

СОДЕРЖАНИЕ

Введение	6
1. ОБЗОР ПОДСИСТЕМ УЧЕТА НАУЧНО-ИССЛЕДОВАТЕЛЬСКИХ РАБОТ.....	9
1.1. Общее представление об информационных системах	11
1.2 Описание пользователей.....	11
1.3 Требования к разрабатываемой системе.....	12
1.4 Обзор инструментальных средств, поддерживающим процесс создания информационных систем.....	14
2. ПРОЕКТИРОВАНИЕ ПОДСИСТЕМЫ УЧЁТА НАУЧНО ИССЛЕДОВАТЕЛЬСКИХ РАБОТ	19
2.1 Функциональная декомпозиция системы	19
2.2 Проектирование архитектуры программного средства	22
2.3 Проектирование интерфейса программного средства	23
2.4 Проектирование информационного обеспечения программного средства	24
3. РЕАЛИЗАЦИЯ ПОДСИСТЕМЫ УЧЁТА НАУЧНО ИССЛЕДОВАТЕЛЬСКИХ РАБОТ	26
3.1 Выбор средств для реализации подсистемы.....	26
3.2 Организация взаимодействия подсистемы с базой данных.....	37
3.3 Руководство пользователя	39
4. БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ.....	41
4.1. Создание требуемых параметров микроклимата в производственных помещениях	41
4.2. Основные принципы и способы обеспечения безопасности населения в чрезвычайных ситуациях.....	49
Заключение	53
Список литературы	54
Приложение	55

Введение

В целях формирования национальной системы информатизации, массового внедрения и использования во всех сферах экономики и жизни общества современных информационных технологий, средств компьютерной техники и телекоммуникаций, наиболее полного удовлетворения растущих информационных потребностей граждан, создания благоприятных условий для вхождения в мировое информационное сообщество и расширения доступа к мировым информационным ресурсам[1].

В Постановлении Президента Республики Узбекистан "О мерах по дальнейшему внедрению и развитию современных информационно-коммуникационных технологий. (Собрание законодательства Республики Узбекистан, 2012 г., № 13, ст. 139) одной из основных задачи дальнейшего внедрения и развития информационно-коммуникационных технологий, в частности, является программа мер по коренному и качественному улучшению функционирования национальной информационно-поисковой системы, увеличению количества ее пользователей [2]

Работа по учету научно исследовательских работ сопряжена с рядом трудностей. Во-первых, механический способ учета очень трудоемок. По мере увеличения количества работ возникают поиск работ отличающихся каким-либо признаком становится все более сложным, не говоря уже о формировании отчета. Данные каждый раз приходится переписывать вручную, процесс их изменения также очень сложен. Единственный способ упрощения учета научно-исследовательских работ - обращение к средствам информационной технологии.

Система автоматизированного учета научно исследовательской работы студентов имеет основным элементом электронную базу данных.

Электронной базой данных называется совокупность связанных данных, организованных по определенным правилам, предусматривающим

общие принципы описания, хранения и манипулирования, независимая от прикладных программ. База данных является информационной моделью предметной области. Обращение к базам данных осуществляется с помощью системы управления базами данных (СУБД).

Современные базы данных являются реляционными. Именно такая модель баз данных необходима для системы учета научно-исследовательской работы студентов.

Реляционная модель данных - разработанная Э. Коддом в 1970г. логическая модель данных, описывающая:

- структуры данных в виде (изменяющихся во времени) наборов отношений;
- теоретико-множественные операции над данными: объединение, пересечение, разность и декартово произведение;
- специальные реляционные операции: селекция, проекция, соединение и деление; а также специальные правила, обеспечивающие целостность данных.

Реляционная модель позволяет вносить изменения для содержимого отдельной части базы данных, например списка преподавателей, не требуя дополнительного изменения данных в других таблицах, где есть связь с изменяемым элементом. Обновление данных в связанных областях базы данных может производиться автоматически.

Целью написания данной выпускной квалификационной работы является создание подсистемы учета научно-исследовательской работы студентов. Эта система должна обеспечивать возможность удобного добавления научно-исследовательских работ поиска работ по заданному критерию и составление отчета в виде таблиц по результатам поиска, а также просмотр и редактирование таблиц базы данных.

Объем научно-исследовательских работ, число конференций увеличивается постоянно. Поэтому просто необходимо внедрять автоматизированную систему учета научно-исследовательских работ студентов.

Принцип системы автоматизированного учета состоит в следующем. Ответственное за учет научно исследовательских работ лицо вводит в систему новые записи о работах. Таким же образом вводятся данные преподавателей, которым может поручаться руководство работ, студентов которым могут назначаться научно-исследовательские задания, новые группы учащихся, новые предстоящие конференции. По существующим данным при заданном критерии может быть осуществлен поиск, результаты поиска могут быть сохранены в отчет на жесткий диск и впоследствии при необходимости распечатаны.

Для внесения изменений есть интерфейс, который позволяет просматривать таблицы базы данных полностью и редактировать их.

1. ОБЗОР ПОДСИСТЕМ УЧЕТА НАУЧНО-ИССЛЕДОВАТЕЛЬСКИХ РАБОТ.

Анализ предметной области. В последнее время огромное распространение получают электронные системы учета.

Электронная система учета - система где для учета применяются средства вычислительной техники. Для хранения данных используется электронная база данных. Для ввода и редактирования программный интерфейс. Как уже говорилось выше, в реляционной модели данных есть возможность определения одного атрибута или их множества в качестве ключа отношения. Это свойство позволяет формировать запросы к базе данных очень компактно с использованием терминов реляционной алгебры и реляционного счисления, что делает реляционную модель очень простой для разработчика прикладного программного обеспечения. С другой стороны, вся информация, которая будет храниться и использоваться в подсистеме учета научно-исследовательских работ студентов представляется в табличной форме, что является характерной чертой представления информации в реляционных базах данных, а в частности, в их разновидности - табличных базах данных. С учетом вышеизложенного можно сделать вывод, что для разработки подсистемы наиболее подходит СУБД, основанная на использовании реляционной модели данных. Пользователю нужно быстро и удобно вести учет, осуществлять поиск в записях, выводить таблицу о поиске просматривать и редактировать базу данных. Разрабатываемая подсистема должна обеспечить реализацию соответствующих функций.

К преимуществам электронных база данных перед физическими относятся:

малый объем (на компьютере можно хранить десятки и сотни тысяч листов с записями);

возможность быстрого поиска;

возможность быстрого изменения нужной записи;
низкая стоимость (не требуются расходные материалы например бумага чернила и папки для ведения учета).

К недостаткам электронных баз данных относятся:
потенциальная несовместимость с новым аппаратным или программным обеспечением;
экраны некоторых устройств быстро переутомляют глаза;

В рамках формирования электронной подсистемы учета можно выделить содержательные и технологические проблемы. Содержательные проблемы - это проблемы, связанные с определением состава записей системы учета, контингентом пользователей, формулированием целей ее создания электронной библиотеки и ее функциональных возможностей. Технологические - это описание электронных таблиц, их учет, организация структуры хранилища данных.

Общесистемные требования определяют общую структуру системы учета, технологию ее функционирования с учетом поставленных задач и определенной специфики.

Все информационное пространство подсистемы учета, доступное пользователю, должно быть представлено в виде совокупности самостоятельных объектов. Электронные объекты представляют собой текстовые произведения, изображения, фонограммы, базы данных или их фрагменты, отдельные имена и т.д. Организация информационного пространства как совокупности объектов и однозначная идентификация последних необходимы для обеспечения эффективной навигации и выполнения некоторых видов информационных поисков.

1.1. Общее представление об информационных системах

В зависимости от конкретной области применения информационные системы могут очень сильно различаться по своим функциям, архитектуре, реализации. Однако можно выделить, по крайней мере, два свойства, которые являются общими для всех информационных систем. Во-первых, любая информационная система предназначена для сбора, хранения и обработки информации. Поэтому в основе любой информационной системы лежит среда хранения и доступа к данным. Среда должна обеспечивать уровень надежности хранения и эффективность доступа, которые соответствуют области применения информационной системы.

Во-вторых, информационные системы ориентируются на конечного пользователя, например, банковского клерка. Такие пользователи могут быть очень далеки от мира компьютеров. Для них терминал, персональный компьютер или рабочая станция представляют собой всего лишь орудие их собственной профессиональной деятельности. Поэтому информационная система обязана обладать простым, удобным, легко осваиваемым интерфейсом, который должен предоставить конечному пользователю все необходимые для его работы функции, но в то же время не дать ему возможность выполнять какие-либо лишние действия. Иногда этот интерфейс может быть графическим с меню, кнопками, подсказками и т.д. В любом случае наличие развитых интерфейсных средств является обязательным для любой современной информационной системы.

1.2 Описание пользователей

Сведения о пользователях . У системы существует один пользователь: преподаватель или ответственный за научно-исследовательской работы. Преподаватель ведет учет научно исследовательской работы студентов.

Пользовательская среда. Операционная система: Microsoft Windows 2000/ME/XP/Vista.

Профили пользователей. Профили пользователей и их описание приведено в таблице 1.

Таблица 1. Профили пользователей.

Типичный представитель	Преподаватель
Описание	Пользователь системы, наделенный правами на поиск добавление записей и редактирование базы данных
Тип	Пользователь
Ответственности	Ведет учет базы данных
Критерий успеха	Возможность быстро автоматизировано вести учет

Ключевые потребности пользователей. Преподаватель затрачивает большое количество времени на ведение учета научно исследовательской работы студентов Поэтому необходима программа которая могла бы автоматизировать и ускорить этот процесс.

1.3 Требования к разрабатываемой системе

Структурированное описание заказа. Возможность удобного учета научно исследовательской работы студентов.

Расчёт нормативного времени выполнения работ заказа. Возможность для каждой из работ заказа автоматически определить, на основании введенных параметров, время исполнения работы с точностью до секунды. Проверка задания не более 1 секунды.

Контекст использования системы. Система является законченной независимой разработкой.

Сводка возможностей. Требуемые возможности подсистемы приведены в таблице 2.

Таблица 2. Возможности подсистемы.

Выгоды заказчика	Поддерживающие возможности
Снижение трудоемкости преподавателя	Автоматическое внесение записей
Повышение эффективности учета	Возможность поиска по заданному критерию
Облегчение внесения изменений	Базу данных можно редактировать

Предположения и зависимости. Система будет использоваться в ВУЗе.

Применяемые стандарты. Система должна соответствовать всем стандартам интерфейса пользователя Microsoft® Windows® и Microsoft.net Framework.

Системные требования:

128 Mb памяти

10 Mb свободного дискового пространства

процессор с тактовой частотой 1.2 GHz

Операционная система Windows 2000/XP.

Эксплуатационные требования. Система работает в однопользовательском режиме работы.

Требования на входные данные. Выходными данными в разрабатываемой системе являются информация о работе, о группе, о студенте, о преподавателе, о конференции и критерий поиска, для поиска работ удовлетворяющих ему.

Таблица 3. Входные данные.

Источник данных	Параметр	Тип данных	Ограничения
Лицо, ведущее учет (преподаватель)	Критерии поиска	Массив строк (string [])	Все символы кроме ' и "
	Информация о студенте	Массив строк (string [])	
	Информация о работе	Массив строк (string [])	
	Информация о группе	Массив строк (string [])	
	Информация о руководителе	Массив строк (string [])	
	Информация о конференции	Дата (DateTime)	Формат даты дд. мм. Гггг

Требования на выходные данные. Выходными данными в разрабатываемой системе являются список работ, удовлетворяющей критериям поиска, в виде таблицы.

1.4 Обзор инструментальных средств, поддерживающим процесс создания информационных систем

Ни одну область деятельности человека, поддерживаемую информационными технологиями, невозможно представить себе без

использования баз данных, помогающих получить быстрый доступ к информации, увеличивая тем самым продуктивность работы. Клиент-серверные приложения, получившие в последнее время широкое распространение, построены на основе баз данных; приложения Internet и intranet могут получать доступ к базам данных, открывая широкие возможности для публикации информации, необходимой широкому кругу пользователей.

С развитием компьютерных технологий и появлением CASE-моделирования (Computer Aided Software Engineering) возникла потребность в инструментах, которые бы поддерживали стандарты моделирования.

Современный инструмент моделирования баз данных должен удовлетворять ряду требований.

- Позволять разработчику сконцентрироваться на самом моделировании, а не на проблемах с графическим отображением диаграммы. Инструмент должен автоматически размещать сущности на диаграмме, иметь развитые и простые в управлении средства визуализации и создания представлений модели.

- Инструмент должен проверять диаграмму на согласованность, автоматически определяя и разрешая несоответствия. Однако инструмент должен быть настраиваемым и при желании предоставлять разработчику некоторую свободу в действиях и право самому разрешать несоответствия или отступления от методологии.

- Инструмент моделирования должен поддерживать как логическое, так и физическое моделирование.

- Современный инструмент должен автоматически генерировать базу данных на СУБД назначения.

Все современные инструменты моделирования в той или иной степени удовлетворяют перечисленным выше общим требованиям, для решения поставленной задачи в выпускной квалификационной работе используется инструмент моделирования баз данных ERwin версии 3.5, продукте

компании Logic Works. Выбор инструмента не случаен, т.к. на нынешний момент ERwin является наиболее мощным средством для разработки структуры данных, как на логическом, так и на физическом уровне. Следует отметить, что существует несколько модификаций ERwin, каждая из которых, помимо моделирования, предназначена для выполнения специфических целей. Здесь мы рассмотрим ERwin3.5/ERX, который предназначен для работы именно с системами управления базами данных. Остальные члены семейства ERwin предназначены для использования с инструментами разработки клиентской части приложения, такими, как Power Builder, Visual Basic и прочими. Продукт Logic Works ERwin 3.5 был выпущен в феврале этого года и сразу же получил признание широкого круга пользователей за многие усовершенствования по сравнению с предыдущими версиями, которые в него были внесены. Этот инструмент моделирования полностью поддерживает стандарт IDEF1X и является лидером на рынке инструментов разработки баз данных.

Описание BPwin 4.0

BPwin - мощный инструмент моделирования, который используется для анализа, документирования и реорганизации сложных бизнес-процессов. Модель, созданная средствами BPwin, позволяет четко документировать различные аспекты деятельности - действия, которые необходимо предпринять, способы их осуществления, требующиеся для этого ресурсы и др. Таким образом, формируется целостная картина деятельности предприятия - от моделей организации работы в маленьких отделах до сложных иерархических структур. При разработке или закупке программного обеспечения модели бизнес-процессов служат прекрасным средством документирования потребностей, помогая обеспечить высокую эффективность инвестиций в сферу ИТ. В руках же системных аналитиков и разработчиков BPwin - еще и мощное средство моделирования процессов при создании корпоративных информационных систем (КИС).

Power Designer

При современном подходе к разработке информационных систем роль этапа проектирования сложно переоценить. При этом возникает необходимость, как в нотации описания проекта, так и в возможности чтения проектной документации всеми участниками разработки - бизнес-аналитиками, IT-аналитиками и разработчиками.

Таким образом, Power Designer 9.5 обладает следующей функциональностью:

- поддержка моделей и диаграмм;
- поддержка методологий UML, моделирования баз данных (CDM и PDM), моделирования хранилищ данных (data warehouse diagram) и моделирования бизнес-процессов;

- единый репозиторий;
- Поддержка командной разработки.

Power Designer 9.5 предлагает:

- полную поддержку всех диаграмм UML;
- собственную нотацию описания бизнес-процессов, легкую для понимания и использования
- удобное средство проектирования данных и их согласования с объектной моделью приложения
- интегрированный репозиторий для командной разработки.

Power Designer 9.5 рассчитан на использование следующими ключевыми ролями в любом проекте:

- Бизнес-аналитиками
- IT-аналитиками
- Проектировщиками и разработчиками баз данных
- Разработчиками приложений

Power Designer 9.5 делает основной упор на разработке бизнес-процесса как отправной точке отсчета проекта. Средство достаточно легко в использовании и освоении, что позволяет удешевить процесс обучения персонала и внедрения его на предприятии.

Для разработчиков предлагается весь инструментарий языка UML с кодогенерацией в наиболее распространенные средства разработки (Java, C#, VB.NET). При этом Power Designer 9.5 обладает удобным генератором отчетов и проектных документов.

Для проектировщиков и разработчиков БД предлагается привычный подход к проектированию (концептуальная и физическая схема), поддержка более 30 реляционных СУБД, наличие обратного проектирования (reverse engineering) и согласование объектной модели и модели данных.

Целью использования Power Designer 9.5 в современном бизнесе является:

- автоматизация бизнес процессов
- обсуждение требований к новым системам и процессам
- контроль информационной архитектуры бизнеса
- понимание существующих систем для наилучшей интеграции и обслуживания
- Координация действий команды.

Power Designer 9.5 достаточно удобен в работе. На его освоение не нужно много времени и средств на обучение пользователей. Это удешевит проект за счет уменьшения финансирования этих пунктов.

Power Designer 9.5 позволяет согласовывать объектно-ориентированную и концептуальную модели данных, ориентированную на реляционные СУБД. При этом согласование также сделано достаточно легко и удобно.

2. ПРОЕКТИРОВАНИЕ ПОДСИСТЕМЫ УЧЁТА НАУЧНО ИССЛЕДОВАТЕЛЬСКИХ РАБОТ

2.1 Функциональная декомпозиция системы

В настоящее время существует ряд методологий, специально предназначенных для упрощения системного анализа. Одним из инструментов системного анализа является CASE–средство верхнего уровня BPWin. Важнейшим этапом современного процесса разработки сложных систем вообще и программного обеспечения в частности является этап системного анализа и моделирования соответствующей предметной области. Данный этап является пред проектным. Его цель заключается в разработке спецификации проекта (технического задания на разработку проекта). От успеха проведения этого этапа зависит успех проекта в целом.

В настоящее время существует ряд методологий, специально предназначенных для упрощения системного анализа и моделирования предметной области. Данные методологии поддерживаются специальными инструментальными средствами автоматизированного анализа, моделирования и разработки сложных систем, получившими название CASE-средств (Computer-Aided Software/System Engineering – компьютерная поддержка проектирования программного обеспечения/систем).

В семействе CASE-средств инструменты для анализа предметной области составляют небольшую часть. Однако именно изучение и моделирование предметной области является наиболее важным этапом при разработке любого приложения, так как позволяет четко и однозначно определить задачи, которые стоят перед разработчиками. Одним из инструментов системного анализа является CASE-средство верхнего уровня BPwin (Business Process for Windows). Термин Business Process («бизнес-

процесс») близок по смыслу к термину «предметная область» и в настоящее время часто заменяет последний.

Основными функциями BPwin являются рисование диаграмм, представляющих собой средства визуального представления отдельных компонентов моделируемой предметной области различных уровней детализации, проверка целостности и согласованности иерархической модели, построенной из диаграмм различных уровней детализации, генерация различного вида отчетов по построенной модели. К достоинствам BPwin следует отнести обеспечение логической четкости в определении и описании элементов диаграмм, проверку целостности связей между диаграммами, локализацию или коррекцию наиболее часто встречающихся ошибок при моделировании.

Цель построения модели некоторого процесса (предметной области) – специфицирование операций и действий, выполняемых в процессе взаимосвязей между ними.

BPwin поддерживает три методологии структурного анализа и моделирования систем – IDEF0, IDEF3 и DFD. В процессе создания модели бизнес-процесса на любой ветви модели можно переключиться на любую из методологий и создать смешанную модель.

В IDEF0-модели операция представляет собой процесс преобразования входных материалов или информации в некоторый результат на выходе с использованием ресурсов в виде механизма и при выполнении условий, представленных в виде управления.

Методология DFD включает такие понятия, как внешняя ссылка и хранилище данных. Это делает её более удобной по сравнению с IDEF0 для моделирования программного обеспечения и систем документооборота.

Методология IDEF3 включает элемент "перекрёсток", что позволяет описать логику взаимодействия компонентов системы.

На основании требований к подсистеме была разработана диаграмма вариантов использования(диаграмма DFD).

Добавление внешних сущностей и элементов памяти происходит в модели DFD. Диаграмму DFD используют там, где нужно предусмотреть занесение или извлечение информации в базу.

Декомпозиция подсистемы «Учет научно исследовательских работ» показана на рисунке 1

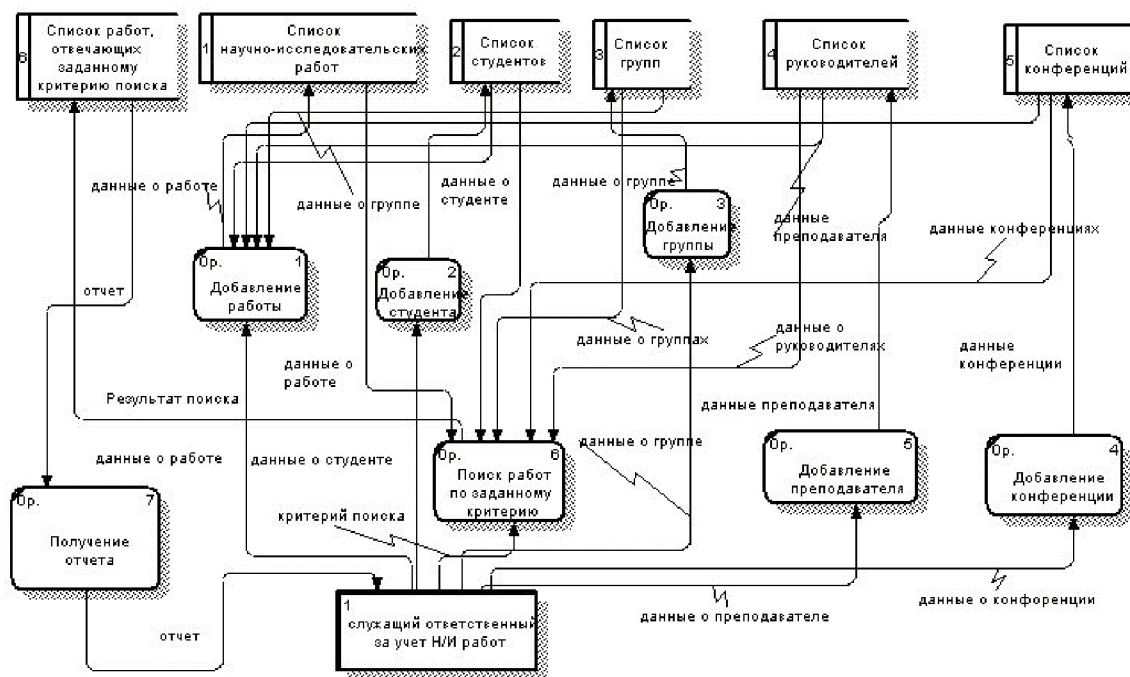


Рис. 1. Диаграмма DFD

Подсистема состоит из следующих таблиц:

- список научно-исследовательских работ;
- список студентов;
- список групп;
- список руководителей;
- список конференций;
- список работ, отвечающих заданному критерию поиска;

Подсистема «Учет научно исследовательских работ» выполняет следующие группы команд:

- добавление работы;
- добавление студента;
- добавление группы;

- добавление конференции;
- добавление преподавателя;
- поиск работ по заданному критерию;

2.2 Проектирование архитектуры программного средства

Проектируемую подсистему учёта научно исследовательских работ студентов предполагается реализовать как Windows-приложение. Оно должно удовлетворять всем требованиям совместимости с версиями Microsoft Windows начиная с версии Microsoft Windows 2000.

Логическая и физическая модели базы данных

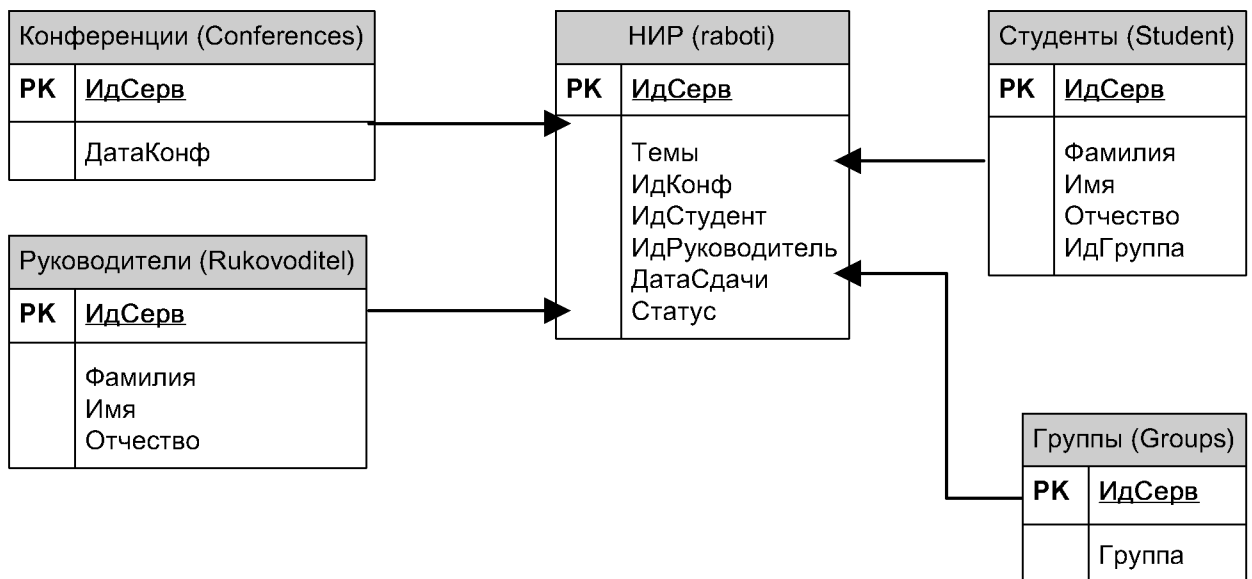


Рис.2. Логическая модель Б.Д.

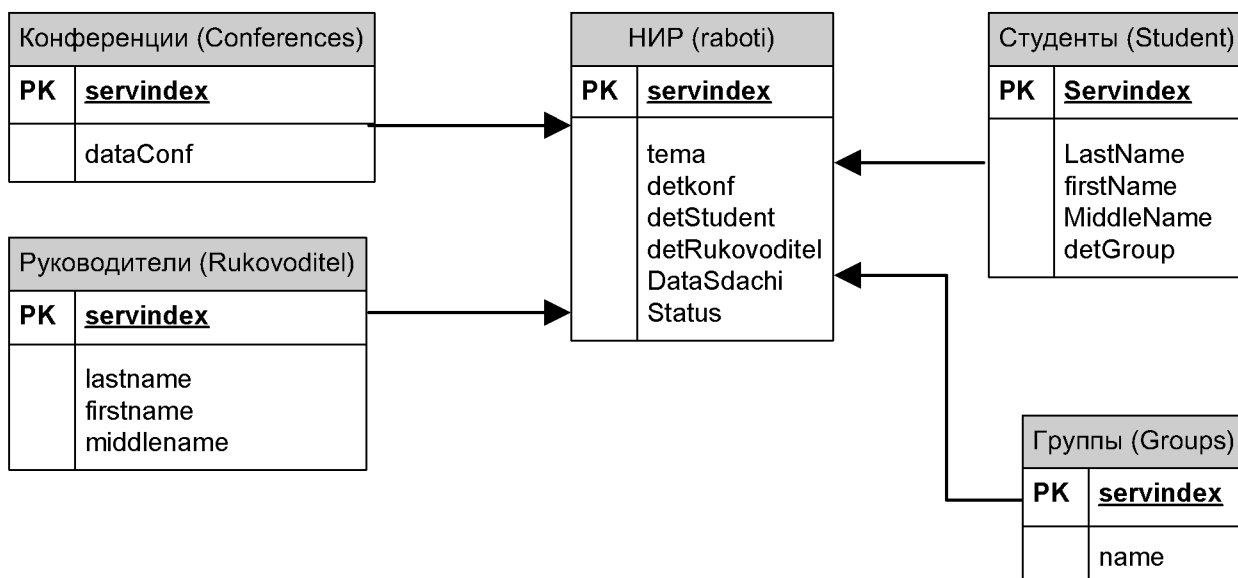


Рис.3. Физическая модель Б.Д.

На основе логической модели выполняется построение физической модели данных(рис. 2). Физическая модель данных содержит полную информацию необходимую для создания всех объектов базы данных информационной системы(рис. 3).

Каждому элементу логической модели соответствует элемент в физической модели. Поэтому каждая сущность становится отношением (таблицей), атрибуты – столбцами, ключи – индексами, логические типы приводятся их к физическим типам данных выбранной целевой Системы управления Базой данных.

2.3 Проектирование интерфейса программного средства

Проектируемая подсистема должна обеспечить возможность удобного внесения данных в базу поиска работ в базе данных, и редактирования таблиц базы данных.

На странице поиска пользователю-студенту необходимо сначала указать критерии поиска, затем после выполнения поиска проанализировать полученный список работ и в случае если результат поиска его удовлетворил

сохранить в виде таблицы на компьютере. Поиск может осуществляться по различным критериям.

На форме редактирования данных в таблицах необходимо при открытии формы отключить возможность редактирования чтобы пользователь случайно не внес неверные изменения в базу данных и не сохранил их. В случае если пользователь захочет перейти в режим редактирования он должен поставить метку в квадратике режим редактирования, тогда он сможет вносить изменения в таблицы. При нажатии на кнопку обновить данные изменения будут сохранены в базе.

На странице добавления записей пользователь должен будет ввести информацию, описывающую добавляемый элемент: работу, студента, преподавателя или конференцию. При этом пользователю должна быть введена вручную только уникальная информация об элементе, а остальная информация может уже храниться в связанных с типом данного элемента таблицах базы и должна быть выбрана из выпадающих списков.

Структурная схема разрабатываемой программы приведена на рис. 4.

2.4 Проектирование информационного обеспечения программного средства

Каждая работа имеет несколько параметров описания. Эти параметры тема, данные руководителя, данные студента, даты завершения и конференции. Таблица Raboty хранит уникальную информацию о работе, а также ссылки на связанные поля из других таблиц. Из других таблиц по связи определяются остальные данные о работе: студент преподаватель жата конференции. Таким образом таблицы students, conferentions и preposts являются вспомогательными для таблицы Raboty и созданы для избегания многократного дублирования в записях одних и тех же данных. Таким же образом таблица groups является вспомогательной для таблицы students. При

поиску данные берутся из всех пяти таблиц, на основании связи записи в основной таблице Raboty с соответствующими данными в других таблицах.

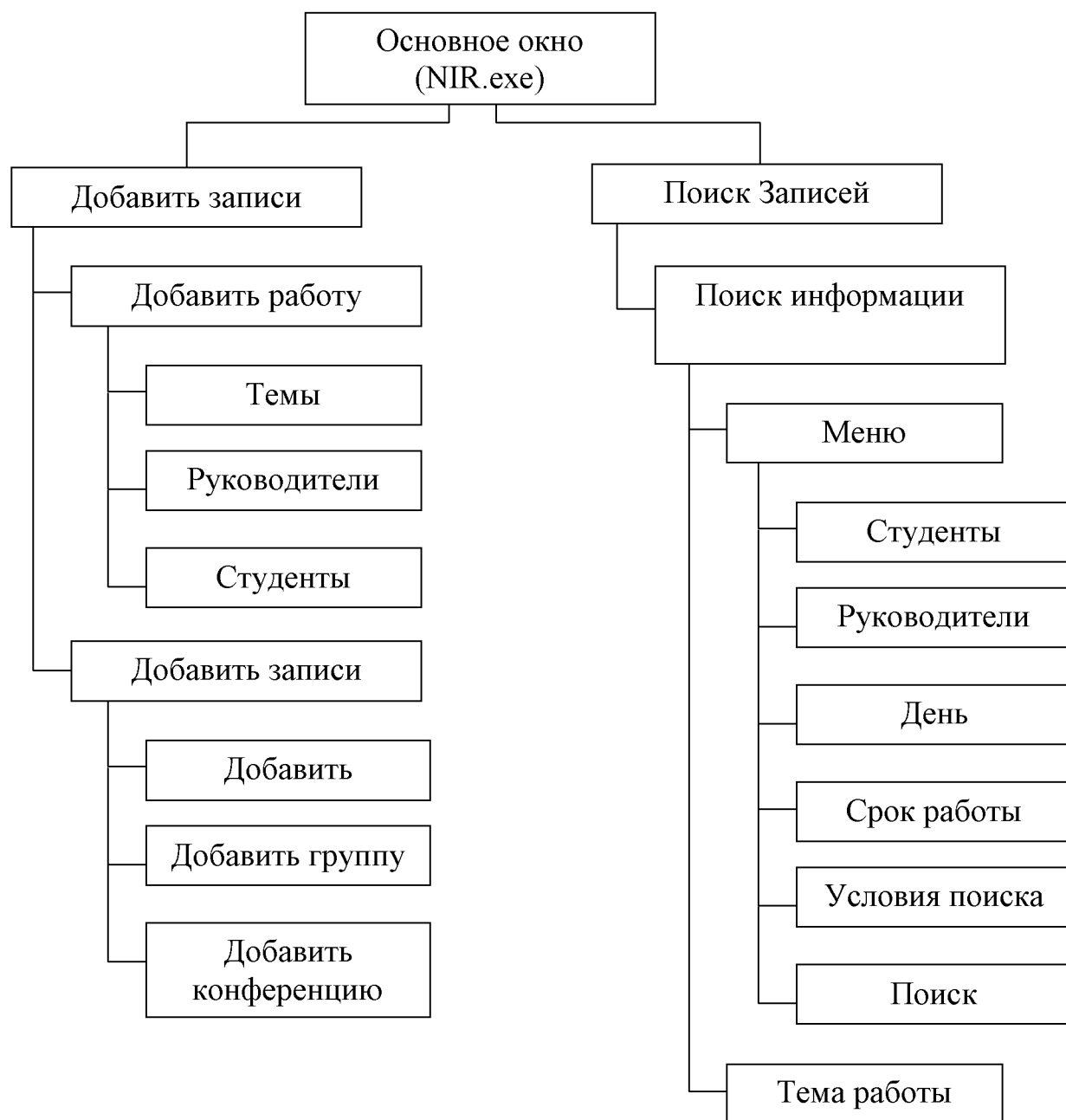


Рис. 4. Структурная схема программы.

3. РЕАЛИЗАЦИЯ ПОДСИСТЕМЫ УЧЁТА НАУЧНО ИССЛЕДОВАТЕЛЬСКИХ РАБОТ

3.1 Выбор средств для реализации подсистемы

Для реализации подсистемы был выбран язык программирования C# и среда программирования Microsoft Visual Studio.net 2003.

C# (Си-шарп) - объектно-ориентированный язык программирования для платформы .net. Разработан в 2000 году Андерсом Хейлсбергом, Скоттом Вилтамутом и Питером Гольде под эгидой Microsoft Research. Основным постулатом C# является высказывание: "всякая сущность есть объект". Язык основан на строгой компонентной архитектуре и реализует передовые механизмы обеспечения безопасности кода.

C# - это полнофункциональный объектно-ориентированный язык, который поддерживает все три "столпа" объектно-ориентированного программирования: инкапсуляцию, наследование и полиморфизм. Он имеет прекрасную поддержку компонентов, надежен и устойчив благодаря использованию "сборки мусора", обработки исключений, безопасности типов.

Язык C# разрабатывался "с нуля" и вобрал в себя много полезных свойств таких языков, как C++, Java, Visual Basic, а также Pascal, Delphy и др. При этом необходимость обратной совместимости с предыдущими версиями отсутствовала, что позволило языку C# избежать многих отрицательных сторон своих предшественников.

Как и Java, C# разрабатывался для Интернет и примерно 75% его синтаксических возможностей аналогичны языку программирования Java, его также называют "очищенной версией Java. 10% подобны языку программирования C++, а 5% - заимствованы из языка программирования Visual Basic. Объем новых концептуальных идей в языке C# около 10%.

Выделение и объединение лучших идей современных языков программирования делает язык C# не просто суммой их достоинств, а языком программирования нового поколения.

Применение C# упрощает разработку, отладку и развертывание Windows-приложений, поскольку все эти задачи удается решать в одной и той же среде разработки - Visual Studio.net.

Основные преимущества, которые дает разработчикам Windows-приложений использование ASP.net:

- компиляция исполняемых частей Windows-приложения, благодаря которой они работают быстрее, чем интерпретируемые сценарии;
- использование нового языка Visual C# - более совершенной версии C, поддерживающей ООП и контроль типов;
- возможность создания новых, нестандартных серверных элементов управления на основе существующих;
- интеграция с ADO.net, позволяющая получать доступ к БД и применять инструменты для конструирования БД из Visual Studio.net;

Кроме того, C# применяет ряд более универсальных компонентов и инструментальных средств из ОС Windows. Хотя они не считаются частью ASP.net, они имеют ключевое значение для программирования с использованием ASP.net:

- .net Framework - полный набор классов, необходимый для Windows-программирования. Помимо классов ASP.net, в него входят классы для решения других задач программирования: файлового доступа, преобразования типов, манипулирования массивами и строками и т. д.;
- инструменты и классы ADO.net для работы с БД, предоставляющие доступ к БД SQL Server и ODBC. Механизмы доступа к данным часто являются ключевыми компонентами Web-приложений.

В качестве СУБД, управляющей базой данных разрабатываемой подсистемы, была выбрана СУБД Microsoft Access, поскольку разрабатываемое программное средство не требует многопользовательского

доступа к базе данных Тесная интеграция с Visual Studio.net предоставляет разработчикам возможность проще создавать безопасные, сильные приложения баз данных при меньших затратах.

Реализация информационного обеспечения. Для создания базы данных воспользуемся программой Microsoft Access.

Создадим новую базу данных и назовем ее Nlraboty. Сохраним ее в паке вместе с файлом Kursovoi NEW. exe

Добавим таблицы в созданную базу данных. Для этого в верхней панели перейдем на закладку "Создание" и нажмем создать таблицу. Создадим пять таблиц с названиями Raboty, prepods, students, groups, conferentions. Перейдя в режим конструктора создадим в каждой таблице соответствующие поля и зададим типы этих полей.

Для таблицы Raboty создадим следующие поля(рис. 5.) и зададим их типы:

	Имя поля	Тип данных
🔑	Код	Числовой
	Тема	Текстовый
	КонфИД	Числовой
	СтудИД	Числовой
	РуководительИД	Числовой
	ДатаСдачи	Дата/время

Рис.5. таблица работы

Для таблицы students создадим следующие поля(рис. 6.) и зададим их типы:

	Имя поля	Тип данных
🔑	Код	Числовой
	Фамилия	Текстовый
	Имя	Текстовый
	Отчество	Текстовый
	ГрупИД	Числовой

Рис. 6. таблица студенты

Для таблицы prepods создадим следующие поля(рис. 7.) и зададим их типы:

	Имя поля	Тип данных
?	Код	Числовой
	Фамилия	Текстовый
	Имя	Текстовый
	Отчество	Текстовый

Рис. 7. таблица преподаватели

Для таблицы groups создадим следующие поля(рис. 8.) и зададим их типы:

	Имя поля	Тип данных
?	КодГруппы	Числовой
	НазваниеГруппы	Текстовый

Рис. 8. таблица группы

Реализация интерфейса пользователя. Для создания системы выбрана среда Microsoft Visual Studio 2003.net. Создадим новый проект C# Windows Application и назовём его NIR New.

Создадим начальную форму. В панели свойств в поле text объекта форма зададим текст для верхней кромки формы "Система учета научно исследовательских работ студентов". Откроем панель элементов и перетащим на форму два элемента типа buton. Пользуясь панелью свойств напишем на них соответствующий текст.

После всех действий форма должна выглядеть так как на рис. 9. Для создания формы поиска из панели элементов переместим на форму элемент tabControl. В панели свойств этого элемента в поле tabPage, создадим 5 страниц tabPage. Свойству text укажем назначим соответствующий текст страницы закладки. Перейдем на элемент tabPage1, предназначенном для ввода информации в базу данных о новой научно-исследовательской работе. На нем расположим поля ввода типа textbox, для значений из связанных таблиц создадим элементы comboBox. В их свойствах dataSource укажем соответствующий источник данных, а в свойствах

displayMember соответствующее поле таблицы. Для ввода даты расположим на форме элемент dateTimePicker.

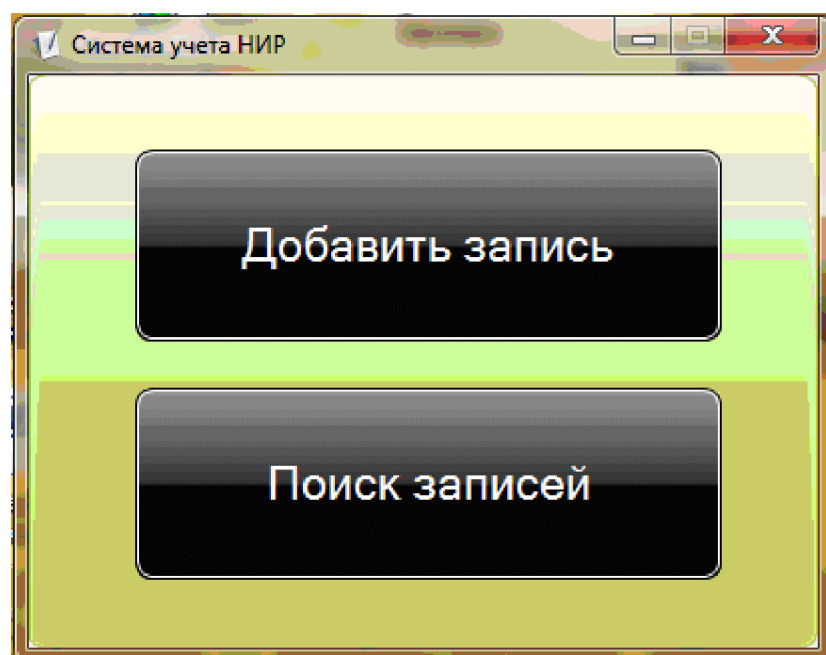


Рис.9. Начальная форма.

Добавим кнопку при нажатии которой осуществляется ввод значений в базу.

Аналогичным образом расположим соответствующие элементы на других закладках tabPage

Форма в итоге будет выглядеть так:

Добавление записей в базу данных

Добавить работу | Добавление записей

Тема
Разработать БД "Издательство".

Руководитель
Иванов Иван Иванович

Конференция
13.05.2013

Дата окончания
29 мая 2013 г.

Студент
Группа: 201-11
Ф.И.О.: Ганижонов Рустам Махмуджон угли

☐ Выполнен

Добавить

Рис.10. Добавление работы

Добавление записей в базу данных

Добавить работу | Добавление записей

Добавить студента

Фамилия:

Отчество:

Имя:

Группа: 201-11

Добавить

Добавить группу

Название группы:

Добавить

Добавить конференцию

Дата конференции: 29 мая 2013 г.

Добавить

Рис.11. Добавление студента, группы и конференции.

На форме для поиска расположим также элемент TabControl, создадим 5 элементов tabPage, соответствующих критериям поиска. Если критерий поиска текст на соответствующей странице создаем элемент textBox, если дата, то dateTimePicker. На странице поиска по информации студента расположим три элемента Edit чтобы включать и отключать критерии поиска данных студента. И кнопки "поиск" на каждом элементе tabPage. На каждом элементе TabControl помещаем элемент dataGrid, в поле свойств dataSource указываем соответствующую таблицу данных, чтобы в нем отображались результаты поиска.

Форма будет выглядеть так:

Поиск данных

Меню

Студент

Фамилия:

Имя:

Отчество:

Руководитель:

Фамилия:

Имя:

Отчество:

Дата конференции:

29 мая 2013 г.

Метод поиска:

☐ Только условные

☒ Показать все

Период работы:

С: 29 мая 2013 г.

По: 29 мая 2013 г.

Поиск

За период работы

Тема НИР:

Тема НИР	Руководитель	Студент	Группа	Дата сдачи	Дата конференции	Статус
Разработать БД "Издательство"	Шакирова Эрика Левченко	Ганижонов Рустам Махмуджон...	201-11	25.06.2013	28.06.2013	☐
-Видишь Суслика? Нет. А он есть!!!	Иванов Иван Иванович	Отажонов Азизбек Муминжон у...	201-11	20.05.2013	23.05.2013	☒
Тема ВКР	Иванов Иван Иванович	Васильев Ильдар Тимурович	210-09	29.05.2013	13.05.2013	☐
Разработка базу данных авиакасы	Кушакова Илона Давтян	Кац Альбина Рахимовна	210-09	20.05.2013	23.05.2013	☐
Разработать ПО, которая следит за ...	Иванов Иван Иванович	Алимов Леонид Якубович	211-09	20.05.2013	23.05.2013	☒
Разработка ПО для фирмы "Record"	Иванов Иван Иванович	Дусмхаммедова Васила Фазы...	211-09	05.06.2013	07.06.2013	☐
hdsjhfdjsfhjdsfh	Иванов Иван Иванович	Жангазова Дилдора Суяркулов...	212-09	15.05.2013	13.05.2013	☒

Записей: 7

Рис.12. Общий поиск, показывает все результаты.

Для просмотра и редактирования таблиц входим в систему нажимая на кнопку «Добавить запись». Выбираем нужные поля и заполняем нужными данными и нажимаем кнопку «Добавить». Если эта новая запись то добавляется в базу, иначе редактируется запись.

Поиск данных

Меню

Студент

Фамилия:

Имя:

Отчество:

Руководитель:

Фамилия:

Имя:

Отчество:

Дата конференции:

мая 2013 г.

Метод поиска:

☒ Только условные

☐ Показать все

Период работы:

С: мая 2013 г.

По: мая 2013 г.

Поиск

За период работы

Тема НИР:

Тема НИР	Руководитель	Студент	Группа	Дата сдачи	Дата конференции	Статус
Видишь Суслика? Нет. А он есть!!!	Иванов Иван Иванович	Отажонов Азизбек Муминжон у...	201-11	20.05.2013	23.05.2013	☒
ghfgh	Иванов Иван Иванович	Васильев Ильдар Тимурович	210-09	10.05.2013	13.05.2013	☒
Разработать ПО, которая следит за оце...	Иванов Иван Иванович	Алимов Леонид Якубович	211-09	20.05.2013	23.05.2013	☒
Разработка ПО для фирмы "Record"	Иванов Иван Иванович	Дусмухаммедова Василя Фазы...	211-09	05.06.2013	07.06.2013	☐
hdsjhfdjsfhjdsfh	Иванов Иван Иванович	Жангазова Дилдора Суяркулов...	212-09	15.05.2013	13.05.2013	☒

Записей: 5

Рис.13. Поиск по руководителю

Поиск данных

Меню

Студент

Фамилия:

Имя:

Отчество:

Руководитель:

Фамилия:

Имя:

Отчество:

Дата конференции:

☒ 23 мая 2013 г.

Метод поиска:

☒ Только условные

☐ Показать все

Период работы:

С: мая 2013 г.

По: мая 2013 г.

Поиск

За период работы

Тема НИР:

Тема НИР	Руководитель	Студент	Группа	Дата сдачи	Дата конференции	Статус
Видишь Суслика? Нет. А он есть!!!	Иванов Иван Иванович	Отажонов Азизбек Муминжон у...	201-11	20.05.2013	23.05.2013	☒
Разработка баз данных авиакасы	Кушакова Илона Давтян	Кац Альбина Рахимовна	210-09	20.05.2013	23.05.2013	☐
Разработать ПО, которая следит за...	Иванов Иван Иванович	Алимов Леонид Якубович	211-09	20.05.2013	23.05.2013	☒

Записей: 3

Рис.14. Поиск по дате конференции

Поиск данных

Меню

Студент

Фамилия:

Имя:

Отчество:

Руководитель:

Фамилия:

Имя:

Отчество:

Дата конференции:

31 мая 2013 г.

Метод поиска:

☒ Только условные

☐ Показать все

Период работы:

С: 1 мая 2013 г.

По: 31 мая 2013 г.

Поиск

☒ За период работы

Поиск

Тема НИР:

Тема НИР	Руководитель	Студент	Группа	Дата сдачи	Дата конференции	Статус
Видишь Суслика? Нет. А он есть!!!	Иванов Иван Иванович	Отажонов Азизбек Муминжон у...	201-11	20.05.2013	23.05.2013	☒
ghfgh	Иванов Иван Иванович	Васильев Ильдар Тимурович	210-09	10.05.2013	13.05.2013	☒
Разработка базу данных авиакасс	Кушакова Илона Давтян	Кац Альбина Рахимовна	210-09	20.05.2013	23.05.2013	☐
Разработать ПО, которая следит з...	Иванов Иван Иванович	Алимов Леонид Якубович	211-09	20.05.2013	23.05.2013	☒
hdsjhfdjsfhjdsfh	Иванов Иван Иванович	Жангазова Дилдора Суяркулов...	212-09	15.05.2013	13.05.2013	☒

Записей: 5

Рис.15. Поиск работ сделанных за период

Реализация функциональности программного средства. Реализация обработки событий windows-формы была проведена с помощью кода на языке C#. Листинг разработанной подсистемы приведён в приложении. Рассмотрим основные глобальные переменные, основные процедуры и функции.

Начальная форма (Form1):

private void button1_Click (object sender, System. EventArgs e) - обработчик события нажатия на элемент button1. При нажатии выполняется открытие формы добавления записей.

private void button2_Click (object sender, System. EventArgs e) - обработчик события нажатия на элемент button2. При нажатии выполняется открытие формы поиска.

private void button3_Click (object sender, System. EventArgs e) - обработчик события нажатия на элемент button3. При нажатии выполняется открытие формы просмотра и редактирования таблиц.

Форма добавления записей (Form2):

private void button4_Click (object sender, System. EventArgs e) - обработчик события нажатия на элемент button4, выполняется добавление данных в таблицу Raboty

private void button3_Click (object sender, System. EventArgs e) - обработчик события нажатия на элемент button3, выполняется добавление данных в таблицу conferentions

private void button5_Click (object sender, System. EventArgs e) - обработчик события нажатия на элемент button5, выполняется добавление данных в таблицу preposds

private void button1_Click (object sender, System. EventArgs e) - обработчик события нажатия на элемент button1, выполняется добавление данных в таблицу students

private void button2_Click (object sender, System. EventArgs e) - обработчик события нажатия на элемент button2, выполняется добавление данных в таблицу groups

private void Form2_Load (object sender, System. EventArgs e) обработчик события загрузки формы. При загрузке формы данные отделяются от базы для возможности действий по дабовлению новых данных.

private void comboBoxGR1_SelectedIndexChanged (object sender, System. EventArgs e) обработчик события изменения выбранного элемента в выпадающем списке для выбора группы. При изменении в выпадающие списки куда посылаются данные студентов загружаются данные конкретно учащихся выбранной группы.

private void comboBoxfam_Enter (object sender, System. EventArgs e) - обработчик события входа в элемент comboBoxfam в котором загружены данные фамилий студентов. При входе в него и в другие выпадающие списки данных студентов загружаются только данные тех студентов которые учатся в группе выбранной в списке comboBoxGR1.

private void comboBoxfam_Leave (object sender, System. EventArgs e) - обработчик события покидания элемента comboBoxfam в котором загружены

данные фамилий студентов. При покидании этого элемента в него и в другие выпадающие списки данных студентов загружаются только данные тех студентов которые учатся в группе выбранной в списке `comboBoxGR1` и с фамилией выбранной в списке `comboBoxfam`.

`private void comboBoxname_Enter (object sender, System. EventArgs e)` - обработчик события входа в элемент `comboBoxname` в котором загружены данные имен студентов. При входе в элемент в него и в другие выпадающие списки данных студентов загружаются только данные тех студентов которые учатся в группе выбранной в списке `comboBoxGR1` и с фамилией выбранной в списке `comboBoxfam`.

`private void comboBoxname_Leave (object sender, System. EventArgs e)` - обработчик события покидания элемента `comboBoxname` в котором загружены данные имен студентов. При входе в элемент в него и в другие выпадающие списки данных студентов загружаются только данные тех студентов которые учатся в группе выбранной в списке `comboBoxGR1` и с фамилией выбранной в списке `comboBoxfam` и именем выбранным в списке `comboBoxname`.

Форма поиска (Form3):

`private void button1_Click (object sender, System. EventArgs e)` - обработчик события нажатия кнопки `button1`. При нажатии кнопки производится поиск работ с заданными данными студента.

`private void button2_Click (object sender, System. EventArgs e)` - обработчик события нажатия кнопки `button2`. При нажатии кнопки производится поиск работ с заданными данными руководителя работы.

`private void button3_Click (object sender, System. EventArgs e)` - обработчик события нажатия кнопки `button3`. При нажатии кнопки производится поиск работ с заданными данными темы работы.

`private void button4_Click (object sender, System. EventArgs e)` - обработчик события нажатия кнопки `button3`. При нажатии кнопки производится поиск работ с заданными данными даты конференции.

private void button5_Click (object sender, System. EventArgs e) - обработчик события нажатия кнопки button3. При нажатии кнопки производится поиск работ с заданными данными фамилии руководителя.

private void button6_Click (object sender, System. EventArgs e) - обработчик события нажатия кнопки button3. При нажатии кнопки производится запись результатов поиска в виде таблицы в файл html.

Форма поиска (Form3):

private void Form4_Load (object sender, System. EventArgs e) - обработчик событий загрузки формы просмотра и редактирования. При загрузке формы данные отсоединяются от базы и заполняются в элементы их представления в виде таблиц интерфейса формы.

private void button3_Click (object sender, System. EventArgs e) - обработчик события нажатия на кнопку button1, который вызывает событие Close закрытия формы.

private void checkBox1_CheckedChanged (object sender, System. EventArgs e) - обработчик события изменения флажка просмотр/редактирование. При отсутствии флажка в элементе checkBox1 редактирование данных не возможно, кнопка для фиксирования изменений не доступна для нажатия, в обратном случае оно возможно и также становится доступной кнопка для фиксирования изменений.

private void button1_Click (object sender, System. EventArgs e) - обработчик события нажатие кнопки button1. При нажатии на кнопку изменения фиксируются в базе данных.

3.2 Организация взаимодействия подсистемы с базой данных

Взаимодействие с базой данных осуществляется посредством следующих классов ADO.net:

System. Data. DataSet - хранит таблицы в оперативной памяти. System. Data. OleDb. OleDbConnection - класс, предоставляющий доступ к основным параметрам подключения.

System. Data. OleDb. OleDbDataAdapter - класс, предоставляющий возможность удобного наполнения таблиц объекта типа DataSet значениями из базы данных.

System. Data. OleDb. OleDbCommand - класс, хранящий строку команды на языке SQL и предоставляющий базовые возможности выполнения этой команды.

В процессе работы сайта при обработке некоторых событий (нажатие кнопок "найти", "добавить документ", "удалить документ") происходит формирование SQL-запроса из некоторых стандартных команд, а также из параметров, введенным пользователем. Например, при нажатии на кнопку "найти" строка SQL-запроса формируется следующим образом. Свойству selectCommand commandText присваивается значение строки первая часть которой всегда выглядит таким образом "SELECT Raboty. Код, Raboty. Тема, prepods. Фамилия AS ФамилияРуководителя, students. Фамилия, students. Имя, students. Отчество, groups. НазваниеГруппы, Raboty. ДатаСдачи, conferentions. Дата AS ДатаКонференции FROM Raboty, students, prepods, groups, conferentions WHERE (Raboty. КонфИД = conferentions. Код) AND (Raboty. СтудИД = students. Код) AND (students. ГрупИД = groups. КодГруппы) AND (Raboty. РуководительИД = prepods. Код)"

Окончание строки зависит от критерия поиска и введенного пользователем параметра который прибавляется к строке, предварительно будучи переведен в строковый тип если изначально к нему не относится.

Для критерия тема работа окончание строки выглядит так: "AND (Raboty. Тема="+st+tema+st+")" tema это строковая переменная значение которой берется из поля ввода куда пользователь водит критерий отбора, в данном случае тексту в элементе формы textBox5. Текст команды передается классу OleDbDataAdapter, который и выполняет SQL-команду.

3.3 Руководство пользователя

Для запуска программы пользователю нужно открыть файл `kursovoi NEW.exe`. Запустится форма, предлагающая выбор одного из трех видов действий: добавление работы данных, поиск работ, и просмотр базы данных. При выборе первого варианта откроется форма для внесения записей в базу данных. Для внесения записей пользователю нужно открыть страницу панели закладок соответствующую данным которые он хочет ввести. Если он хочет добавить студента он должен зайти на страницу "добавить студента", если он хочет добавить новую группу учащихся, ему нужно зайти на страницу "добавить группу", и таким же образом, для добавления руководителя, конференции и работы.

После перехода на соответствующую страницу, пользователь вводит имеющиеся данные, часть данных он выбирает из выпадающего списка. На странице добавление конференции данные дату он выбирает из выпадающего календаря.

После заполнения всех полей, пользователю для записи введенных данных в базу нужно нажать кнопку "добавить", данные будут внесены в базу.

Для поиска работ пользователю нужно выбрать критерий поиска. Доступные критерии: фамилия, имя, отчество студента, фамилия преподавателя, дата конференции, тема работы, и период времени, за который были выполнены работы. Каждому критерию соответствует страница панели закладок. При выборе соответствующей страницы появляется поле ввода критерия отбора. На страницах поиска по дате конференции и поиска за период выпадает календарь, в котором пользователь отмечает дату. При нажатии на кнопку "поиск" система производит поиск и выдает результат в виде таблицы с записями удовлетворяющими критерию. В этой таблице можно сортировать результат

поиска по определенному столбцу нажав на название столбца. После ее нажатия появляется диалог сохранения файла, где пользователь должен указать название файла и его расположение в системе каталогов где он будет сохранен.

Нажатие кнопки "просмотр всех записей" соответствует третьему доступному варианту действий пользователя. Впоследствии открывается форма на которой можно просматривать и редактировать все таблицы с данными. Для просмотра нужной таблицы нужно переместиться на соответствующую страницу панели закладок. Чтобы внести изменения в базу данных, нужно поставить флажок редактировать. Чтобы сохранить изменения пользователь должен нажать на кнопку "обновить данные" данные будут обновлены в базе. Для выхода пользователь может нажать кнопку "завершить просмотр".

4. БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ

4.1. Создание требуемых параметров микроклимата в производственных помещениях.

Для создания требуемых параметров микроклимата в производственном помещении применяют системы вентиляции и кондиционирования воздуха, а также различные отопительные устройства. Вентиляция представляет собой смену воздуха в помещении, предназначенную поддерживать в нем соответствующие метеорологические условия и чистоту воздушной среды.

Вентиляция помещений достигается удалением из них нагретого или загрязненного воздуха и подачей чистого наружного воздуха. Поскольку в данной главе рассматриваем системы вентиляции, предназначенные для обеспечения заданных метеорологических условий, рассмотрим общеобменную вентиляцию, которая осуществляет смену воздуха во всем помещении. Другие типы вентиляции рассмотрены далее.

Общеобменная вентиляция предназначена для поддержания требуемых параметров воздушной среды во всем объеме помещения. Схема такой системы вентиляции представлена на рис. 15.

Для эффективной работы системы общеобменной вентиляции при поддержании требуемых параметров микроклимата количество воздуха, поступающего в помещение ($L_{\text{пр}}$), должно быть практически равно количеству воздуха, удаляемого из него ($L_{\text{выт}}$).

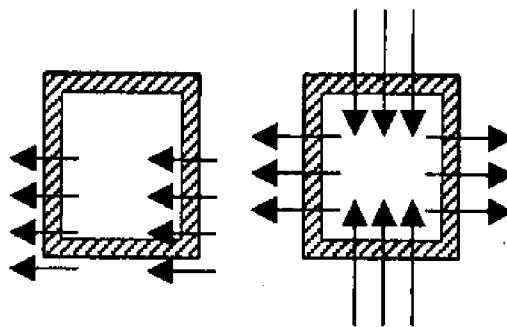


Рис.15. Схема общеобменной вентиляции
(стрелками показано направление движения воздуха)

Количество приточного воздуха, требуемого для удаления избытков явной теплоты из помещения ($Q_{изб} > \text{кДж/ч}$), определяется выражением:

$$L_{np} = \frac{Q_{изб}}{C \rho_{np} (t_{выт} - t_{np})}, \quad (4.1)$$

где: L_{np} – требуемое количество приточного воздуха, $\text{м}^3/\text{ч}$; C – удельная теплоемкость воздуха при постоянном давлении, равная $1 \text{ кДж}/(\text{кг} \cdot \text{град})$; ρ_{np} – плотность приточного воздуха, $\text{кг}/\text{м}^3$; $t_{выт}$ – температура удаляемого воздуха, $^{\circ}\text{C}$; t_{np} – температура приточного воздуха, $^{\circ}\text{C}$.

Для эффективного удаления избытков явной теплоты температура приточного воздуха должна быть на $5\text{--}8^{\circ}\text{C}$ ниже температуры воздуха в рабочей зоне.

Количество приточного воздуха, необходимого для удаления влаги, выделившейся в помещении, рассчитывают по формуле:

$$L_{np} = \frac{G_{ВП}}{\rho_{np} (d_{выт} - d_{прит})}, \quad (4.2)$$

где $G_{ВП}$ – масса водяных паров, выделяющихся в помещении, $\text{г}/\text{ч}$; $d_{выт}$ – содержание влаги в удаляемом из помещения воздухе, $\text{г}/\text{кг}$; $d_{прит}$ – содержание влаги в наружном воздухе, $\text{г}/\text{кг}$; ρ_{np} – плотность приточного воздуха, $\text{кг}/\text{м}^3$.

При одновременном выделении в производственном помещении паров влаги и избыточной теплоты последовательно проводят расчет по формулам

(4.1) и (4.2) и в качестве искомого результата используют большее из полученных значений.

По способу перемещения воздуха вентиляция может быть как естественной, так и с механическим побуждением, возможно также сочетание этих двух способов. При *естественной* вентиляции воздух перемещается за счет разности температур в помещении и наружного воздуха, а также в результате ветрового давления (действия ветра). Способы естественной вентиляции: инфильтрация, проветривание, аэрация, с использованием дефлекторов.

При *механической* вентиляции воздух перемещается с помощью специальных воздуходувных машин-вентиляторов, создающих определенное давление и служащих для перемещения воздуха в вентиляционной сети. Чаще всего на практике используют осевые и радиальные вентиляторы.

По месту действия вентиляция бывает общеобменной и местной. *Общеобменная* вентиляция обеспечивает поддержание требуемых параметров воздушной среды во всем объеме помещения, а *местная* – в определенной его части.

Воздух, всасываемый вентиляторами из атмосферы, после очистки и подогрева поступает в специальные каналы, называемые воздуховодами, и разводится по производственному помещению. Такая вентиляция называется *приточной*. Нагретый воздух из помещения, содержащий водяные пары, отводится из помещения с помощью системы вытяжной вентиляции.

Приточная и вытяжная ветвь вентиляции могут быть объединены, в этом случае система вентиляции называется *приточно-вытяжной*. Большое распространение на практике получила приточно-вытяжная вентиляция с рециркуляцией воздуха. Для нее характерно использование части воздуха, удаляемого из помещения и прошедшего очистку в системе приточной вентиляции. При этом рециркулирующий воздух разбавляется частью свежего воздуха, поступающего из атмосферы. Использование такой системы

вентиляции позволяет снизить расходы на очистку воздуха, поступающего из атмосферы, и на его нагрев в холодное время года.

Как уже сказано выше, для создания требуемых параметров микроклимата на определенном участке производственного помещения служит местная приточная вентиляция. В отличие от общеобменной приточной вентиляции она подает воздух не во все помещения, а лишь в ограниченную часть. Различают следующие устройства местной приточной вентиляции: воздушные души и оазисы, а также воздушно-тепловые завесы.

Воздушные души применяются для защиты работающих от воздействия теплового излучения интенсивностью 350 Вт/м^2 и более. Принцип действия этого устройства основан на обдуве работающего струей увлажненного воздушного потока, скорость которого составляет $1\text{--}3,5 \text{ м/с}$. При этом увеличивается теплоотдача от организма человека в окружающую среду.

В *воздушных оазисах*, представляющих собой часть производственного помещения, ограниченного со всех сторон переносными перегородками, создаются требуемые параметры микроклимата. Указанные источники используются в горячих цехах.

Для защиты людей от переохлаждения в холодное время года в дверных проемах и воротах устраивают *воздушные* и *воздушно-тепловые завесы*. Принцип их работы основан на том, что под углом к холодному воздушному потоку, поступающему в помещение, направлен воздушный поток (комнатной температуры или подогретый), который либо снижает скорость и изменяет направление холодного воздушного потока, уменьшая вероятность возникновения сквозняков в производственном помещении, либо подогревает холодный поток (в случае воздушно-тепловой завесы). Такие воздушно-тепловые завесы установлены на входах на станции метрополитена, а также в дверях крупных магазинов.

В настоящее время для поддержания требуемых параметров микроклимата широко применяются установки для кондиционирования воздуха (кондиционеры). *Кондиционированием воздуха* называется создание

и автоматическое поддержание в производственных или бытовых помещениях независимо от внешних метеорологических условий постоянных или изменяющихся по определенной программе температуры, влажности, чистоты и скорости движения воздуха, сочетание которых создает комфортные условия труда или требуется для нормального протекания технологического процесса. Кондиционер – это автоматизированная вентиляционная установка, которая поддерживает в помещении заданные параметры микроклимата. Эксплуатация установок для кондиционирования воздуха обычно дороже, чем вентиляционных систем.

Для поддержания заданной температуры воздуха в помещениях в холодное время года используют различные системы отопления: водяная, паровая, воздушная и комбинированная.

В системах *водяного отопления* в качестве теплоносителя используется вода, нагретая либо до 100°C либо перегретая выше этой температуры. Эти системы отопления наиболее эффективны в санитарно-гигиеническом отношении.

Системы *парового отопления* используются, как правило, в промышленных помещениях. Теплоносителем в них является водяной пар низкого или высокого давления.

В *воздушных системах* для отопления используется нагретый в специальных установках (калориферах) воздух. *Комбинированные системы* отопления используют в качестве элементов рассмотренные выше системы отопления.

Параметры микроклимата в производственных помещениях контролируются различными контрольно-измерительными приборами. Для измерения температуры воздуха в производственных помещениях применяют ртутные (для измерения температуры выше 0°C) и спиртовые (для измерения температуры ниже 0°C) термометры. Если требуется постоянная регистрация изменения температуры во времени, используют приборы, называемые термографами. Например, отечественный прибор –

термограф типа М-16 – регистрирует изменение температуры за определенный период (сутки или неделю). Существуют и другие устройства для измерения температуры воздуха, например, термопары.

Для измерения относительной влажности воздуха используются приборы, называемые психрометрами и гигрометрами, а для регистрации изменения этого параметра во времени служит гигрограф.

Простейший психрометр – это устройство, состоящее из сухого и влажного термометров. У влажного термометра резервуар обернут гигроскопической тканью, конец которой опущен в стаканчик с дистиллированной водой. Сухой термометр показывает температуру воздуха в производственном помещении, а влажный – более низкую температуру, так как испаряющаяся с поверхности влажной ткани вода отнимает тепло у резервуара термометра. Существуют специальные переводные психрометрические таблицы, позволяющие по температурам сухого и влажного термометров определять относительную влажность воздуха в помещении.

Более сложным по конструкции, но и более точным является так называемый аспирационный психрометр, который также состоит из сухого и влажного термометров, помещенных в металлические трубки и обдуваемых воздухом со скоростью 3–4 м/с, в результате чего повышается стабильность показаний термометров и практически устраняется влияние теплового излучения. Определение относительной влажности осуществляется также с использованием психрометрических таблиц. Аспирационные психрометры, например МВ-4М или М-34, могут быть использованы для одновременного измерения в помещении температуры воздуха и относительной влажности.

Другим устройством для определения относительной влажности служит гигрометр, действие которого основано на свойстве некоторых органических веществ (органических мембран, человеческого волоса) удлиняться во влажном воздухе и укорачиваться в сухом. Измеряя деформацию чувствительного элемента (мембраны или волоса), можно

судить о величине относительной влажности в производственном помещении. Гигрографы записывают изменения величины относительной влажности как функцию времени. Примером такого гигрографа может служить прибор типа М-21, который осуществляет суточную или недельную запись регистрируемого параметра.

Скорость движения воздуха в производственном помещении измеряется приборами – анемометрами.

Работа крыльчатого анемометра основана на изменении скорости вращения специального колеса, оснащенного алюминиевыми крыльями, расположенными под углом 45° к плоскости, перпендикулярной оси вращения колеса. Ось колеса соединена со счетчиком оборотов. При изменении скорости воздушного потока изменяется и скорость вращения колеса, т. е. увеличивается (уменьшается) число оборотов за определенный промежуток времени. По этой информации можно определить скорость воздушного потока.

Крыльчатые анемометры рекомендуется применять для измерения скорости воздушного потока в интервале 0,4–10 м/с, при скоростях 1–35 м/с применяются чашечные анемометры, в которых крылья заменены чашечками. Примером крыльчатого анемометра служит прибор АСО-3 тип Б, чашечного – тип МС-13.

Существуют и другие приборы для измерения скорости движения воздуха: шаровые или цилиндрические кататермометры и термоанемометры.

Интенсивность теплового излучения в отечественной практике измеряют актинометрами, действие которых основано на поглощении теплового излучения и регистрации выделившейся тепловой энергии. Простейший тепловой приемник – термopара. Она представляет собой электрический контур из двух проволок, изготовленных из различных материалов (как металлов, так и полупроводников), например медь–константан, серебро–палладий, серебро–висмут, висмут–сурьма, вольфрам–рений и др. Две проволоки из различных материалов сваривают

или спаивают между собой. Тепловое излучение нагревает один из спаев двух проволок, в то время как другой спай служит для сравнения и поддерживается при постоянной температуре (T_0). Электрическая схема термопары представлена на рис. 16.

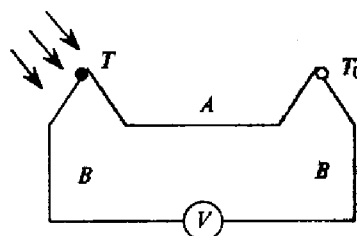


Рис.16. Термопара, состоящая из материала А и В

Две проволоки из материалов А и В составляют электрический контур. При нагреве одного из спаев тепловым излучением до температуры T возникает термоЭДС V_{AB} , величина которой измеряется вольтметром. ТермоЭДС в большом интервале температур прямо пропорциональна разности $T - T_0$ (где, T_0 – температура холодного слоя термопары):

$$V_{AB} = \varepsilon_{AB} (T - T_0).$$

Величина ε_{AB} носит название коэффициента Зеебека для веществ А и В. Этот эффект называют термоэлектрическим или эффектом Зеебека в честь его открывателя (1821 г.). Иногда n термопар соединяют между собой последовательно, получая при этом термоэлектрическую батарею. ТермоЭДС и соответственно чувствительность этого прибора в n раз выше, чем у обычной термопары, что позволяет измерять тепловое излучение малой интенсивности.

В основу промышленных приборов для измерения интенсивности теплового излучения – актинометров – положен принцип термоэлектрической батареи. Чувствительный элемент актинометра состоит из алюминиевой пластинки, на которой в шахматном порядке расположены зачерненные и блестящие секции. Зачерненные полосы интенсивно

поглощают тепловое излучение, а блестящие отражают его, поэтому первые из них нагреваются значительно сильнее, чем вторые. Положительные спаи термопар, соединенные между собой последовательно присоединены к зачерненным полоскам алюминиевой фольги и нагреваются под воздействием теплового излучения значительно сильнее, чем отрицательные спаи, присоединенные к блестящим полоскам. Под воздействием разности температур возникает термоЭДС, которая измеряется чувствительным прибором, отградуированным в единицах тепловой радиации (Вт/м^2).

При отклонении параметров микроклимата от величин, создающих комфортные условия, большое значение имеет правильный выбор спецодежды. При работе в помещениях с пониженной температурой воздуха необходимо использовать утепленную спецодежду. Для персонала, занятого в горячих цехах, используют спецодежду, изготовленную из материалов с низкой теплопроводностью.

4.2. Основные принципы и способы обеспечения безопасности населения в чрезвычайных ситуациях.

К основным мероприятиям по обеспечению безопасности населения в чрезвычайных ситуациях относятся следующие: прогнозирование и оценка возможности последствий чрезвычайных ситуаций; разработка мероприятий, направленных на предотвращение или снижение вероятности возникновения таких ситуаций, а также на уменьшение их последствий. Кроме того, очень важным является обучение населения действиям в чрезвычайных ситуациях и разработка эффективных способов его защиты.

Прогнозирование чрезвычайных ситуаций — это метод ориентировочного выявления и оценки обстановки, складывающейся в результате стихийных бедствий, аварий и катастроф. Различают долгосрочные и краткосрочные прогнозы. Долгосрочные прогнозы

направлены на изучение и определение сейсмических районов, территорий, где возможны селевые потоки или оползни, границ зон вероятного затопления при авариях плотин или природных наводнениях, а также границ очагов поражения при техногенных авариях. Краткосрочные прогнозы используются для ориентировочного определения времени возникновения чрезвычайной ситуации.

Для составления прогнозов используются различные статистические данные, а также сведения о некоторых физических и химических характеристиках окружающих природных сред. Так, для прогнозирования землетрясений в сейсмоопасных районах изучают изменение химического состава природных вод, проводят наблюдение за изменением уровня воды в колодцах, определяют механические и физические (электрические и магнитные) свойства грунта. Значительную информацию для прогноза землетрясений может дать наблюдение за поведением некоторых животных.

Разработаны методы прогнозирования пожаров – лесных, торфяных и др. Для прогнозирования влияния скрытых очагов пожара (подземных или торфяных) на возможность возникновения лесных пожаров используется фотосъемка в инфракрасной части спектра, осуществляемая с самолетов или космических аппаратов.

Для прогнозирования обстановки, возникающей при развитии различных чрезвычайных ситуаций, применяют математические методы (математическое моделирование).

При прогнозировании чрезвычайной ситуации планируют постоянно проводимые, фоновые и защитные мероприятия.

К постоянно проводимым мероприятиям относятся постоянный контроль за качеством строительно-монтажных работ при возведении зданий и сооружений, создание надежной системы оповещения о возникновении чрезвычайной ситуации, строительство защитных укрытий и убежищ, снабжение населения средствами индивидуальной защиты (например, противогазами), обязательное обучение населения правилам поведения в

чрезвычайных ситуациях, разработка планов ликвидации последствий чрезвычайных ситуаций и их финансовое и материальное обеспечение и др.

При предсказании момента чрезвычайной ситуации проверяются и приводятся в готовность система оповещения населения, а также аварийно-спасательные службы, разворачивается система наблюдения и разведки, нейтрализуются особоопасные производства и объекты (химические предприятия, атомные электростанции и др.), проводится частичная эвакуация населения.

Способы защиты населения в чрезвычайных ситуациях следующие: эвакуация, укрытие в защитных сооружениях (убежищах), использование средств индивидуальной защиты. Под э в а к у а ц и е й понимают вывоз населения или его части из очага поражения при чрезвычайной ситуации. Защитные сооружения – это специально разработанные инженерные сооружения, предназначенные для защиты от воздействия различных физических, химических и биологических опасных и вредных факторов, вызванных чрезвычайной ситуацией. Защитные сооружения могут быть использованы для защиты населения как при боевых действиях, так и при техногенных авариях, сопровождающихся выбросами в окружающую среду радиоактивных и токсичных химических веществ, а также бактериологических агентов (вирусов, микроорганизмов и др.).

С р е д с т в а и н д и в и д у а л ь н о й з а щ и т ы населения предназначены для исключения попадания внутрь организма, на кожу и на одежду перечисленных выше веществ, а также бактериологических агентов. Это средства защиты органов дыхания (респираторы, противогазы), специальные защитные одежда и обувь. Медицинские средства индивидуальной защиты предназначены для профилактики и оказания первой помощи населению в чрезвычайных ситуациях. Они включают вещества, ослабляющие или предотвращающие воздействие на организм человека токсичных веществ (антидоты) или ионизирующих излучений (радиопротекторы), противобактериальные средства (антибиотики, вакцины и др.), а также

средства частичной санитарной обработки (индивидуальные перевязочные и противохимические пакеты).

Заключение

При написании данной выпускной квалификационной работы была создана подсистема учёта научно исследовательской работы студентов. Данная система реализует возможность поиска научно исследовательских работ по ряду критериев и получения экземпляра, а также есть возможность изменения, редактирование базы данных и добавления новых задания для студентов.

В рамке выпускной квалификационной работы получены следующие основные результаты:

- проектирована архитектура подсистемы;
- интерфейс подсистемы;
- информационное обеспечение подсистемы;
- разработана диаграмма вариантов использования;
- выбрана средства для реализации подсистемы;
- организована взаимодействия подсистемы с базой данных;
- разработана руководство пользователя;

Подсистема показало, что она успешно реализует поставленные перед ней задачи, то есть сокращает количество времени на то, чтобы получить добавить запись, осуществить поиск, а также экономит время, затрачиваемое на то, чтобы внести изменения в базу данных.

Список литературы

1. И.А.Каримов. Узбекистан на пороге достижения независимости. Государственное издательство, 2011г.
2. Каримов И.А.: Постановление Президента Республики Узбекистан «О мерах по дальнейшему внедрению развитию современных информационно-коммуникационных технологий» / Собрание законодательства Республики Узбекистан, 2012 г., № 13, стр. 139. Ташкент, 2012
3. Каратыгин С.А., Тихонов А.Ф., Тихонова Л.Н., Ильина М.М., Долголаптев В.Г. Электронный офис: В 2-х томах. – М.: БИНОМ, 2007. – 768
3. Дарнелл Рик и др. HTML 4. Энциклопедия пользователя: Пер. с англ. – К.: Издательство «ДиаСофт», 2008. – 688 с.
4. Валда Хиллей. Секреты Windows NT Server 4.0. – К.: Диалектика, 2009. – 528 с.
5. Айден К., Фибельман Х., Крамер М. Аппаратные средства РС.: Перевод с немецкого - СПб BHV Санкт-Петербург, 2009. - 544с.
6. Кириллов В.В., Громов Г.Ю. Структуризированный язык запросов (SQL). Учебное пособие, – С-П.: Санкт-Петербургский технический университет, 2010. – 210 с.
7. Белов С.В. Безопасность жизнедеятельности М.: Высшая школа. 2009.
8. Ёрматов Ғ.Ё., Исамухамедов Ё.У. Меҳнатни муҳофаза қилиш. Дарслик. Ўзбекистан нашриёти. Тошкент 2002.
9. www.ziynet.uz/ru/library/view/alllibs/libid/31813
10. www.ziynet.uz/ru/library/libid/31802
11. <http://msdn.microsoft.com/ru-ru/library/6a71f45d.aspx>

Приложение

Список сокращений

АИС – автоматизированная информационная система
АС – автоматизированная система
ИС – информационная система
ИУС – информационно-управляющая система;
ИАС – информационная автоматизированная система
АСУ – автоматизированная система управления
БД - база данных;
ИО - информационное обеспечение;
ПК – персональный компьютер;
ПО - программное обеспечение;
ПП - программный продукт;
ОС – операционная система
ЭВМ - электронная вычислительная машина;
СУБД - система управления базами данных.
ГОСТ – государственный стандарт
ИВС – информационно-вычислительная сеть
НСД – несанкционированный доступ
РБД – распределенная база данных
РИОС – распределенная информационная обучающая система
СУРБД - система управления распределенными базами данных
ФКЗ – функциональный комплекс задач
ФП – функциональная подсистема
КИС - корпоративных информационных систем

Листинг программы

Листинг Form1.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace NIR
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void vistaButton1_Click(object sender, EventArgs e)
        {
            this.Visible = false;
            AddData ad = new AddData();
            ad.ShowDialog();
            this.Visible = true;
            this.Select();
        }

        private void vistaButton2_Click(object sender, EventArgs e)
        {
```

```

        this.Visible = false;
        searchData sd = new searchData();
        sd.ShowDialog();
        this.Visible = true;
        this.Select();
    }
}

```

Листинг AddData

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Data.OleDb;

namespace NIR
{
    public partial class AddData : Form
    {
        public AddData()
        {
            InitializeComponent();

            OleDbDataAdapter oda;
            OleDbCommand ocom;

```

```

DataTable dt;

void selectData()
{
    try
    {
        string src = "";
        src = "select (lastname+' '+firstname+' '+middlename) as name
from rukovoditel";
        dataAdap(src);
        comboBox_rukov.DataSource = dt;
        comboBox_rukov.DisplayMember = "name";

        src = "select * from conferences order by dateConf";
        dataAdap(src);
        comboBox_conf.DataSource = dt;
        comboBox_conf.DisplayMember = "dateConf";

        src = "select * from groups order by name";
        dataAdap(src);
        comboBox_group.DataSource = dt;
        comboBox_group.DisplayMember = "name";
        dataAdap(src);
        comboBox_addSt.DataSource = dt;
        comboBox_addSt.DisplayMember = "name";

    }
    catch (Exception ex)
    {

```

```

        MessageBox.Show(ex.Message, "Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void AddData_Load(object sender, EventArgs e)
{
    selectData();
}

private void comboBox_group_SelectedIndexChanged(object sender,
EventArgs e)
{
    try
    {
        string src = "select servindex from groups where name = '" +
comboBox_group.Text + "'";
        dataAdap(src);
        comboBox_group.Tag = dt.Rows[0][0];
        selectFIO((int)comboBox_group.Tag);
    }
    catch
    {
    }
}

void selectFIO(int detgroup)
{
    try
    {

```

```

        comboBox_st.DataSource = null;
        comboBox_st.Items.Clear();
        string src = "select (lastname+' '+firstname+' '+middlename) as
name from student where detgroup = " + detgroup + """;
        dataAdap(src);
        comboBox_st.DataSource = dt;
        comboBox_st.DisplayMember = "name";
    }
    catch
    {
    }
}

private void comboBox_st_SelectedIndexChanged(object sender,
EventArgs e)
{
    try
    {
        string[] fio = comboBox_st.Text.Split(' ');
        if (fio.Length == 4)
            fio[2] = fio[2] + " " + fio[3];
        if (fio[0] != "")
        {
            string src = "select s.servindex, t.tema from student s, raboti t "
+
            "where s.lastname = '" + fio[0] + "' and s.firstname = '" +
fio[1] + "' and s.middlename = '" + fio[2] + "' and t.detstudent = s.servindex";
            dataAdap(src);
            comboBox_st.Tag = dt.Rows[0][0];
            richTextBox_tema.Text = dt.Rows[0][1].ToString();

```

```

    }
}
catch
{
}
}

```

```

private void comboBox_conf_SelectedIndexChanged(object sender,
EventArgs e)
{
    try
    {
        string[] w = comboBox_conf.Text.Split('.');
        string wh = "#" + w[1] + "/" + w[0] + "/" + w[2] + "#";
        string src = "select servindex from conferences where dateConf =
" + wh + ";

        dataAdap(src);
        comboBox_conf.Tag = dt.Rows[0][0];
    }
    catch
    {
    }
}

```

```

private void comboBox_rukov_SelectedIndexChanged(object sender,
EventArgs e)
{
    try
    {
        string[] fio = comboBox_rukov.Text.Split(' ');

```

```

        if (fio.Length == 3)
        {
            string src = "select servindex from rukovoditel " +
                "where lastname = '" + fio[0] + "' and firstname = '" + fio[1] +
                "' and middlename = '" + fio[2] + "'";
            dataAdap(src);
            comboBox_rukov.Tag = dt.Rows[0][0];
        }
    }
    catch
    {
    }
}

```

```

private void button_addTheme_Click(object sender, EventArgs e)
{
    try
    {
        string th = richTextBox_tema.Text;
        int detconf = 0, detstud = 0, detruk = 0, sts = 0;
        string dataS = "";
        if (comboBox_conf.Tag != null)
            detconf = (int)comboBox_conf.Tag;
        if (comboBox_st.Tag != null)
            detstud = (int)comboBox_st.Tag;
        if (comboBox_rukov.Tag != null)
            detruk = (int)comboBox_rukov.Tag;
        dataS = dateTimePicker_okonDate.Value.ToShortDateString();
        if (checkBox1.Checked)
            sts = 1;
    }
}

```

```

else
    sts = 0;
if (detconf != 0 && detruk != 0 && detstud != 0 && th != "")
{
    ocom = new OleDbCommand("select servindex from raboti
where detstudent = " + detstud + "", Program.odc);
    ocom.Connection.Open();
    int k = 0;
    if (ocom.ExecuteScalar() == null)
        k = 0;
    else
        k = 1;
    ocom.Connection.Close();
    string str = "";
    if (k == 0)
        str = "insert into raboti select " + th + " as tema, " + detconf
+ " as detkonf, " + detstud + " as detstudent, " + detruk + " as detrukovoditel, " +
dataS + " as DataSdachi, " + sts + " as status;";
    else
        str = "update raboti set tema = " + th + "", detkonf = " +
detconf + ", detrukovoditel = " + detruk + ", DataSdachi = " + dataS + "", Status =
" + sts + " where detstudent = " + detstud + "";
    ocom = new OleDbCommand(str, Program.odc);
    ocom.Connection.Open();
    ocom.ExecuteNonQuery();
    ocom.Connection.Close();
    if (k == 0)
        MessageBox.Show("Новая НИР добавлена!",
"Information", MessageBoxButtons.OK, MessageBoxIcon.Information);
    else

```

```

        MessageBox.Show("Данные изменены!", "Information",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else
        MessageBox.Show("Форма не заполнена!", "Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    catch (Exception ex)
    {
        Program.odc.Close();
        MessageBox.Show(ex.Message, "Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void comboBox_addSt_SelectedIndexChanged(object sender,
EventArgs e)
{
    try
    {
        string s = "select servindex from groups where name = '" +
comboBox_addSt.Text + "'";
        dataAdap(s);
        comboBox_addSt.Tag = dt.Rows[0][0];
    }
    catch
    {
    }
}

```

```

        private void tabControl1_SelectedIndexChanged(object sender,
EventArgs e)
        {
            selectData();
        }

        void dataAdap(string src)
        {
            oda = new OleDbDataAdapter(src, Program.odc);
            dt = new DataTable();
            oda.Fill(dt);
        }

        private void button_addSt_Click(object sender, EventArgs e)
        {
            try
            {
                if (textBox_fam.Text != "" && textBox_ism.Text != "" &&
textBox_otch.Text != "")
                {
                    int detgroup = 0;
                    if (comboBox_addSt.Tag != null)
                        detgroup = (int)comboBox_addSt.Tag;
                    string s = "insert into student select " + textBox_fam.Text + "
as lastname, " + textBox_ism.Text + " as firstname, " +
" " + textBox_otch.Text + " as middlename, " + detgroup + "
as detgroup";

                    ocom = new OleDbCommand(s, Program.odc);
                    ocom.Connection.Open();
                    ocom.ExecuteNonQuery();

```

```

        ocom.Connection.Close();
        textBox_fam.Clear();
        textBox_ism.Clear();
        textBox_otch.Clear();
        MessageBox.Show("Данные студента успешно добавлен!",
"Information", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else
        MessageBox.Show("Форма не заполнена!", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    catch (Exception ex)
    {
        Program.odc.Close();
        MessageBox.Show(ex.Message, "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

```

private void button_addGr_Click(object sender, EventArgs e)
{
    try
    {
        if (textBox_gr.Text != "")
        {
            string s = "insert into groups select '" + textBox_gr.Text + "' as
name";

            ocom = new OleDbCommand(s, Program.odc);
            ocom.Connection.Open();
            ocom.ExecuteNonQuery();

```

```

        ocom.Connection.Close();
        textBox_fam.Clear();
        textBox_ism.Clear();
        textBox_otch.Clear();
        selectData();
        MessageBox.Show("Группа успешно добавлена!",
"Information", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else
        MessageBox.Show("Форма не заполнена!", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    catch (Exception ex)
    {
        Program.odc.Close();
        MessageBox.Show(ex.Message, "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

```

private void button_konf_Click(object sender, EventArgs e)
{
    try
    {
        string d = dateTimePicker_conf.Value.ToShortDateString();
        if (d != "")
        {
            string s = "insert into conferences select '" + d + "' as dateConf";
            ocom = new OleDbCommand(s, Program.odc);
            ocom.Connection.Open();

```

```

        ocom.ExecuteNonQuery();
        ocom.Connection.Close();
        textBox_fam.Clear();
        textBox_ism.Clear();
        textBox_otch.Clear();
        MessageBox.Show("Дата конференции успешно
добавлена!", "Information", MessageBoxButtons.OK,
MessageBoxIcon.Information);
    }
    else
        MessageBox.Show("Форма не заполнена!", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    catch (Exception ex)
    {
        Program.odc.Close();
        MessageBox.Show(ex.Message, "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
}

```

Листинг Program.

```

using System;
using System.Collections.Generic;
using System.Windows.Forms;
using System.Data.OleDb;

```

```

namespace NIR
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            string path = Application.StartupPath;
            odc = new
OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" +
path + @"\Data\NIRaboti.accdb");
            Application.Run(new Form1());
        }

        public static OleDbConnection odc;
    }
}

```

Листинг SearchData.

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

```

```
using System.Data.OleDb;
```

```
using System.IO;
```

```
namespace NIR
```

```
{
```

```
    public partial class SearchData : Form
```

```
    {
```

```
        public SearchData()
```

```
        {
```

```
            InitializeComponent();
```

```
        }
```

```
        OleDbCommand ocom;
```

```
        OleDbDataAdapter oda;
```

```
        DataTable dt;
```

```
        private void checkBox2_CheckedChanged(object sender, EventArgs e)
```

```
        {
```

```
            dateTimePicker_s.Enabled = checkBox_per.Checked;
```

```
            dateTimePicker_po.Enabled = checkBox_per.Checked;
```

```
        }
```

```
        private void выходToolStripMenuItem_Click(object sender, EventArgs
```

e)

```
        {
```

```
            this.Close();
```

```
        }
```

```
        private void button_search_Click(object sender, EventArgs e)
```

```
        {
```

```

        string bdate = "", edate = "", bdConf = "", edConf = "", filter = "",
theme = "";
        string famSt = "", ismSt = "", otchSt = "", famRuk = "", ismRuk = "",
otchRuk = "";
        famSt = textBox_famSt.Text.Trim();
        famRuk = textBox_famRuk.Text.Trim();
        ismRuk = textBox_ismRuk.Text.Trim();
        ismSt = textBox_ismSt.Text.Trim();
        otchRuk = textBox_otchRuk.Text.Trim();
        otchSt = textBox_otchSt.Text.Trim();
        theme = richTextBox_theme.Text.Trim();
        if (checkBox_per.Checked)
        {
            bdate = dateTimePicker_s.Value.ToShortDateString().Replace('.', '-');
            edate = dateTimePicker_po.Value.ToShortDateString().Replace('.', '-');
        }
        else
        {
            bdate = "01-01-1900";
            edate = "01-01-5000";
        }

        if (dateTimePicker_konf.Checked && radioButton_usl.Checked)
        {
            bdConf =
dateTimePicker_konf.Value.ToShortDateString().Replace('.', '-');
            edConf = bdConf;
        }

```

```

else
{
    bdConf = "01-01-1900";
    edConf = "01-01-5000";
}

if (radioButton_usl.Checked)
    filter = "tema like '%" + theme + "%' and rl like '%" + famRuk +
"%'" + rf like '%" + ismRuk + "%' and " +
        "ro like '%" + otchRuk + "%' and sl like '%" + famSt + "%'
and sf like '%" + ismSt + "%' and " +
        "so like '%" + otchSt + "%'";
else
    filter = "";

string src = "select r.tema, (ru.lastname + ' ' + ru.firstname + ' ' + ru.middlename) as rukov," +
        " (s.lastname + ' ' + s.firstname + ' ' + s.middlename) as student,
g.name as groups," +
        " r.DataSdachi, c.dateConf," +
        " ru.lastname as rl, ru.firstname as rf, ru.middlename as ro," +
        " s.lastname as sl, s.firstname as sf, s.middlename as so,
r.status" +
        " from student s, groups g, raboti r, rukovoditel ru,
conferences c" +
        " where r.detkonf = c.servindex" +
        " and r.detstudent = s.servindex" +
        " and r.detrukovoditel = ru.servindex" +
        " and s.detgroup = g.servindex" +

```

```

" and r.DataSdachi between #" + bdate + "# and #" + edate +
"#" +
" and c.dateConf between #" + bdConf + "# and #" + edConf
+ " #" +

```

```

" order by g.name, s.lastname, r.tema";
try
{
    dataGrid View1.Columns.Clear();
    oda = new OleDbDataAdapter(src, Program.odc);
    dt = new DataTable();
    oda.Fill(dt);

    DataView dv = new DataView(dt);
    dv.RowFilter = filter;
    dataGrid View1.DataSource = dv;

    dataGrid View1.Columns[0].HeaderText = "Тема НИР";
    dataGrid View1.Columns[1].HeaderText = "Руководитель";
    dataGrid View1.Columns[2].HeaderText = "Студент";
    dataGrid View1.Columns[3].HeaderText = "Группа";
    dataGrid View1.Columns[4].HeaderText = "Дата сдачи";
    dataGrid View1.Columns[5].HeaderText = "Дата конференции";
    dataGrid View1.Columns[12].HeaderText = "Статус";

    dataGrid View1.Columns[6].Visible = false;
    dataGrid View1.Columns[7].Visible = false;
    dataGrid View1.Columns[8].Visible = false;
    dataGrid View1.Columns[9].Visible = false;
    dataGrid View1.Columns[10].Visible = false;
    dataGrid View1.Columns[11].Visible = false;

```

```

        dataGridView1.Columns[0].Width = 350;
        dataGridView1.Columns[1].Width = 200;
        dataGridView1.Columns[2].Width = 200;
        dataGridView1.Columns[3].Width = 70;
        dataGridView1.Columns[4].Width = 100;
        dataGridView1.Columns[5].Width = 100;
        label_itemCount.Text = dataGridView1.RowCount.ToString();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void toolStripMenuItem1_Click(object sender, EventArgs e)
{
    saveFileDialog1.FileName = "Отчет по НИР";
    saveFileDialog1.Filter = "Microsoft Office Excel (*.csv)|*.csv";
    saveFileDialog1.Title = "Сохранить отчет";
    if (dataGridView1.RowCount != 0)
    {
        int cols;
        if (saveFileDialog1.ShowDialog() == DialogResult.OK)
        {
            //open file
            StreamWriter wr = new
StreamWriter(saveFileDialog1.FileName, false, Encoding.Unicode);

            //determine the number of columns and write columns to file
            cols = dataGridView1.Columns.Count;

```

```

        for (int i = 0; i < cols - 1; i++)
        {
            wr.Write(dataGridView1.Columns[i].Name.ToString().ToUpper() + ",");
        }
        wr.WriteLine();
        //write rows to excel file
        for (int i = 0; i < (dataGridView1.Rows.Count - 1); i++)
        {
            for (int j = 0; j < cols; j++)
            {
                if (dataGridView1.Rows[i].Cells[j].Value != null)
                {
                    wr.Write(dataGridView1.Rows[i].Cells[j].Value + ",");
                }
                else
                {
                    wr.Write(",");
                }
            }
            wr.WriteLine();
        }
        //close file
        wr.Close();
        MessageBox.Show("Отчет сохранен!", "Information",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}
}
}
}

```