

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
РЕСПУБЛИКИ КАЗАХСТАН  
ИННОВАЦИОННЫЙ ЕВРАЗИЙСКИЙ УНИВЕРСИТЕТ  
КАФЕДРА «АСОИиУ»**

Магистерская диссертация  
«Применение альтернативного метода учета, обработки и хранения  
электронной документации на основе .NET технологий на примере ИнЕУ»

**Специальность 6N0704 – «Вычислительная техника и программное  
обеспечение»**

Исполнитель \_\_\_\_\_ В.А. Шудров  
(подпись, дата)

Научный руководитель

Профессор \_\_\_\_\_ Т.М. Салий  
(подпись, дата)

Допущена к защите  
Зав. кафедрой АСОИиУ  
профессор \_\_\_\_\_ В.В. Нармат  
(подпись, дата)

Печать \_\_\_\_\_

## Определения

В настоящей диссертации применяются следующие термины с соответствующими определениями:

**Документ** – материальный носитель, содержащий, выражающий информацию в зафиксированном виде и специально предназначенный для её передачи во времени и пространстве.

**Электронный документооборот** – единый механизм по работе с документами, представленными в электронном виде, с реализацией концепции «безбумажного делопроизводства».

**Атрибут** – необходимое, существенное, неотъемлемое свойство объекта.

**Объектно-ориентированное программирование** – парадигма программирования, в которой основными концепциями являются понятия объектов и классов.

## Обозначения и сокращения

В настоящей диссертации применяются следующие обозначения и сокращения:

ИМО – информационно-методическое обеспечение.

ИнЕУ – Инновационный Евразийский Университет.

НИОКР – научно-исследовательские и опытно-конструкторские работы.

АСОИиУ – автоматизированные системы обработки информации и управления.

ИТ – информационные технологии.

СУД – система управления документами.

СУЭД – система управления электронными документами.

EDMS – electronic document management system, см. СУЭД.

OCR – optical character recognition (оптическое распознавание символов) – автоматическое распознавание с помощью специальных программ графических изображений символов печатного текста (например, введённого в компьютер с помощью сканера) и преобразование их в формат, пригодный для обработки текстовыми процессорами, редакторами текстов и т. д.

ICR – intelligent character recognition (интеллектуальное распознавание символов) – технология, обеспечивающая распознавание изображений текстов из символов, вписываемых в специальные формы.

БД – база данных.

ПО – программное обеспечение.

PDM – product data management – управление производственной документацией.

КИС – корпоративная информационная система.

OLE – object linking and embedding – связывание и встраивание объектов. Набор протоколов Microsoft для обмена данными между отдельными приложениями в Windows 3.1 и старше. Определяют, как одно приложение может использовать данные, подготовленные другим приложением.

DDE – dynamic data exchange – динамический обмен данными.

ODMA – open document management API – открытый API управления документами (документооборотом). Промышленный стандарт для связи прикладных программ с системой управления документами и другим групповым ПО.

MAPI – messaging API – интерфейс прикладного программирования систем передачи сообщений (электронной почты).

СУБД – система управления базами данных.

САПР – система автоматизированного проектирования.

OMS – output management system – система управления выводом документов.

IDARS – integrated document archive and retrieval systems – интегрированные системы архивации и поиска документов.

ERP – enterprise resource planning – планирование бизнес-ресурсов. Программное обеспечение, объединяющее все ресурсы предприятия, необходимые для его работы, включая планирование заказов, финансы и пр.

ТЭО – технико-экономическое обоснование.

ИС – информационная система.

НСИ – нормативно-справочная информация.

АТС – автоматическая телефонная станция.

CLR – common language runtime – общезыковая исполняющая среда.

FCL – framework class library – библиотека классов.

CIL – common intermediate language – общий промежуточный язык.

ЕСМА – european computer manufacturers association – Европейская ассоциация по стандартизации информационных и вычислительных систем.

ООП – объектно-ориентированное программирование

## Введение

**Актуальность исследования.** Сегодня все больше внимания на предприятиях уделяется эффективной организации и управлению существующими процессами производства и обслуживания, не стали исключением и высшие учебные заведения Казахстана. Стоит отметить, что проблема документооборота в высших учебных заведениях является одной из наиболее острых среди проблем управления.

Вуз – это многопрофильная организация, включающая в себя управление, образовательный процесс, научную и инновационную деятельность, а также различные обеспечивающие процессы (бухгалтерия, кадровый учет, делопроизводство и др.). Таким образом, «продукцией», которую выпускает вуз, с различных точек зрения, являются:

- услуги по реализации закона об образовании;
- услуги по подготовке специалистов для организаций;
- выпуск учебно-методической продукции (учебные курсы, книги);
- выпуск научно-технической продукции (реализация инновационных проектов, проведение НИОКР и т.п.).

Деятельность вуза связана с большим объемом бумажной документации, большую часть которой занимают документы, относящиеся к управлению образовательным процессом и его организации.

Особенностью автоматизации процессов в высших учебных заведениях является то, что деятельность университета как хозяйствующего субъекта и как образовательной организации происходит параллельно и эти процессы обладают тесной взаимосвязью. К примеру, информация о студентах используется как при учете успеваемости, так и в работе финансовых подразделений вуза, ответственных за информацию об оплате обучения.

В современных условиях для повышения эффективности управления необходимо совершенствование работы с документами, так как всякое управленческое решение всегда базируется на информации, на служебном документе.

Значение делопроизводственного обслуживания как одной из важнейших сфер деятельности определяется рядом факторов и, прежде всего, его универсальностью. Основу информационной среды любого предприятия, организации или учреждения составляют документы, созданные как «традиционным» рукописным, машинописным, типографическим способом, так и полученные с использованием компьютерных технологий. Содержащаяся в них информация будет обладать юридической силой и может быть использована только при совершении ряда обязательных делопроизводственных операций. В правильной организации делопроизводства заложена основа сохранности и эффективного использования документной информации в будущем.

На большинстве предприятий документооборот существует в бумажной форме. Безусловно, в таком виде документооборот более привычен и

традиционен, но при этом он обладает рядом недостатков, существенно перекрывающих достоинство его привычности.

В работе с документами большую часть сил и времени отнимают рутинные вещи типа доставки, поиска и систематизации. В случае с бумажными носителями их приходится осуществлять людям. Существует возможность переложить эти обязанности на электронные системы.

Информационные и коммуникационные технологии используются в образовании уже давно, но раньше этот процесс развивался стихийно, без комплексного подхода к автоматизации. В результате подготовка специалистов выполнялась на основе устаревших технологий, а современные информационные системы почти не применялись. В последние годы в связи с увеличением финансирования у вузов появились возможности для реформирования технического и информационного обеспечения процесса функционирования, а, следовательно, и для повышения качества образования.

Компьютеры в системе образования применяются довольно широко. Они используются в процессе обучения, в научных исследованиях, информационных технологиях. Однако первое, с чего в большинстве случаев начинается автоматизация учреждения, это бухгалтерский учет и прочие операции, совершаемые финансовыми и кадровыми подразделениями вуза [1].

Документированная информация составляет основу управления, его эффективность в значительной степени базируется на производстве и потреблении информации. В современном обществе информация стала полноценным ресурсом производства, важным элементом социальной и политической жизни общества. Качество информации определяет качество управления. В современных условиях для повышения эффективности управления необходимо уделять достаточное внимание совершенствованию работы с документами, так как всякое управленческое решение всегда базируется на информации, на служебном документе.

Организация работы с документами влияет на качество работы аппарата управления, организацию и культуру труда управленческих работников. От того, насколько профессионально ведется документация, зависит успех управленческой деятельности в целом. Деловая информация представляется в виде разного рода документов, исследования показывают, что 75% рабочего времени сотрудников организаций тратится на их подготовку, сопровождение, заполнение, копирование и передачу [2].

В современных условиях деятельности университета такое положение вещей нельзя назвать удовлетворительным. Рассмотрим более детально эту ситуацию на примере работы отдела ИМО ИнЕУ. Как показывает практика, основная часть нагрузки отдела ИМО – составление плана публикаций учебно-методической документации для кафедр вуза. Процесс утверждения плана публикаций состоит из следующих ступеней:

1. Создание плана публикаций для каждой из кафедр ВУЗа.
2. Утверждение плана публикаций с указанием авторов и даты публикации для каждой из кафедр на заседании совета кафедры.

3. Утверждение публикаций в различных инстанциях ВУЗа (Научно-методический совет, проректорат и т.д.)

4. Проверка каждой из публикаций на соответствие требованиям к публикациям.

5. Утверждение каждой из публикаций в ИМО.

6. Сдача работы в копировально-множительный отдел ИнЕУ для печати необходимого числа копий каждой публикации.

7. Сдача электронного варианта работы в медиатеку ИнЕУ для свободного доступа работников и учащихся вуза.

8. Рассылка необходимого числа копий публикации по кафедрам ИнЕУ.

9. Отметка о готовой публикации в отделе ИМО.

Несмотря на кажущуюся простоту процесса утверждения публикаций, он достаточно протяженный по времени, и каждый из этапов процесса характеризуется большим количеством бумажной документации, связанной с прохождением публикации через каждую из соответствующих инстанций.

Наряду с подготовкой плана публикаций, в задачи ИМО входит контроль исполнения плана кафедрами вуза. В частности, это выражается в составлении отчета о задолженностях кафедр по учебно-методическим публикациям по специальностям, что также является достаточно трудоемким процессом.

Соответственно, в случае необходимости нахождения какой-либо информации, касающейся плана или конкретной публикации, сотрудникам приходится просматривать большое количество документов прежде, чем информация будет найдена. Такой способ работы с документацией противоречит фундаментальным свойствам информации – ее актуальности и доступности.

Все это не может не сказаться на качестве учебного процесса. Своевременная подготовка методического обеспечения по образовательной программе должна позволить нормально функционировать кафедре и осуществлять преподавателям вуза свою деятельность.

Актуальность выбранной темы обусловлена тем, что деятельность отдела ИМО является важным звеном в организации учебного процесса в ИнЕУ, и в условиях развивающегося университета своевременное и полное исполнение возложенных на него обязанностей будет способствовать повышению качества организации учебного процесса и полноты его обеспечения необходимой научно-методической документацией.

Можно выделить следующие ключевые проблемы, связанные с подготовкой и контролем выполнения плана публикаций:

1. Протяженность во времени.
2. Большое количество сопутствующей бумажной документации.
3. Долгое время доступа к документации.
4. Разрозненность информации.

Стоит отметить, что вышеперечисленные проблемы характерны не только для ИнЕУ, но и для всех организаций образования, ведущих бумажный документооборот. Несмотря на удовлетворительное обеспечение вуза новыми техническими средствами, позволяющими эффективно собирать, обрабатывать,

передавать, хранить информацию, до сих пор проблема автоматизации внутреннего документооборота находится в начальной стадии. И, в первую очередь, направлена на автоматизацию финансовых, кадровых и прочих управленческих процессов и не затрагивает процесс управления учебной деятельностью вуза.

**Целью** данного исследования является выработка и применение альтернативного метода обработки информации о научно-методических публикациях в ИнЕУ.

Для достижения поставленной цели необходимо решить следующие **основные задачи**:

- изучить существующие разработки по предметной области исследования;
- изучить методику предпроектного обследования предприятия;
- разработать альтернативный метод обработки информации о научно-методических публикациях.

**Направление научного исследования** – детальное рассмотрение деятельности ИМО в процессе создания и утверждения плана научно-методических публикаций и ее влияния на указанный процесс. На основе исследования строится выработка альтернативного метода учета и обработки информации о научно-методических публикациях.

**Теоретическая значимость.** Положения, разработанные в диссертации, могут служить основой для дальнейших исследований по проблеме организации учебного процесса в вузах, а также могут быть использованы при чтении курса лекций кафедры АСОИиУ ИнЕУ по дисциплинам «Технология программирования», «Надежность компьютерных систем» и «Системы управления базами данных».

**Практическая значимость** исследования заключается в возможности применения результатов исследования в вузах Республики Казахстан для автоматизации деятельности подразделений с похожими задачами.

**Новизна** исследования состоит в разработке альтернативного метода учета и обработки информации о научно-методических публикациях с разработкой информационной системы на основе .NET технологий.

**Ожидаемые результаты.** Применение альтернативного метода обработки информации о научно-методических публикациях позволит достигнуть следующих результатов:

- облегчить работу подразделения ИМО;
- обеспечить быстрое получение информации, необходимой для своевременного реагирования на ту или иную ситуацию;
- исключить выполнение одной и той же работы дважды;
- уменьшить влияния человеческого фактора на степень контролируемости процесса утверждения;
- ввести единую информационную базу;
- обеспечить доступность информации разным подразделениям вуза на местах;



- контролировать процесс обеспечения научно-методической документацией с различных позиций;
- обеспечить быстрое выявление ошибок и легкость их исправления.

# 1 Обзор и анализ научной технической литературы по проблеме электронного документооборота

## 1.1 Основные виды систем электронного документооборота

На сегодняшний день тема электронного документооборота разработана достаточно глубоко и теоретически и практически обоснованные решения с различной степенью эффективности используются на предприятиях разного рода деятельности. Однако, в высших учебных заведениях степень автоматизации процессов находится на невысоком уровне и зачастую начинается и заканчивается на автоматизации процессов управления финансовыми и кадровыми потоками вуза.

Спектр существующих решений в области документооборота достаточно широк, поэтому в первую очередь рассмотрим существующие типы систем электронного документооборота.

Прежде всего, необходимо дать определения того, о каком типе программного обеспечения будет идти речь. Наиболее полное определение EDMS даёт аналитическая компания IDC – СУД обеспечивают процесс создания, управления доступом и распространения больших объемов документов в компьютерных сетях, а также обеспечивают контроль над потоками документов в организации. [3] Часто эти документы хранятся в специальных хранилищах или в иерархии файловой системы. Типы файлов, которые, как правило, поддерживают системы СУД включают, текстовые документы, образы, электронные таблицы, аудио-, видеоданные, и документы Web. Общими возможностями систем СУД являются создание документов, управление доступом, преобразование и безопасность. Предназначены СУД для хранения, обработки и поиска неструктурированной (текстов, рисунков) и произвольно структурированной (чертежей, схем, аудио- или видеозаписей) информации. [4]

На рынке средств электронного управления документами, условно, выделим пять категорий технологий. Следует заметить, что ни одна классификация, видимо, не является идеальной. В результате некоторые продукты одновременно попадают в несколько категорий и имеют возможности, характерные для продуктов из разных категорий.

Ниже перечислены категории технологий СУД с примерами наиболее известных поставщиков и продуктов в каждом классе:

- системы СУД, ориентированные на бизнес-процессы: Documentum, FileNet (Panagon и Watermark), Hummingbird (PC DOCS) – как правило, предназначены для специфических вертикальных и горизонтальных приложений, иногда ориентированные на использование в определенной индустрии. Эти решения, как правило, обеспечивают полный жизненный цикл работы с документами, включая технологии работы с образами, управления записями и потоками работ, управление контентом и т.д.

- корпоративные СУД: Lotus (Domino.Doc), дополнения к Novell GroupWise, Opent Text (LiveLink), Keyfile Corp., Oracle (Context) –

обеспечивают корпоративную инфраструктуру для создания, совместной работы над документами и их публикации, доступную, как правило, всем пользователям в организации. Основные возможности этих систем аналогичны системам, ориентированным на бизнес-процессы. Однако их отличительной особенностью является способ использования и распространения. Аналогично таким средствам как текстовые редакторы и электронные таблицы, корпоративные СУД являются стандартным, «приложением по умолчанию» для создания и публикации документов в организации. Как правило, эти средства не ориентированы на использование только в какой-то определенной индустрии или для узко определенной задачи. Они предлагаются и внедряются как общекорпоративные технологии, доступные практически любой категории пользователей.

- системы управления контентом (от англ. content – содержание, суть): Adobe, Excalibur – обеспечивают процесс отслеживания создания, доступа, контроля и доставки информации вплоть до уровня разделов документов и объектов для их последующего повторного использования и компиляции. Потенциально доступность информации не в виде документов, а в виде меньших объектов облегчает процесс обмена информацией между приложениями.

- системы управления образами – преобразуют информацию с бумажных носителей в цифровой формат, как правило, это TIFF (Tagged Image File Format), после чего документ может быть использован в работе уже в электронной форме.

- системы управления потоками работ (Workflow management): Lotus (Domino/Notes и Domino Workflow), Jetform, FileNet, Action Technologies, Staffware – эти системы в основном рассчитаны на обеспечение движения неких объектов по заранее заданным маршрутам (так называемая «жесткая маршрутизация»). На каждом этапе объект может меняться, поэтому его называют общим словом «работа» (work). Системы такого типа называют системами workflow – «поток работ» (к сожалению, для этого термина нет точного эквивалента в русском языке). К работам могут быть привязаны документы, но не документы являются базовым объектом этих систем. С помощью таких систем можно организовать определенные работы, для которых заранее известны и могут быть прописаны все этапы.

СУЭД используются для достижения следующих целей:

- интеграции процессов;
- повышения информированности руководства и специалистов за счет увеличения объемов информационного хранения, централизованной обработки информации, уменьшения времени поиска документов и, соответственно, подготовки отчетов и докладов, а также за счет повышения полноты и достоверности отчетов;
- уменьшения стоимости документационного обеспечения управления предприятием за счет перехода от бумажного делопроизводства к

электронному, снижения стоимости копирования и передачи бумажных документов;

- уменьшения стоимости и сокращения времени поиска бумажных оригиналов документов в архивном хранении за счет получения точной адресации в электронном виде;

- интеграции информационных процессов в рамках кооперации предприятий;

- создание качественно новой информационной базы для последующего совершенствования процессов документационного обеспечения управления и технологии работы с документами.

СУЭД должна соответствовать существующей информационно – организационной структуре предприятия (группы предприятий) и обеспечивать модификацию по мере совершенствования этой структуры.

СУЭД предназначена для автоматизации процессов документационного обеспечения управления предприятием, включая:

1. Процессы подготовки, ввода, хранения, поиска и вывода организационно-распорядительных документов (подсистема «Электронный архив»);

2. Процессы подготовки, ввода, хранения, поиска и вывода стандартных форм документов (подсистема «Ввод стандартных форм документов»);

3. Управление делопроизводством (создание, обработка и систематизация архивного хранения документов – подсистема «Делопроизводство»).

Объектом автоматизации являются процессы:

- создание документов;

- подготовка учета, систематизации и архивного хранения, поиска и получения организационно-распорядительных, отчетно-статистических, учетных, плановых, информационно-справочных и других управленческих документов;

- работа с документами (передача, учет, контроль исполнения и др.).

## **1.2 Основные требования к системам электронного документооборота**

Основные требования к системе состоят в следующем. Для информационного обмена в системе должна использоваться корпоративная вычислительная сеть, в которую включаются как локальные, так и удаленные пользовательские системы. СУЭД должна взаимодействовать с электронной почтовой системой и автоматизированными системами управления предприятием. Система должна допускать круглосуточный режим функционирования; иметь средства диагностики и индикации текущего состояния системы, используемых ресурсов; позволять наращивать количественные и расширять функциональные характеристики системы.

СУЭД должна обеспечивать изменение своих характеристик, параметров, адресатов передачи документов и т.п. в зависимости от изменения регламента документооборота предприятия (норм, правил, порядка, форматов и т.п.).

Система должна строиться на основе открытых технологий, обеспечивающих ее дальнейшую модернизацию и развитие без переработки. [5] В системе должна быть предусмотрена возможность использования масштабируемых аппаратных платформ, в частности технологически простой замены серверов и устройств хранения информации на более производительные (емкие).

Система должна обеспечивать с достаточной точностью распознавания документов на русском и основных европейских языках текстового и табличного формата А3, А4, а также полнотекстовую индексацию текстового документа. Особое внимание должно уделяться объемам хранения, одновременному доступу к документам и срокам хранения документов.

Подсистема «Электронный архив» предназначена для следующих целей:

- оперативного и потокового ввода документов, полученных из бумажных документов путем сканирования (получение электронных графических образов) и OCR-распознавания (преобразование их в текстовые документы);
- оперативного и потокового ввода электронных документов из других источников;
- передачи электронных документов в подсистему «Делопроизводство»;
- реквизитного и полнотекстового индексирования документов;
- хранения электронных документов всех видов и представлений (в том числе графических образов), их поисковых образов и индексов (атрибутивных и полнотекстовых);
- адресация оригиналов документов в бумажном архиве;
- поиска и извлечения документов из электронного архива;
- вывода документов из системы в виде твердой (бумажной) копии или в электронном виде, в том числе для дальнейшей доставки средствами связи;
- связи представлений хранимых документов с фактографической частью базы данных;

Подсистема «Ввод стандартных форм документов» обеспечивает:

- проектирование представлений стандартных форм документов для их последующего ввода;
- потоковый ввод стандартных форм документов в фактографическую часть подсистемы «Электронный архив» (включая сканирование, OCR (ICR), стилизованное распознавание, контекстный контроль и экспорт в БД).

Подсистема «Делопроизводство» обеспечивает:

- создание электронных документов средствами подсистемы;
- ведение версий (редакций) документов и их авторства;
- защиту от конфликтов при попытке одновременной модификации документа;
- электронное визирование документов (электронная подпись);
- связывание документов в логические группы;

- организацию потока работ с документами: задание маршрута (регламентного или ситуационного) движения документов и условий осуществления работ с ними;
- допуск к документам и расширенным видам работ с ними в соответствии с установленными правами пользователей;
- протоколирование видов работ, производимых пользователями с документами;
- управление маршрутами движения документов;
- контроль местонахождения документа и состояния его исполнения;
- передачу и извлечение документов в/из подсистемы «Электронный архив».

Возможный состав программных инструментальных и технологических средств, ориентированных на управление документами и документооборотом, а также средств реализации процедур работы с документами может быть представлен следующим образом:

- средства для ввода бумажных документов и распознавание образов;
- средства для создания электронных документов;
- средства для организации и работы с электронным архивом;
- технологические средства, ориентированные на управление документооборотом;
- технологические средства, ориентированные на управление документами;
- инструментальные средства разработки приложений, реализующих специфические функции и технологии работы с документами.

Программные технологические пакеты, ориентированные на управление документами и документооборотом, должны быть открытыми для интеграции с приложениями, реализующими специфические функции, характерные при работе с документами на предприятии.

Инструментальные средства для разработки приложений должны быть такими, чтобы приложения, разработанные с их помощью, интегрировались в программную среду управления документами и документооборотом.

Следует сказать, что в настоящее время существует немало подходов к решению проблем управления электронным документооборотом. Кроме так называемых универсальных систем электронного документооборота (EDMS) решить задачу управления документами можно при помощи следующих основных видов программных средств [6]:

- средств групповой работы (типа ПО Lotus Notes, Novell GroupWise и MS Exchange);
- систем управления особыми видами документов (в частности, PDM-систем);
- специальных модулей управления документооборотом в составе корпоративных информационных систем для предприятий (SAP R/3, Baan и др.).

Чем же универсальные системы электронного документооборота отличаются от них? И в чем преимущества их использования по сравнению с другими подходами?

Пожалуй, в каждом конкретном случае нужно исходить из особенностей целей и задач, которые ставит предприятие при внедрении системы управления документооборотом. Например, для одних задач наилучшим решением будет ПО Novell GroupWise, для других же оптимальным будет внедрение именно EDMS-системы. Важным свойством универсальных систем электронного документооборота является то, что они предоставляют возможность решения большого числа задач управления документами. Тем не менее, следует отметить, что полностью все задачи управления документооборотом они все равно не смогут решить (из-за их динамически меняющегося и, практически, неисчерпаемого многообразия). Кроме того, внедрение специализированного ПО может оказаться дешевле и эффективней.

У EDMS-систем, по сравнению с альтернативными решениями, есть ряд серьезных достоинств. В частности, по сравнению с ПО поддержки групповой работы, большинство наиболее распространенных в мире EDMS-систем обладает следующими достоинствами:

- практическая неограниченность размера электронного архива;
- встроенная поддержка полнотекстовой индексации;
- поддержка версий и подверсий документов;
- возможность хранения документов в архиве в исходном формате;
- возможность ведения журналов действий пользователей.

Возникает вопрос: можно ли на основе ПО поддержки групповой работы создать полноценную систему управления документооборотом предприятия? Следует сказать, что это вполне возможно, т.к. в некоторых программных средствах поддержки групповой работы есть развитые среды разработки приложений (например, в Lotus Notes). Более того, уже существуют приложения (разработанные, к примеру, на базе ПО Lotus Notes), в которых вышеперечисленные возможности EDMS-систем также реализованы. Однако надо заметить, что эти возможности реализованы в EDMS-системах уже изначально. Кроме того, при работе с групповым ПО в ряде случаев возникают вопросы снижения быстродействия приложений при работе с большими и сверхбольшими базами данных. Групповое ПО ориентировано, прежде всего, на решение задач организации взаимодействия между пользователями. По своей сути, оно является расширением программ электронной почты и не всегда подходит для работы с крупными архивами документов. На рынке сформировалось мнение, что применение группового ПО (в качестве основы для создания систем электронного документооборота) наиболее уместно на малых и средних предприятиях. Следует также отметить, что EDMS-системы в большинстве случаев имеют шлюзы в групповое ПО и, при необходимости, могут использовать имеющиеся в нем почтовые программы.

В свою очередь, специализированные системы управления документами можно условно разделить на 2 категории [7]. В первую категорию входят

программные средства (достаточно давно разработанные), изначально ориентированные на узкопрофильное применение и не имеющие средств интеграции с другими информационными системами. Подобные системы еще есть на предприятиях, и, в основном, они представлены местными разработчиками. Помимо управления документооборотом, такие программные системы предназначены для решения конкретных финансовых задач, управления персоналом, ведения складского учета и т. д. Отметим, что у большинства таких систем есть существенные ограничения при их применении на предприятиях. В частности, для каждого из приложений в такой системе необходимо поддерживать свою базу данных (нередко эти базы данных никак не связаны друг с другом). Затраты на поддержание инфраструктуры таких систем настолько велики, что они постепенно изживают себя. Во вторую категорию входят специализированные PDM-системы, предназначенные для управления производственной информацией и имеющие средства интеграции с другими программными системами. PDM-системы работают уже и с электронными образами сложнейших объектов.

Что касается КИС, то в большинство из них встроены модули поддержки делопроизводства. [8] Однако возможности этих модулей достаточно ограничены, т.к. практически невозможно создать универсальную и полнофункциональную КИС. Кроме того, цена западных КИС настолько велика, что не всякое предприятие сможет позволить себе ее внедрение. Следует все же отметить, что в большинстве распространенных EDMS-систем реализована интеграция с наиболее известными КИС (в частности, с SAP R/3, Oracle Applications и др.). Именно возможность интеграции с различными приложениями является одним из характерных свойств EDMS-систем. Благодаря наличию этого свойства, EDMS-системы могут выступать в качестве связующего звена между различными системами, функционирующими на предприятии, создавая, тем самым, основу для организации всего его делопроизводства.

### **1.3 Свойства универсальных систем электронного документооборота**

У EDMS-систем есть и другие свойства, благодаря наличию которых применение универсальных систем электронного документооборота позволяет существенно оптимизировать управление современным предприятием.

В литературе [9] можно встретить описание основных свойств EDMS-систем.

1. Открытость. Все EDMS-системы построены по модульному принципу, а их API-интерфейсы являются открытыми. Это позволяет добавить к EDMS-системам (при необходимости) новые функции или расширять уже имеющиеся. В настоящее время разработка приложений, хорошо интегрируемых с EDMS-системами, стала отдельным видом бизнеса в отрасли промышленного производства ПО, и множество третьих фирм готовы предложить здесь свои услуги. Возможность относительно простого добавления к EDMS-системам множества модулей разработки третьих фирм значительно расширяет их



функциональные возможности. Например, для EDMS-систем разработаны модули ввода документов со сканера, связи с электронной почтой, с программами пересылки факсов и др.

2. Высокая степень интеграции с прикладным ПО. Важной особенностью EDMS-систем является также высокая степень их интеграции с прикладными программами за счет использования технологий OLE Automation, DDE, ActiveX, ODMA, MAPI и др. А непосредственно при работе с документами вообще нет необходимости пользоваться утилитами EDMS-системы. Пользователи имеют дело только с обычными прикладными программами: в момент инсталляции клиентской части EDMS-системы прикладные программы дополняются новыми функциями и элементами меню. Например, пользователь текстового процессора Microsoft Word, открывая файл, сразу видит библиотеки и папки с документами EDMS-системы (откуда он и выбирает необходимый ему документ). При сохранении документ автоматически помещается в базу данных EDMS-системы. То же относится и к другим офисным и специализированным программам.

3. Особенности хранения документов. EDMS-системы работают, преимущественно, на распределенных архитектурах и используют разнообразные комбинации технологий сбора, индексирования, хранения, поиска и просмотра электронных документов. Во всех EDMS-системах реализована иерархическая система хранения документов. Можно сказать, что применяется принцип «шкаф/полка/папка». Каждый документ помещается в папку, которая, в свою очередь, находится на полке и т. д. Количество уровней вложения при хранении документов не ограничено. Один и тот же документ может входить в состав нескольких папок и полок за счет применения механизма ссылок (исходный документ, в этом случае, остается неизменным и хранится в определенном администратором EDMS-системы месте). В ряде EDMS-систем реализованы еще более мощные возможности хранения за счет установления связей между документами (так называемые связанные документы). Причем эти связи можно устанавливать и редактировать в графическом виде.

Любому документу в EDMS-системе присущ определенный набор атрибутов (например, его название, автор документа, время его создания и др.). Набор атрибутов может меняться от одного типа документа к другому (в пределах одного типа документов он сохраняется неизменным). В EDMS-системах атрибуты документа хранятся в реляционной базе данных. Для каждого типа документов с помощью визуальных средств создается шаблон карточки, где в понятном графическом виде представлены наименования атрибутов документа. При введении документа в EDMS-систему берется необходимый шаблон и заполняется карточка (вносятся значения атрибутов). После заполнения карточка оказывается связанной с самим документом.

В большинстве случаев серверная часть EDMS-системы состоит из следующих логических компонентов (которые могут располагаться как на одном, так и на нескольких серверах):

- хранилища атрибутов документов (карточек);

- хранилища документов;
- сервисы полнотекстовой индексации.

Под хранилищем документов обычно понимается хранилище содержимого документов. Хранилище атрибутов и хранилище документов часто объединяют под общим названием «архив документов». [10] Для хранения атрибутов в большинстве EDMS-систем используются СУБД Oracle, Sybase, MS SQL Server и Informix, обеспечивающие поиск документов по атрибутам.

Для хранения непосредственно содержимого документов в большинстве EDMS-систем применяются файл-серверы MS Windows NT, Novell NetWare, UNIX и др. В этом случае могут быть реализованы и гетерогенные комбинации сетевых сред. Например, база данных с атрибутами документов может работать под управлением ОС UNIX в сети TCP/IP, а сами документы могут храниться под ОС Novell NetWare в сети IPX/SPX. Следует отметить, что большим преимуществом EDMS-систем является хранение документов в исходном формате (и автоматическое распознавание EDMS-системами множества форматов файлов).

Для обеспечения функционирования хранилищ данных и просмотра содержащейся в них информации, Корпорация Microsoft совместно с партнерами предлагает тесно интегрированный комплекс программных средств: [11]

- операционные системы Microsoft Windows и Microsoft Windows Server;
- система управления базами данных (СУБД) Microsoft SQL Server;
- сервер обмена сообщениями Microsoft Exchange Server;
- программный пакет Microsoft Office.

В последнее время все большую популярность приобретает хранение документов вместе с атрибутами в базе данных. Такой подход имеет свои преимущества и недостатки. Преимуществом является значительное повышение безопасности доступа к документам, а основным недостатком – низкая эффективность работы с документами при большом объеме хранимой информации. При данном подходе также требуется использование мощных серверов с большими объемами оперативной памяти и жестких дисков. Кроме того, в случае сбоя базы данных восстановить хранившиеся в ней документы будет очень непросто. Необходимо также строго привязываться к конкретной СУБД.

Сервис полнотекстовой индексации обычно является опциональным. Полнотекстовая индексация предназначена для индексации содержимого документов, чтобы, при необходимости, пользователь мог найти документы с указанными в строке поиска словами. Данный сервис должен работать с документами в их исходных форматах, т. е. должен понимать все поддерживаемые в EDMS-системе форматы файлов. Он также должен поддерживать язык, с которым работает пользователь (для казахстанских предприятий это казахский и русский языки).

Для физического хранения информации в пределах 5 лет вполне надежны любые современные носители информации. [12]

4. Особенности маршрутизации документов. Модули EDMS-систем, отвечающие за документооборот, принято называть модулями маршрутизации документов. В общем случае используется свободная и жесткая маршрутизация документов. При свободной маршрутизации любой участвующий в документообороте пользователь может по своему усмотрению изменить существующий (или задать новый) маршрут прохождения документов. При жесткой маршрутизации маршруты прохождения документов строго регламентированы, и пользователи не вправе их менять. Однако при жесткой маршрутизации может выполняться обработка логических операций, когда маршрут изменяется при выполнении каких-либо заранее заданных условий (например, отправке документа руководству при превышении конкретным пользователем своих должностных полномочий, предположим, финансовых). В большинстве EDMS-систем модуль маршрутизации входит в комплект поставки, в некоторых его необходимо приобретать отдельно. В частности, полнофункциональные модули маршрутизации разрабатывают и поставляют третьи фирмы.

5. Средства разграничения доступа. В EDMS-системах реализованы надежные средства разграничения полномочий и контроля над доступом к документам. В большинстве случаев с их помощью определяются следующие виды доступа (набор задаваемых полномочий зависит от конкретной EDMS-системы):

- полный контроль над документом;
- право редактировать, но не уничтожать документ;
- право создавать новые версии документа, но не редактировать его;
- право аннотировать документ, но не редактировать его и не создавать новые версии;
- право читать документ, но не редактировать его;
- право доступа к карточке, но не к содержимому документа;
- полное отсутствие прав доступа к документу.

Во время работы с EDMS-системой каждое действие пользователя протоколируется, и, таким образом, вся история его работы с документами может быть легко проконтролирована.

6. Отслеживание версий и подверсий документов. При одновременной работе с документом сразу нескольких пользователей (особенно, когда его необходимо согласовывать в различных инстанциях) очень удобным свойством EDMS-систем является использование версий и подверсий документа. Предположим, исполнитель создал первую версию документа и передал ее на рассмотрение следующему пользователю. Второй пользователь изменил документ и создал на его основе уже новую версию. Затем он передал свою версию документа в следующую инстанцию третьему пользователю, который создал уже третью версию. Спустя определенное время, ознакомившись с замечаниями и исправлениями, первый исполнитель документа решает доработать исходную версию и на ее основе создает подверсию первой версии документа. Достоинством EDMS-систем является реализованная в них

возможность автоматического отслеживания версий и подверсий документов (пользователи всегда могут определить, какая именно версия или подверсия документа является наиболее актуальной по порядку или времени их создания).

7. Наличие утилит просмотра документов разных форматов. В состав большинства EDMS-систем входят утилиты для просмотра документов (так называемые viewers – просмотрщики), поддерживающие многие форматы файлов. С их помощью очень удобно работать, в частности, с графическими файлами (например, с файлами чертежей в САПР). Помимо базового комплекта утилит просмотра (входящего в каждую EDMS-систему), у третьих фирм можно приобрести дополнительные утилиты, хорошо интегрируемые с EDMS-системами.

8. Возможность аннотирования документов. При организации групповой работы над документами обычно весьма полезна возможность их аннотирования. Так как в некоторых случаях пользователи лишены прав на внесение каких-либо изменений в документ в процессе его согласования, то они могут воспользоваться возможностью его аннотирования. В большинстве EDMS-систем аннотирование реализуется за счет включения в карточку документа атрибута для аннотации и передачи пользователям прав на редактирование такого поля карточки. Но такое решение не всегда приемлемо (особенно при аннотировании графического документа). В связи с этим в некоторых EDMS-системах реализована так называемая функция «красного карандаша», с помощью которой можно графически указать недостатки на самом изображении. Программные средства, в которых реализована функция «красного карандаша», широко предлагаются третьими фирмами.

9. Поддержка различных клиентских программ. Клиентами большинства EDMS-систем могут быть ПК с ОС Microsoft Windows 95, 98, ME, NT, 2000, XP или Vista. В некоторых EDMS-системах допускается также использование платформ UNIX и Macintosh. Кроме того, все современные EDMS-системы позволяют работать с документами через стандартные Web-браузеры. Так как Web-браузеры могут быть размещены на разнообразных клиентских платформах, то это облегчает решение проблемы обеспечения работы EDMS-систем в гетерогенных сетевых средах. При использовании Интернет-технологий, у EDMS-систем появляется еще один серверный компонент, отвечающий за доступ к документам через Web-браузер.

10. Системы управления выводом документов. Одним из особых сегментов современного рынка СУД являются так называемые OMS-системы, основным предназначением которых является генерация выходных документов. Ряд OMS-систем отвечает только за распределение и доставку выходных документов (в электронном виде – в форматах DOC, HTML, XML и PDF). Очень часто OMS-системы интегрированы с программными пакетами сканирования документов и изображений. Полезной возможностью некоторых OMS-систем является и взаимодействие с унаследованными системами.

В некоторых OMS-системах дополнительно реализована возможность архивации и долговременного хранения выходных отчетов и документов. В связи с этим, многие из OMS-систем классифицируются Gartner Group как

IDARS. Однако главной причиной популярности OMS-систем все же является занимаемая ими рыночная ниша – генерация документов и отчетов в информационных системах предприятий и организаций, построенных с использованием ERP-систем. По мнению аналитиков Gartner Group, одним из слабых мест современных ERP-систем является плохое управление генерацией выходных документов [13] (разработчики ERP-систем больше сосредоточены на повышении функциональности ключевых модулей своего ПО, чем на «второстепенных» вопросах обеспечения генерации выходных отчетов, не имеющих, по их мнению, хороших рыночных возможностей). Именно этот недостаток ERP-систем и послужил основным фактором появления и быстрого развития рынка OMS-систем.

11. Информационно-поисковые технологии для электронного делопроизводства. В системах должны быть предусмотрены развитые возможности индексирования, хранения и поиска документов, включая полнотекстовое индексирование. [14]

Индексирование документов означает:

- реквизитную индексацию;
- ручное управление структурами хранения документов;
- настройку состава полей, типа и длины контрольных карточек документа в архиве;
- автоматическую полнотекстовую индексацию документов;
- автоматическое распознавание формата файла электронного документа;
- индексацию текста документа сразу при его вводе;
- морфологический анализ индексируемых текстов для приведения помещаемых в индекс слов к канонической форме (нормализация текста в индексе);
- наличие стоп-словарей при полнотекстовой индексации и обеспечение их настройки;
- выделение из текста сложных (составных) терминов и понятий;
- поддержку и настройку словарей и тезауруса;
- технологию реквизитной разметки документов;
- настройку набора реквизитов;
- возможность различных режимов реквизитной разметки документов (ручной, автоматический, полуавтоматический);
- интеграцию с произвольными программами создания, просмотра и изменения электронных документов.

Хранение документов включает в себя возможность:

- работы с библиотеками магнитооптических дисков, магнитных лент, а также различных оптических носителей для хранения больших объемов информации;
- управление внешними носителями;
- хранение электронных документов произвольного формата, аннотированных текстов, по которым производится поиск;

- хранения в системе для каждой страницы как исходного изображения, так и распознанного текста.

Поиск предусматривает:

- наличие стандартного языка запросов к БД;
- реквизитный поиск документов;
- полнотекстовый поиск, точность поиска;
- поиск по именам документов и аннотациям;
- формирование сложных запросов с использованием логических операций;
- контекстный поиск и меру близости слов при поиске;
- адекватность результата поиска запросу;
- морфологический анализ при построении запроса;
- возможность работы с системой удаленных пользователей.

Поиск может быть составным и включать несколько поисковых атрибутов из перечисленных выше типов. Результатом поиска служат образ найденных документов и содержание аннотаций, построенных в процессе индексирования или его ручного атрибутирования. Найденные образы могут быть распечатаны или переданы по корпоративной сети для просмотра с другого рабочего места.

Запрос строится путем заполнения шаблона ключевыми словами, содержание которых пользователь может задавать произвольно. Поиск документов по запросу осуществляется на двух уровнях: сначала происходит поиск в справочнике данных, а по ссылкам ищется (при задании пользователем соответствующих атрибутов) сам документ. Как правило, хранение документов осуществляется автономно из-за больших объемов архива и необходимости использования оптических и прочих накопителей информации. Такая архитектура, кроме всего, позволяет минимизировать время, затрачиваемое на физический доступ к документам в режиме множественных запросов, а также обеспечивает доступ данных из неоднородных (гетерогенных) сетей.

Следует отметить, что образ документа может храниться как в текстовом, так и в графическом формате. В последнем случае он, как правило, является результатом использования OCR-технологий.

## 2 Методика предпроектного обследования организации

В современных условиях оценка эффективности проектов внедрения информационных систем становится обычной практикой, и внедрение систем электронного документооборота – не исключение. Доросинский и Кузьмин [16] отмечают, что организация и внедрение СЭД – один из самых популярных проектов в области информационных технологий. При разработке систем управления документацией предприятия в качестве методики исследования используется методика обследования предприятия и составления схем существующих процессов. При этом следует учитывать, что организационная и технологическая стороны процесса автоматизации являются взаимопроникающими и не могут рассматриваться обособленно. [17]

Барановская и др. [18] дают следующее определение предпроектному обследованию предприятия – это изучение и диагностический анализ существующей системы обработки информации. Материалы, полученные в результате обследования, должны быть использованы:

- для обоснования разработки и поэтапного внедрения систем;
- для составления технического задания на разработку систем;
- для разработки технического и рабочего проектов систем.

Обследование проводится разработчиками совместно с заказчиком после издания приказа заказчика о проведении работ по предпроектному обследованию.

Обследование начинается с изучения производственно-экономических характеристик объекта, основных функций, осуществляемых подразделениями и их руководителями. [19] Далее изучаются задачи, обеспечивающие реализацию функций управления, организационная структура, штаты и содержание работ по управлению на предприятии и в объединении, а также характер подчиненности вышестоящим органам управления. В процессе обследования должны быть выявлены:

- инструктивно-методические и директивные материалы, на основе которых определяются состав подсистемы и перечень задач;
- возможности применения новых методов решения задач.

При изучении каждой функциональной задачи управления рассматриваются:

- назначение задачи; сроки и периодичность ее решения;
- степень формализуемости задачи;
- источники информации, необходимые для решения задачи;
- показатели и их количественные характеристики;
- порядок корректировки информации;
- действующие алгоритмы расчета показателей и возможные методы контроля;
- действующие средства сбора, передачи и обработки информации;
- действующие средства связи;
- принятая точность решения задачи;

- трудоемкость решения задачи;
- действующие формы представления исходных данных и результатов их обработки в виде документов;
- потребители результатной информации по задаче.

При обследовании документооборота следует составить схему маршрута движения документов, которая должна отразить:

- количество документов;
- место формирования показателей документа;
- взаимосвязь документов при их формировании;
- маршрут и длительность движения документа;
- место использования и хранения данного документа;
- внутренние и внешние информационные связи;
- объем документа в знаках.

По результатам обследования следует установить перечень задач управления, решение которых целесообразно автоматизировать, и очередность их разработки. [20] В отчете по обследованию, называемом ТЭО, приводятся: характеристика материально-технической базы производства предприятия (объединения), численность работников по категориям, основные технико-экономические показатели производства и реализации продукции, краткое описание функций подразделений и должностных лиц, схемы информационных связей и объем информации по периодам, схемы маршрутов движения документов, данные об уровне автоматизации управленческого труда и методах управления.

В ТЭО обосновываются предложения по совершенствованию системы управления, выделяются функции, подлежащие автоматизации, указываются первоочередной комплекс задач и предварительный перечень средств системы, проводится ориентировочная оценка экономической эффективности создания ИС.

Техническое задание на систему разрабатывается заказчиком при непосредственном участии разработчика. Техническое задание – это документ, утвержденный в установленном порядке, определяющий цели, требования и основные исходные данные, необходимые для разработки автоматизированной системы управления, и содержащий предварительную оценку экономической эффективности системы.

Утвержденное техническое задание является документом, которым разработчики должны руководствоваться на всех этапах создания системы, и проектирования задач. Изменения, вносимые в техническое задание, должны оформляться протоколом, являющимся частью технического задания. Протокол должен утверждаться заказчиком.

При разработке технического задания следует:

- установить общую цель создания ИС, определить состав подсистем и задач;
- разработать и обосновать требования, предъявляемые к информационным подсистемам;



- разработать и обосновать требования, предъявляемые к информационной базе, математическому и программному обеспечению, комплексу технических средств (включая средства связи и передачи данных);
- установить общие требования к проектируемой системе;
- определить перечень задач и исполнителей;
- определить этапы создания системы и сроки их выполнения;
- провести предварительный расчет затрат на создание системы и определить уровень экономической эффективности ее внедрения.

Техническое задание должно включать следующие разделы: [21]

1. Введение.
2. Основание для разработки системы.
3. Общие положения.
4. Функциональная часть системы.
5. Обеспечивающая часть системы.
6. Организация работ и исполнители.
7. Этапы разработки и внедрения системы.
8. Предварительный расчет затрат на создание системы и экономической эффективности от ее внедрения.

После утверждения технического задания разрабатываются координационный план создания системы, сетевой график работ и проводится расчет затрат на разработку системы.

Основанием для разработки технического проекта системы служит техническое задание, утвержденное заказчиком.

Технический проект системы – это техническая документация, утвержденная в установленном порядке, содержащая общесистемные проектные решения, алгоритм решения задач, а также оценку экономической эффективности автоматизированной системы управления и перечень мероприятий по подготовке объекта к внедрению.

Технический проект разрабатывается в целях определения основных проектных решений по созданию системы. На этом этапе осуществляется комплекс научно-исследовательских и экспериментальных работ для выбора наилучших вариантов решений, проводятся экспериментальная проверка основных проектных решений и расчет экономической эффективности системы. Фактически технический проект содержит комплекс экономико-математических и алгоритмических моделей.

Полный комплект технического проекта на систему включает в себя 10 документов:

1. Пояснительная записка.
2. Функциональная и организационная структура системы.
3. Постановка задач и алгоритм решения.
4. Организация информационной базы.
5. Альбом форм документов.
6. Система математического обеспечения.
7. Принцип построения комплекса технических средств.

8. Расчет экономической эффективности системы.
9. Мероприятия по подготовке объекта к внедрению системы.
10. Ведомость документов.

Из приведенного перечня документ № 3 «Постановка задач и алгоритм решения» выполняется для каждой отдельной задачи, включаемой в ИС, остальные документы – общие для всей системы. Кроме того, документы №№ 1, 2, 5, 8 и 9 могут разрабатываться для отдельных подсистем. Все перечисленные документы можно сгруппировать и представить в виде четырех основных частей технического проекта: экономико-организационная, информационная, математическая, техническая.

Экономико-организационная часть технического проекта содержит пояснительную записку с обоснованием разработки системы, перечень организаций разработчиков, краткую характеристику объекта с указанием основных технико-экономических показателей его функционирования и связей с другими объектами, краткие сведения об основных проектных решениях по функциональной и обеспечивающей частям системы.

В разделе технического проекта, посвященном организационной и функциональной структуре системы, даются:

- обоснование выделяемых подсистем, их перечень и назначение;
- перечень задач, решаемых в каждой подсистеме, с краткой характеристикой их содержания; схема информационных связей между подсистемами и между задачами в рамках каждой подсистемы.

Для каждой задачи, включенной в комплекс первоочередных задач, выполняются ее постановка и алгоритм решения. В этот раздел технического проекта включаются:

- организационно-экономическая сущность задачи (наименование, цель решения, краткое содержание, метод, периодичность и время решения задачи, способы сбора и передачи данных, связь задачи с другими задачами, характер использования результатов решения, в которых они используются);
- экономико-математическая модель задачи (структурная и развернутая форма представления);
- входная оперативная информация (характеристика показателей, их разрядность и диапазон изменения, формы представления);
- НСИ – содержание и формы представления;
- информация, хранимая для связи с другими задачами;
- информация, накапливаемая для последующих решений данной задачи;
- информация по внесению изменений (система внесения изменений и перечень информации, подвергающейся изменениям);
- алгоритм решения задачи (последовательность этапов расчета, схема, расчетные формулы);
- контрольный пример (набор заполненных данными форм входных документов, условные документы с накапливаемой и хранимой информацией, формы выходных документов, заполненные по результатам решения

экономико-технической задачи и в соответствии с разработанным алгоритмом расчета).

В документе «Расчет экономической эффективности системы» содержится сводная смета затрат, связанных с эксплуатацией систем, приводится расчет годовой экономической эффективности, источниками которой являются оптимизация производственной структуры хозяйства (объединения), снижение себестоимости продукции за счет рационального использования производственных ресурсов и уменьшения потерь, улучшения принимаемых управленческих решений.

В документе «Мероприятия по подготовке объекта к внедрению системы» приводятся перечень организационных мероприятий по совершенствованию сложившейся структуры управления, перечень работ по внедрению системы, которые необходимо выполнить на стадии рабочего проектирования, с указанием сроков и ответственных лиц.

Информационная часть технического проекта объединяет документы №№ 4 и 5. В документе «Организация информационной базы» отражаются:

- источники поступления информации и способы ее передачи для решения первоочередного комплекса функциональных задач;
- совокупность показателей, используемых в системе;
- состав документов, сроки и периодичность их поступления;
- основные проектные решения по организации фонда НСИ;
- состав НСИ, включая перечень реквизитов, их определение, разрядность, диапазон изменения и перечень документов НСИ;
- перечень массивов НСИ, их объем, порядок и частота корректировки информации;
- предложения по унификации документации, контрольный пример по внесению изменений в НСИ;
- структурная форма НСИ с описанием связи между элементами;
- требования к технологии создания и ведения фонда;
- методы хранения, поиска, внесения изменений и контроля, определения объемов и потоков информации НСИ.

«Альбом форм документов» содержит формы НСИ.

Математическая часть технического проекта содержит обоснование структуры математического обеспечения, обоснование выбора системы программирования, в том числе перечень стандартных программ.

Техническая часть технического проекта включает: описание и обоснование схемы технического процесса обработки данных; обоснование требований к разработке нестандартного оборудования; обоснование и выбор структуры комплекса технических средств и его функциональных групп; комплекс мероприятий по обеспечению надежности функционирования технических средств.

Рабочее проектирование заключается в разработке материалов, обеспечивающих эксплуатацию автоматизированной системы обработки информации. [22]

Рабочий проект – это техническая документация, утвержденная в установленном порядке, содержащая уточненные данные и детализированные общесистемные проектные решения, программы и инструкции по решению задач, а также уточненную оценку экономической эффективности автоматизированной системы управления и уточненный перечень мероприятий по подготовке объекта к внедрению.

Рабочий проект разрабатывается на основе технического проекта, утвержденного заказчиком.

На этапе рабочего проектирования заказчик должен закончить работы по подготовке объекта к внедрению системы, подготовить помещения для установки компьютеров, организовать учебу работников всех звеньев организационной структуры, разместить заказы на изготовление нестандартного оборудования.

Разработчиками на этом этапе создания системы уточняется сетевой график выполнения рабочего проекта, проводятся экспериментальные исследования для изыскания путей реализации принятых проектных решений, обосновываются дополнительные проектные решения, разрабатывается технологический процесс сбора и обработки информации, составляется рабочая документация, уточняются расчеты экономической эффективности системы.

В состав рабочей документации проекта входят документы:

1. Пояснительная записка.
2. Функциональная и организационная структура.
3. Должностные инструкции.
4. Инструкция по заполнению входных оперативных документов.
5. Инструкция по использованию выходных документов.
6. Инструкция по организации и ведению нормативно-справочной информации.
7. Инструкция по организации хранения информации в архиве.
8. Инструкция по подготовке информации к вводу в ПК.
9. Расчет экономической эффективности системы.
10. Мероприятия по подготовке объекта к внедрению.
11. Ведомость документов.

Экономико-организационная часть рабочего проекта содержит уточненный перечень задач, решаемых каждой подсистемой, с указанием периодичности и сроков их решения; инструкции каждому должностному лицу с описанием действий при нормальном режиме функционирования системы и при его нарушениях; порядок и правила использования входных документов и маршруты их движения.

Расчет экономической эффективности проводится на основе уточненных сметно-финансовых расчетов на создание системы. Мероприятия по подготовке объекта и внедрению системы включают общий перечень работ, наименование подразделений и ответственных исполнителей, срок исполнения и формы завершения отдельных этапов.

Информационная часть рабочего проекта включает материалы с перечнем показателей, используемых в задачах различных подсистем; описание порядка

формирования массивов информации; описание методов внесения изменений в информацию и методов организации контроля информации; перечень показателей, выдаваемых по запросу аппарата управления.

Кроме того, приводятся альбомы документов и рабочие инструкции: по формированию исходных данных для решения задач, по организации массивов информации, внесению в них изменений, по хранению и обновлению информации.

Математическая часть рабочего проекта содержит уточнение в составе экономико-математических моделей; описание методов, алгоритмы и программы решения задач; описание методов организации массивов информации; выбранную систему программирования; используемую операционную систему; библиотеку стандартных программ и инструкции для их использования; эталоны программ для решения задач и для работы с НСИ.

Техническая часть рабочего проекта предусматривает определение технических средств (конфигурация ПК, периферийные устройства, средства связи и передачи данных); описание технологического процесса обработки данных; расчет и составление графика загрузки комплекса технических средств; описание режима функционирования комплекса технических средств.

Рабочий проект служит основой для внедрения системы. Внедрение системы представляет собой процесс, включающий подготовку объекта, опытную эксплуатацию и приемку ИС в промышленную эксплуатацию.

Внедрение системы – это процесс постепенного перехода от существующей системы учета и анализа к новой, предусмотренной документацией рабочего проекта на всю систему. Внедрение отдельных задач и подсистем может проводиться параллельно с разработкой рабочего проекта на всю систему.

Основными этапами внедрения системы являются:

- подготовка объекта к внедрению системы;
- сдача задач и подсистем в опытную эксплуатацию;
- проведение опытной эксплуатации;
- сдача задач, подсистем, системы в целом в промышленную эксплуатацию.

Опытная эксплуатация ИС заключается в проверке алгоритмов, программ и звеньев технологического процесса обработки данных в реальных условиях. Она проводится для окончательной отладки программ и отработки технологического процесса решения задач; проверки подготовленности информационной базы; отработки взаимосвязи задач системы; приобретения навыков работы персоналом предприятия.

Сдача задач в опытную эксплуатацию осуществляется после представления рабочей документации заказчику и оформляется актом.

Началу опытной эксплуатации должно предшествовать издание приказа, определяющего степень участия и ответственность заказчика и разработчика, а также сроки ее проведения. Опытная эксплуатация задач проводится на основе реальной информации о производственно-финансовой деятельности предприятия в установленном режиме функционирования с дублированием

работ персонала объекта. На данном этапе разработчик проводит обучение персонала работе на компьютере по конкретным программам.

Срок проведения опытной эксплуатации устанавливается в каждом конкретном случае.

После окончания опытной эксплуатации задач составляется протокол о ходе и результатах опытной эксплуатации. Сдача задач в промышленную эксплуатацию оформляется актом, подписываемым заказчиком и разработчиком.

Опытная эксплуатация подсистем проводится в целях комплексной проверки всех ее элементов, подготовленности информационной базы, отладки технологического процесса сбора и обработки информации, обучения персонала работе в условиях функционирования подсистемы. Опытную эксплуатацию подсистемы следует осуществлять на основе полного объема реальной информации в установленном режиме функционирования с необходимым дублированием работ.

Сдача подсистемы в промышленную эксплуатацию проводится после сдачи в промышленную эксплуатацию задач пускового комплекса данной подсистемы. Опытная эксплуатация всей информационной системы проводится в целях комплексной проверки функционирования ее задач, проверки подготовленности обеспечивающей части системы к функционированию, окончательной отладки технологического процесса сбора и обработки информации.

Опытная эксплуатация системы должна осуществляться на основе необходимого объема информации о деятельности объекта в установленном режиме функционирования.

После окончания опытной эксплуатации системы составляется отчет о внедрении. При положительных результатах опытной эксплуатации система сдается в промышленную эксплуатацию.

В ходе промышленной эксплуатации ИС проводится анализ функционирования системы. Целями анализа функционирования системы являются проверка эффективности реализованных проектных решений в условиях ее промышленной эксплуатации, выработка рекомендаций по дальнейшему развитию системы и формирование типовых решений.

Анализ функционирования системы предусматривает проверку:

- функционирования технических средств;
- функционирования задач и подсистем в условиях автоматизированной обработки;
- действий персонала в условиях функционирования системы.

Результаты анализа используются для оценки качества системы и ее реальной экономической эффективности.

Работы по анализу функционирования системы проводятся разработчиком в порядке авторского надзора на основе договора с заказчиком после некоторого периода эксплуатации ИС (не менее 6 месяцев). Проведение авторского надзора осуществляется за счет средств, выделенных на создание системы.

Программа работ по проведению анализа составляется разработчиком и согласовывается с заказчиком. Анализ функционирования системы начинается после издания приказа о проведении этой работы. В приказе указываются сроки и объекты обследования (согласно программе), а также назначаются представители заказчика, участвующие в этой работе, и лица, ответственные за своевременное и полное представление необходимых материалов разработчику системы. Сбор всех данных (заполнение необходимых форм, регистрация в журнале и др.) осуществляется представителем заказчика и контролируется разработчиком. Накопленные данные передаются в сроки, указанные в программе, представителям разработчиков для разработки.

Результаты обработки данных по каждому исследованному элементу ИС (или групп однотипных элементов) протоколируются разработчиком с участием представителей заказчика. На основе оформленных протоколов разработчик после завершения всех работ, предусмотренных программой, составляет отчет по анализу функционирования ИС.

Сдача заказчику отчета по анализу функционирования системы является завершающим этапом работы разработчика.

В процессе анализа функционирования задач, подсистем и действий персонала в условиях внедрения ИС проводятся работы, аналогичные обследованию объекта по параметрам каждой функции подсистем ИС, с учетом применяемого комплекса технических средств и следующих факторов:

- своевременности поступления к управленческому персоналу необходимой информации;
- повышения достоверности информации;
- улучшения технико-экономических показателей работы предприятия.

Качество функционирования отдельных задач и подсистем оценивается по показателям достоверности и своевременности информации, повышения качества соответствующих управленческих решений.

По результатам анализа функционирования системы разрабатываются предложения для дальнейшего развития ИС.

### 3 Разработка альтернативного метода обработки информации

#### 3.1 Разработка метода контроля над процессом утверждения научно-методических публикаций

В соответствии с выбранной методикой, было проведено исследование деятельности отдела ИМО ИнЕУ. Результатом исследования явилось документированное описание процесса утверждения научно-методических публикаций, представленное на рис. 3.1.

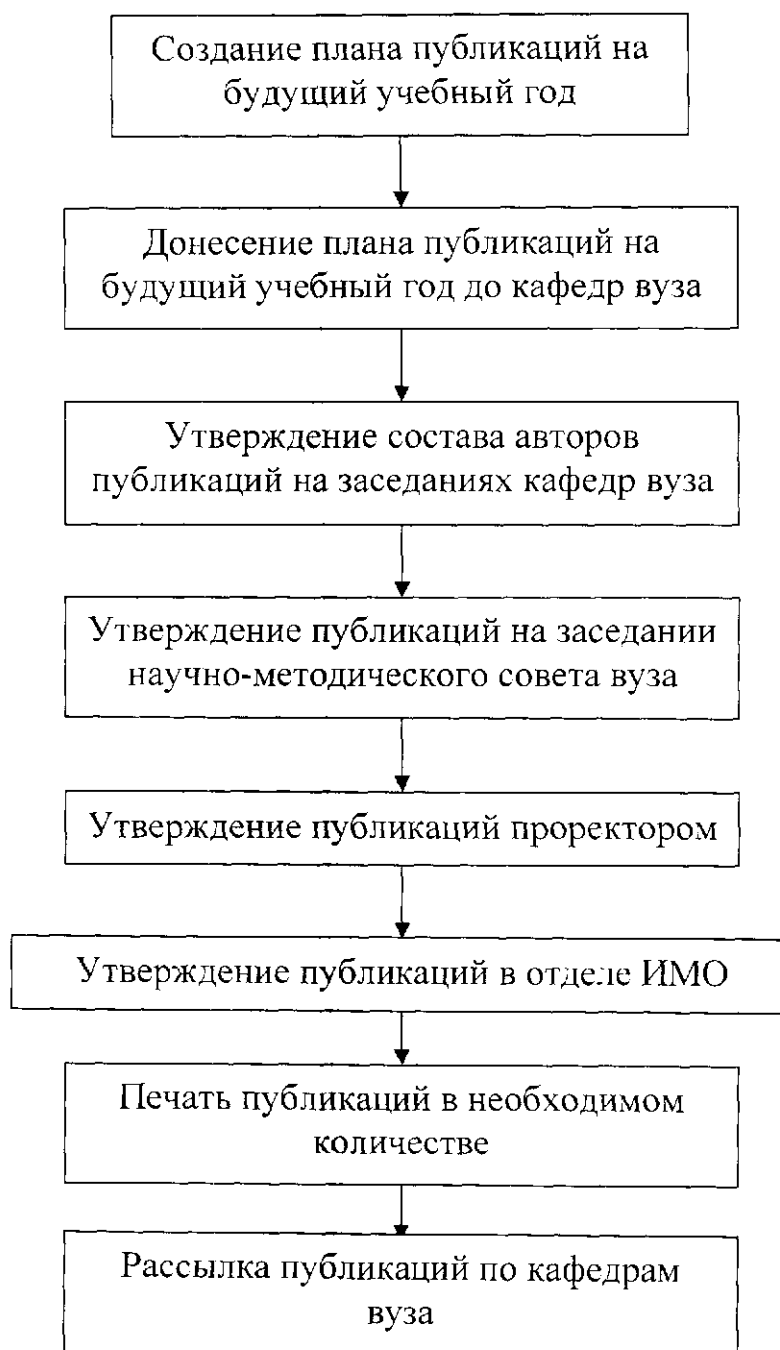


Рисунок 3.1 – Схема процесса утверждения научно-методических публикаций



Утверждение каждой научно-методической публикации проходит согласно рис. 3.1, отклонение от установленного порядка не допускается.

Вся информация, касающаяся продвижения документа по стадиям утверждения передается в отдел ИМО кураторами на кафедрах по телефону, либо при личной встрече. При этом не исключаются неточности при передаче информации, т.е. возникают случаи искажения передаваемой информации, к примеру, при перебоях в работе АТС ИнЕУ и т.п. Принимая во внимание занятость участников процесса, зачастую переданная информация уже не является актуальной, и, следовательно, снижается степень контроля над процессом утверждения научно-методических публикаций.

Наиболее ресурсоемкой относительно времени, среди перечисленных стадий процесса утверждения, является стадия утверждения публикации в ИМО. Это обусловлено тем, что куратор приносит документ в распечатанном виде и получает ответ на бланке установленной формы. Этот процесс повторяется до тех пор, пока документ не будет соответствовать всем нормам, принятым в ИМО.

Еще одним существенным моментом является то, что после утверждения авторов публикации на кафедре вуза, зачастую состав авторов впоследствии меняется. Смена авторского состава, вследствие загруженности персонала, не всегда адекватно отражается в документах ИМО, что вносит дополнительную неразбериху в процесс, снижает степень адекватности информации о публикации, что ведет к потере контроля над процессом со стороны работников ИМО.

Рациональная организация контроля над процессом утверждения научно-методических публикаций позволит в значительной мере сэкономить время и ресурсы, необходимые на его осуществление. Можно выделить следующие факторы, влияющие на ресурсоемкость процесса:

- использование бумажных версий документов в процессе утверждения в ИМО;
- личное присутствие кураторов;
- использование бумажных бланков в процессе утверждения;
- изменение информации о документах (публикациях) без уведомления работников ИМО.

Рассмотрев указанные факторы, можно предложить альтернативный метод контроля над процессом утверждения научно-методических публикаций. Выделим ключевые моменты альтернативного метода контроля над процессом утверждения:

1. Создание информационной системы контроля исполнения плана научно-методических публикаций.
2. Исключение личного присутствия кураторов в ИМО для утверждения публикации.
3. Использование электронной почты в качестве среды для передачи документов на утверждение в ИМО.
4. Ограничение прав кафедр вуза в процессе утверждения авторского состава научно-методических публикаций.

### 3.2 Практическая реализация метода обработки информации

В современном понимании ИС – это организационные, технические, программные и информационные средства, объединенные в единую систему с целью сбора, хранения, обработки и выдачи информации, предназначенной для решения определенного множества задач. [23] Информационная система контроля исполнения плана публикаций позволит:

- значительно сократить время, необходимое на составление плана публикаций;
- снизить вероятность возникновения ошибки при заполнении информации о публикациях;
- сократить количество используемой бумажной документации.
- использовать локальную сеть вуза в качестве рабочей среды для участников процесса, тем самым исключив личное присутствие кураторов в ИМО как необходимую составляющую процесса утверждения;
- контролировать ход процесса утверждения каждой публикации;
- ограничить возможности кафедр по редактированию информации о научно-методических публикациях.

В результате внедрения информационной системы отпадет необходимость в личном присутствии кураторов в ИМО, также сократится количество бумажной документации – стандартных бланков выдачи результата проверки. Подобная ИС будет уникальным решением, использование которого будет возможно только для решения подобного класса задач, т.к. для автоматизации работы вуза подходит далеко не любое, а специализированное ИТ-решение. [24]

Использование электронной почты в качестве среды для передачи документов в ИМО значительно сократит время, необходимое для передачи документа на проверку, избавит от необходимости личного присутствия куратора в ИМО и сократит количество бумажной документации – публикации сдаются на проверку в распечатанном виде.

Ограничение прав на изменение информации кафедрами вуза в значительной мере увеличит актуальности информации о публикациях, содержащейся в ИМО.

Информационная система контроля процесса публикации учебно-методических работ «ИСКРА» была создана с применением Microsoft .Net Framework 2.0 на базе СУБД Microsoft SQL Server 2005 и является 100% совместимой с операционной системой Microsoft Windows, используемой в вузе. Структурная схема информационной системы представлена на рис. 3.2.

База данных системы реализована на базе СУБД Microsoft SQL Server. В качестве программного средства, обеспечивающего функциональность СУБД, была выбрана бесплатная версия Microsoft SQL Server – Microsoft SQL Server 2005 Express Edition. Эта версия SQL Server бесплатна для загрузки и предоставляет достаточную функциональность для обеспечения работоспособности информационной системы «ИСКРА».

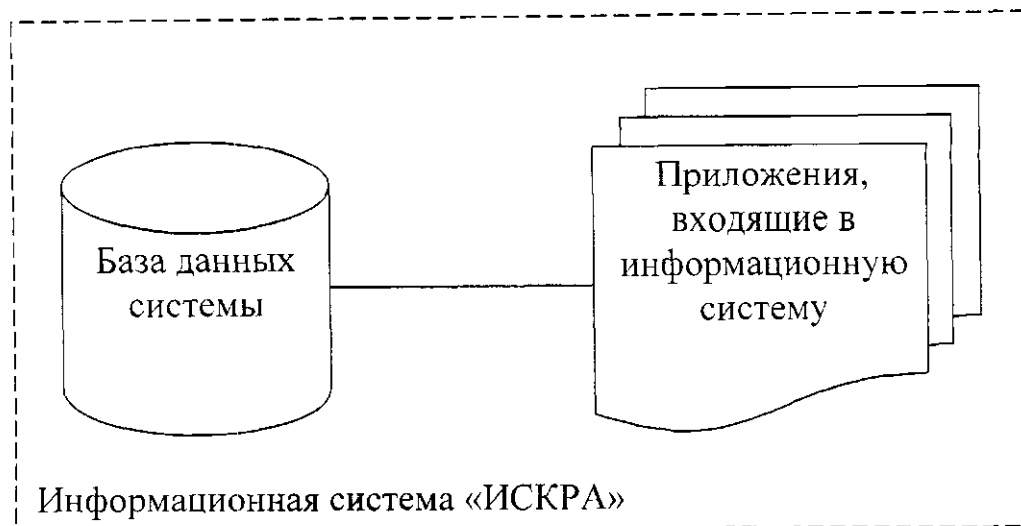


Рисунок 3.2 – Структурная схема информационной системы.

В качестве среды разработки была выбрана Microsoft Visual Studio 2005 Express Edition, язык разработки – C#.

C# изначально разрабатывался как простой и безопасный высокопроизводительный объектно-ориентированный язык программирования [25] для платформы .NET.

.NET Framework состоит из двух частей: CLR и FCL.

CLR предоставляет модель программирования, используемую во всех типах приложений. У CLR собственный загрузчик файлов, диспетчер памяти (сборщик мусора), система безопасности (безопасность доступа к коду), пул потоков и др. Кроме того, CLR предоставляет объектно-ориентированную модель программирования, определяющую, как выглядят и ведут себя типы и объекты.

FCL предоставляет объектно-ориентированный API-интерфейс, используемый всеми моделями приложений. В ней содержатся определения типов, которые позволяют разработчикам выполнять ввод/вывод, планирование задач в других потоках, создавать графические образы, сравнивать строки и т. п. Естественно, что все эти определения типов соответствуют существующей CLR в модели программирования.

CLR – основа .NET Framework и представляет собой виртуальную среду для исполнения приложений, находящуюся поверх операционной системы. Программный код, предназначенный для CLR называется управляемым кодом и состоит из команд псевдомашинного языка CIL. Команды CIL компилируются в машинный код по запросу в период выполнения. Компиляция метода происходит лишь один раз – при первом вызове, и затем кэшируется в памяти, чтобы впоследствии быть вызванным без задержки. [26] Несмотря на то, что компиляция по запросу отрицательно сказывается на производительности приложения, это компенсируется тем, что каждый вызываемый метод компилируется не более одного раза, а так же усилиями разработчиков по оптимизации компилятора.

CLR обладает следующими достоинствами: [27]

- единая программная модель, весь прикладной сервис представлен общей объектно-ориентированной программной моделью;
- упрощенная модель программирования CLR, избавляющая от недостатков, присущих работе с Win32 и COM;
- контроль версий, позволяющий изолировать прикладные компоненты, так что приложение всегда загружает компоненты, с которыми оно строилось и тестировалось;
- упрощенное развертывание, т.к. компоненты .NET Framework не связаны с реестром Windows и установка приложения зачастую представляет собой копирование исполняемых файлов в нужную директорию и создание ярлыков;
- многоплатформенность, обеспечиваемая генерацией компилятором команд CIL;
- интеграция языков программирования, означающая, что .NET-ориентированный язык может использовать типы, созданные на других языках;
- облегченное повторное использование кода, т.к. описанные выше механизмы позволяют создавать собственные классы, которые могут быть легко использованы в сторонних приложениях;
- автоматическое управление памятью, т.н. «сборка мусора» – CLR следит за использованием ресурсов приложением и в случае необходимости автоматически освобождает их;
- проверка безопасности типов – CLR может проверять безопасность использования типов в коде, что гарантирует корректное обращение к существующим типам;
- развитая поддержка отладки, позволяющая использовать разные языки при создании приложения;
- единый принцип обработки программных сбоев – CLR генерирует исключение при возникновении сбоя, содержащее детальную информацию о возникшей неполадке, упрощающую ее отладку;
- безопасность – доступ к программному коду может быть ограничен в соответствии с правилами безопасности, установленными системным администратором;
- взаимодействие с существующим кодом – в .NET Framework реализована полная поддержка доступа к COM-компонентам и Win32-функциям в существующих DLL.

Являясь надстройкой над операционной системой, платформа .NET Framework может быть использована в любой существующей операционной системе. [28]

Корпорация Microsoft выпустила следующие версии .NET Framework:

- версия 1.0 выпущена в 2002 г.;
- версия 1.1 выпущена в 2003 г.;
- версия 2.0 выпущена в 2005 г.;
- версия 3.0 выпущена в 2006 г.;
- версия 3.5 выпущена в 2007 г.;

– версия 4.0 анонсирована в 2008 г., публичная бета-версия выпущена 20 мая 2009 г.

При разработке ИС использовалась платформа .NET Framework 2.0., являющаяся 100% совместимой с Microsoft Windows XP SP2, используемой в ИнЕУ. Несмотря на обеспечение обратной совместимости, более старшие версии .NET Framework предназначены для операционных систем Windows Vista и старше. .NET Framework версии 2.0 поддерживает 32-разрядные версии x86, а также 64-разрядные версии x64 и IA64 ОС Windows. Для КПК (с ОС Windows CE) и небольших устройств предоставляется «облегченная» версия .NET Framework – .NET Compact Framework. 13 декабря 2001 года ЕСМА приняла в качестве стандартов язык программирования C#, определенные компоненты CLR и FCL. [29]

Язык программирования C# достаточно прост – в нем насчитывается порядка 80 ключевых слов и 10 основным типов данных. [30] Тем не менее, он обеспечивает полную поддержку структурного, компонентно-ориентированного и объектно-ориентированного программирования. При разработке этого языка корпорацией Microsoft был учтен опыт похожих языков программирования, развивавшихся в течение последних 20 лет, совместно с зарождением и развитием принципов объектно-ориентированного программирования. [31]

Синтаксически C# похож на Java и C++, однако он не является клоном этих языков. При его создании был учтен опыт разработки на этих языках, синтаксические конструкции были переработаны и улучшены. Синтаксические конструкции C# унаследованы не только от Java и C++, но и от Visual Basic. Например, в C#, как и в Visual Basic, используются свойства классов. Как C++, C# позволяет производить перегрузку операторов для созданных. C# – это, фактически, гибрид разных языков. При этом, C# синтаксически не менее чист, чем Java, так же прост, как Visual Basic, и обладает практически той же мощностью и гибкостью, что и C++. [32]

C# обладает следующими достоинствами:

- отсутствие необходимости работы с указателями, Однако в случае возникновения такой необходимости, в C# присутствуют возможности для работы с указателями;
- автоматическое управление памятью;
- наличие встроенных синтаксических конструкций для работы с перечислениями, структурами и свойствами классов;
- наличие возможности перегрузки операторов;
- полная поддержка использования программных интерфейсов;
- полная поддержка аспектно-ориентированных программных технологий. [33]

Особенностью языка программирования C# является его изначальная разработка как языка программирования для платформы .NET, что обеспечивает его высокую эффективность использования для создания приложений в среде .NET Framework. [34]

СУБД Microsoft SQL Server 2005, использованная для создания и управления базой данных ИС, совместима с .NET Framework и предоставляет программисту все возможности платформы:

- поддержка библиотек .NET Framework;
- встроенный CLR, позволяющий использовать для написания запросов не только Transact-SQL, но и другие .NET-ориентированные языки (C#, Visual Basic.NET);

Интеграция с CLR обладает следующими достоинствами: [35]

- расширение возможностей для программиста благодаря возможности использования объектно-ориентированных языков программирования и компонентов .NET Framework;
- повышение безопасности приложений;
- определение пользовательских типов и агрегатов;
- общая оболочка для разработки – возможности создания управляемых объектов для SQL Server 2005 присутствуют в Visual Studio 2005.

ИС «ИСКРА» построена на двухуровневой архитектуре «клиент-сервер». Использование данной архитектуры предназначено для разрешения проблем путем разделения компонентов приложений и размещения их там, где они будут функционировать наиболее эффективно. [36] Так, за управление базой данных и обеспечение работы логики системы отвечает СУБД, а клиентская часть ИС предназначена для установки на компьютер конечного пользователя.

База данных системы состоит из 14 таблиц, краткое описание которых представлено в таблице 3.1.

Таблица 3.1 – Краткое описание таблиц базы данных.

№	Имя таблицы	Краткое описание
1	ACADEMIES	Содержит список академий университета, используется для представления структуры университета в системе.
2	CHAIRS	Содержит список кафедр университета, используется для представления структуры университета в системе.
3	DISCIPLINES	Содержит список дисциплин, преподаваемых в университете, используется для представления структуры университета в системе.
4	DOCAUTHORS	Содержит связи типа «многие-ко-многим» между таблицами TEACHERS и DOCUMENTS, используется для представления списка авторов документа.

Продолжение таблицы 3.1.

5	DOCDISCIPLINE	Содержит связи типа «многие-ко-многим» между таблицами DISCIPLINES и DOCUMENTS, используется для представления списка авторов документа.
6	DOCHOURS	Содержит информацию о почасовой нагрузке по семестрам, используется для представления информации о почасовой нагрузке, используемой в документе.
7	DOCHOURTYPES	Содержит информацию о типах почасовой нагрузки. Служебная таблица, используется для обеспечения работоспособности системы.
8	DOCSPEC	Содержит связи типа «один-ко-многим» между таблицами SPECIALITIES и DOCUMENTS, используется для представления принадлежности документа к определенной дисциплине.
9	DOCTYPE	Содержит информацию о типах документов, используемых в системе. Служебная таблица, используется для обеспечения работоспособности системы.
10	DOCUMENTS	Содержит информацию о документах, присутствующих в системе. Основная таблица, используемая в системе.
11	FACULTIES	Содержит список факультетов университета, используется для представления структуры университета в системе.
12	IMOHISTORY	Содержит записи истории утверждения документа в отделе ИМО.
13	SPECIALITIES	Содержит список специальностей, присутствующих в университете, используется для представления структуры университета в системе.
14	TEACHERS	Содержит список преподавателей, работающих или работавших в университете, используется для представления авторства документа.

Рассмотрим назначение и структуру каждой из таблиц более детально.

В таблице ACADEMIES, структура которой представлена в таблице 3.2, хранится полная информация обо всех академиях университета, используемая в системе при работе с документами. Данная таблица является частью таблиц, обеспечивающих представление организационной структуры университета в информационной системе.

В таблице CHAIRS, структура которой представлена в таблице 3.3, хранится полная информация обо всех кафедрах университета, используемая в системе при работе с документами. Данная таблица является частью таблиц, обеспечивающих представление организационной структуры университета в информационной системе.

Таблица 3.2 – Структура полей таблицы ACADEMIES.

№	Название	Тип	Размер поля	Описание
1	acId	Целое число	Длинное целое	Уникальный численный идентификатор академии, используемый для обеспечения целостности связей.
2	acName	Текстовый	50	Название академии
3	acName4Doc	Текстовый	50	Название академии, используемое в документах
4	acDirector	Текстовый	50	Фамилия и инициалы директора академии

Таблица 3.3 – Структура полей таблицы CHAIRS.

№	Название	Тип	Размер поля	Описание
1	chairId	Целое число	Длинное целое	Уникальный численный идентификатор кафедры, используемый для обеспечения целостности связей.
2	chairName	Текстовый	50	Название академии
3	chairChair	Текстовый	50	Название академии, используемое в документах
4	chairFaculty	Целое число	Длинное целое	Уникальный численный идентификатор факультета, к которому относится кафедра, используемый для обеспечения межтабличных связей.

В таблице DISCIPLINES, структура которой представлена в таблице 3.4, хранится полная информация обо всех дисциплинах, преподаваемых в университете, используемая в системе при работе с документами. Данная таблица является частью таблиц, обеспечивающих представление организационной структуры университета в информационной системе.



В таблице DOCAUTHORS, структура которой представлена в таблице 3.5, хранится информация о списке авторов указанного документа. Данная таблица является частью таблиц, обеспечивающих представление документа в информационной системе.

В таблице DOCDISCIPLINE, структура которой представлена в таблице 3.6, хранится информация о принадлежности указанного документа к какой-либо из дисциплин, преподаваемых в вузе. Данная таблица является частью таблиц, обеспечивающих представление документа в информационной системе.

Таблица 3.4 – Структура полей таблицы DISCIPLINES.

№	Название	Тип	Размер поля	Описание
1	disciplineId	Целое число	Длинное целое	Уникальный численный идентификатор дисциплины, используемый для обеспечения целостности связей.
2	disciplineChair	Текстовый	50	Уникальный численный идентификатор кафедры, к которой относится дисциплина, используемый для обеспечения межтабличных связей.
3	disciplineName	Текстовый	150	Название дисциплины

Таблица 3.5 – Структура полей таблицы DOCAUTHORS.

№	Название	Тип	Размер поля	Описание
1	docId	Целое число	Длинное целое	Уникальный численный идентификатор документа, используемый для обеспечения межтабличных связей.
2	teacherId	Целое число	Длинное целое	Уникальный численный идентификатор преподавателя – автора документа, используемый для обеспечения межтабличных связей.
3	recID	Целое число	Длинное целое	Уникальный численный идентификатор записи, используемый для обеспечения целостности представления данных.

Таблица 3.6 – Структура полей таблицы DOCDISCIPLINE.

№	Название	Тип	Размер поля	Описание
1	docId	Целое число	Длинное целое	Уникальный численный идентификатор документа, используемый для обеспечения межтабличных связей.
2	discId	Целое число	Длинное целое	Уникальный численный идентификатор дисциплины, к которой относится документ, используемый для обеспечения межтабличных связей.
3	recID	Целое число	Длинное целое	Уникальный численный идентификатор записи, используемый для обеспечения целостности представления данных.

В таблице DOCHOURS, структура которой представлена в таблице 3.7, хранится информация о почасовой нагрузке, указанной в документе, по какой-либо из дисциплин, преподаваемых в вузе. Данная таблица является частью таблиц, обеспечивающих представление документа в информационной системе.

Таблица 3.7 – Структура полей таблицы DOCHOURS.

№	Название	Тип	Размер поля	Описание
1	recId	Целое число	Длинное целое	Уникальный численный идентификатор записи, используемый для обеспечения целостности представления данных.
2	docId	Целое число	Длинное целое	Уникальный численный идентификатор документа, к которому относится информация, используемый для обеспечения межтабличных связей.
3	hourType	Текстовый	80	Тип почасовой нагрузки.
4	doc1term	Целое число	Длинное целое	Количество часов нагрузки на 1 семестр
5	doc2term	Целое число	Длинное целое	Количество часов нагрузки на 2 семестр
6	doc3term	Целое число	Длинное целое	Количество часов нагрузки на 3 семестр

Продолжение таблицы 3.7.

7	doc4term	Целое число	Длинное целое	Количество часов нагрузки на 4 семестр
8	doc5term	Целое число	Длинное целое	Количество часов нагрузки на 5 семестр
9	doc6term	Целое число	Длинное целое	Количество часов нагрузки на 6 семестр
10	doc7term	Целое число	Длинное целое	Количество часов нагрузки на 7 семестр
11	doc8term	Целое число	Длинное целое	Количество часов нагрузки на 8 семестр

В таблице DOCHOURTYPES, структура которой представлена в таблице 3.8, хранится информация о типах почасовой нагрузки. Данная таблица является частью таблиц, обеспечивающих работоспособность информационной системы.

Таблица 3.8 – Структура полей таблицы DOCHOURTYPES.

№	Название	Тип	Размер поля	Описание
1	hourType	Текстовый	80	Тип почасовой нагрузки.

В таблице DOCSPEC, структура которой представлена в таблице 3.9, хранится информация о принадлежности указанного документа к какой-либо из специальностей, имеющих в вузе. Данная таблица является частью таблиц, обеспечивающих представление документа в информационной системе.

Таблица 3.9 – Структура полей таблицы DOCSPEC.

№	Название	Тип	Размер поля	Описание
1	docId	Целое число	Длинное целое	Уникальный численный идентификатор документа, используемый для обеспечения межтабличных связей.
2	specId	Целое число	Длинное целое	Уникальный численный идентификатор специальности, к которой относится документ, используемый для обеспечения межтабличных связей.
3	recID	Целое число	Длинное целое	Уникальный численный идентификатор записи, используемый для обеспечения целостности представления данных.

В таблице DOCTYPE, структура которой представлена в таблице 3.10, хранится информация о типах документов, используемых при добавлении в

базу данных нового документа. Данная таблица является частью таблиц, обеспечивающих работоспособность информационной системы.

Таблица 3.10 – Структура полей таблицы DOCTYPE.

№	Название	Тип	Размер поля	Описание
1	docType	Текстовый	80	Тип документа.

В таблице DOCUMENTS, структура которой представлена в таблице 3.11, хранится основная информация о документах, зарегистрированных в системе. Данная таблица является частью таблиц, обеспечивающих представление документа в информационной системе.

Таблица 3.11 – Структура полей таблицы DOCUMENTS.

№	Название	Тип	Размер поля	Описание
1	docId	Целое число	Длинное целое	Уникальный численный идентификатор документа, используемый для обеспечения межтабличных связей и целостности представления данных.
2	docType	Текстовый	50	Описание типа документа.
3	docTechnology	Текстовый	100	Описание образовательной технологии, к которой относится документ.
4	docCourse	Целое число	Короткое целое	Учебный курс, на который рассчитан документ
5	docAddedDate	Дата	-	Дата добавления документа в базу данных
6	docApprovedDate	Дата	-	Дата утверждения документа проректором вуза

Продолжение таблицы 3.11.

7	docEdForm	Текстовый	80	Образовательная форма, на которую рассчитан документ
8	docEdBase	Текстовый	100	Образовательная база, на которой строится документ
9	docGosStandart	Текстовый	500	Государственный образовательный стандарт, на котором основан документ
10	docTipProg	Текстовый	500	Типовая образовательная программа, на которой основан документ
11	docStatus	Текстовый	50	Статус документа
12	docNMSProtocolNum	Текстовый	50	№ протокола, в котором документ был утвержден на заседании научно-методического совета вуза
13	docNMSDate	Дата	-	Дата утверждения документа на заседании научно-методического совета вуза
14	docChairProtocolNum	Текстовый	50	№ протокола, в котором документ был утвержден на заседании кафедры вуза
15	docChairDate	Дата	-	Дата утверждения документа на заседании кафедры вуза
16	docControlDate	Дата	-	Контрольная дата, к которой документ должен пройти все инстанции и быть утвержден
17	docPrintDate	Дата	-	Дата отправления документа в отдел множительных и копировальных машин для печати

Продолжение таблицы 3.11.

18	docPrintReturnDate	Дата	-	Дата возврата печатных копий документа из отдела множительных и копировальных машин
19	docDeliveredDate	Дата	-	Дата рассылки печатных копий документа по кафедрам вуза
20	docControlType	Текстовый	100	Форма контроля по дисциплине, к которой относится документ
21	docCourseProject	Логический	2	Индикация наличия или отсутствия курсовой работы по дисциплине, к которой относится документ

В таблице FACULTIES, структура которой представлена в таблице 3.12, хранится информация о факультетах, имеющих в университете. Данная таблица является частью таблиц, обеспечивающих представление организационной структуры университета в информационной системе.

Таблица 3.12 – Структура полей таблицы FACULTIES.

№	Название	Тип	Размер поля	Описание
1	facultyId	Целое число	Длинное целое	Уникальный численный идентификатор факультета, используемый для обеспечения межтабличных связей и целостности представления данных.
2	facultyAcademy	Целое число	Длинное целое	Уникальный численный идентификатор академии, к которой относится факультет, используемый для обеспечения межтабличных связей.
3	facultyFullName	Текстовый	100	Полное название факультета.
4	facultyShortName	Текстовый	50	Сокращенное название (аббревиатура) факультета

Продолжение таблицы 3.12.

5	facultyDean	Текстовый	50	Фамилия и инициалы декана факультета
6	facultyNMSName	Текстовый	300	Название факультета, используемое в документе

В таблице IMOHISTORY, структура которой представлена в таблице 3.13, хранится информация о процессе утверждения документа в отделе ИМО. Данная таблица является частью таблиц, обеспечивающих представление документа в информационной системе.

Таблица 3.13 – Структура полей таблицы IMOHISTORY.

№	Название	Тип	Размер поля	Описание
1	docId	Целое число	Длинное целое	Уникальный численный идентификатор документа, к которому относится информация, используемый для обеспечения межтабличных связей.
2	historyId	Целое число	Длинное целое	Уникальный численный идентификатор записи, используемый для обеспечения целостности представления данных.
3	historyDate	Текстовый	80	Дата добавления записи
4	historyResult	Целое число	Длинное целое	Результат проверки документа
5	historyComment	Целое число	Длинное целое	Комментарий

В таблице SPECIALITIES, структура которой представлена в таблице 3.14, хранится информация о специальностях, существующих в вузе. Данная таблица является частью таблиц, обеспечивающих представление организационной структуры в информационной системе.

Таблица 3.14 – Структура полей таблицы SPECIALITIES.

№	Название	Тип	Размер поля	Описание
1	specId	Целое число	Длинное целое	Уникальный численный идентификатор специальности, используемый для обеспечения целостности представления данных.
2	specChair	Целое число	Длинное целое	Уникальный численный идентификатор кафедры, к которой относится специальность, используемый для обеспечения межтабличных связей.
3	specName	Текстовый	200	Название специальности
4	specCode	Текстовый	50	Шифр специальности

В таблице TEACHERS, структура которой представлена в таблице 3.15, хранится информация о преподавателях – авторах документа, работающих в вузе. Данная таблица является частью таблиц, обеспечивающих представление документа в информационной системе.

Таблица 3.15 – Структура полей таблицы TEACHERS.

№	Название	Тип	Размер поля	Описание
1	teacherId	Целое число	Длинное целое	Уникальный численный идентификатор преподавателя, используемый для обеспечения целостности представления данных.
2	teacherChair	Целое число	Длинное целое	Уникальный численный идентификатор кафедры, к которой относится преподаватель, используемый для обеспечения межтабличных связей.
3	teacherName	Текстовый	100	Фамилия и инициалы преподавателя
4	teacherStatus	Текстовый	200	Статус преподавателя

Основную часть функциональности информационной системы обеспечивают хранимые процедуры, расположенные на сервере баз данных. Основные операции, выполняемые процедурами:

- добавление новой информации;



- редактирование информации;
- удаление информации;
- получение информации по определенным критериям.

Все существующие процедуры можно разделить на четыре вида:

1. Редактирование структуры университета.
2. Редактирование информации о документе.
3. Редактирование служебной информации.
4. Вспомогательные процедуры.

Опишем подробнее каждую из групп процедур, а также процедуры, входящие в нее. Первая группа хранимых процедур представлена в таблице 3.16.

Таблица 3.16 – Хранимые процедуры, отвечающие за редактирование структуры университета.

№	Название	Кол-во параметров	Описание
1	addAcademy	3	Хранимая процедура, обеспечивающая добавление новой академии в организационную структуру университета.
2	editAcademy	4	Хранимая процедура, обеспечивающая редактирование информации об академиях, входящих в организационную структуру университета.
3	deleteAcademy	1	Хранимая процедура, обеспечивающая удаление информации об академиях, входящих в организационную структуру университета.
4	addChair	3	Хранимая процедура, обеспечивающая добавление новой кафедры в организационную структуру университета.
5	editChair	4	Хранимая процедура, обеспечивающая редактирование информации о кафедрах, входящих в организационную структуру университета.
6	deleteChair	1	Хранимая процедура, обеспечивающая удаление информации о кафедрах, входящих в организационную структуру университета.

Продолжение таблицы 3.16.

7	addDiscipline	2	Хранимая процедура, обеспечивающая добавление новой дисциплины, преподаваемой в университете.
8	editDiscipline	3	Хранимая процедура, обеспечивающая редактирование информации о дисциплинах, преподаваемых в университете.
9	deleteDiscipline	1	Хранимая процедура, обеспечивающая удаление информации о дисциплинах, преподаваемых в университете.
10	addFaculty	5	Хранимая процедура, обеспечивающая добавление нового факультета в организационную структуру университета.
11	editFaculty	6	Хранимая процедура, обеспечивающая редактирование информации о факультетах, входящих в организационную структуру университета.
12	deleteFaculty	1	Хранимая процедура, обеспечивающая удаление информации о факультетах, входящих в организационную структуру университета.
13	addSpec	3	Хранимая процедура, обеспечивающая добавление новой специальности.
14	editSpec	4	Хранимая процедура, обеспечивающая редактирование информации о специальностях, имеющихся в университете.
15	deleteSpec	1	Хранимая процедура, обеспечивающая удаление информации о специальностях, имеющихся в университете.
16	addTeacher	3	Хранимая процедура, обеспечивающая добавление нового преподавателя в список преподавателей университета.
17	editTeacher	4	Хранимая процедура, обеспечивающая редактирование информации о преподавателях, работающих и работавших в университете.
18	deleteTeacher	1	Хранимая процедура, обеспечивающая удаление информации о преподавателях, работающих и работавших в университете.

Следующая группа хранимых процедур, описанная в таблице 3.17, отвечает за добавление, редактирование и удаление информации о документах.

Таблица 3.17 – Хранимые процедуры, отвечающие за редактирование информации о документах.

№	Название	Кол-во параметров	Описание
1	addDocument	13	Хранимая процедура, обеспечивающая добавление нового документа в систему.
2	editDocument	14	Хранимая процедура, обеспечивающая редактирование информации о документах, имеющихся в системе.
3	deleteDocument	1	Хранимая процедура, обеспечивающая удаление информации о документах, имеющихся в системе.
4	docApproval	3	Хранимая процедура, обеспечивающая добавление отметки об утверждении выбранного документа проректором вуза.
5	docChairApproval	4	Хранимая процедура, обеспечивающая добавление отметки об утверждении выбранного документа на заседании кафедры вуза с указанием номера протокола заседания.
6	docDelivery	2	Хранимая процедура, обеспечивающая добавление отметки о рассылке выбранного документа по кафедрам вуза.
7	docMedia	2	Хранимая процедура, обеспечивающая добавление отметки о добавлении выбранного документа в медиатеку вуза.
8	docNMSApproval	4	Хранимая процедура, обеспечивающая добавление отметки об утверждении выбранного документа на заседании научно-методического совета факультета с указанием номера протокола заседания.

Продолжение таблицы 3.17.

9	docPrint	2	Хранимая процедура, обеспечивающая добавление отметки о передаче выбранного документа в отдел множительно-копировальной техники для создания необходимого количества печатных копий.
10	docPrintReturn	2	Хранимая процедура, обеспечивающая добавление отметки о возвращении выбранного документа из отдела множительно-копировальной.
11	getDocAuthors	1	Хранимая процедура, обеспечивающая выборку списка авторов документа.
12	getDocsBySpec	1	Хранимая процедура, обеспечивающая выборку списка документов, относящихся к выбранной специальности.
13	getIMOHistory	1	Хранимая процедура, обеспечивающая выборку списка записей, относящихся к истории утверждения выбранного документа в отделе ИМО.
14	setDocAuthors	2	Хранимая процедура, обеспечивающая добавление преподавателя к списку авторов документа.
15	setDocHours	10	Хранимая процедура, обеспечивающая добавление в базу данных информации о почасовой нагрузке, указанной в документе.
16	clearDocAuthors	1	Хранимая процедура, обеспечивающая удаление из базы данных списка авторов документа.
17	clearDocHours	1	Хранимая процедура, обеспечивающая удаление из базы данных информации о почасовой нагрузке, указанной в документе.

Продолжение таблицы 3.17.

18	addIMOHHistory	4	Хранимая процедура, обеспечивающая добавление в базу данных информации о процессе утверждения документа в отделе ИМО.
19	editIMOHHistory	3	Хранимая процедура, обеспечивающая редактирование хранящейся в базе данных информации о процессе утверждения документа в отделе ИМО.
20	deleteIMOHHistory	1	Хранимая процедура, обеспечивающая удаление хранящейся в базе данных информации о процессе утверждения документа в отделе ИМО.

За редактирование служебной информации отвечают хранимые процедуры, перечисленные в таблице 3.18.

Таблица 3.18 – Хранимые процедуры, отвечающие за редактирование служебной информации.

№	Название	Кол-во параметров	Описание
1	addHourType	1	Хранимая процедура, обеспечивающая добавление нового типа почасовой нагрузки в базу данных.
2	addDocType	1	Хранимая процедура, обеспечивающая добавление нового типа документа в базу данных.
3	editHourTypes	2	Хранимая процедура, обеспечивающая редактирование информации о типах почасовой нагрузки, хранящейся в базе данных.
4	editDocType	2	Хранимая процедура, обеспечивающая редактирование информации о типах документов, хранящейся в базе данных.
5	deleteHourType	1	Хранимая процедура, обеспечивающая удаление информации о типах почасовой нагрузки, хранящейся в базе данных.

Продолжение таблицы 3.18.

6	deleteDocType	1	Хранимая процедура, обеспечивающая удаление информации о типах документов, хранящейся в базе данных.
---	---------------	---	------------------------------------------------------------------------------------------------------

Последний тип хранимых процедур, перечисленных в таблице 3.19 – вспомогательные хранимые процедуры, используемые в других хранимых процедурах и обеспечивающие реализацию логики работы системы.

Таблица 3.19 – Вспомогательные хранимые процедуры.

№	Название	Кол-во параметров	Описание
1	deleteDocsByAcademy	1	Хранимая процедура, обеспечивающая удаление информации о документах, относящихся к указанной академии.
2	deleteDocsByChair	1	Хранимая процедура, обеспечивающая удаление информации о документах, относящихся к указанной кафедре.
3	deleteDocsByDiscipline	1	Хранимая процедура, обеспечивающая удаление информации о документах, относящихся к указанной дисциплине.
4	deleteDocsByFaculty	1	Хранимая процедура, обеспечивающая удаление информации о документах, относящихся к указанному факультету.
5	deleteDocsBySpeciality	1	Хранимая процедура, обеспечивающая удаление информации о документах, относящихся к указанной специальности.

Одним из факторов, обеспечивающих работу логики системы и целостность данных, хранящихся в базе данных, являются межтабличные связи. В базе данных используются два основным типа связей:

- один-ко-многим;
- многие-ко-многим.

Структура межтабличных связей (диаграмма базы данных) показана в приложении А.

Безопасность при работе с базой данных обеспечивается средствами Microsoft SQL Server – для подключения к серверу баз данных необходимо указать имя пользователя и пароль. В системе присутствуют три основных типа пользователей:

- работник отдела ИМО, серверное имя входа – imo\_admin;
- работник кафедры, серверное имя входа – chair\_user;
- куратор кафедры, серверное имя входа – curator.

Права пользователей распределены в соответствии с исполняемыми ими обязанностями по редактированию информации. Ограничения на доступ к информации касаются как таблиц базы данных, так и отдельных столбцов таблиц.

Для работы с базой данных был разработан комплекс программного обеспечения, получивший название «Информационная Система «ИСКРА». Информационная система состоит из трех основных и одного служебного модуля, выполняющих следующие функции:

1. Модуль «АРМ ИМО» – автоматизированное рабочее место работника ИМО. Предоставляет полный доступ к информационному содержанию БД (добавление, редактирование, удаление любой информации, касающейся утверждения публикации), поиск необходимой информации, составление плана публикаций, составление отчетов.

2. Модуль «Кафедра» – модуль для работников кафедры ИнЕУ. Предоставляет доступ к информации, касающейся публикаций кафедры (доступ ограничен рамками соответствующей кафедры), позволяет редактировать информацию, касающуюся утверждения публикации на заседании совета кафедры (утверждение списка авторов, указанием номера протокола и даты проведения заседания совета кафедры и т.д.).

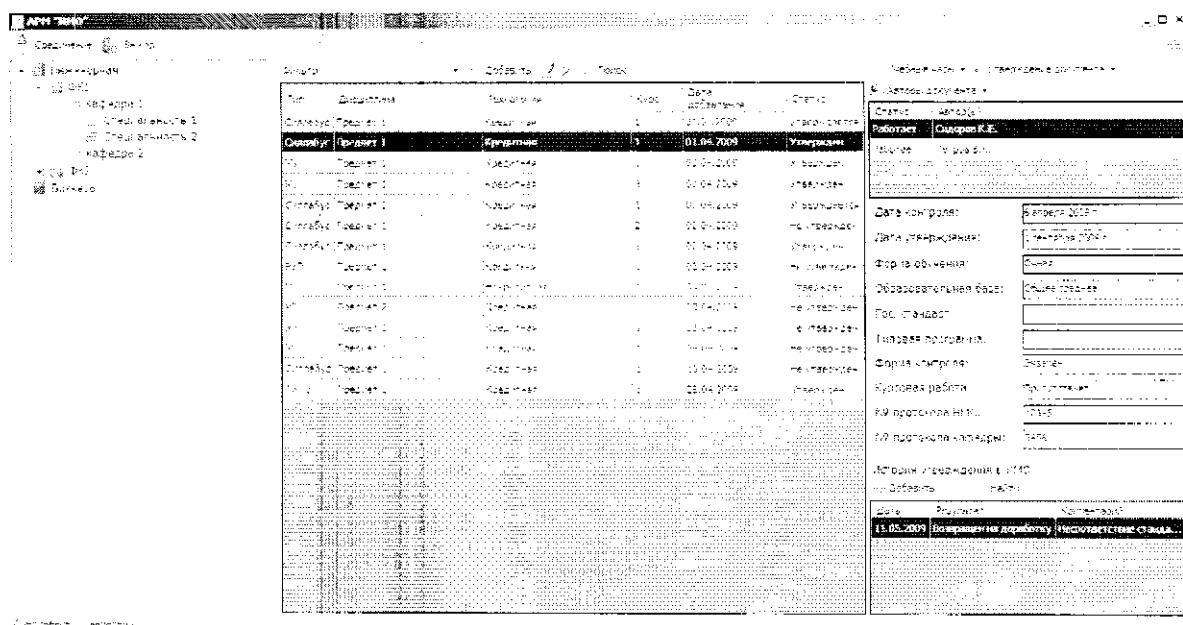
3. Модуль «Куратор» – модуль для работников кафедры, ответственных за взаимодействие с ИМО. Предоставляет доступ к информации, касающейся публикаций кафедры (доступ ограничен рамками соответствующей кафедры), позволяет редактировать информацию, касающуюся процесса утверждения публикации в ИМО (дата внесения на рассмотрение, результат рассмотрения, комментарии).

4. Модуль «Конфигурация системы» – служебный модуль для редактирования информации, касающейся структуры подразделений вуза, типа публикаций и прочих данных, обеспечивающих работу информационной системы.

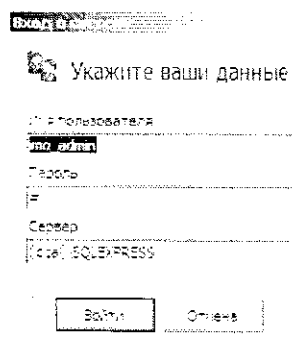
В плане удобства пользовательского интерфейса и облегчения поиска необходимой информации система «ИСКРА» обладает следующими особенностями:

- цветовое выделение записей в зависимости от статуса документа («Не утвержден», «Утверждается» и «Утвержден»);

- Основная часть информационной системы – модуль «Автоматизированное рабочее места работника ИМО» (АРМ «ИМО»). Модуль АРМ «ИМО» позволяет работать с документами, составляющими основу для создания плана издания научно-методической документации кафедрами вуза. Для удобства навигации, в программе наглядно представлена структура университета в древовидной форме, с указанием списка специальностей, относящихся к кафедрам. Главное окно программы представлено на рис. 3.3.



Для начала работы с программой пользователю необходимо подключиться к серверу баз данных. Подключение осуществляется нажатием на кнопку «Соединение», после чего на экране появляется форма подключения к серверу с указанием имени пользователя, пароля и сетевого адреса сервера, показанная на рис. 3.4.



57



Добавление нового документа происходит путем нажатия на кнопку «Добавить», после чего появляется окно добавления нового документа, где пользователю необходимо ввести базовую информацию о документе. Форма добавления нового документа представлена на рис. 3.5.

**Новый документ**

Детали документа

Тип документа: РИП

Специальность: Специальность 1

Дисциплина: Предмет 1

Технология: Кредитная

Курс: 1

Дата контроля: 11 мая 2009 г.

Форма обучения: Очная

Образовательная база: Общее среднее

Гос. стандарт:

Типовая программа:

Курсовая работа: ☐ Присутствует

Форма контроля:

Сохранить Отмена

Рисунок 3.5 – Форма добавления нового документа

После этого документ добавляется в список документов с присвоенным статусом «Не утвержден». Далее в процессе работы с программой пользователь может редактировать различную информацию, относящуюся к процессу утверждения документа. Информация об учебной нагрузке редактируется путем выбора пункта «Редактировать» в меню «Учебная нагрузка» (рис. 3.6). Состав авторов документа редактируется путем выбора пункта «Редактировать» в меню «Авторы документа» (рис. 3.7). Информация об утверждении документа редактируется путем выбора пункта «Редактировать» в меню «Авторы документа» (рис. 3.8).

**Редактирование учебной нагрузки**

Сем.	1 сем.	2 сем.	3 сем.	4 сем.	5 сем.	6 сем.	7 сем.	Всего
Полная	0	0	0	0	0	0	0	0
Полуполная	0	0	0	0	0	0	0	0
ОГЭ	0	0	0	0	0	0	0	0

Добавить Удалить Очистить

Сохранить Отмена

Рисунок 3.6 – Форма редактирования учебной нагрузки

При добавлении новой строки, пользователь из выпадающего списка выбирает тип учебной нагрузки и заполняет количество часов по семестрам.

Рисунок 3.7 – Форма редактирования списка авторов документа

В левом списке показываются преподаватели, относящиеся к кафедре вуза. Добавление происходит путем нажатия кнопки «Добавить», либо путем нажатия кнопки «Добавить всех». При попытке добавления преподавателей со статусом «Уволен(а)», программа выдает запрос на подтверждение действия.

Рисунок 3.8 – Форма редактирования деталей процесса утверждения документа

При утверждении документа в одной из инстанций, напротив соответствующей инстанции ставится отметка и указывается дата утверждения. При попытке поставить отметку с нарушением порядка утверждения (например, установить отметку о получении копий раньше, чем документ утвержден проректором), программа выдает сообщение с уведомлением о

нарушении порядка. После установки отметок об утверждении на заседании кафедры и НМС, документу присваивается статус «Утверждается». После утверждения проректором документ считается утвержденным и ему присваивается статус «Утвержден».

Поиск по документам осуществляется путем вызова формы поиска кнопкой «Поиск». Форма поиска показана на рис. 3.9.

Поиск по документам

Условие поиска

Поле: кафедра

Критерий: Содержание

Условие: инженер

Найти Закрыть

Переключить: Инженерная (ЭК) Кафедра 1 Специальность 1

Академия	Факультет	Кафедра	Специальность	Курс	Тип	Технология	Форма	Дата
Инженерная	ЭК1	Кафедра 1	Специальность 1	1	Силлабус	Кредитная	Очная	01.04.2009
Инженерная	ЭК1	Кафедра 1	Специальность 2	2	МУ	Своб	Свободная	01.04.2009
Инженерная	ЭК1	Кафедра 1	Специальность 1	1	Силлабус	Кредитная	Очная	01.04.2009
Инженерная	ЭК1	Кафедра 1	Специальность 1	1	МУ	Кредитная	Свободная	01.04.2009
Инженерная	ЭК1	Кафедра 1	Специальность 1	1	Силлабус	Кредитная	Очная	01.04.2009
Инженерная	ЭК1	Кафедра 1	Специальность 1	1	Силлабус	Кредитная	Очная	01.04.2009
Инженерная	ЭК1	Кафедра 1	Специальность 1	1	Силлабус	Кредитная	Очная	01.04.2009
Инженерная	ЭК1	Кафедра 1	Специальность 1	1	МУ	Кредитная	Очная	01.04.2009

Найдено документов: 13

Рисунок 3.9 – Форма поиска по документам

Из списка «Поиск» пользователь указывает поле, по которому будет осуществляться поиск среди документов. Из списка «Критерий» пользователь выбирает критерий совпадения условия поиска с информацией, содержащейся в базе данных.

Второй основной модуль информационной системы – модуль «Конфигурация системы». Этот модуль предназначен для редактирования информации, составляющей основу работоспособности всей информационной системы. Данный модуль не предназначен для использования обычными пользователями системы – пользоваться им должен только администратор системы, т.к. при неправильном использовании данного модуля присутствует риск потери информации, содержащейся в базе данных.

На рис. 3.10 представлена главная форма программы.

Для начала работы с программой пользователю необходимо установить подключение к серверу баз данных. Эта процедура является стандартной при начале работы с любым из модулей информационной системы. Форма подключения к серверу показана на рис. 3.4.

Дальнейшая работа с модулем происходит путем выбора необходимой закладки из списка закладок и редактирования информации, относящейся к этой части системы.

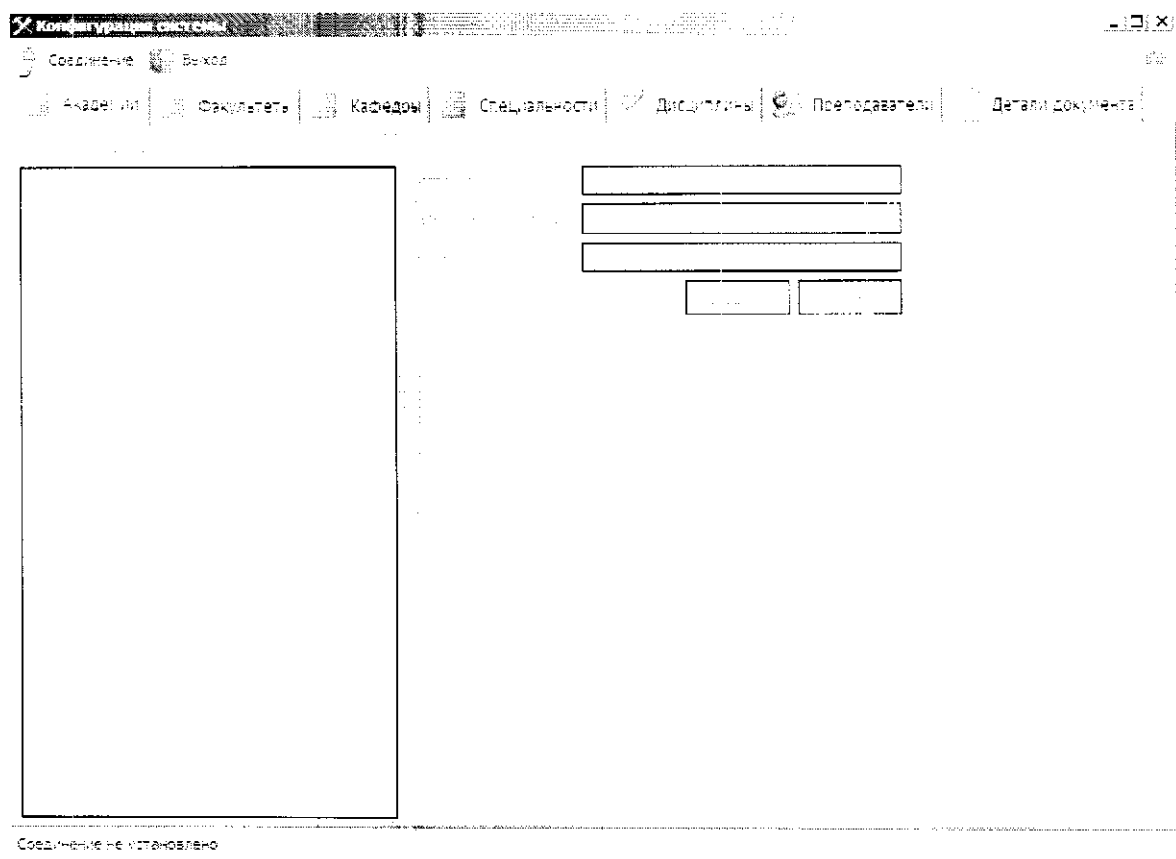


Рисунок 3.10 – Главное окно модуля «Конфигурация системы»

Основное предназначение модуля – редактирование структуры университета и деталей документов, используемых при добавлении новых документов в базу данных.

Закладки, предназначенные для работы со структурой университета:

- Академии;
- Факультеты;
- Кафедры;
- Специальности.

Для редактирования списка дисциплин, к которым может принадлежать документ, предназначена закладка «Дисциплины».

Для редактирования списка преподавателей – авторов документа – предназначена закладка «Преподаватели».

Закладка «Детали документа» предназначена для редактирования двух видов деталей документа:

- видов почасовой нагрузки;
- типов документов.

При разработке модулей информационной системы был использован объектно-ориентированный подход и в основе функциональности приложений лежит класс для работы с базой данных. Для каждого из модулей приведем описание классов. Описание структур данных, принимающихся в качестве параметров при вызове методов и в качестве возвращаемых методами значений, приведено в таблице 3.20.

Таблица 3.20 – Описание структур данных.

№	Название	Кол-во полей	Описание
1	Academy	4	Структура данных, представляющая академию университета. Поля структуры представляют идентификатор академии, название академии, название академии для документа, ФИО директора академии
2	Chair	4	Структура данных, представляющая кафедру университета. Поля структуры представляют идентификатор кафедры, идентификатор факультета, к которому принадлежит кафедра, название кафедры, ФИО заведующего кафедрой.
3	Faculty	6	Структура данных, представляющая факультет университета. Поля структуры представляют идентификатор факультета, идентификатор академии, к которой принадлежит факультет, полное название факультета, краткое название факультета, ФИО декана факультета.
4	loginInfo	3	Структура данных, представляющая информацию о подключении к серверу баз данных. Поля структуры представляют имя пользователя, пароль, сетевой адрес сервера.
5	Speciality	4	Структура данных, представляющая специальность, существующую на кафедре университета. Поля структуры представляют идентификатор специальности, идентификатор кафедры, к которой относится специальность, название специальности, шифр специальности.
6	Discipline	2	Структура данных, представляющая дисциплину, преподаваемую в университете. Поля структуры представляют идентификатор дисциплины, название дисциплины.
7	Author	4	Структура данных, представляющая преподавателя университета. Поля структуры представляют идентификатор преподавателя, идентификатор кафедры, ФИО преподавателя, статус преподавателя.

8	theDoc	14	Структура данных, представляющая документ. Поля структуры представляют идентификатор документа, тип документа, образовательную технологию, курс, форму обучения, образовательную базу, типовую программу, государственный образовательный стандарт, дату контроля, идентификатор дисциплины, идентификатор специальности, идентификатор кафедры, наличие курсового проекта, форму контроля.
9	imoHistory	4	Структура данных, представляющая запись из истории утверждения документа в отделе ИМО. Поля структуры представляют идентификатор записи, идентификатор документа, результат, комментарий.
10	docHours	9	Структура данных, представляющая информацию о почасовой нагрузке, указанной в документе. Поля структуры представляют обозначение типа почасовой нагрузки, количество часов на каждый из 8 семестров.
11	docApproval	10	Структура данных, представляющая движение документа по различным инстанциям в процессе утверждения. Поля структуры представляют идентификатор документа, дату утверждения документа на заседании кафедры, номер протокола заседания кафедры, дату утверждения документа на заседании НМС факультета, номер протокола заседания НМС, дату утверждения документа проректором, дату отправления документа в отдел множительно-копировальной техники, дату возвращения документа из отдела множительно-копировальной техники, дату рассылки копий документа по кафедрам вуза, дату отправления копии документа в медиатеку вуза.
12	searchResult	5	Структура данных, представляющая строку результат поиска по документам. Поля структуры представляют идентификатор академии, идентификатор факультета, идентификатор кафедры, идентификатор специальности, идентификатор документа.

Класс `clsDBWorker` модуля «АРМ «ИМО» состоит из 20 public методов, одного конструктора, одного private поля данных и одного public свойства для доступа к private полю.

Поля данных класса:

- `private SqlConnection con` – переменная типа `SqlConnection` для хранения подключения к серверу баз данных.

Свойства класса:

- `public SqlConnection connection` – свойство, предоставляющее доступ к private полю данных `con`. Поддерживает только чтение, без возможности изменения поля (реализован только метод `get`).

Конструкторы класса:

- `public clsDBWorker(string serverPath, string username, string userpassword)` – конструктор класса, принимающий три параметра: путь к серверу, имя пользователя и пароль пользователя. При создании экземпляра класса в теле конструктора создается подключение к серверу баз данных и, если подключение прошло успешно, экземпляр соединения сохраняется в поле данных `con`.

Методы класса:

- `public Academy[] getAcademies()` – метод, отвечающий за получение списка академий университета. Возвращает в качестве результата работы массив элементов типа `Academy`, который представляет собой список академий университета;

- `public Faculty[] getFaculties(int acId)` – метод, отвечающий за получение списка факультетов для указанной академии. В качестве параметров принимает числовое значение, обозначающее идентификатор академии в базе данных системы. Возвращает в качестве результата работы массив элементов типа `Faculty`, который представляет собой список факультетов академии;

- `public Chair[] getChairs(int facId)` – метод, отвечающий за получение списка кафедр для указанного факультета. В качестве параметров принимает числовое значение, обозначающее идентификатор факультета в базе данных системы. Возвращает в качестве результата работы массив элементов типа `Chair`, который представляет собой список кафедр факультета;

- `public Speciality[] getSpecialities(int chairId)` – метод, отвечающий за получение списка специальностей для указанной кафедры. В качестве параметров принимает числовое значение, обозначающее идентификатор кафедры в базе данных системы. Возвращает в качестве результата работы массив элементов типа `Speciality`, который представляет собой список специальностей кафедры;

- `public Discipline[] getDisciplines(int chairId)` – метод, отвечающий за получение списка дисциплин, преподаваемых на указанной кафедре. В качестве параметров принимает числовое значение, обозначающее идентификатор кафедры в базе данных системы. Возвращает в качестве результата работы массив элементов типа `Discipline`, который представляет собой список дисциплин, преподаваемых на указанной кафедре;

– `public Author[] getAuthors(int chairId)` – метод, отвечающий за получение списка преподавателей указанной кафедры. В качестве параметров принимает числовое значение, обозначающее идентификатор кафедры в базе данных системы. Возвращает в качестве результата работы массив элементов типа `Author`, который представляет собой список преподавателей указанной кафедры;

– `public Author[] getDocAuthors(int docId)` – метод, отвечающий за получение списка преподавателей – авторов указанного документа. В качестве параметров принимает числовое значение, обозначающее идентификатор документа в базе данных системы. Возвращает в качестве результата работы массив элементов типа `Author`, который представляет собой список авторов указанного документа;

– `public string[] getDocTypes()` – метод, отвечающий за получение списка типов документов из базы данных. Возвращает в качестве результата массив строковых элементов;

– `public string[] getHourTypes()` – метод, отвечающий за получение списка типов почасовой нагрузки из базы данных. Возвращает в качестве результата массив строковых элементов;

– `public docHours[] getDocHours(int docId)` – метод, отвечающий за получение информации о почасовой нагрузке, указанной в документе. В качестве параметров принимает числовое значение, обозначающее идентификатор документа в базе данных системы. Возвращает в качестве результата работы массив элементов типа `docHours`, который представляет собой информацию о почасовой нагрузке;

– `public void addDocument(theDoc doc)` – метод, обеспечивающий добавление нового документа в систему. В качестве параметров принимает структуру типа `theDoc`, содержащую информацию о новом документе;

– `public void editDocument(theDoc doc)` – метод, обеспечивающий редактирование информации о документе, имеющимся в системе. В качестве параметров принимает структуру типа `theDoc`, содержащую информацию о документе;

– `public void deleteDocument(int docId)` – метод, обеспечивающий удаление информации о документе, имеющимся в системе. В качестве параметров принимает числовое значение, обозначающее идентификатор документа в базе данных системы.

– `public void addIMOHistory(imoHistory historyRec)` – метод, обеспечивающий добавление новой записи к истории утверждения документа в отделе ИМО. В качестве параметров принимает структуру типа `imoHistory`, содержащую всю необходимую информацию;

– `public void editIMOHistory(imoHistory historyRec)` – метод, обеспечивающий редактирование записи из истории утверждения документа в отделе ИМО. В качестве параметров принимает структуру типа `imoHistory`, содержащую всю необходимую информацию;



– `public void deleteIMOHHistory(int historyId)` ) – метод, обеспечивающий удаление записи из истории утверждения документа в отделе ИМО. В качестве параметров принимает числовое значение, обозначающее идентификатор записи в базе данных системы.

– `public void clearDocAuthors(int docId)` – метод, обеспечивающий удаление списка авторов указанного документа. В качестве параметров принимает числовое значение, обозначающее идентификатор документа в базе данных системы.

– `public void setDocAuthors(int docId, int[] authors)` – метод, обеспечивающий добавление авторов к списку авторов указанного документа. В качестве параметров принимает числовое значение, обозначающее идентификатор документа в базе данных системы и массив числовых значений, состоящий из идентификаторов преподавателей;

– `public void docApproval(docApproval doc)` – метод, обеспечивающий редактирование информации о процессе утверждения документа в различных инстанциях вуза. В качестве параметров принимает структуру типа `docApproval`, содержащую всю необходимую информацию;

– `public void setDochoours(int docId, docHours[] hours)` – метод, обеспечивающий добавление информации о почасовой нагрузке, указанной в документе. В качестве параметров принимает числовое значение, обозначающее идентификатор документа в базе данных системы и массив структур `docHours`, содержащий всю необходимую информацию.

Класс `clsDBWorker` модуля «Конфигурация системы» состоит из 32 `public` методов, одного конструктора, одного `private` поля данных и одного `public` свойства для доступа к `private` полю.

Поля данных класса:

– `private SqlConnection con` – переменная типа `SqlConnection` для хранения подключения к серверу баз данных.

Свойства класса:

– `public SqlConnection connection` – свойство, предоставляющее доступ к `private` полю данных `con`. Поддерживает только чтение, без возможности изменения поля (реализован только метод `get`).

Конструкторы класса:

– `public clsDBWorker(string serverPath, string username, string userpassword)` – конструктор класса, принимающий три параметра: путь к серверу, имя пользователя и пароль пользователя. При создании экземпляра класса в теле конструктора создается подключение к серверу баз данных и, если подключение прошло успешно, экземпляр соединения сохраняется в поле данных `con`.

Методы класса:

– `public Academy[] getAcademies()` – метод, отвечающий за получение списка академий университета. Возвращает в качестве результата работы массив элементов типа `Academy`, который представляет собой список академий университета;

- `public Faculty[] getFaculties(int acId)` – метод, отвечающий за получение списка факультетов для указанной академии. В качестве параметров принимает числовое значение, обозначающее идентификатор академии в базе данных системы. Возвращает в качестве результата работы массив элементов типа `Faculty`, который представляет собой список факультетов академии;
- `public Chair[] getChairs(int facId)` – метод, отвечающий за получение списка кафедр для указанного факультета. В качестве параметров принимает числовое значение, обозначающее идентификатор факультета в базе данных системы. Возвращает в качестве результата работы массив элементов типа `Chair`, который представляет собой список кафедр факультета;
- `public Speciality[] getSpecialities(int chairId)` – метод, отвечающий за получение списка специальностей для указанной кафедры. В качестве параметров принимает числовое значение, обозначающее идентификатор кафедры в базе данных системы. Возвращает в качестве результата работы массив элементов типа `Speciality`, который представляет собой список специальностей кафедры;
- `public Discipline[] getDisciplines(int chairId)` – метод, отвечающий за получение списка дисциплин, преподаваемых на указанной кафедре. В качестве параметров принимает числовое значение, обозначающее идентификатор кафедры в базе данных системы. Возвращает в качестве результата работы массив элементов типа `Discipline`, который представляет собой список дисциплин, преподаваемых на указанной кафедре;
- `public Author[] getAuthors(int chairId)` – метод, отвечающий за получение списка преподавателей указанной кафедры. В качестве параметров принимает числовое значение, обозначающее идентификатор кафедры в базе данных системы. Возвращает в качестве результата работы массив элементов типа `Author`, который представляет собой список преподавателей указанной кафедры;
- `public Author[] getDocAuthors(int docId)` – метод, отвечающий за получение списка преподавателей – авторов указанного документа. В качестве параметров принимает числовое значение, обозначающее идентификатор документа в базе данных системы. Возвращает в качестве результата работы массив элементов типа `Author`, который представляет собой список авторов указанного документа;
- `public string[] getDocTypes()` – метод, отвечающий за получение списка типов документов из базы данных. Возвращает в качестве результата массив строковых элементов;
- `public string[] getHourTypes()` – метод, отвечающий за получение списка типов почасовой нагрузки из базы данных. Возвращает в качестве результата массив строковых элементов;
- `public void addAcademy(Academy ac)` – метод, отвечающий за добавление новой академии в организационную структуру университета. В качестве параметров принимает структуру типа `Academy`, содержащую всю необходимую информацию;

- `public void editAcademy(Academy ac)` – метод, отвечающий за редактирование информации об академии. В качестве параметров принимает структуру типа `Academy`, содержащую всю необходимую информацию;
- `public void deleteAcademy(Academy ac)` – метод, отвечающий за удаление информации об академии. В качестве параметров принимает структуру типа `Academy`, содержащую всю необходимую информацию;
- `public void addFaculty(Faculty fac)` – метод, отвечающий за добавление нового факультета в организационную структуру университета. В качестве параметров принимает структуру типа `Faculty`, содержащую всю необходимую информацию;
- `public void editFaculty(Faculty fac)` – метод, отвечающий за редактирование информации о факультете. В качестве параметров принимает структуру типа `Faculty`, содержащую всю необходимую информацию;
- `public void deleteFaculty(Faculty fac)` – метод, отвечающий за удаление информации о факультете. В качестве параметров принимает структуру типа `Faculty`, содержащую всю необходимую информацию;
- `public void addChair(Chair ch)` – метод, отвечающий за добавление новой кафедры в организационную структуру университета. В качестве параметров принимает структуру типа `Chair`, содержащую всю необходимую информацию;
- `public void editChair(Chair ch)` – метод, отвечающий за редактирование информации о кафедре. В качестве параметров принимает структуру типа `Chair`, содержащую всю необходимую информацию;
- `public void deleteChair(Chair ch)` – метод, отвечающий за удаление информации о кафедре. В качестве параметров принимает структуру типа `Chair`, содержащую всю необходимую информацию;
- `public void addSpec(Speciality spec)` – метод, отвечающий за добавление новой специальности. В качестве параметров принимает структуру типа `Speciality`, содержащую всю необходимую информацию;
- `public void editSpec(Speciality spec)` – метод, отвечающий за редактирование информации о специальности. В качестве параметров принимает структуру типа `Speciality`, содержащую всю необходимую информацию;
- `public void deleteSpec(Speciality spec)` – метод, отвечающий за удаление информации о специальности. В качестве параметров принимает структуру типа `Speciality`, содержащую всю необходимую информацию;
- `public void addDiscipline(Discipline disc)` – метод, отвечающий за добавление новой дисциплины. В качестве параметров принимает структуру типа `Discipline`, содержащую всю необходимую информацию;
- `public void editDiscipline(Discipline disc)` – метод, отвечающий за редактирование информации о дисциплине. В качестве параметров принимает структуру типа `Discipline`, содержащую всю необходимую информацию;
- `public void deleteDiscipline(Discipline disc)` – метод, отвечающий за удаление информации о дисциплине. В качестве параметров принимает структуру типа `Discipline`, содержащую всю необходимую информацию;

– `public void addTeacher(Author teacher)` – метод, отвечающий за добавление нового преподавателя. В качестве параметров принимает структуру типа `Author`, содержащую всю необходимую информацию;

– `public void editTeacher(Author teacher)` – метод, отвечающий за редактирование информации о преподавателе. В качестве параметров принимает структуру типа `Author`, содержащую всю необходимую информацию;

– `public void deleteTeacher(Author teacher)` – метод, отвечающий за удаление информации о преподавателе. В качестве параметров принимает структуру типа `Author`, содержащую всю необходимую информацию;

– `public void addDocType(string docType)` – метод, отвечающий за добавление нового типа документов. В качестве параметра принимает строковую переменную, содержащую всю необходимую информацию;

– `public void editDocType(string oldType, string newType)` – метод, отвечающий за редактирование информации о типе документов. В качестве параметров принимает две строковые переменные, содержащие всю необходимую информацию;

– `public void deleteDocType(string docType)` – метод, отвечающий за удаление информации о типе документов. В качестве параметра принимает строковую переменную, содержащую всю необходимую информацию;

– `public void addHourType(string hourType)` – метод, отвечающий за добавление нового типа почасовой нагрузки. В качестве параметра принимает строковую переменную, содержащую всю необходимую информацию;

– `public void editHourType(string oldType, string newType)` – метод, отвечающий за редактирование информации о типах почасовой нагрузки. В качестве параметров принимает две строковые переменные, содержащие всю необходимую информацию;

– `public void deleteHourType(string hourType)` – метод, отвечающий за удаление информации о типе почасовой нагрузки. В качестве параметра принимает строковую переменную, содержащую всю необходимую информацию;

Использование объектного подхода при проектировании и создании приложения позволило значительно упростить понимание структуры программы. Применение ООП позволяет строить приложения для работы с базами данных, обладающие необходимым уровнем функциональности для выполнения всех необходимых действий. Использование ООП позволит снизить затраты на дальнейшее развитие приложения и изменения реализации отдельных методов или классов в случае необходимости. [37]

## Заключение

Рассмотрев существующий процесс утверждения плана научно-методических публикаций, мы выделили ключевые проблемы, связанные с его подготовкой и выполнением. Был предложен альтернативный метод контроля над исполнением плана публикаций, состоящий из следующих элементов:

1. Создание информационной системы контроля исполнения плана научно-методических публикаций.
2. Исключение личного присутствия кураторов в ИМО для утверждения публикации.
3. Использование электронной почты в качестве среды для передачи документов на утверждение в ИМО.
4. Ограничение прав кафедр вуза в процессе утверждения авторского состава научно-методических публикаций.

Результатом практической реализации данного метода явилось:

- облегчение работы сотрудников отдела ИМО;
- сокращение количества сопутствующей процессу бумажной документации вследствие использования электронной почты в качестве среды передачи документов;
- снижение влияния случайных факторов на контроль над процессом утверждения публикаций;
- ликвидация разрозненности информации о состоянии документов;
- обеспечение быстрого доступа к необходимой информации;
- обеспечение быстрого выявления и исправления допущенных ошибок, снижение влияния человеческого фактора.

Практическое применение разработанного метода показало сокращение времени:

- на составление плана научно-методических публикаций – до 40%;
- на поиск необходимой информации – до 75%;
- на передачу документов – до 85%;
- на исправление допущенных ошибок – до 30%.

Информационная система, разработанная в рамках предложенного метода, была внедрена в отделе ИМО и прошла апробацию на II Международной летней методической школе по теме «Современные образовательные технологии» (июнь, 2009). Акт о внедрении приведен в приложении В, программа летней школы – в приложении С.

На основании проведенного исследования можно сделать вывод о низкой степени автоматизации рутинных процессов документооборота вуза, касающегося организации учебного процесса. Проблема документооборота является актуальной для любой организации и особенно остро она стоит в организациях системы образования, т.к. их деятельность характеризуется большим количеством сопутствующей документации.

Методы работы с документацией, основанные на применении информационных технологий являются наиболее эффективными в

современных условиях и обеспечивают, при незначительных затратах, резкое повышение эффективности работы подразделений, чья деятельность связана с обработкой документации. Исследования по данной, теме, направленные на выработку практических решений, являются показателем заинтересованности высших учебных заведений Казахстана в решении существующих проблем управления потоками документов и выработке оптимальных стратегий управления образовательным процессом.

Выбор эффективного метода обработки документации является одним из определяющих факторов успешного управления предприятием. Возможность производить оперативный и эффективный обмен информацией между всеми участками производственного и организационного процесса, позволяет сократить время, требуемое на подготовку конкретных задач, исключить возможных появлений ошибок на всех этапах работы с документацией. Применение современных информационных технологий в процессе обработки информации позволяет значительно сократить время, необходимое на ее первичную и последующую обработку. Поскольку актуальность и доступность информации являются ее основополагающими свойствами, обеспечение своевременного доступа к ней имеет определяющее значение при организации процесса ее обработки.

Набор задач, требующих автоматизации, при этом, будет только расширяться. Это делает невозможным эффективное внедрение комплекса приложений автоматизации документооборота при отсутствии их тесной интеграции. Эту задачу призвано обеспечить создание комплексной системы автоматизации документооборота. При подобном подходе задача разбивается на два этапа, а именно – внедрение платформы для автоматизации документооборота и реализация на ее базе комплекса интегрированных приложений. [38] Внедрение базовой платформы автоматизации документооборота позволит:

- удешевить разработку и внедрение конечных приложений;
- обеспечить удобство пользователя и унификацию интерфейса всех приложений;
- сократить стоимость эксплуатации и сопровождения комплекса приложений;
- обеспечить общее информационное пространство, возможности интегрированного поиска и извлечения знаний, накапливаемых в различных приложениях;
- обеспечить унифицированные средства мониторинга процессов и контроля исполнения;
- обеспечить возможности сбора статистической и аналитической информации о скорости и своевременности исполнения этапов управленческих процессов;
- обеспечить возможность постепенного расширения автоматизированных процессов, а также возможностей их модификации по мере изменения процессов.

Предложенный в данной работе метод учета информации о документах и созданная в его рамках ИС являются основой для внедрения базовой системы электронного документооборота. Развитие методов интеллектуального анализа данных на основе применения концепций информационных хранилищ, экспертных систем, систем моделирования бизнес-процессов и систем управления документооборотом, реализованных в контуре общей информационной системы, способствуют усилению обоснованности принимаемых управленческих решений. [39] Таким образом, современные ИС обеспечивают оперативность коммуникации и интеграцию участников управленческих процессов и повышают качество принимаемых решений на всех уровнях управления организацией. Применительно к организациям образования, это приведет к повышению качества подготовки специалистов, т.к. рационально организованный контроль над процессом и результатами учебно-познавательной деятельности [40], а так же полное и своевременное обеспечение необходимой учебно-методической документацией всех участников образовательного процесса являются залогом успеха в сфере образовательной деятельности.

## Список использованных источников

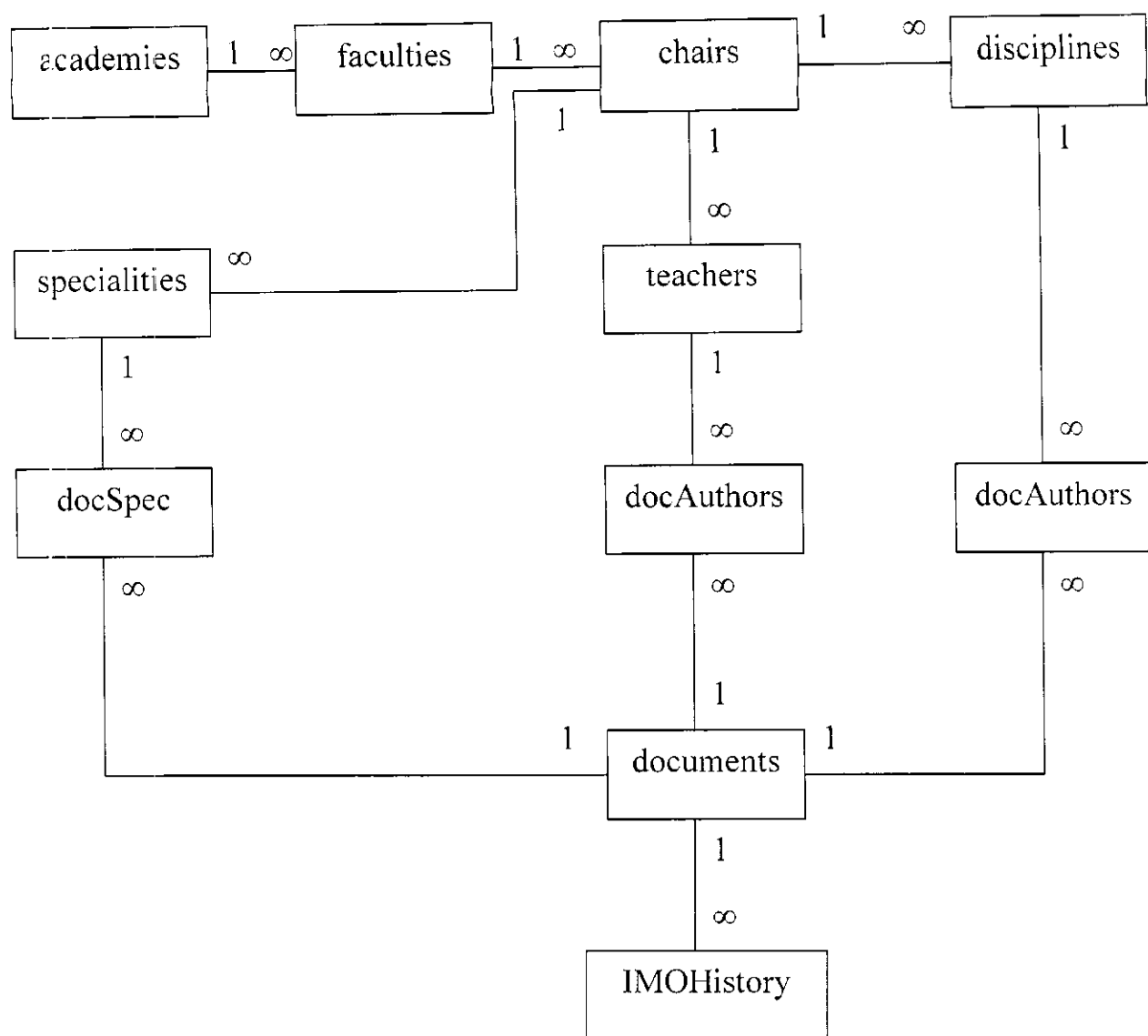
- 1 Бобылева М.П. На пути к информационному менеджменту (вопросы оценки деятельности служб документационного обеспечения управления в условиях автоматизации документооборота) // Делопроизводство. 1999. №1.
- 2 Кузнецов С.В., Шур Е.С. Введение в проблему автоматизации бухгалтерского учета в вузе // Университетское управление. 1998. № 1(4). С. 45-48.
- 3 Системы электронного управления документами: обзор, классификация и оценка возврата от внедрения – <http://www.mdi.ru/library/analit/sysel.html>
- 4 Аляева Ю.П., Малахов А.А., Мелихов В.О., Фан Ч.В. Исследование рациональной структуры организаций информационного профиля – Телематика'2000. Труды VII Всероссийской научно-методической конференции (Санкт-Петербург, 29.05-01.06.2000). – СПб.: СПбГИТМО (ТУ), 2000. – 202 с.
- 5 Карминский А. М., Нестеров П. В. Информатизация бизнеса. – М.: Финансы и статистика, 1997. – 416 с.
- 6 Глинских А. Современные системы электронного документооборота – [http://www.ci.ru/inform09\\_01/p223edoc.htm](http://www.ci.ru/inform09_01/p223edoc.htm)
- 7 Пьянзин К. Универсальные системы управления документами // Lan/Журнал сетевых решений. 1998. №11, том 4
- 8 Короткий С. Концепция построения комплексных информационных систем – <http://www.cfin.ru/vernikov/kias/vest.shtml>
- 9 Мировой рынок систем электронного документооборота – <http://www.naudoc.ru/info/analitics/index.php?article=622>
- 10 Фертман И., Попов К., Тучков А. Аппаратное обеспечение электронного конструкторского документооборота – [http://www.alee-archive.ru/page.jsp?pk=node\\_100100](http://www.alee-archive.ru/page.jsp?pk=node_100100)
- 11 Корпорация Microsoft. Автоматизация бумажного документооборота и делопроизводства – <http://www.microsoft.com/Rus/Government/Newsletters/Docflow/04.mspx>
- 12 Тихонов В.И. Архивное хранение электронных документов: проблемы и решения – <http://www.delo-press.ru/magazines/documents/issue/2006/2/1554/>
- 13 Анализ современного состояния мирового рынка КИС – <http://www.interface.ru/fset.asp?Url=/erp/analiz.htm>
- 14 Идеальное средство организации групповой работы – [http://www.ci.ru/inform10\\_97/pol\\_pro.htm](http://www.ci.ru/inform10_97/pol_pro.htm)
- 15 Марков А. Концепция построения электронного архива – [http://www.alee-archive.ru/page.jsp?pk=node\\_1129899705713](http://www.alee-archive.ru/page.jsp?pk=node_1129899705713)
- 16 Доросинский Л.Г., Кузьмин Д.В. Методика обследования системы электронного документооборота при реализации современного бизнес-процесса – <http://www.jurnal.org/articles/2008/inf43.html>
- 17 Игнатович Р.В., Кондраков Д.Ю. Методологическая сущность проблемы автоматизации управления // Сборник научных трудов СевКавГТУ. Серия "Экономические науки", 2000 г., №1. С. 99-104.



- 18 Барановская Т.П., Лойко В.И., Семенов М.И., Трубилин А.И. Информационные системы и технологии в экономике: Учебник. – 2-е издание, дополненное и переработанное – М.: Финансы и статистика, 2005. – 416 с.
- 19 Спецификация функциональных требований к ИС – <http://www.excode.ru/art6054p1.html>
- 20 Каноническое проектирование ИС – [www.itgrussia.ru/facilities/proektirovanie\\_is/](http://www.itgrussia.ru/facilities/proektirovanie_is/)
- 21 ГОСТ 19.201-78. Техническое задание. Требования к содержанию и оформлению – <http://www.nist.ru/hr/doc/gost/19201-78.htm>
- 22 Власов А.И., Лыткин С.Л., Яковлев В.Л. Краткое практическое руководство разработчика информационных систем на базе СУБД Oracle: Библиотечка журнала "Информационные технологии" – М.: Машиностроение, 2000. – 120 с.
- 23 Братищенко В.В. Проектирование информационных систем: Учебное пособие. – Иркутск: Изд-во БГУЭП, 2004. – 84 с.
- 24 Гореткина Е. «АйТи-Университет»: системный подход к автоматизации вуза // PC Week Review, 2009 г., №3
- 25 Либерти Дж. Создание .NET-приложений. Программирование на C#. 2-е изд. – СПб.: Символ-плюс, 2003 г. – 684 с.
- 26 Просиз Дж. Программирование для Microsoft .Net /Пер. с англ. – М.: Издательско-торговый дом «Русская редакция», 2003. – 704 с.
- 27 Рихтер Дж. CLR via C#. Программирование на платформе Microsoft .Net Framework 2.0 на языке C#. Мастер-класс. / Пер. с англ. – СПб.: Питер, 2007. – 656 с.
- 28 Либерти Дж. Создание .NET-приложений. Программирование на C#. 2-е изд. – СПб.: Символ-плюс, 2003 г. – 684 с.
- 29 .NET Framework – [http://en.wikipedia.org/wiki/.NET\\_Framework](http://en.wikipedia.org/wiki/.NET_Framework)
- 30 Введение в язык C# и .NET Framework – <http://msdn.microsoft.com/ru-ru/library/z1zx9t92.aspx>
- 31 Робинсон С., Корнес О., Глинн Дж., Харвей Б., Макквин К., Моемека Дж., Нагель К., Скиннер М., Ватсон К. C# для профессионалов. Том I. – М.: Лори, 2003. – 478 с.
- 32 Троелсен Э. C# и платформа .NET. Библиотека программиста. – СПб.: Питер, 2004. – 796 с.
- 33 C Sharp – [http://ru.wikipedia.org/wiki/C\\_Sharp](http://ru.wikipedia.org/wiki/C_Sharp)
- 34 Шилдт Г. C#. Учебный курс. – СПб.: Питер, 2003. – 512 с.
- 35 Байдачный С., Маленко Д., Лозинский Ю. - SQL Server 2005. Новые возможности для разработчиков – М.: СОЛОН-Пресс, 2006. – 208 с.
- 36 Петров В.Н. Информационные системы – СПб.: Питер, 2003. – 688 с.
- 37 Шудров В.А. Применение объектного подхода при разработке приложений на языке C# для автоматизации учета документооборота ИМО // Вестник Инновационного Евразийского Университета. 2009. №1.
- 38 Андреев В. Автоматизация документооборота компании – непростой выбор // IT Manager, 2003, №6 (12)

39 Смирнова Г.Н., Сорокин А.А., Тельнов Ю.Ф. Проектирование экономических информационных систем: Учебник – М.: Финансы и статистика, 2003. – 512 с.

40 Шхацва К.Л. Автоматизация управления учебным процессом. Программный комплекс «АСУ-ВУЗ» – <http://www.ito.su/2002/IV/IV-0-562.html>





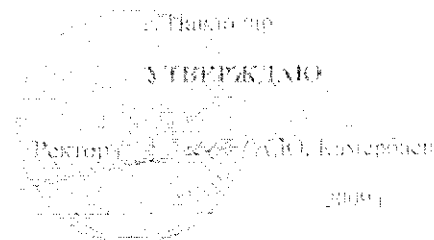
АКТ

2009 г. №

АКТ

Названием:

[Одобрения результатов научно-исследовательской работы и организации учебного процесса]



Мы, нижеподписавшиеся, председатели комиссии – Никитин Е.Б., ректор по учебной работе директор Инженерной академии,

1. Комардина Л.С., директор по учебно-методической работе и качеству образования;
2. Наумов В.В., заведующий кафедрой АСОНБМ;
3. Ушкова Н.М., преподаватель ИМО;
4. Сидий Т.М., преподаватель ИМО;
5. Абраменко А.Н., преподаватель организационно-управленческой работы.

составили настоящий акт о том, что в Инженерном Инновационном университете с 1 июля 2009 г. внедряются в организации учебного процесса результаты, полученные в ходе выполнения научно-исследовательской работы с 1 сентября 2008 г. по 1 июля 2009 г. по теме «Применение авторского метода учета, обработки и хранения электронной документации на основе XML-технологий» (авторские ИИУ) в сочетании с собственной инициативой под научным руководством канд. техн. наук, профессора кафедры АСОНБМ Т.М. Сидий, разработчик – магистрант В.А. Шурков.

Использование результатов ИИУ заключается в следующем:

- создание единой базы данных учета выполнения кафедрами учебно-методических материалов, созданных программой «Автоматизированное рабочее место ИИУ» с использованием разработанной работы с базой данных документов;
- разработан автоматизированной системой создания отчетов, необходимых в работе ИИУ;
- Сотрудники ИМО имеют доступ к разработанной программе;
- Результатом внедрения результатов ИИУ являются:
- сокращение времени на поиск и обработку документации сотрудниками ИМО, без использования разработанной программы;
- повышение качества организации учебного процесса вследствие увеличения контроля исполнения учебно-методических указаний.

Настоящий акт составлен в 5 экз. и передан на хранение: первый экз. – на кафедру, второй экз. – в деканат, третий экз. – в отдел информационно-методического обеспечения, четвертый экз. – в отдел организационно-управленческой работы, пятый экз. – в ИИУ.

Приложение (схема программы) – четвертому экземпляру акта на 7 лист в 1 экз.

Председатель комиссии

Члены комиссии:

Е.Б. Никитин

Л.С. Комардина

В.В. Наумов

Н.М. Ушкова

Т.М. Сидий

А.Н. Абраменко

**Площадка 5**  
**Информационные технологии обучения**  
**ауд. 220 , корпус 1, 14.30-17.00 час.**

**Модератор:** Салий Т.М., к.п.н., профессор кафедры «АСОИиУ»,  
**Секретарь:** Смолкина М.Н., магистрант ИнЕУ

**Участники:** заведующие кафедрами, ответственные за методическую работу кафедр,  
молодые преподаватели, магистранты ИнЕУ

1. Основные направления информатизации в высшей школе .  
Ходарев А... магистрант по специальности  
«Информационные системы»  
научный руководитель: Деревягин С.И., к.т.н.,  
зав.кафедрой ИВТ
2. Мобильная связь, компьютеры и теле-коммуникационные технологии в вузе.  
Бекетов А.О., магистрант по специальности  
«Вычислительная техника и программное обеспечение», научный  
руководитель: Салий Т.М., к.п.н., профессор каф. «АСОИиУ»
3. Информационная модель прогнозирования результатов ПГК на основе текущей  
успеваемости студентов  
Смолкина М.Н., магистрант по специальности  
Информатика ,  
научный руководитель Зайцева Н.М., к.т.н., проф. каф.  
ИВТ
4. Общие принципы организации работы с интерактивной доской.  
Тукманов С.К., магистрант по специальности  
«Вычислительная техника и программное обеспечение» научный  
руководитель: Наумов В.В., к.т.н., зав.каф. «АСОИиУ»
5. Применение альтернативного метода учета, обработки и хранения электронной  
документации на основе NET технологий на примере ИнЕУ  
Шудров В.А., магистрант по специальности  
«Вычислительная техника и программное обеспечение»  
научный руководитель: Салий Т.М., к.п.н., профессор  
каф. «АСОИиУ»
6. Разработка автоматизированной подсистемы управления базами данных Центра  
тестирования ИнЕУ.  
Саринова А., специалист центра тестирования ИнЕУ,  
научный руководитель: Ляшенко И.Н., доцент кафедры  
АСОИиУ
7. Методические рекомендации по составлению контента в «Moodle»: на примере  
дисциплины «Экономическое право»  
Голубева Т.Н., преподаватель кафедры  
«Юридический, исторический и социально-правовых дисциплин»

**Мастер-класс** Салий Т.М. на тему «Система Moodle»

Приложение С – Программа II Международной летней методической школы: