоретического материала от преподавателя; учебные и методические материалы, загруженные в систему, выступают в этом случае как справочные материалы. Характер учебных и методических материалов, которые могут быть загружены в систему, как и форма их представления, могут быть любыми. В зависимости от изучаемого курса и темы, преподаватель, разрабатывающий учебные материалы, может использовать для их создания любое программное обеспечение.

Отличительной особенностью разработанной системы является планирование обучения. Предусмотрена возможность записи обучающегося (группы обучающихся) на учебный курс и средства контроля прохождения обучающимися занятий учебного курса.

Проведенное тестирование разработанной конфигурации показывает, что она полностью удовлетворяет поставленной задаче, работает стабильно и без ошибок, обладает большой функциональностью и удобным интерфейсом. Разработанную конфигурацию можно внедрять в Учебном центре для организации процесса подготовки слушателей курсов повышения квалификации.

Внедрение разработанной системы должно резко увеличить эффективность процесса подготовки сотрудников, проходящих обучение в учебном центре, и контроля их знаний.

Список используемой литературы и используемых источников

1. 1С:Предприятие 8. Система программ [Электронный ресурс]. – Режим доступа: <http://v8.1c.ru>. – Заглавие с экрана. – (дата обращения 15.05.2022).
2. Ажеронок, В. А. Профессиональная разработка в системе «1С: Предприятие 8» (+DVD-ROM) / В. А. Ажеронок, А. П. Габец, Д. И. Гончаров, Д. В. Козырев, Д. С. Кухлевский, А. В. Островерх, М. Г. Радченко, Е. Ю. Хрусталева. – Москва: 1С-Паблишинг, 2013. – 870 с.
3. Ажеронок, В. А. Разработка управляемого интерфейса (+CD). Серия «1С: Профессиональная разработка» / В. А. Ажеронок, А. В. Островерх, М. Г. Радченко, Е. Ю. Хрусталева. – Москва: 1С-Паблишинг, 2015. – 740 с.
4. Айзенк, Г. Классические IQ тесты / Г. Айзенк. – Москва: Эксмо, 2012.
5. Ашарина, И. В. Основы программирования на языках С и С++: Курс лекций для высших учебных заведений / И. В. Ашарина. – Москва: Гор. линия-Телеком, 2018. – 208 с.
6. Бабэ Б. Просто и ясно о Borland C++: Пер. с англ. – Москва : Бином. – 416 с.: илл.
7. Баженова, И. Ю. Языки программирования: Учебник для студентов учреждений высш. проф. образования / И. Ю. Баженова; Под ред. В. А. Сухомлин. – Москва : ИЦ Академия, 2018. – 368 с.
8. Вендров, А. М. CASE технологии. Современные методы и средства проектирования информационных систем / А. М. Вендров. – Москва: Финансы и статистика, 2014. – 176 с.
9. Вендров, А. М. Проектирование программного обеспечения экономических информационных систем / А. М. Вендров. – Москва: Финансы и статистика, 2012. – 187с.
10. Габец, А. П. Реализация прикладных задач в системе «1С: Предприятие 8.2» (+CD). Серия «1С: Профессиональная разработка» / А. П.

Габец, Д. В. Козырев, Д. С. Кухлевский, Е. Ю. Хрусталева. – Москва: 1С-Публишинг, 2014. – 830 с.

11. Гончаров, Д. И. Решение специальных прикладных задач в «1С: Предприятия 8.2». Серия «1С: Профессиональная разработка» / Д. И. Гончаров, Е. Ю. Хрусталева. – Москва: 1С-Публишинг, 2015. – 935 с.

12. Диго, С. М. Проектирование и использование баз данных : учебник / С. М. Диго. – Москва: Финансы и статистика, 2015.

13. Дрещинский, В. А. Методология научных исследований [Электронный ресурс] : учебник для бакалавриата и магистратуры / В. А. Дрещинский. – 2-е изд., перераб. и доп. – Москва: Издательство Юрайт, 2018. – 324 с.

14. Дроздова, Г. И. Научно-исследовательская и творческая работа в семестре [Электронный ресурс] : учебное пособие / Г. И. Дроздова. – Омск: Омский государственный институт сервиса, 2013. – 66 с.

15. Емельянова, И. Н. Основы научной деятельности студента. Магистерская диссертация [Электронный ресурс] : учебное пособие для вузов / И. Н. Емельянова. – Москва: Издательство Юрайт, 2018. – 115 с.

16. Ивашина, А. С. Тестовый контроль знаний / А. С. Ивашина, Т. С. Свиридова // Специалист. – 2013. – №12.

17. Котлер, Ф. Маркетинг менеджмент / Ф. Котлер. – Санкт-Петербург: Питер, 2012. – 756 с.

18. Матищев, А. Н. Эффективность рекламы / А. Н. Матищев. – Москва: Издательство «Финпресс», 2012. – 416 с.

19. Мокий, М. С. Методология научных исследований [Электронный ресурс] : учебник для магистратуры / М. С. Мокий, А. Л. Никифоров, В. С. Мокий. – Москва: Издательство Юрайт, 2018. – 255 с.

20. Новости и технологии торговли [Электронный ресурс]. – Режим доступа: <http://www.torgrus.com>. – Заглавие с экрана. – (дата обращения 15.05.2022).

21. Основы построения баз данных / под ред. А. Д. Хомоненко. – Санкт-Петербург, 2014.
22. Попов, В. А. Размышление учителя над итогами тестирования / В. А. Попов // Математика в школе. – 2013. – №3.
23. Программирование для всех [Электронный ресурс]. – Режим доступа: <http://www.realcoding.net>. – Заглавие с экрана. – (дата обращения 15.05.2022).
24. Радченко, М. Г. 1С: Предприятие 8.2. Практическое пособие разработчика. Примеры и типовые приемы (+диск) / М. Г. Радченко, Е. Ю. Хрусталева. – Москва: 1С-Паблишинг, 2014. – 968 с.
25. Радченко, М. Г. 1С: Предприятие 8.2. Коротко о главном. Новые возможности версии 8.2 / М. Г. Радченко. – Москва: 1С-Паблишинг, 2012. – 690 с.
26. Радченко, М. Г. 1С: Предприятие 8.2. Практическое пособие разработчика. Примеры и типовые приемы / М. Г. Радченко, Е. Ю. Хрусталева. – Москва: 1С-Паблишинг, 2012. – 620 с.
27. Радченко, М. Г. Архитектура и работа с данными «1С: Предприятия 8.2». Серия «1С: Профессиональная разработка» / М. Г. Радченко, Е. Ю. Хрусталева. – Москва: 1С-Паблишинг, 2013. – 650 с.
28. Титов, С. В. Недоработки централизованного тестирования / С. В. Титов // Математика в школе. – 2012. – №7.
29. Хрусталева, Е. Ю. 101 совет начинающим разработчикам в системе «1С: Предприятие 8» / Е. Ю. Хрусталева. – Москва: 1С-Паблишинг, 2011. – 214 с.
30. Alwan, M. Automation of manufacturing processes / M. Alwan // Manufacturing, Engineering and Technology. – San Francisco, 2006. – №6. – PP. 1051-1061.
31. Automating Software Development Processes [Article] / T. Kitchens // developer.* Magazine. – Canada, 2006. – PP. 002-009.

32. Blethyn, S. G. *Designing Information Systems* / S. G. Blethyn, C. Y. Parker. – London: Butterworth-Heinemann, 2014 – PP. 380-382.

33. Durkovic, O. *Risks in Information Systems Development Projects* / O. Durkovic, L. Rakovic // *Management Information Systems*. – Serbia, 2009. – PP. 013-019.

34. Madhav, S. *Game Programming Algorithms and Techniques* / S. Madhav // Addison-Wesley. – California, 2009. – PP. 02-012.

Приложение А

Текст программных модулей справочников

Справочники. Вопросы. Форма элемента. Модуль формы

```
&НаКлиенте  
Процедура ПриОткрытии(Отказ)  
ВидимостьЭлементов();// галочка НужныВсеПравильные доступна  
только если пр оставлена галочка НесколькоПравильных  
КонецПроцедуры
```

```
// галочка НужныВсеПравильные доступна только если пр оставлена  
галочка НесколькоПравильных
```

```
&НаКлиенте  
Процедура ВидимостьЭлементов()
```

```
Элементы.НужныВсеПравильные.Доступность =  
Объект.НесколькоПравильных;  
КонецПроцедуры
```

```
// если сняли галочку НесколькоПравильных - то и галочку  
НужныВсеПравильные снимаем
```

```
&НаКлиенте  
Процедура НесколькоПравильныхПриИзменении(Элемент)
```

```
Если Не Объект.НесколькоПравильных Тогда  
    Объект.НужныВсеПравильные = Ложь;  
КонецЕсли;
```

```
ВидимостьЭлементов();  
КонецПроцедуры
```

Справочники. Тесты. Форма элемента. Модуль формы

```
// при щелчке на поле Учебные материалы - открыть файл  
&НаКлиенте  
Процедура УчебныеМатериалыВыбор(Элемент, ВыбраннаяСтрока,  
Поле, СтандартнаяОбработка)
```

```
Если Поле.Имя = "УчебныеМатериалыФайл" Тогда
```

Продолжение приложения А

ОбщийКлиент.ОткрытьХранимыйФайл(Элемент.ТекущиеДанные.Файл);

 Возврат;
КонецЕсли;

КонецПроцедуры

Справочники.Хранимые файлы. Форма списка. Модуль формы

// щелкнули по имени файла - открыть файл
&НаКлиенте
Процедура СписокВыбор(Элемент, ВыбраннаяСтрока, Поле,
СтандартнаяОбработка)

 Если Поле.Имя = "Наименование" Тогда
 СтандартнаяОбработка = Ложь;

ОбщийКлиент.ОткрытьХранимыйФайл(Элемент.ТекущиеДанные.Ссылка);

 Возврат;
КонецЕсли;

КонецПроцедуры

Справочники. Хранимые файлы. Модуль объекта

// заполнить некоторые реквизиты по умолчанию
Процедура ОбработкаЗаполнения(ДанныеЗаполнения,
ТекстЗаполнения, СтандартнаяОбработка)
 Автор = ПараметрыСеанса.глПользователь;
 // автор - текущий пользователь

 Если РольДоступна("Тестер") Тогда
 КатегорияФайла =
Справочники.КатегорииФайлов.ФайлОтСтудента;
 Наименование = "Выполненное задание";
КонецЕсли;

КонецПроцедуры

Приложение Б

Текст общих программных модулей

Общие команды. Записаться на занятие. Модуль команды

```
&НаКлиенте  
Процедура ОбработкаКоманды(ПараметрКоманды,  
ПараметрыВыполненияКоманды)
```

```
ОбщийСрв.ЗаписатьсяНаТест(ПараметрКоманды,  
ОбщийСрв.ПолучитьГлПользователя());  
Оповестить("ПланОбучения");// переоткрыть форму регистра  
сведений, если произошли изменения в регистре  
КонецПроцедуры
```

Общие команды. Записаться на курс. Модуль команды

```
&НаКлиенте  
Процедура ОбработкаКоманды(ПараметрКоманды,  
ПараметрыВыполненияКоманды)  
ОбщийСрв.ЗаписатьсяНаТему(ПараметрКоманды,  
ОбщийСрв.ПолучитьГлПользователя());  
Оповестить("ПланОбучения");// переоткрыть форму регистра  
сведений, если произошли изменения в регистре  
КонецПроцедуры
```

Общие команды. Открыть файл. Модуль команды

```
&НаКлиенте  
Процедура ОбработкаКоманды(ПараметрКоманды,  
ПараметрыВыполненияКоманды)
```

```
ОбщийКлиент.ОткрытьХранимыйФайл(ПараметрКоманды);
```

```
КонецПроцедуры
```

Продолжение приложения Б

Общие модули. Общий клиент. Модуль

```
// открыть файл из содержимого ХранимыйФайл
&НаКлиенте
Процедура ОткрытьХранимыйФайл(ХранимыйФайл) Экспорт
Если ХранимыйФайл = Неопределено Тогда
```

```
    Возврат;
КонецЕсли;
```

```
Если ХранимыйФайл.Пустая() Тогда
    Возврат;
КонецЕсли;
```

```
ИмяФайла = "";
ХФСсылка = Неопределено;
Гипер      = Ложь;
ОбщийСрв.ПараметрыХранимыйФайл(ХранимыйФайл, ИмяФайла,
ХФСсылка, Гипер); // получить реквизиты из ХранимыйФайл
```

```
Если Гипер Тогда // если это гиперссылка на внешний интернет-ресурс
    ЗапуститьПриложение(ИмяФайла); // то в ИмяФайла хранится
    сама ссылка
    Возврат;
КонецЕсли;
```

```
Файл = Новый Файл(ИмяФайла);
ИмяФайла = ПолучитьИмяВременногоФайла(Файл.Расширение);
Адрес = ПолучитьНавигационнуюСсылку(ХФСсылка,
"ДанныеФайла");
ПолучитьФайл(Адрес, ИмяФайла, Ложь);
ЗапуститьПриложение(ИмяФайла);
```

КонецПроцедуры

Общие модули. ОбщийСрв.Модуль

```
// вернуть пользователя, зарегистрировавшегося при входе
Функция ПолучитьГлПользователя() Экспорт
Возврат ПараметрыСеанса.глПользователь;
```


Продолжение приложения Б

КонецФункции

```
// создать описание типа
Функция обСоздатьОписаниеТипов(НазваниеТипа, Длина = "",
Точность = "") Экспорт
    мас = Новый Массив;
    мас.Добавить(Тип(НазваниеТипа));
    Если Длина <> "" Тогда
        Если Точность = "" Тогда
            КвалифСтроки = Новый КвалификаторыСтроки(Длина,
ДопустимаяДлина.Переменная);
            Иначе
                КвалифЧисла = новый КвалификаторыЧисла(Длина,
Точность, ДопустимыйЗнак.Любой);
            КонецЕсли;
        КонецЕсли;
    Возврат Новый ОписаниеТипов(мас, КвалифСтроки, КвалифЧисла);
КонецФункции
```

```
// вернуть реквизит с именем ИмяРеквизита из Вопрос
Функция РеквизитВопроса(Вопрос, ИмяРеквизита) Экспорт
```

```
Если ИмяРеквизита = "НесколькоПравильных" Тогда
    Возврат Вопрос.НесколькоПравильных;
КонецЕсли;
```

```
Если ИмяРеквизита = "НужныВсеПравильные" Тогда
    Возврат Вопрос.НужныВсеПравильные;
КонецЕсли;
```

```
Если ИмяРеквизита = "ТребуетсяФайл" Тогда
    Возврат Вопрос.ТребуетсяФайл;
КонецЕсли;
```

```
Если ИмяРеквизита = "ТребуетсяРазвернутыйОтвет" Тогда
    Возврат Вопрос.ТребуетсяРазвернутыйОтвет;
КонецЕсли;
```

КонецФункции

```
// вернуть реквизит с именем ИмяРеквизита из Тест
Функция РеквизитТеста(Тест, ИмяРеквизита) Экспорт
```

Продолжение приложения Б

Если ИмяРеквизита = "ВсеВопросыОбязательно" Тогда
 Возврат Тест.ВсеВопросыОбязательно;
КонецЕсли;

Если ИмяРеквизита = "ТребуетсяРучнаяПроверка" Тогда
 Возврат Тест.ТребуетсяРучнаяПроверка;
КонецЕсли;

Если ИмяРеквизита = "РежимОбучения" Тогда

 Возврат Тест.РежимОбучения;
КонецЕсли;

Если ИмяРеквизита = "Пересдача" Тогда
 Возврат Тест.Пересдача;
КонецЕсли;

Если ИмяРеквизита = "Таймер" Тогда
 Возврат Тест.Таймер;
КонецЕсли;

КонецФункции

// вернуть владельца (тема) из Тест
Функция ПолучитьТемуТеста(Тест) Экспорт

 Если Не Тест = Неопределено Тогда
 Возврат Тест.Владелец;
 КонецЕсли;
КонецФункции

// создать в регистре сведений Обучение записи по всем занятиям
(тестам) из Темы для ФизЛицо

 Процедура ЗаписатьсяНаТему(Тема, ФизЛицо) Экспорт
 ТестыТемы = Справочники.Тесты;
 НаборЗаписейОбучение =

РегистрыСведений.Обучение.СоздатьНаборЗаписей();

 Выборка = ТестыТемы.Выбрать(Тема, "Код");// выбрать тесты,
подчиненные Теме

 Пока Выборка.Следующий() Цикл // для каждого теста из выборки

 НаборЗаписейОбучение.Отбор.Занятие.Установить(Выборка.Ссылка);

Продолжение приложения Б

```
НаборЗаписейОбучение.Отбор.ФизЛицо.Установить(ФизЛицо);
НаборЗаписейОбучение.Прочитать(); // прочитать записи по
условиям отбора
    Если НаборЗаписейОбучение.Количество() = 0 Тогда // нет ни
одной записи
        НоваяЗапись = НаборЗаписейОбучение.Добавить(); //
создать новую
        НоваяЗапись.Тема      = Тема;    // заполнить все
измерения

        НоваяЗапись.Занятие = Выборка.Ссылка;
        НоваяЗапись.Порядок = Выборка.Ссылка.Код;
        НоваяЗапись.ФизЛицо = ФизЛицо;
        НоваяЗапись.Период  = ТекущаяДата();
    Иначе
        Сообщить("Ошибка: Вы уже записаны на занятие /" +
Выборка.Ссылка + "/"); // есть как минимум одна такая запись
        Продолжить;
    КонецЕсли;
    НаборЗаписейОбучение.Записать(Истина);
    Сообщить("Запись на занятие /" + Выборка.Ссылка + "/ - ОК");
КонецЦикла;

КонецПроцедуры// ЗаписатьсяНаТему

// создать в регистре сведений Обучение запись по всем занятию Тест
для ФизЛицо
Процедура ЗаписатьсяНаТест(Тест, ФизЛицо) Экспорт
// ищем и записываем через набор записей
НаборЗаписейОбучение =
РегистрыСведений.Обучение.СоздатьНаборЗаписей();
    НаборЗаписейОбучение.Отбор.Занятие.Установить(Тест); // установить
условия отбора
    НаборЗаписейОбучение.Отбор.ФизЛицо.Установить(ФизЛицо);
    НаборЗаписейОбучение.Прочитать(); // считать записи согласно отбора
    Если НаборЗаписейОбучение.Количество() = 0 Тогда // если таких
записей нет
        НоваяЗапись = НаборЗаписейОбучение.Добавить(); // то добавить
        НоваяЗапись.Тема      = Тест.Владелец; // запомнить все
измерения
        НоваяЗапись.Занятие = Тест;
        НоваяЗапись.ФизЛицо = ФизЛицо;
```

Продолжение приложения Б

```
НоваяЗапись.Период = ТекущаяДата();
Иначе
    Сообщить("Ошибка: Вы уже записаны на занятие /" + Тест
+ "/""); // такая запись уже есть
    Возврат;
КонецЕсли;
НаборЗаписейОбучение.Записать(Истина);

КонецПроцедуры

// вернуть из ХранимыйФайл его реквизиты
Процедура ПараметрыХранимыйФайл(ХранимыйФайл, ИмяФайла,
ХФСсылка, Гипер) Экспорт

ИмяФайла = ХранимыйФайл.ИмяФайла;
ХФСсылка = ХранимыйФайл.Ссылка;
Гипер      = ХранимыйФайл.ГиперСсылка;

КонецПроцедуры

// проверить - можно ли еще раз создавать документ для прохождения
ВыбФизЛицомтеста ВыбТест
Функция ТакойУжеЕсть(ВыбФизЛицо, ВыбТест) Экспорт

Если ВыбТест.Пересдача = 0 Тогда // если можно сдавать сколько
угодно раз
    Возврат Ложь;
КонецЕсли;

Запрос = Новый Запрос; // для поиска документов сдачи теста
Запрос.Текст =
"ВЫБРАТЬ
|     Тестирование.ФизЛицо КАК ФизЛицо,
|     Тестирование.Тест КАК Тест
|ИЗ
|     Документ.Тестирование КАК Тестирование
|ГДЕ
|     Тестирование.ФизЛицо = &ВыбФизЛицо
|     И Тестирование.Тест = &ВыбТест
|";
Запрос.УстановитьПараметр("ВыбФизЛицо", ВыбФизЛицо);
Запрос.УстановитьПараметр("ВыбТест", ВыбТест);
```

Продолжение приложения Б

```
Результат = Запрос.Выполнить();  
Выборка = Результат.Выбрать();  
Возврат Выборка.Количество() >= ВыбТест.Пересдача; // если задано  
количество возможных сдач
```

```
КонецФункции //ТакойУжеЕсть
```

```
Конфигурация. Модуль сеанса
```

```
// при начале сеанса - проверить пользователя
```

```
Процедура УстановкаПараметровСеанса(ТребуемыеПараметры)
```

```
// попытка найти пользователя в справочнике ФизическиеЛица
```

```
Пользователь =
```

```
Справочники.ФизическиеЛица.НайтиПоКоду(ИмяПользователя());
```

```
Если Пользователь.Пустая() Тогда // этого пользователя нет в  
спрвочнике Физ лиц - значит он входит первый раз
```

```
Пользователь =
```

```
Справочники.ФизическиеЛица.СоздатьЭлемент(); // создать новую запись
```

```
Пользователь.Код = ИмяПользователя();
```

```
Пользователь.Наименование = ИмяПользователя();
```

```
Пользователь.Записать();
```

```
Сообщить("Пользователь зарегистрирован в справочнике  
физических лиц");
```

```
КонецЕсли;
```

```
// вошедший (или созданный) пользователь (запись справочника  
ФизЛица) - текущий пользователь
```

```
ПараметрыСеанса.глПользователь = Пользователь.Ссылка;
```

```
КонецПроцедуры
```

Приложение В

Текст программных модулей документов и регистров

Документы.Тестирование.Формы.Заметка.Модуль формы

```
// форма ввода заметки студентом
// при открытии формы заполнить из переданных извне Параметров
нужные реквизиты этой формы
&НаСервере
Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

Если Параметры.Свойство("Ответ") Тогда
    Ответ = Параметры.Ответ;
КонецЕсли;
Если Параметры.Свойство("Заметка") Тогда
    Заметка = Параметры.Заметка;
КонецЕсли;
КонецПроцедуры
// при нажатии кнопки Сохранить - послать оповещение в вызывающую
форму
&НаКлиенте
Процедура СохранитьЗаметку(Команда)
    Закрыть(Заметка); // послать оповещение и передать параметр
КонецПроцедуры
```

Документы.ТИестиоование.Формы.Проверка.Модуль формы

```
// форма вызывается при нажатии кнопки Проверить в форме документа
Тестирование
// позволяет преподавателю прикрепить свой файл и заметку к ответу
студента. Также показывает файл и заметку студента
// заполнить переданные параметры из вызывающей формы в реквизиты
этой формы
&НаСервере
Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)
    Если Параметры.Свойство("Ответ") Тогда
        Ответ = Параметры.Ответ;
    КонецЕсли;
Если Параметры.Свойство("Балл") Тогда
    Балл = Параметры.Балл;
```

Продолжение приложения В

```
КонецЕсли;
Если Параметры.Свойство("Файл") Тогда
    Файл = Параметры.Файл;
КонецЕсли;
Если Параметры.Свойство("Заметка") Тогда
    Заметка = Параметры.Заметка;
КонецЕсли;
Если Параметры.Свойство("ФайлП") Тогда
    ФайлП = Параметры.ФайлП;
КонецЕсли;
Если Параметры.Свойство("ЗаметкаП") Тогда
    ЗаметкаП = Параметры.ЗаметкаП;
КонецЕсли;
```

```
// доступность элементов
```

```
Элементы.Балл.ТолькоПросмотр      = РольДоступна("Тестер");
Элементы.ЗаметкаП.ТолькоПросмотр  = РольДоступна("Тестер");
Элементы.ФайлП.ТолькоПросмотр     = РольДоступна("Тестер");
```

```
КонецПроцедуры
```

```
// при нажати на кнопку Сохранить - записать результаты в структуру и
послать оповещение с этой структурой в вызывающую форму
```

```
&НаКлиенте
```

```
Процедура Сохранить(Команда)
```

```
    Результат = Новый Структура;
```

```
    Результат.Вставить("Балл", Балл);
```

```
    Результат.Вставить("ФайлП", ФайлП);
```

```
    Результат.Вставить("ЗаметкаП", ЗаметкаП);
```

```
    Закрыть(Результат); // послать оповещение
```

```
КонецПроцедуры
```

```
// при начале выбора файла преподавателя - открыть форму выбора
```

```
Спр. Хранимые файлы
```

```
&НаКлиенте
```

```
Процедура ФайлПНачалоВыбора(Элемент, ДанныеВыбора,
СтандартнаяОбработка)
```

```
    СтандартнаяОбработка = Ложь;
```

```
    ПараметрыФормы = Новый Структура("РежимВыбора,
ЗакрыватьПриВыборе", Истина, Истина);
```

Продолжение приложения В

```
    П = Новый Структура("Владелец",  
ОбщийСрв.ПолучитьГлПользователя()); // текущего пользователя передаем  
через параметр
```

```
    ПараметрыФормы.Вставить("Отбор", П);
```

```
    // после окончания выбора оповещение придет в функцию  
ВыбратьФайл
```

```
    ОбработкаВыбора = Новый ОписаниеОповещения("ВыбратьФайл",  
ЭтаФорма);
```

```
    ОткрытьФорму("Справочник.ХранимыеФайлы.ФормаВыбора",  
ПараметрыФормы,  
ЭтаФорма, , , ОбработкаВыбора);
```

```
КонецПроцедуры
```

```
    // сюда приходит оповещение при выборе файла в форме выбора Спр  
Хранимые файлы
```

```
    &НаКлиенте
```

```
Процедура ВыбратьФайл(Значение, ДопПараметры) Экспорт
```

```
Если Значение = Неопределено Тогда
```

```
    Возврат;
```

```
КонецЕсли;
```

```
ФайлП = Значение; // поместить выбранный файл в соответствующий  
реквизит формы
```

```
КонецПроцедуры
```

```
Документы.Тестирование.Формы.Форма документа. Модуль формы
```

```
    // при создании формы - установить начальные параметры
```

```
    &НаСервере
```

```
Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)
```

```
    // если форма вызвана из внешней формы (элемента или списка) Спр  
Тесты - то передаем параметр Тест
```

```
Если Параметры.Свойство("Ключ") Тогда
```

```
    Если Параметры.Ключ.Пустая() Тогда // если это новый  
документ
```

```
        Если Параметры.Свойство("Тест") Тогда // если передан  
параметр Тест
```


Продолжение приложения В

Если Не Параметры.Тест.Пустая() Тогда

```
        Объект.Тест = Параметры.Тест; // то
записываем его в реквизит документа
        КонецЕсли;
        КонецЕсли;
        КонецЕсли;
        КонецЕсли;

        Элементы.ТекущийВопросКартинка.РазмерКартинки =
РазмерКартинки.АвтоРазмер;

        // видимость и доступность элементов управления в зависимости от
настроек теста
        Элементы.ОтветыБалл.Видимость = Объект.Проведен И
Объект.Тест.РежимОбучения;
        Элементы.Проверка.Видимость = Объект.Проведен И
Объект.Тест.ТребуетсяРучнаяПроверка;
        Элементы.ОтветыБалл.ТолькоПросмотр = РольДоступна("Тестер");
        Элементы.КомментарийП.ТолькоПросмотр = РольДоступна("Тестер");

        ТекущийВопросНомер = 1; // номер текущего вопроса

        КонецПроцедуры

// при открытии формы
&НаКлиенте
Процедура ПриОткрытии(Отказ)

// проверка на повторную сдачу данного теста
Если Объект.Ссылка.Пустая() Тогда // если новый документ
        Если ОбщийСрв.ТакойУжеЕсть(Объект.ФизЛицо, Объект.Тест)
Тогда // проверяет количество допустимых персдач
                Отказ = Истина;
                Сообщить("Данный тест пересдавать нельзя");
                Возврат;
        КонецЕсли;
        КонецЕсли;

        Если Есть Таймер(Объект.Тест) Тогда // если задано ограничение
времени для теста
```

Продолжение приложения В

```
ПодключитьОбработчикОжидания("СработалТаймер",  
ОбщийСрв.РеквизитТеста(Объект.Тест, "Таймер")*60); //подключить таймер  
КонецЕсли;
```

```
УстановитьФлагПроверкиП(); // выставить реквизит Проверка  
преподавателем
```

```
// вызвать процедуру как будто при ручном выборе теста  
ТестПриИзменении(Неопределено);
```

```
КонецПроцедуры
```

```
// проверка необходимости включения таймера  
&НаСервере  
Функция ЕстьТаймер(ВыбТест)
```

```
Возврат Не ВыбТест.Пустая() И РольДоступна("Тестер") И Не  
Объект.Проведен  
И ВыбТест.Таймер > 0;
```

```
КонецФункции
```

```
// при срабатывании таймера  
&НаКлиенте  
Процедура СработалТаймер()
```

```
ОтключитьОбработчикОжидания("СработалТаймер"); // отключить  
таймер  
Сообщить("Время истекло");  
ПараметрыЗаписи = Новый Структура;  
ПараметрыЗаписи.Вставить("РежимЗаписи",  
РежимЗаписиДокумента.Проведение);  
Записать(ПараметрыЗаписи); // записать документ с проведением - тест  
завершен  
ОбновитьОтображениеДанных();
```

```
КонецПроцедуры
```

```
// установить реквизит Проверка преподавателем  
Процедура УстановитьФлагПроверкиП()
```

```
Элементы.ПроверкаП.Видимость = Объект.Проведен;
```

Продолжение приложения В

```
// выставляем значение этого реквизита также как в рег сведений
Обучение
  Запрос = Новый Запрос;
  Запрос.Текст =
  "ВЫБРАТЬ
  |   ОбучениеСрезПоследних.Занятие КАК Занятие,
  |   ОбучениеСрезПоследних.ФизЛицо КАК ФизЛицо,
  |   ОбучениеСрезПоследних.Документ КАК Документ,
  |   ОбучениеСрезПоследних.Проверка КАК Проверка
  |ИЗ
  |   РегистрСведений.Обучение.СрезПоследних КАК
ОбучениеСрезПоследних
  |ГДЕ
  |   ОбучениеСрезПоследних.Документ = &ВыбДДокумент
  ";

  Запрос.УстановитьПараметр("ВыбДДокумент", Объект.Ссылка);
  Результат = Запрос.Выполнить();
  Выборка = Результат.Выбрать();
  Пока Выборка.Следующий() Цикл
      ПроверкаП = Выборка.Проверка;
      Возврат;
  КонецЦикла;

  ПроверкаП = 0;

  КонецПроцедуры

// по команде Следующий - перейти на следующий вопрос
&НаКлиенте
Процедура Следующий(Команда)

  СохранитьОтветНаТекущийВопрос(); // записать ответ на текущий
вопрос в таб. часть ДанныеОтветы

  ТекущийВопросНомер = ТекущийВопросНомер + 1; // номер
следующего вопроса
  Если ТекущийВопросНомер = Вопросы.Количество()+1 Тогда
      ТекущийВопросНомер = 1;
  КонецЕсли;
  СчетчикВопросов(); // отобразить счетчик вопросов
```

Продолжение приложения В

ЗаполнитьТекущийВопрос();// заполнить текст вопроса, варианты ответов и картинку вопроса

КонецПроцедуры

// по команде Предыдущий - перейти на следующий вопрос
&НаКлиенте
Процедура Предыдущий(Команда)

СохранитьОтветНаТекущийВопрос();// записать ответ на текущий вопрос в таб. часть ДанныеОтветы

ТекущийВопросНомер = ТекущийВопросНомер - 1; // номер предыдущего вопроса

Если ТекущийВопросНомер = 0 Тогда
 ТекущийВопросНомер = Вопросы.Количество();

КонецЕсли;

СчетчикВопросов();// отобразить счетчик вопросов

ЗаполнитьТекущийВопрос();// заполнить текст вопроса, варианты ответов и картинку вопроса

КонецПроцедуры

// записать ответ на текущий вопрос в таб. часть ДанныеОтветы
&НаКлиенте
Процедура СохранитьОтветНаТекущийВопрос()

УдалитьОтветыНаВопрос(ТекущийВопрос);// удалить все ранее данные ответы на текущий вопрос

Для Каждого ТекСтрока Из Объект.Ответы Цикл // цикл по таб части Ответы

 Если ТекСтрока.Отметка Тогда // если галочка стоит
 // сохранить текущий ответ вместе с файлом и

заметкой в таб часть ДанныеОтветы

 СохранитьОтвет(ТекущийВопрос, ТекСтрока.Ответ,
ТекСтрока.Балл, ТекСтрока.Файл, ТекСтрока.Заметка, ТекСтрока.ФайлП,
ТекСтрока.ЗаметкаП);

 КонецЕсли;

КонецЦикла;

КонецПроцедуры

Продолжение приложения В

```
// удалить из таб. части Данные ответы все записи на вопрос Вопрос
&НаСервере
Процедура УдалитьОтветыНаВопрос(Вопрос)
```

```
СтруктураОтбора = Новый Структура("Вопрос", Вопрос); // параметр
отбора
```

```
МассивУдаляемыхСтрок =
Объект.ДанныеОтветы.НайтиСтроки(СтруктураОтбора); // сформировать
массив
```

```
Для Каждого ЭлементМассива Из МассивУдаляемыхСтрок Цикл //
цикл по всем элементам массива
```

```
Объект.ДанныеОтветы.Удалить(ЭлементМассива); //
удалить текущий элемент массива из таб. части ДанныеОтветы
КонецЦикла;
```

КонецПроцедуры

```
// сохранить Ответ на Вопрос в таб. части ДанныеОтветы
&НаКлиенте
```

```
Процедура СохранитьОтвет(Вопрос, Ответ, Балл, Файл, Заметка,
ФайлП, ЗаметкаП)
```

```
НоваяСтрока = Объект.ДанныеОтветы.Добавить(); // создать строку в
ТЧ
```

```
НоваяСтрока.Вопрос = Вопрос;
```

```
НоваяСтрока.Ответ = Ответ;
```

```
НоваяСтрока.Балл = Балл;
```

```
НоваяСтрока.Файл = Файл;
```

```
НоваяСтрока.Заметка = Заметка;
```

```
НоваяСтрока.ФайлП = ФайлП;
```

```
НоваяСтрока.ЗаметкаП = ЗаметкаП;
```

КонецПроцедуры

```
// заполнить на форме все параметры текущего вопроса теста
&НаКлиенте
```

```
Процедура ЗаполнитьТекущийВопрос()
```

```
ТекущийВопрос = ПолучитьТекущийВопрос(); // получить ссылку на
текущий вопрос теста
```

Продолжение приложения В

```
СЗОтветов =
СформироватьСписокОтветовТекущегоВопроса(ТекущийВопрос); //
сформировать СЗ ответов
    Объект.Ответы.Очистить(); // очистить ТЧ Ответы
    Для Каждого ТекСтрока Из СЗОтветов Цикл // цикл по списку
значений
        НоваяСтрока = Объект.Ответы.Добавить(); // создать строку в ТЧ
        НоваяСтрока.Ответ = ТекСтрока.Значение;
        НоваяСтрока.Отметка =
НайтиВДанныхОтветах(ТекущийВопрос, ТекСтрока.Значение, // заполнить
данный ранее пользователем ответ

НоваяСтрока.Файл, НоваяСтрока.Заметка,

НоваяСтрока.ФайлП, НоваяСтрока.ЗаметкаП);
        НоваяСтрока.Балл =
НайтиБаллВДанныхОтветах(ТекущийВопрос, ТекСтрока.Значение); //
заполнить балл ранее данного ответа
        КонецЦикла;

// заполнить картинку вопроса
ТекущийВопросКартинка = Неопределено;
КартинкаФайл = ПолучитьКартинкуВопроса(ТекущийВопрос);
Если Не КартинкаФайл = Неопределено Тогда
    ТекущийВопросКартинка =
ПолучитьНавигационнуюСсылку(КартинкаФайл, "ДанныеФайла");
    КонецЕсли;

// видимость и доступность элементов в зависимости от реквизитов
вопроса
    Элементы.ОтветыОтметка.ТолькоПросмотр =
ОбщийСрв.РеквизитВопроса(ТекущийВопрос, "ТребуетсяФайл")
Или
ОбщийСрв.РеквизитВопроса(ТекущийВопрос,
"ТребуетсяРазвернутыйОтвет");;

    Элементы.ОтветыФайл.Видимость =
ОбщийСрв.РеквизитВопроса(ТекущийВопрос, "ТребуетсяФайл");
    Элементы.ОтветыЗаметка.Видимость =
ОбщийСрв.РеквизитВопроса(ТекущийВопрос,
"ТребуетсяРазвернутыйОтвет");;
```

Продолжение приложения В

КонецПроцедуры

```
// получить из Спр ХранимыеФайлы ссылку на запись с файлом,  
прикрепленном к Вопросу, у которой стоит реквизит Основной  
&НаСервере  
Функция ПолучитьКартинкуВопроса(Вопрос)
```

```
Выборка = Справочники.ХранимыеФайлы.Выбрать(, Вопрос); //  
выбрать все записи из Спр ХранимыеФайлы, подчиненные Вопросу  
Пока Выборка.Следующий() Цикл // по всем записям выборки  
Если Выборка.Основной Тогда // если стоит галочка  
Основной
```

```
Возврат Выборка.Ссылка; // то это искомый файл  
КонецЕсли;  
КонецЦикла;
```

```
Возврат Неопределено; // нет файлов или ни один из них не Основной  
КонецФункции
```

```
// найти в ТЧ ДанныеОтветы запись с реквизитами Впорос и Ответ  
&НаКлиенте  
Функция НайтиВДанныхОтветах(Вопрос, Знач Ответ, Файл, Заметка,  
ФайлП, ЗаметкаП)
```

```
Для Каждого ТекСтрока Из Объект.ДанныеОтветы Цикл // по всем  
записям ТЧ ДанныеОтветы  
Если (ТекСтрока.Вопрос = Вопрос) И (ТекСтрока.Ответ = Ответ)  
Тогда // если оба реквизита равны искомым  
Файл = ТекСтрока.Файл; // то заполняем  
параметры, перданные в функцию
```

```
Заметка = ТекСтрока.Заметка;  
ФайлП = ТекСтрока.ФайлП;  
ЗаметкаП = ТекСтрока.ЗаметкаП;  
Возврат Истина; // значит нашли
```

```
КонецЕсли;  
КонецЦикла;
```

```
// не нашли  
Файл = Неопределено;  
Заметка = Неопределено;  
ФайлП = Неопределено;  
ЗаметкаП = Неопределено;
```

Продолжение приложения В

Возврат Ложь;

КонецФункции

// найти в ТЧ ДанныеОтветы запись с реквизитами Вопрос и Ответ и вернуть Балл

Функция НайтиБаллВДанныхОтветах(Вопрос, Ответ)

Для Каждого ТекСтрока Из Объект.ДанныеОтветы Цикл // по всем записям ТЧ ДанныеОтветы

Если (ТекСтрока.Вопрос = Вопрос) И (ТекСтрока.Ответ = Ответ) Тогда // если оба реквизита равны искомым

Возврат ТекСтрока.Балл; // значит нашли

КонецЕсли;

КонецЦикла;

Возврат Ответ.Вес; // если не нашли - то вес ответа по-умолчанию

КонецФункции

// при попытке выбора нового теста - проверить возможность персдачи &НаКлиенте

Процедура ТестОбработкаВыбора(Элемент, ВыбранноеЗначение, СтандартнаяОбработка)

// проверить возможность персдачи

Если ОбщийСрв.ТакойУжеЕсть(Объект.ФизЛицо, ВыбранноеЗначение) Тогда // проверить кол-во уже сданных и возможность новой сдачи

Сообщить("Данный тест пересдавать нельзя");

СтандартнаяОбработка = Ложь;

Возврат; // нельзя больше персдавать

КонецЕсли;

// если в выбранном тесте есть ограничение по времени

Если ЕстьТаймер(ВыбранноеЗначение) Тогда //то включить таймер
ПодключитьОбработчикОжидания("СработалТаймер",
ОбщийСрв.РеквизитТеста(Объект.Тест, "Таймер")*60);;

КонецЕсли;

КонецПроцедуры

Продолжение приложения В

```
// при выборе нового теста (выбор уже свершился)
&НаКлиенте
Процедура ТестПриИзменении(Элемент)

Если Не Объект.Тест.Пустая() Тогда // если тест выбран
    Объект.Тема = ОбщийСрв.ПолучитьТемуТеста(Объект.Тест);
    ТекущийВопросНомер = 1; // то начинаем его с вопроса 1
    СформироватьСписокВопросов(); // сформировать СЗ вопросов
текущего теста
    СчетчикВопросов(); // выставить счетчик вопросов на форме
    ЗаполнитьТекущийВопрос(); // отобразить параметры текущего
вопроса
    КонецЕсли;

КонецПроцедуры

// выставить счетчик вопросов на форме
&НаКлиенте
Процедура СчетчикВопросов()

    // заголовок группы = номер текущего вопроса/ из общего количества
вопросов теста
    Элементы.ГруппаТекущийВопрос.Заголовок = "Вопрос " +
Строка(ТекущийВопросНомер) + "/" +
        Строка(Вопросы.Количество());

КонецПроцедуры

// сформировать СЗ Вопросы текущего теста (Вопросы - реквизит
формы)
&НаСервере
Процедура СформироватьСписокВопросов()

    Вопросы.Очистить(); // очистить СЗ
    Выборка = Справочники.Вопросы.Выбрать(, Объект.Тест, "Код
Возр"); // выбрать все вопросы текущего теста по возрастанию кода
    Пока Выборка.Следующий() Цикл // по всем вопросам выборки
        Вопросы.Добавить(Выборка.Ссылка); // включить вопрос в СЗ
Вопросы
    КонецЦикла;

КонецПроцедуры
```

Продолжение приложения В

```
// сформировать СЗ всех ответов на ТекущийВопрос
&НаСервере
Функция
СформироватьСписокОтветовТекущегоВопроса(ТекущийВопрос)

    СЗ = Новый СписокЗначений;    // СЗ, который возвращаем

    Выборка = Справочники.Ответы.Выбрать(, ТекущийВопрос,, "Код
    Возр"); // выборка ответов на ТекущийВопрос по возрастанию кодов
    Пока Выборка.Следующий()Цикл // по всем ответам выборки
        СЗ.Добавить(Выборка.Ссылка); // добавить в результирующий
список
    КонецЦикла;

    Возврат СЗ; // возвращаем сформированный список значений

КонецФункции

// по номеру ТекущийВопросНомер (реквизит формы) вернуть запись
из СЗ Вопросы (реквизит формы)
&НаКлиенте
Функция ПолучитьТекущийВопрос()

    Если Вопросы.Количество() Тогда // если в СЗ есть хоть что-то
        Возврат Вопросы[ТекущийВопросНомер-1].Значение; // то
вернуть значение СЗ по индексу
    КонецЕсли;

    Возврат Неопределено; // если в тесте нет ни одного вопроса
КонецФункции

// при проставлении галочки в списке ответов текущего вопроса
// если не допускается несколько вариантов ответа на текущий вопрос -
то стереть все остальные галочки
&НаКлиенте
Процедура ОтветыОтметкаПриИзменении(Элемент)

    // если допускается несколько вариантов
    Если ОбщийСрв.РеквизитВопроса(ТекущийВопрос,
"НесколькоПравильных") Тогда
        Возврат; // то ничего не делать
    КонецЕсли;
```

Продолжение приложения В

// если допускается только один ответ - то нужно удалить все
остальные ответы и оставить только этот

// запомнить текущие значения

ЗапомнитьОтметку = Элементы.Ответы.ТекущиеДанные.Отметка;

ЗапомнитьФайл = Элементы.Ответы.ТекущиеДанные.Файл;

ЗапомнитьЗаметку = Элементы.Ответы.ТекущиеДанные.Заметка;

// погасить галочки и файлы у всех ответов

Для Каждого ТекСтрока Из Объект.Ответы Цикл

ТекСтрока.Отметка = Ложь;

ТекСтрока.Файл = "";

ТекСтрока.Заметка = "";

КонецЦикла;

// у текущего ответа восстановить значения (так как они были
погашены вместе со всеми

Элементы.Ответы.ТекущиеДанные.Отметка = ЗапомнитьОтметку;

Элементы.Ответы.ТекущиеДанные.Файл = ЗапомнитьФайл;

Элементы.Ответы.ТекущиеДанные.Заметка = ЗапомнитьЗаметку;

КонецПроцедуры

// перед записью документа тоже сохранить ответ на текущий вопрос
&НаКлиенте

Процедура ПередЗаписью(Отказ, ПараметрыЗаписи)

СохранитьОтветНаТекущийВопрос();

КонецПроцедуры

// при редактировании/вводе/удалении файла студента

&НаКлиенте

Процедура ОтветыФайлПриИзменении(Элемент)

// ставить отметку (галочку) ответа на вопрос или нет

Элементы.Ответы.ТекущиеДанные.Отметка = СтавитьОтметку();

КонецПроцедуры

Продолжение приложения В

```
// начало выбора файла студента - открыть форму выбора Спр.  
ХранимыеФайлы  
  &НаКлиенте  
  Процедура ОтветыФайлНачалоВыбора(Элемент, ДанныеВыбора,  
  СтандартнаяОбработка)  
  
  СтандартнаяОбработка = Ложь;  
  
  ПараметрыФормы = Новый Структура("РежимВыбора,  
  ЗакрыватьПриВыборе", Истина, Истина);  
  П = Новый Структура("Владелец", Объект.ФизЛицо); // текущего  
  студента передаем через параметр - он будет Владельцем  
  ПараметрыФормы.Вставить("Отбор", П);  
  
  ОбработкаВыбора = Новый ОписаниеОповещения("ВыбратьФайл",  
  ЭтаФорма); // при закрытии той формы вызов придет в ВыбратьФайл  
  
  ОткрытьФорму("Справочник.ХранимыеФайлы.ФормаВыбора",  
  ПараметрыФормы, // открыть ту форму  
  ЭтаФорма, , , , ОбработкаВыбора);  
  КонецПроцедуры  
  
  // при создании нового файла студента (нажали кнопку Новый) -  
  открыть форму элемента Спр. ХранимыеФайлы как вновь создаваемого  
  файла  
  &НаКлиенте  
  Процедура ОтветыФайлСоздание(Элемент, СтандартнаяОбработка)  
  
  СтандартнаяОбработка = Ложь;  
  
  ПараметрыФормы = Новый  
  Структура("ЗакрыватьПриЗакрытииВладельца", Истина);  
  ПараметрыФормы.Вставить("Владелец", Объект.ФизЛицо);  
  
  ОбработкаВыбора = Новый ОписаниеОповещения("ВыбратьФайл",  
  ЭтаФорма); // при закрытии той формы оповещение придет в  
  
  ОткрытьФорму("Справочник.ХранимыеФайлы.ФормаОбъекта",  
  ПараметрыФормы, // открытие той формы  
  ЭтаФорма, , , , ОбработкаВыбора);  
  
  КонецПроцедуры
```

Продолжение приложения В

```
// при удалении файла студента (нажали кнопку Удалить)
&НаКлиенте
Процедура ОтветыФайлОчистка(Элемент, СтандартнаяОбработка)

    Элементы.Ответы.ТекущиеДанные.Отметка = СтавитьОтметку();//
    ставить или не ставить отметку
    КонецПроцедуры

// эта функция вызывается из открываемых форм Спр ХранимыеФайлы
при выполнении в них выбора либо при закрытии
&НаКлиенте
Процедура ВыбратьФайл(Значение, ДопПараметры) Экспорт

    Если Значение = Неопределено Тогда // если ничего не выбрали
        Возврат;
    КонецЕсли;

// записываем выбранный файл в поля текущей записи таб. Части
Ответы
    Элементы.Ответы.ТекущиеДанные.Файл = Значение;
    Элементы.Ответы.ТекущиеДанные.Отметка = СтавитьОтметку();//
    ставить или не ставить галочку
    ОтветыОтметкаПриИзменении(Значение); // те же действия что и при
    ручном проставлении отметки
    ЭтотОбъект.Модифицированность = Истина;

    КонецПроцедуры

// по команде (кнопке) Проверка - открыть спец форму Проверка этого
документа
&НаКлиенте
Процедура Проверить(Команда)

    // все нужные парметры передаем в вызываемую форму через
    Параметры
    ПараметрыФормы = Новый
    Структура("ЗакрыватьПриЗакрытииВладельца", Истина);
    ПараметрыФормы.Вставить("Ответ",
    Элементы.Ответы.ТекущиеДанные.Ответ);
    ПараметрыФормы.Вставить("Балл",
    Элементы.Ответы.ТекущиеДанные.Балл);
```

Продолжение приложения В

```
ПараметрыФормы.Вставить("Файл",  
Элементы.Ответы.ТекущиеДанные.Файл);  
ПараметрыФормы.Вставить("Заметка",  
Элементы.Ответы.ТекущиеДанные.Заметка);  
ПараметрыФормы.Вставить("ФайлП",  
Элементы.Ответы.ТекущиеДанные.ФайлП);  
ПараметрыФормы.Вставить("ЗаметкаП",  
Элементы.Ответы.ТекущиеДанные.ЗаметкаП);
```

```
ОбработкаПроверки = Новый  
ОписаниеОповещения("ВыполнитьПроверку", ЭтаФорма); // при закрытии  
оповещение придет в функцию ВыполнитьПроверку
```

```
ОткрытьФорму("Документ.Тестирование.Форма.Проверка",  
ПараметрыФормы, ЭтаФорма, , , ОбработкаПроверки);
```

КонецПроцедуры

```
// при закрытии формы Проверка в эту функцию придет оповещение  
&НаКлиенте  
Процедура ВыполнитьПроверку(Значение, ДопПараметры) Экспорт
```

```
Если Значение = Неопределено Тогда // если ничего не выбрано  
    Возврат;  
КонецЕсли;
```

```
// записать перданные из той формы параметры в поля текущец записи  
таб. части ОТветы
```

```
Элементы.Ответы.ТекущиеДанные.Балл = Значение.Балл;  
Элементы.Ответы.ТекущиеДанные.ФайлП = Значение.ФайлП;  
Элементы.Ответы.ТекущиеДанные.ЗаметкаП = Значение.ЗаметкаП;  
ЭтотОбъект.Модифицированность = Истина;
```

КонецПроцедуры

```
// при вводе текста в поле Заметка студентом  
&НаКлиенте  
Процедура ОтветыЗаметкаПриИзменении(Элемент)
```

```
Элементы.Ответы.ТекущиеДанные.Отметка = СтавитьОтметку(); //  
ставить или не ставить галочку
```

Продолжение приложения В

ОтветыОтметкаПриИзменении(Элемент); // как если бы галочку поставили вручную

КонецПроцедуры

// нажали кнопку Открытие - открыть форму Заметка этого документа
&НаКлиенте

Процедура ОтветыЗаметкаОткрытие(Элемент, СтандартнаяОбработка)

СтандартнаяОбработка = Ложь;

// все нужные параметры передаем через Параметры

ПараметрыФормы = Новый

Структура("ЗакрыватьПриЗакрытииВладельца", Истина);

ПараметрыФормы.Вставить("Ответ",

Элементы.Ответы.ТекущиеДанные.Ответ);

ПараметрыФормы.Вставить("Заметка",

Элементы.Ответы.ТекущиеДанные.Заметка);

// после закрытия той формы оповещение придет в функцию

НаписатьЗаметку

ОбработкаПроверки = Новый

ОписаниеОповещения("НаписатьЗаметку", ЭтаФорма);

ОткрытьФорму("Документ.Тестирование.Форма.Заметка",
ПараметрыФормы, ЭтаФорма, , , ОбработкаПроверки);

КонецПроцедуры

// обработчик оповещения при закрытии формы Заметка

&НаКлиенте

Процедура НаписатьЗаметку(Значение, ДопПараметры) Экспорт

Если Значение = Неопределено Тогда // если ничего не выбрано

Возврат;

КонецЕсли;

// записываем набранный текст в поле Заметка текуще записи таб.
части Ответы

Элементы.Ответы.ТекущиеДанные.Заметка = Значение;

Элементы.Ответы.ТекущиеДанные.Отметка = СтавитьОтметку(); //

ставить или не ставить галочку

ОтветыОтметкаПриИзменении(Значение); // как если бы поставили
галочку вручную

Продолжение приложения В

ЭтотОбъект.Модифицированность = Истина;

КонецПроцедуры

// при удалении заметки студента

&НаКлиенте

Процедура ОтветыЗаметкаОчистка(Элемент, СтандартнаяОбработка)

Элементы.Ответы.ТекущиеДанные.Отметка = СтавитьОтметку();

КонецПроцедуры

// ставить или не ставить галочку в поле Отметка ответа

&НаКлиенте

Функция СтавитьОтметку()

Возврат Не(Элементы.Ответы.ТекущиеДанные.Файл.Пустая())

Или (Не

ПустаяСтрока(Элементы.Ответы.ТекущиеДанные.Заметка));

КонецФункции

// запрет удаления строк из таб. части ОТветы

&НаКлиенте

Процедура ОтветыПередУдалением(Элемент, Отказ)

Отказ = Истина;

КонецПроцедуры

// щелчок на гиперссылке файла студента - открытие файла

// щелчок в другом месте - вызов команды (кнопки) Проверить

&НаКлиенте

Процедура ОтветыВыбор(Элемент, ВыбраннаяСтрока, Поле,
СтандартнаяОбработка)

Если Объект.Проведен Тогда

Если Поле.Имя = "ОтветыФайл" Тогда // если в поле Файл

СтандартнаяОбработка = Ложь;

ОбщийКлиент.ОткрытьХранимыйФайл(Элемент.ТекущиеДанные.Фай
л); // открыть связанный файл

Возврат;

КонецЕсли;

КонецЕсли;

Продолжение приложения В

// если в любом другом месте - то тоже самое, что при нажатии на кнопку Проверить

Если Объект.Проведен И ОбщийСрв.РеквизитТеста(Объект.Тест, "ТребуетсяРучнаяПроверка") Тогда

 Проверить(Элемент);

КонецЕсли;

КонецПроцедуры

// после закрытия документа - обновить все формы, которые с ним связаны

&НаКлиенте

Процедура ПослеЗаписи(ПараметрыЗаписи)

Оповестить("ПланОбучения", ,ЭтотОбъект); // всем формам послать оповещение о проведенном изменении

КонецПроцедуры

// при закрытии формы отключить таймер, чтобы он не кртился

&НаКлиенте

Процедура ПередЗакрытием(Отказ, ЗавершениеРаботы, ТекстПредупреждения, СтандартнаяОбработка)

ОтключитьОбработчикОжидания("СработалТаймер");

КонецПроцедуры

Документы.Тестирование.Формы.Форма списка.Модуль

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

Если РольДоступна("Тестер") Тогда

 ЭлементОтбора =

Список.Отбор.Элементы.Добавить(Тип("ЭлементОтбораКомпоновкиДанных"));

 ЭлементОтбора.ЛевоеЗначение = Новый

ПолеКомпоновкиДанных("ФизЛицо");

 ЭлементОтбора.ВидСравнения =

ВидСравненияКомпоновкиДанных.Равно;

 ЭлементОтбора.Использование = Истина;

Продолжение приложения В

```
ЭлементОтбора.РежимОтображения =
РежимОтображенияЭлементаНастройкиКомпоновкиДанных.Обычный;
    ЭлементОтбора.ПравоеЗначение =
ПараметрыСеанса.глПользователь;
    КонецЕсли;

КонецПроцедуры

Документы.Тестирование.Модуль объекта

// при вводе нового документа
Процедура ОбработкаЗаполнения(ДанныеЗаполнения,
СтандартнаяОбработка)

// если вводится на основании теста
Если ТипЗнч(ДанныеЗаполнения) = Тип("СправочникСсылка.Тесты")
Тогда
    Тест = ДанныеЗаполнения.Ссылка;
    Тема = Тест.Владелец;
КонецЕсли;

// тестируемый - пользователь, вошедший в систему
ФизЛицо = ПараметрыСеанса.глПользователь;

КонецПроцедуры

// расчет итогового балла
Процедура ПередЗаписью(Отказ, РежимЗаписи, РежимПроведения)

Кол = 0;           // кол-во вопросов теста
Балл = 0;         // итоговый балл
Выборка = Справочники.Вопросы.Выбрать(, Тест); // по всем вопросам
теста
Пока Выборка.Следующий()Цикл
    ВопросТеста = Выборка.Ссылка; // текущий вопрос теста

    ЕстьОтветНаВопросТеста = Ложь; // признак того, что на
текущий вопрос теста вообще дан хоть какой-то ответ
    БаллЗаВопросТеста = 0;           // балл за текущий
вопрос
```

Продолжение приложения В

```
Для Каждого ТекДанныйОтвет Из ДанныеОтветы Цикл
// по всем данным ответам
    Если ТекДанныйОтвет.Вопрос <> ВопросТеста Тогда//
ищем данные ответы на текущий вопрос теста
        Продолжить;
        КонецЕсли;
        ЕстьОтветНаВопросТеста = Истина; // хоть какой-то
ответ был дан
        БаллЗаВопросТеста = БаллЗаВопросТеста +
ПолучитьВесОтвета(ТекДанныйОтвет); // суммируем все ответы на
текущий вопрос теста
        КонецЦикла;

    Если Не ЕстьОтветНаВопросТеста Тогда// если никакой ответ не
был дан
        Если Тест.ВсеВопросыОбязательно Тогда // если в
настройках теста стоит что на все вопросы дать ответ обязательно
            БаллЗаВопросТеста = -1; // то за этот вопрос
минус 1 балл (иначе ноль баллов)
            КонецЕсли;
        КонецЕсли;

        Балл = Балл + БаллЗаВопросТеста; // итоговый балл
        Кол = Кол+1; // кол-во вопросов
        КонецЦикла;
    Если Не Кол = 0 Тогда
        Балл = Балл/Кол; // средний балл
    КонецЕсли;

КонецПроцедуры

// получить вес данного ответа в зависимости от условий
Функция ПолучитьВесОтвета(ТекДанныйОтвет)

    Если ТекДанныйОтвет.Балл <=0 Тогда // если вес отрицательный -
то возвращаем его без вариантов
        Возврат ТекДанныйОтвет.Балл;
    КонецЕсли;

// если в настройках вопроса задано что можно дать несколько
правильных ответов
```

Продолжение приложения В

// и при этом суммировать веса правильных ответов нужно только если перечислены все правильные

Если ТекДанныйОтвет.Вопрос.НесколькоПравильных И
ТекДанныйОтвет.Вопрос.НужныВсеПравильные Тогда

Если ВсеПравильныеПрисутствуют(ТекДанныйОтвет.Вопрос)
Тогда // проверить - все ли правильные ответы на этот вопрос присутствуют
Возврат ТекДанныйОтвет.Балл; // то суммируем вес этого
ответа

Иначе
Возврат 0; // иначе не суммируем, не смотря на то, что
ответ был дан

КонецЕсли;
Иначе
Возврат ТекДанныйОтвет.Балл; // только один ответ, или
несколько, но необязательно перечислить все правильные
КонецЕсли;

КонецФункции

// проверка того, что пользователь перечислил все правильные (вес > 0)
ответы на Вопрос

Функция ВсеПравильныеПрисутствуют(Вопрос)

ВыборкаПравильныхОтветов = Справочники.Ответы.Выбрать(
Вопрос); // выбрать все ответы на Вопрос

Пока ВыборкаПравильныхОтветов.Следующий()Цикл // по всем
ответам

Если ВыборкаПравильныхОтветов.Вес > 0 Тогда // если это
правильный ответ

Если
ДанныеОтветы.Найти(ВыборкаПравильныхОтветов.Ссылка, "Ответ")=
Неопределено Тогда // то если его нет в списке данных ответов

Возврат Ложь; // то ложь

КонецЕсли;

КонецЕсли;

КонецЦикла;

Возврат Истина; // все правильные ответы пользователем даны

КонецФункции

Продолжение приложения В

```
// при проведении сделать запись в плане обучения (рег. сведений
Обучение)
Процедура ОбработкаПроведения(Отказ, РежимПроведения)

    НаборЗаписейОбучение =
РегистрыСведений.Обучение.СоздатьНаборЗаписей();
    НаборЗаписейОбучение.Отбор.Занятие.Установить(Тест); //
установить отбор по данному Тесту
    НаборЗаписейОбучение.Отбор.ФизЛицо.Установить(ФизЛицо); // и
проходившему его ФизЛицу
    НаборЗаписейОбучение.Прочитать();
    Если НаборЗаписейОбучение.Количество() = 0 Тогда // если такой
записи не было
        НоваяЗапись = НаборЗаписейОбучение.Добавить(); // то создать
ее
        НоваяЗапись.Тема = Тема; // и заполнить все измерения
        НоваяЗапись.Занятие = Тест;
        НоваяЗапись.ФизЛицо = ФизЛицо;
        НоваяЗапись.Период = Дата;
        ИначеЕсли НаборЗаписейОбучение.Количество() = 1 Тогда // если
такая запись только одна
            НоваяЗапись = НаборЗаписейОбучение[0]; // то это она и есть
искомая
        Иначе // если таких записей больше одной
            Сообщить("Ошибка: на одно занятие не может быть несколько
записей в регистре Обучение");
            Возврат;
        КонецЕсли;

        НоваяЗапись.Документ = ЭтотОбъект.Ссылка; // новая запись - либо
новая, либо найденная
        НоваяЗапись.Балл = Балл; // записать реквизит Балл
        Если РольДоступна("Тестер") Тогда // если проводит студент
            НоваяЗапись.Попыток = НоваяЗапись.Попыток+1; // то
увеличить кол-во попыток
        КонецЕсли;
        НоваяЗапись.Проверка = ПроверкаПройдена(); // заполнить реквизит
Проверка
        НаборЗаписейОбучение.Записать(Истина); // сохранить запись

    КонецПроцедуры
```

Продолжение приложения В

```
// при распроведении в рег. сведений Обучение изменить реквизит
Проверка - если был проверен преподадом - то снять эту проверку
Процедура ОбработкаУдаленияПроведения(Отказ)
```

```
Если Не Тест.ТребуетсяРучнаяПроверка Тогда // если проверка вообще
не нужна, то оставляем как есть (там ноль)
```

```
    Возврат;
```

```
КонецЕсли;
```

```
НаборЗаписейОбучение =
РегистрыСведений.Обучение.СоздатьНаборЗаписей();
НаборЗаписейОбучение.Отбор.Занятие.Установить(Тест);
НаборЗаписейОбучение.Отбор.ФизЛицо.Установить(ФизЛицо);
НаборЗаписейОбучение.Прочитать();
Если НаборЗаписейОбучение.Количество() = 1 Тогда
    НоваяЗапись = НаборЗаписейОбучение[0];
Иначе
    Возврат;
```

```
КонецЕсли;
```

```
НоваяЗапись.Проверка = 2; // вернуть в предыдущее состояние
НаборЗаписейОбучение.Записать(Истина);
```

```
КонецПроцедуры
```

```
// логика заполнения реквизита Проверка рег. сведений Обучение
Функция ПроверкаПройдена()
```

```
Если Не Тест.ТребуетсяРучнаяПроверка Тогда // если ручная проврка
преподавателем не требуется
```

```
    Возврат 0;
```

```
КонецЕсли;
```

```
Если РольДоступна("Тестер") Тогда // если документ проводил студент
    Возврат 2;
```

```
КонецЕсли;
```

```
Возврат 1; // если документ проводил не студент (препод или админ)
```

```
КонецФункции
```

Продолжение приложения В

Регистры сведений. Обучение. Формы. Форма записи. Модуль

// при нажатии на кнопку Просмотреть (историю) - открыть форму списка документов Тестирование с отбором

// по всем этой Тесту и Физлицу - т.е. просмотреть все документы прохождения этого теста этим студентом

&НаКлиенте

Процедура Просмотреть(Команда)

Отбор = Новый Структура();

Отбор.Вставить("ФизЛицо", Запись.ФизЛицо);

Отбор.Вставить("Тест", Запись.Занятие);

ПараметрыФормы = Новый

Структура("ЗакрыватьПриЗакрытииВладельца", Истина);

ПараметрыФормы.Вставить("Отбор", Отбор);

ОткрытьФорму("Документ.Тестирование.ФормаСписка",

ПараметрыФормы, ЭтаФорма);

КонецПроцедуры

Регистры сведений. Обучение. Формы. Форма списка. Модуль

// при открытии показывать текущему пользователю только "его" записи (если это студент)

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

Если РольДоступна("Тестер") Тогда

ЭлементОтбора =

Список.Отбор.Элементы.Добавить(Тип("ЭлементОтбораКомпоновкиДанных"));

ЭлементОтбора.ЛевоеЗначение = Новый
ПолеКомпоновкиДанных("ФизЛицо");

ЭлементОтбора.ВидСравнения =
ВидСравненияКомпоновкиДанных.Равно;

ЭлементОтбора.Использование = Истина;

ЭлементОтбора.РежимОтображения =
РежимОтображенияЭлементаНастройкиКомпоновкиДанных.Обычный;

Продолжение приложения В

ЭлементОтбора.ПравоеЗначение = ПараметрыСеанса.глПользователь;
КонецЕсли;

КонецПроцедуры

// установить сортировку по полю Порядок - чтобы показывать план
обучения в соответствии с заданным порядком прохождения занятия по теме
&НаКлиенте
Процедура ПриОткрытии(Отказ)

ЭлементыПользовательскихНастроек =
Список.КомпоновщикНастроек.ПользовательскиеНастройки.Элементы;
Для Каждого Элемент Из ЭлементыПользовательскихНастроек Цикл
Если ТипЗнч(Элемент) =
ТипЗнч(Список.КомпоновщикНастроек.Настройки.Порядок) Тогда
Элемент.Элементы.Очистить();
УсловиеСортировки =
Элемент.Элементы.Добавить(Тип("ЭлементПорядкаКомпоновкиДанных"));
УсловиеСортировки.Поле = Новый
ПолеКомпоновкиДанных("Порядок");
УсловиеСортировки.ТипУпорядочивания =
НаправлениеСортировкиКомпоновкиДанных.Возр;
УсловиеСортировки.Использование = Истина;
КонецЕсли;
КонецЦикла;
КонецПроцедуры

// при щелчке на полях текущей записи
&НаКлиенте
Процедура СписокВыбор(Элемент, ВыбраннаяСтрока, Поле,
СтандартнаяОбработка)

// если на поле Документ - то открыть этот документ
Если Поле.Имя = "Документ" Тогда
СтандартнаяОбработка = Ложь;
П = Новый Структура();
П.Вставить("Ключ", Элемент.ТекущиеДанные.Документ);
П.Вставить("Тест", Элемент.ТекущиеДанные.Занятие);
ОткрытьФорму("Документ.Тестирование.ФормаОбъекта", П);
КонецЕсли;

// если на поле Занятие - то открыть это занятие

Продолжение приложения В

```
Если Поле.Имя = "Занятие" Тогда
    СтандартнаяОбработка = Ложь;
    П = Новый Структура();
    П.Вставить("Ключ", Элемент.ТекущиеДанные.Занятие);
    ОткрытьФорму("Справочник.Тесты.ФормаОбъекта", П);
КонецЕсли;
```

```
// если на поле Попыток - то открыть форму истории - т.е. список
документво Тестирование с отбором
```

```
Если Поле.Имя = "Попыток" Тогда
    СтандартнаяОбработка = Ложь;
    Отбор = Новый Структура();
    Отбор.Вставить("ФизЛицо",
Элементы.Список.ТекущиеДанные.ФизЛицо);
    Отбор.Вставить("Тест",
Элементы.Список.ТекущиеДанные.Занятие);
    ПараметрыФормы = Новый
Структура("ЗакрыватьПриЗакрытииВладельца", Истина);
    ПараметрыФормы.Вставить("Отбор", Отбор);
```

```
    ОткрытьФорму("Документ.Тестирование.ФормаСписка",
ПараметрыФормы, ЭтаФорма);
КонецЕсли;
```

КонецПроцедуры

```
// обработка оповещения из внешних форм
&НаКлиенте
Процедура ОбработкаОповещения(ИмяСобытия, Параметр, Источник)
```

```
Если ИмяСобытия = "ПланОбучения" Тогда // если дали понять что
нужно обновиться
```

```
    Элементы.Список.Обновить(); // обновить содержимое списка
КонецЕсли;
```

КонецПроцедуры