

Министерство образования и науки Российской Федерации
Филиал Федерального государственного автономного образовательного учреждения
высшего образования
«Южно-Уральский государственный университет»
(национальный исследовательский университет)
в г. Нижневартовске

Кафедра «Информатика»

РАБОТА ПРОВЕРЕНА

РЕПЕНЗЕНТ
Инженер МБУ «ЦБС»

/ А.В. Булыгин

« ____ » _____ 2018 г.

ДОПУСТИТЬ К ЗАЩИТЕ

И.о. зав.кафедрой «Информатика»
к.ф-м.н, доцент

/ А.В. Ялаев

« ____ » _____ 2018 г.

**Разработка программного комплекса
по учету компьютерного оборудования**

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУрГУ–09.03.01. 2018.356.ПЗ ВКР

Консультанты
Экономическая часть

к.э.н., доцент

/А.В.Прокопьев/

« ____ » _____ 2018г.

Безопасность жизнедеятельности
к.ф-м.н., доцент

/ А.В. Ялаев

« ____ » _____ 2018 г.

Руководитель работы

к.п.н., доцент

/Е.З. Никонова/

« ____ » _____ 2018 г.

Автор работы
обучающийся группы НвФл-526

/ Г.И. Романов/

« ____ » _____ 2018г.

Нормоконтролер

старший преподаватель

/Л.Н.Буйлушкина/

« ____ » _____ 2018г.

Нижневартовск 2018

АННОТАЦИЯ

Романов Г.И. Разработка программного комплекса по учету компьютерного оборудования – Нижневартовск: филиал ЮУрГУ, Информатика: 2018, 110 с., 13 ил., 17 табл., библиогр. список – 20 наим., 3 прил.

Данная выпускная квалификационная работа представляет собой описание разработки программного комплекса по учету компьютерного оборудования.

Практическая значимость выпускной квалификационной работы состоит в том, что использование средств автоматизации сокращает трудозатраты, требуемые для учета техники и оргтехники, за счет упрощения операции пополнения и представления информации по ремонтам, добавлениям, использованию и отказам, постоянного контроля за корректностью вводимой информации.

Данный программный комплекс рассматривается как один из этапов разработки автоматизированной системы «Учет компьютерной техники и оргтехники», поэтому он решает следующие задачи:

1. ведение справочников техники, используемой на предприятии;
2. учет по местам хранения и установки;
3. получение сводных данных о комплектующих, используемых в компьютерном оборудовании;
4. подготовка данных для проведения инвентаризаций.

					<i>09.03.01.2018.356.ПЗ</i>			
			<i>№до</i>	<i>Подпис</i>	<i>Дата</i>			
<i>Разраб.</i>	<i>Романов Г.И.</i>				<i>Разработка программного комплекса по учету</i>	<i>Лит</i>	<i>Лист</i>	<i>Листов</i>
<i>Проверил</i>	<i>Никонова Е.З.</i>					<i>ЭП</i>	<i>5</i>	<i>110</i>
<i>Рецензент</i>	<i>Булыгин А.В.</i>					Филиал ФГАОУ ВО «ЮУрГУ (НИУ)» в г. Нижневартовске		
<i>Н.контр.</i>	<i>Буйлушкина</i>							
<i>Утверди</i>	<i>Ялаев А.В.</i>							

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	8
1 АНАЛИТИЧЕСКИЙ РАЗДЕЛ.....	10
1.1 Анализ предметной области.....	10
1.2 Назначение программного комплекса.....	12
1.3 Формальная постановка задачи.....	13
1.4 Структура решения задачи, декомпозиция задачи на подзадачи...	14
1.5 Обзор существующих программных продуктов.....	16
2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО КОМПЛЕКСА.....	23
2.1 Проектирование базы данных.....	23
2.2 Инфологическая модель базы данных.....	24
2.3 Структура базы данных.....	26
2.4 Проектирование форм.....	28
2.5 Структура программного комплекса.....	29
2.6 Описание программного интерфейса.....	30
2.7 Сообщения оператору.....	33
3 ОРГАНИЗАЦИОННО-ЭКОНОМИЧЕСКИЙ РАЗДЕЛ.....	34
3.1 Постановка задачи.....	34
3.2 Расчет затрат на создание графического приложения.....	34
3.3 Определение себестоимости приложения.....	35
3.4 Экономическая эффективность разработки.....	37
4 БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ.....	39
4.1 Организация рабочего места программиста.....	39
4.2 Микроклимат рабочего места.....	43
4.3 Освещенность рабочего места.....	44
4.4 Нормирование шума.....	45
4.5 Электробезопасность.....	46
4.6 Излучения ПЭВМ.....	47
4.7 Анализ опасных и вредных производственных факторов.....	48

4.8 Требования к пожарной безопасности.....	49
4.9 Рекомендации по организации режима труда и отдыха.....	50
ЗАКЛЮЧЕНИЕ.....	52
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	54
ПРИЛОЖЕНИЯ	
ПРИЛОЖЕНИЕ А. ТЕХНИЧЕСКОЕ ЗАДАНИЕ.....	57
ПРИЛОЖЕНИЕ Б. КОМПАКТ ДИСК.....	61
ПРИЛОЖЕНИЕ В. ИСХОДНЫЙ КОД ПРОГРАММНОГО КОМПЛЕКСА ПО УЧЕТУ КОМПЬЮТЕРНОГО ОБОРУДОВАНИЯ.....	62

ВВЕДЕНИЕ

Автоматизация различных сфер деятельности предприятия, будь то сбор данных или управление технологическими процессами, позволяет ускорить работу, сделать ее более точной и эффективной, избежать потерь нужной информации, ошибок персонала, дублирования документов, запутанного порядка их прохождения. Для эффективности работы по учету техники необходимо проводить мероприятия по обновлению материально-технической и информационной базы, внедрению современных методов управления с использованием автоматизированных подсистем и автоматизированных рабочих мест работников, созданию банка данных о технике предприятия, его своевременному пополнению, оперативному представлению необходимой информации пользователям. Именно эти требования и определили актуальность данной разработки.

Целью выпускной квалификационной работы является разработка программного комплекса по учету компьютерного оборудования для МБУ «Централизованной библиотечной системы» города Мегион. (далее МБУ «ЦБС»).

Задачи выпускной квалификационной работы:

- проанализировать деятельность предприятия МБУ «ЦБС» города Мегион и выявить объект автоматизации;
- сравнить отечественные и зарубежные готовые коммерческие программные продукты по учету компьютерной техники, а также средства разработки автоматизированных систем;
- построить информационную модель предметной области;
- разработать физическую модель базы данных и пользовательское приложение доступа к данным;
- провести расчет организационно-экономических показателей для разработанного программного комплекса;

– изложить требования к организации рабочего места программиста.

Объектом выпускной квалификационной работы является процесс учета материальных активов в деятельности МБУ «ЦБС» города Мегион.

Предметом выпускной квалификационной работы является автоматизация учета компьютерной техники и оргтехники указанного предприятия.

Практическая значимость выпускной квалификационной работы состоит в том, что использование средств автоматизации сокращает трудозатраты, требуемые для учета техники и оргтехники, за счет упрощения операции пополнения и представления информации по ремонтам, добавлениям, использованию и отказам, постоянного контроля за корректностью вводимой информации. Это значительно упрощает работу сотрудников и исключает ошибки, часто встречающиеся при обычной организации работы.

Структура выпускной квалификационной работы состоит из введения, трех разделов, заключения и библиографического списка.

1 АНАЛИТИЧЕСКИЙ РАЗДЕЛ

1.1 Анализ предметной области

Городские библиотеки на безвозмездной основе оказывают основному контингенту основные виды библиотечных и справочно-библиографических и информационных услуг.

Центральная городская библиотека является методическим, справочно-информационным центром на территории города Мегиона, выполняет функции книжной палаты муниципального уровня по работе с местными краеведческими изданиями, организует взаимное использование библиотечных ресурсов, осуществляет административные и другие функции.

Предметом деятельности МБУ «ЦБС» города Мегиона являются:

- реализация прав граждан на свободный доступ к информационным ресурсам, документному фонду и сведениям о его составе;
- сохранение, накопление и распространение знаний, памяти человечества в виде документов, как на бумажных, так и на аудиовизуальных и электронных носителях информации;
- сохранение и развитие культурного и духовного потенциала населения городского округа город Мегион.

Цели деятельности бюджетного учреждения:

1. Удовлетворение универсальных информационных потребностей общества, организация библиотечной, библиографической и научно-информационной деятельности в интересах пользователей, развития культуры, науки, образования.

2. Собираение, сохранение и предоставление в пользование обществу универсального фонда материальных объектов с зафиксированной на них информацией в виде текста, звукозаписи или изображения, предназначенной для передачи во времени и пространстве в целях хранения и общественного пользования. Максимально полное формирование фонда местных документов.

3. Участие в проведении региональной библиотечной политики, разработке и реализации библиотечных, информационных и культурных программ как на территории городского округа город Мегион, так и на территории Ханты-Мансийского автономного округа – Югры.

4. Для достижения целей Бюджетное учреждение осуществляет следующие основные виды деятельности.

5. Формирование в соответствии с профилем комплектования библиотеки максимально полного фонда документов на основе обязательного экземпляра документов, а также путем покупки документов и иных поступлений в соответствии с законодательством Российской Федерации, обеспечение его постоянного хранения.

6. Обработка, учет и раскрытие фондов библиотеки с помощью системы каталогов на различных носителях информации, информационных баз и банков данных, организация доступа к ним.

7. Осуществление деятельности по библиотековедению, библиографоведению, а также деятельности в качестве методического, координационного и культурного центра муниципального значения в этих и других смежных областях.

8. Библиотечное и информационное, в том числе справочно-библиографическое, обслуживание пользователей по следующим направлениям: бесплатное предоставление полной информации о составе библиотечных фондов через систему каталогов и другие формы библиотечного информирования.

9. Проведение культурно-просветительских и образовательных мероприятий в установленной сфере деятельности.

10. Осуществление в установленном порядке издательской деятельности в области библиотечного дела и библиографии.

11. Проведение исследовательской, в том числе социологической, работы по направлениям деятельности учреждения. Мониторинг потребностей пользователей.

12. Удовлетворение культурных, информационных, образовательных потребностей пользователей, приобщение населения, взрослых и детей к культурным ценностям общества.

13. Осуществление книгообмена как внутри библиотечной системы, так и посредством межбиблиотечного абонементов.

14. Внедрение новых технологий, автоматизация систем поиска и предоставления информации, предоставление пользователям доступа в корпоративные и глобальные информационные сети, обслуживание пользователей в режиме локального и удаленного доступа.

1.2 Назначение программного комплекса

Данный программный комплекс рассматривается как один из этапов разработки автоматизированной системы «Учет компьютерной техники и оргтехники», поэтому он будет решать следующие задачи:

1. ведение справочников техники, используемой на предприятии;
2. учет по местам хранения и установки;
3. получение сводных данных о комплектующих, используемых в компьютерном оборудовании;
4. подготовка данных для проведения инвентаризаций.

Создание этого продукта преследует следующие цели:

1. автоматизация и совершенствование учета и управления;
2. снижение трудоемкости и упорядочение деятельности по управлению.

При этом необходимо учитывать следующие требования:

1. база данных должна быть способна хранить информацию о более 100 учетных единицах техники;
2. стиль интерфейса пользователя должен включать диалоговые окна в виде графических форм изменяемого размера с удобной навигацией по ней с помощью мыши;

3. простота в обращении;
4. невысокие программно-аппаратные требования.

1.3 Формальная постановка задачи

Целью данной работы является разработка программного комплекса для повышения эффективности учета компьютерного оборудования учреждения.

Формальной постановке задачи соответствует контекстная диаграмма методологии IDEF0, представленная на рисунке 1.1.

Входные данные:

- Информация о компьютерах, находящихся на предприятии.
- Информация о комплектующих (видеокарты, процессоры, корпуса и т.д., необходимые для модернизации компьютеров).

Выходные потоки:

- Отчет (готовый отчет по учету компьютерной техники предприятия).

- База данных.

Входные управления:

- Правила (условия, которые необходимо соблюсти, что бы достичь поставленной цели).

- Приказы (поставленная задача предприятию, т.е. провести учет компьютерной техники на предприятии с помощью тех или иных информационных систем).

- Руководители (директора и главные управляющие предприятием).

Входные ресурсы:

- ПК (компьютеры, с помощью которых ведется учёт).
- Сотрудники (специалисты, выполняющие назначенные руководством указания).

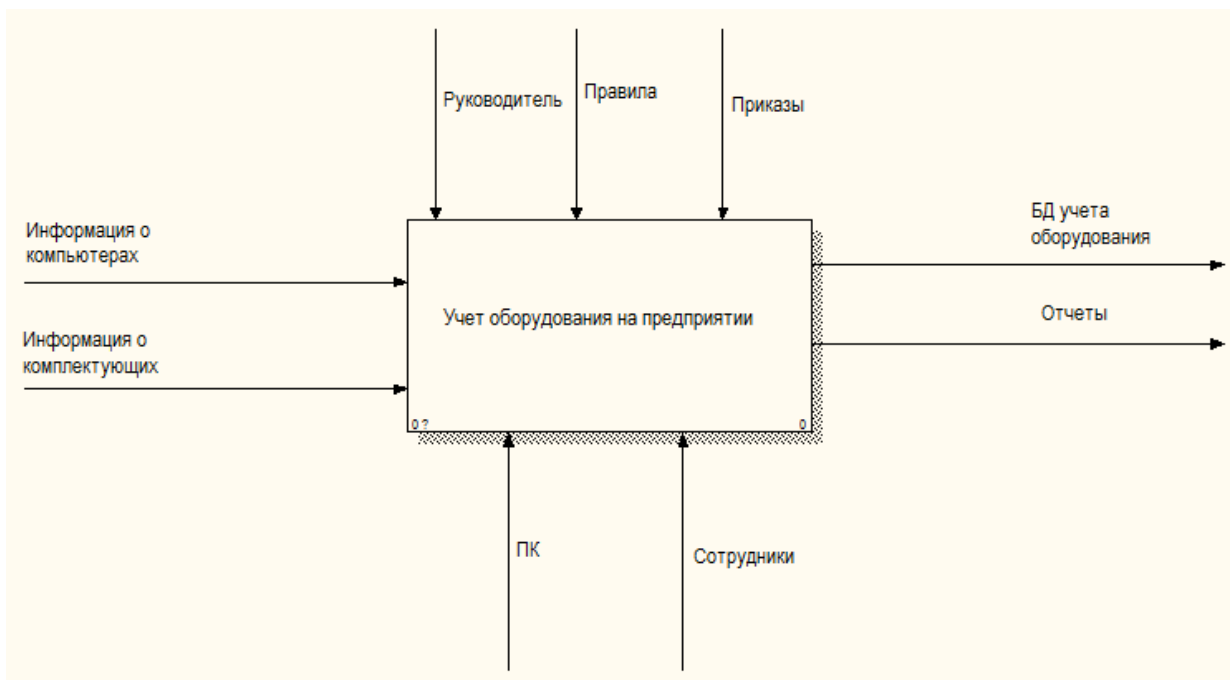


Рисунок 1.1 – Диаграмма IDEF0

1.4 Структура решения задачи, декомпозиция задачи на подзадачи

Для решения поставленной задачи необходимо провести следующие операции:

- ввод информации в базу данных;
- редактирование данных в БД;
- ввод информации о размещении оборудования;
- составление отчета.

Декомпозиция процесса учета оборудования представлена на рисунке 1.2.

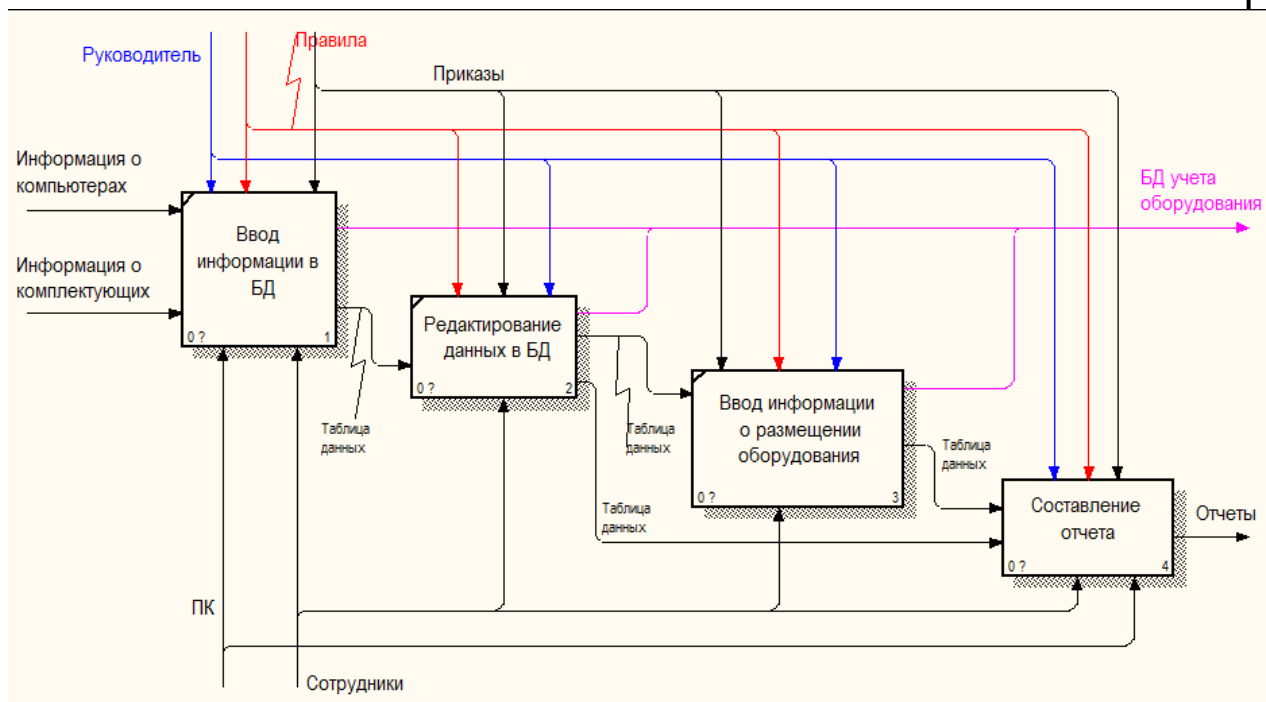


Рисунок 1.2 – Декомпозиция процесса учета оборудования

Также была сформирована информационная модель с использованием диаграмм потоков данных (DFD) декомпозиции работы «Обслуживание компьютера».

В декомпозиции работы «Составление отчета» (рисунок 1.3) были определены:

Внутренние работы – это:

- Сбор данных (собрание сведений о компьютерах и комплектующих).
- Проверка (данные проверяются на точность).
- Отчет (написание отчета о проделанной работе).

Внешние сущности – это комплектующие, компьютеры, руководитель.

Хранилище данных – это данные о компьютерах и комплектующих, данные отчета.

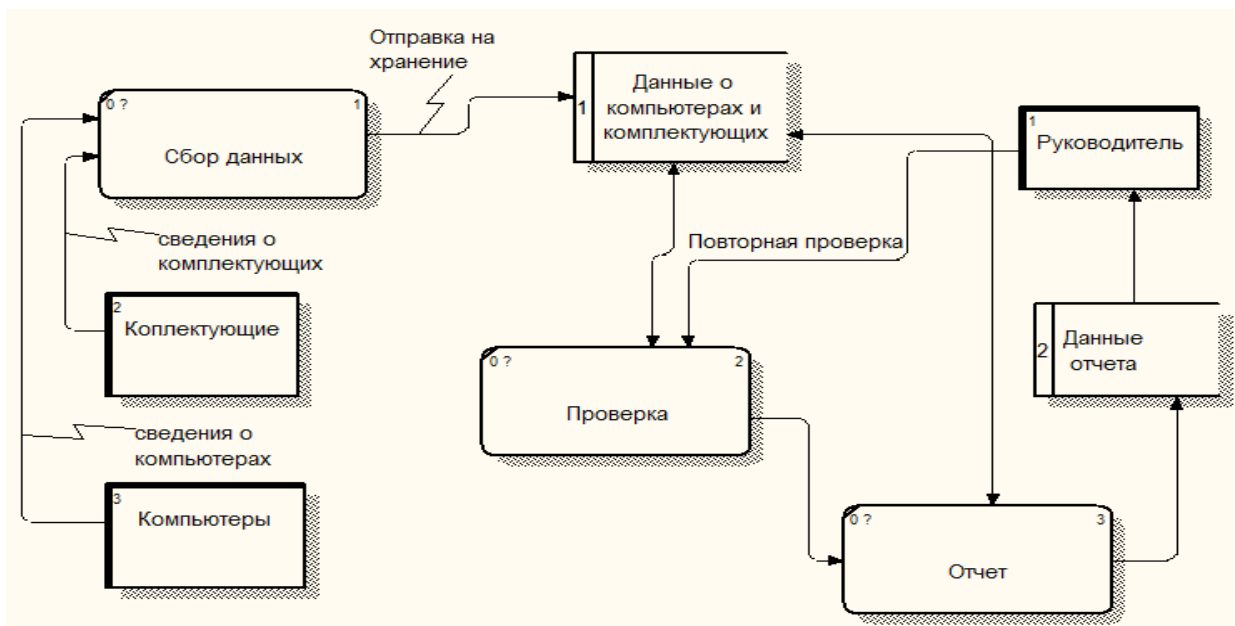


Рисунок 1.3 – Декомпозиция работы «Составление отчета»

При выполнении задачи «Составление отчета» производится сбор сведений о компьютерах и комплектующих, затем происходит отправка их на хранение. После чего данные проверяются на точность, составляется отчет и данные снова отправляются на хранение в первое хранилище данных (см. рисунок 1.3), либо данные отчета отправляются во второе хранилище данных (см. рисунок 1.3), после чего отчет предоставляется на проверку руководителю.

Руководитель проверяет, делает пометки, исправления и отправляет на повторную проверку. После чего отчет отправляется на хранение до повторной проверки руководителя.

1.5 Обзор существующих программных продуктов

Анализ подобных информационных систем проводится для выявления у систем достоинств и недостатков, так же для сравнения функционала, интерфейса, дизайна и удобства её использования. На рынке программных продуктов были выявлены следующие информационные системы:

1. Программный комплекс ITInvent (it-invent.ru).
2. Программный комплекс Hardware Inspector (hwinspector.com).
3. Конфигурация 1С:Учёт компьютеров и оборудования 8.1.

IT Invent – позволяет осуществлять не только учет компьютеров, принтеров, программ и комплектующих, но и учет ремонтов и обслуживаний, работ по поддержке техники, заказов поставщикам, поступлений и перемещений оборудования, учет контрагентов, сотрудников и многое другое. Основная форма программы IT Invent показана на рисунке 1.4.

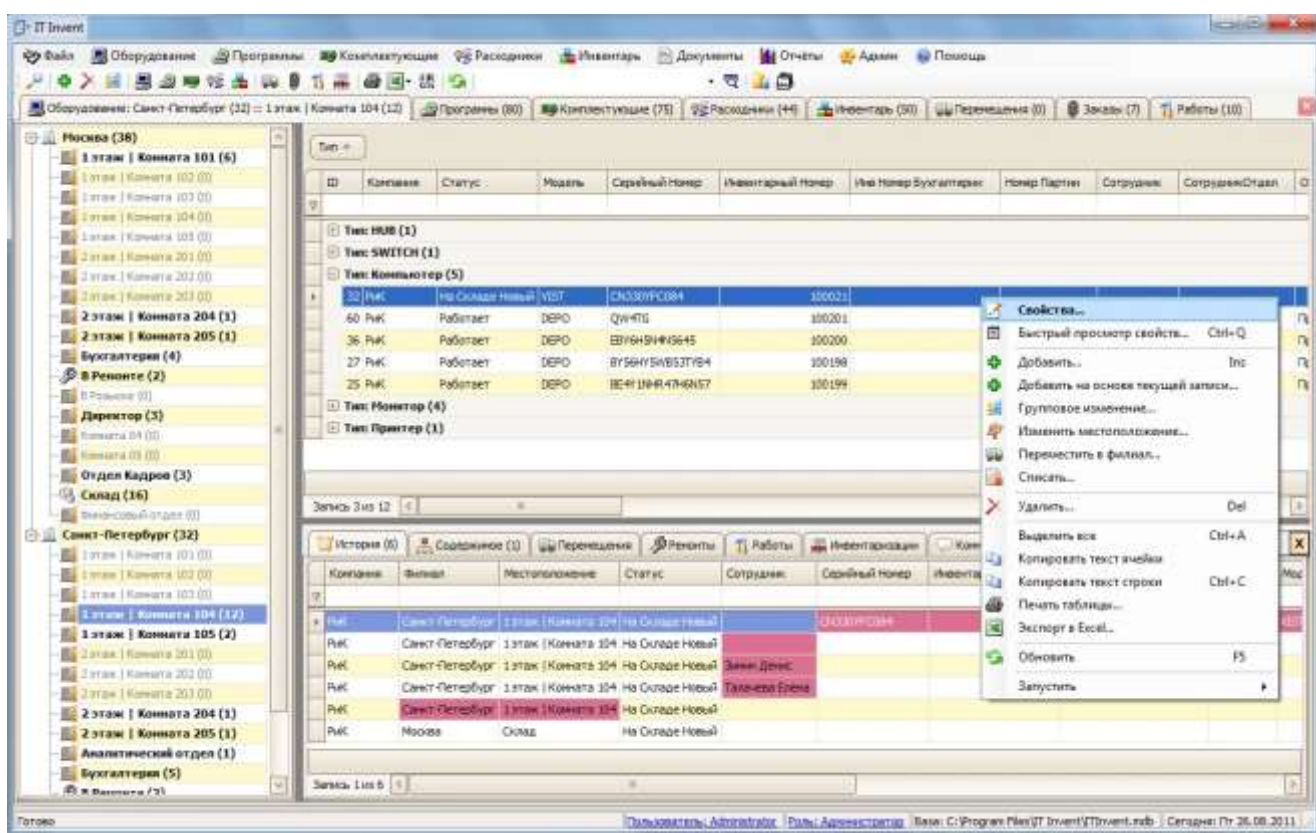


Рисунок 1.4 – Основная форма программы IT Invent

IT Invent – это гибкая и настраиваемая система, которая обладает интуитивно понятным интерфейсом, за счет чего хорошо воспринимается пользователем в плане дизайна. Программа довольно многофункциональна. Хотелось бы отметить следующие ключевые особенности программы:

1. Поддержка базы данных MS Access и MS SQL Server.

2. Многопользовательский режим работы – все филиалы работают с единой базой.

3. Возможность создания и настройки собственных дополнительных свойств различных типов.

4. Учет выполнения работ любых видов внутри организации.

5. Уникальная система создания и печати инвентарных этикеток.

Поддержка принтеров штрих-кодов.

6. Поддержка работы со сканером штрих-кодов. Поиск записей в базе по штрих-коду.

7. Ведение истории изменений по оборудованию.

8. Учет ремонтов и профилактических обслуживаний оборудования и компьютеров.

9. Логическое связывание программ и комплектующих с оборудованием.

10. Учет расходных материалов, комплектующих запчастей и канцелярии.

11. Закрепление учетных единиц за сотрудниками организации. Акты приёма-передачи.

12. Ведение базы поставщиков, сервисных организаций и прочих контрагентов.

13. Гибкое разграничение прав доступа для пользователей системы.

14. Настройка E-Mail оповещений по событиям в программе.

15. Большое количество встроенных печатных форм и отчетов с возможностью их редактирования.

16. Импорт и просмотр данных напрямую из ActiveDirectory.

Программа IT Invent является сетевой. Для работы по сети с единой базой данных, необходимо у каждого пользователя программы в файле 'DBPath.ini' прописать путь для подключения к файлу базы данных или указать этот путь, выбрав пункт меню 'Файл' -> 'Выбор базы данных'. При этом нужно не забыть

выставить каталогу с базой данных права на чтение и запись для всех пользователей программы.

HardwareInspector предназначена для автоматизированного учета и инвентаризации компьютерной техники и иного оборудования в организациях. Уникальность программы HardwareInspector заключается в возможности вести учет не просто текущего состояния параметров компьютера, а всей истории жизни отдельных комплектующих. На рисунке 1.5, показано наглядное представление устройств в дереве рабочих мест.

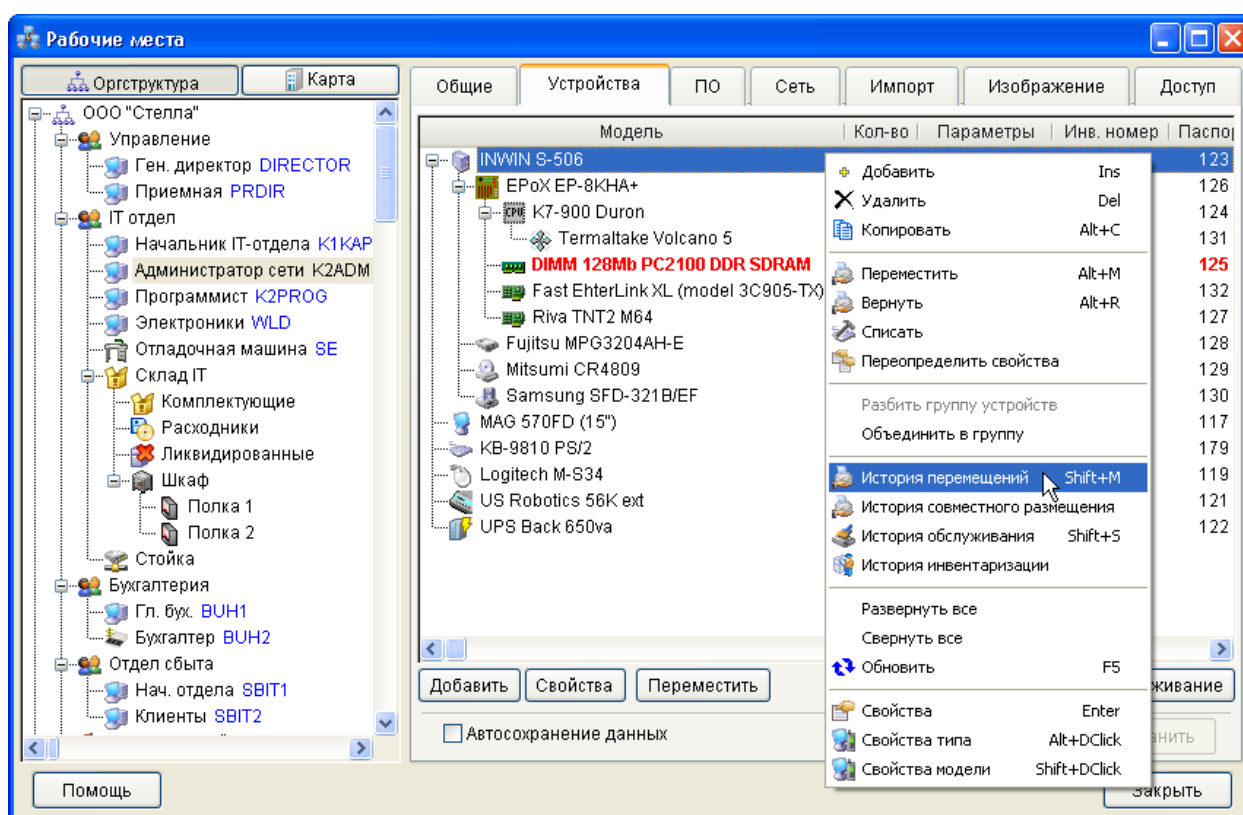


Рисунок 1.5 – Представление устройств в дереве рабочих мест

Интерфейс простой, интуитивно понятный. Что касается дизайна, то он приемлемый. Программа многофункциональна. Хотелось бы отметить следующие ключевые возможности:

1. Детальный учёт компьютеров и ПО.
2. Жизненный цикл учетных объектов.

3. Импорт устройств, ПО, рабочих мест и настроек сети.
4. Автоматизированный аудит рабочих мест.
5. Трассировка сети.
6. Учет и планирование расходных материалов.
7. Учет заявок от пользователей.
8. Инвентаризация учетных объектов.
9. Гибкое разграничение доступа.
10. Поиск информации.
11. Более 30 встроенных настраиваемых отчетов.
12. Подробные справочники по всем аспектам учета.

Программа HardwareInspector распространяется на коммерческой основе, одна лицензия на нее дает право инсталляции программы на любом количестве компьютеров внутри одной локальной сети одной организации.

Конфигурация 1С:Учёт компьютеров и оборудования 8.1. Учет оборудования основан, главным образом, на штрих-кодировании, в результате чего любая операция поиска, подбора или техники становится гораздо проще. С помощью этой конфигурации удобно учитывать и проводить инвентаризации компьютеров, оргтехники и любых других материальных ценностей (оборудование, телефоны, мебель), а также автоматизировать другие сферы деятельности предприятия. На рисунке 1.6 показана основная форма конфигурации 1С.

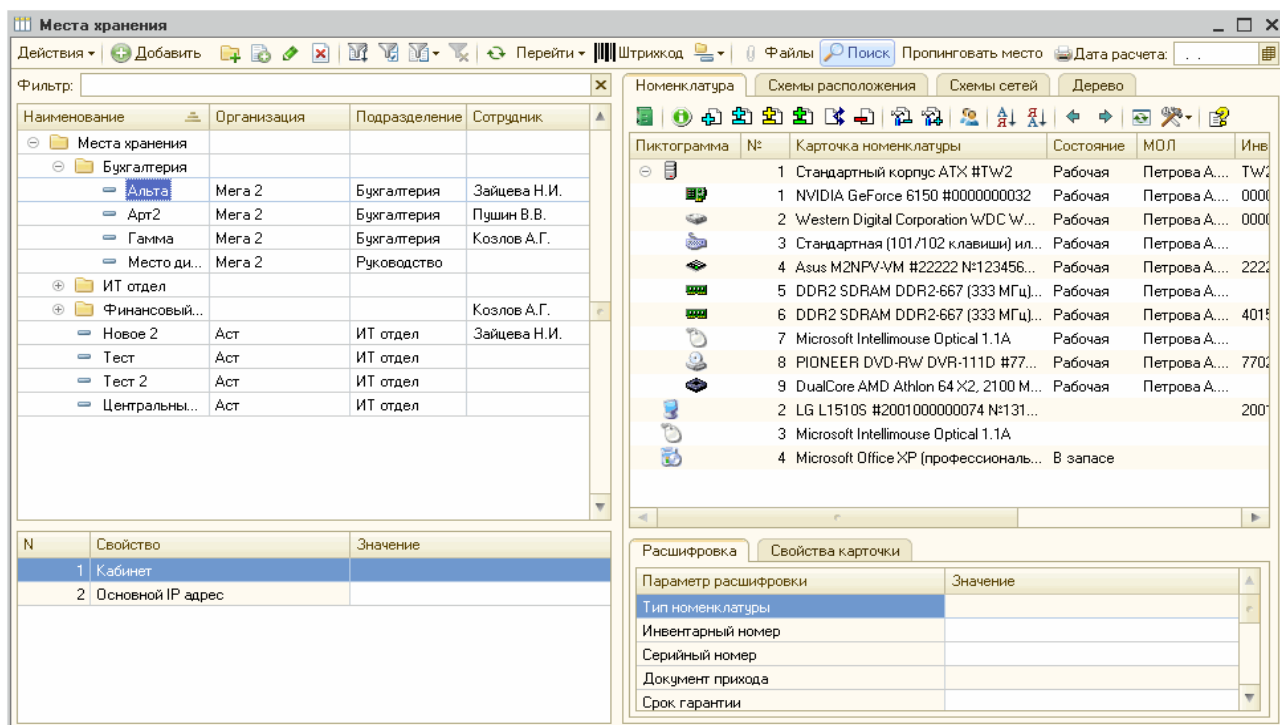


Рисунок 1.6 – 1С:Учёт компьютеров и оборудования 8.1

Основные характеристики продукта:

1. Учёт любой техники, мебели, программного обеспечения.
2. Учёт серийных, инвентарных номеров оборудования.
3. Импорт из системы аппаратного аудита Everest (автоматический сбор данных).
4. Максимально удобный пользовательский интерфейс.
5. Учёт заявок поставщикам.
6. Учёт заявок пользователей и работы с ними.
7. Учёт расходных материалов.
8. Автоматический поиск при сканировании.
9. Индивидуальные наборы настроек и др.

Сравнение выбранных информационных систем представлено в таблице 1 по следующим критериям: функциональность, интерфейс, дизайн, удобство для пользователя, достоинства.

Таблица 1.1 – Сравнение информационных систем

Критерий ПО	«IT Invent»	«Hardware Inspector»	«Конфигурация IC»
Функциональность	Многофункциональна	Многофункциональна	Многофункциональна
Интерфейс	Интуитивно понятный	Простой - интуитивно понятный	Максимально удобный
Дизайн	Хороший	Приемлемый	Стандартный
Удобство для пользователя	Удобно	Проста в использовании	Индивидуальные наборы настроек
Достоинства	Работает по сети	Работает по локальной сети. Обновление 2 раза в месяц. Одну лицензию можно установить на любое количество компьютеров, внутри одной локальной сети, одной организации	Действует свободная линия консультаций по электронной почте и ICQ, а в случае необходимости консультации по телефону.

Выводы по разделу один:

В данном разделе проанализирована предметная область, выделены основные задачи и цели, которые должен решать программный комплекс. Определены требования к программному комплексу. Описана формальная постановка задачи. Также проведен обзор существующих программных продуктов.

2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО КОМПЛЕКСА

2.1 Проектирование базы данных

Для проектируемой системы учета все данные необходимо заносить в базу данных MySQL, управлять которой можно посредством SQL-запросов.

В качестве базы данных выбрана MySQL как одна из широко распространенных баз без ограничений на размер и в то же время достаточно быстрая.

С точки зрения программы база данных MySQL представляет собой удачно организованный набор поименованных таблиц. Каждая таблица – массив (возможно, очень большой) из однородных элементов, которые называются записями. Запись является неделимой единицей информации в базе данных, хотя по запросу можно получать и не всю ее целиком, а только какую-то часть.

Одна из самых популярных сетевых систем управления базой данных Система управления базами данных (далее – СУБД) – это продукт MySQL. Она предназначена для создания небольших баз данных (не более 100 МБ) и поддерживает некоторое подмножество языков запросов SQL. SQL – это специально разработанный стандарт языка запросов к базам данных. В нем присутствуют такие команды, как:

1. создание/удаление таблицы;
2. создание записей в созданной таблице;
3. поиск/удаление записей;
4. обновление некоторых полей созданной записи.

MySQL – это программа-сервер, постоянно работающая на компьютере.

Клиентские программы посылают ей специальные запросы через механизм сокетов (то есть при помощи сетевых средств), она их обрабатывает и запоминает результат. Затем, также по специальному запросу клиента, весь этот результат или его часть передается обратно клиентскому приложению.

Размер результирующего набора данных может быть слишком большим и на его передачу по сети уйдет чересчур много времени, да и редко когда бывает нужно получать весь вывод запроса (то есть все записи, удовлетворяющие выражению запроса). Например, нам может потребоваться лишь подсчитать, сколько записей удовлетворяет тому или иному условию, или же выбрать из данных только первые 10 записей.

Механизм использования сокетов подразумевает технологию клиент-сервер, а это означает, что в системе должна быть запущена специальная программа MySQL-сервер, которая принимает и обрабатывает запросы от программ. Так как вся работа происходит в действительности на одной машине, накладные расходы по работе с сетевыми средствами незначительны (установка и поддержка соединения с MySQL-сервером обходится довольно дешево).

Структура MySQL трехуровневая: базы данных – таблицы – записи. Один сервер MySQL может поддерживать сразу несколько баз данных, доступ к которым может разграничиваться логином и паролем. Зная эти логин и пароль, можно работать с конкретной базой данных. Например, можно создать или удалить в ней таблицу, добавить записи и т.д. Обычно имя-идентификатор и пароль назначаются хостинг-провайдерами, которые и обеспечивают поддержку MySQL для своих пользователей.

2.2 Инфологическая модель базы данных

Цель инфологического моделирования – обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных. Поэтому инфологическую модель данных пытаются строить по аналогии с естественным языком. Основными конструктивными элементами инфологических моделей являются сущности, связи между ними и их свойства (атрибуты).

Прежде чем проектировать структуру базы данных, необходимо создать инфологическую модель требуемой базы данных, в которой дать назначение всем сущностям (таблица 2.1).

Таблица 2.1 – Описание инфологической модели

№	Название сущности	Описание
1	Сотрудники	Отражает информацию о сотрудниках предприятия
2	Устройства	Отражает информацию об устройствах предприятия
3	Отдел	Отражает информацию об отделах
4	Производители	Отражает информацию о производителях компьютеров и комплектующих
5	Свойства	Отражает информацию о свойствах устройств
6	ТипУстройства	Отражает информацию о типах устройств
7	Параметры	Отражает информацию о параметрах свойств

Таблица 2.2 – Атрибуты сущности «Устройства»

Имя столбца	Тип	Ключевое поле
id_устройства	bigint	*
id_тип	bigint	
id_производитель	bigint	
id_сотрудники	bigint	
Инвентарный номер	nvarchar(50)	
Серийный номер	nvarchar(50)	
Наименование	nvarchar(50)	
Дата установки	date	

Таблица 2.2 – Атрибуты сущности «Сотрудники»

Имя столбца	Тип	Ключевое поле
id_сотрудника	bigint	*
id_отдела	bigint	
Табельный номер	bigint	
Фамилия	nvarchar(50)	
Имя	nvarchar(50)	
Отчество	nvarchar(50)	
Должность	nvarchar(50)	

Таблица 2.3 – Атрибуты сущности «Отдел»

Имя столбца	Тип	Ключевое поле
id_отдела	bigint	*
Отдел	nvarchar(50)	

Таблица 2.4 – Атрибуты сущности «Производители»

Имя столбца	Тип	Ключевое поле
id_производителя	bigint	*
производитель	nvarchar(50)	

Таблица 2.5 – Атрибуты сущности «ТипУстройства»

Имя столбца	Тип	Ключевое поле
id_тип	bigint	*
Тип устройства	nvarchar(50)	

Таблица 2.6 – Атрибуты сущности «Параметры»

Имя столбца	Тип	Ключевое поле
id_параметра	bigint	*
Параметр	nvarchar(50)	

Таблица 2.7 – Атрибуты сущности «Свойства»

Имя столбца	Тип	Ключевое поле
id_параметра	bigint	
id_устройства	bigint	
Свойство	nvarchar(50)	

2.3 Структура базы данных

Этап физического проектирования базы данных предусматривает принятие разработчиком окончательного решения о способах реализации создаваемой базы. Поэтому физическое проектирование обязательно производится с учетом всех особенностей выбранной СУБД.

Для реализации программного комплекса будет использоваться база данных MySQL.

Создадим базу данных под именем Svod, в ней создадим семь таблиц, названия которых соответствуют именам сущностей инфологической модели, строки (поля) таблиц соответствуют атрибутам соответствующей сущности. Ниже приведены структуры созданных таблиц:

1. Таблица «Отдел» содержит информацию об отделах предприятия.
2. Таблица «Производители» содержит информацию о производителях устройств.
3. Таблица «Устройства» содержит информацию об устройствах предприятия.
4. Таблица «Сотрудники» содержит информацию о сотрудниках.
5. Таблица «Параметры» содержит информацию обо всех возможных параметрах устройства.
6. Таблица «ТипУстройства» содержит информацию о типах устройства.
7. Таблица «Свойства» содержит информацию обо всех свойствах любого параметра устройства.

База данных проектируемой системы имеет разветвленную структуру с несколькими отношениями «один-ко-многим». На рисунке 2.1 приведена схема структуры.

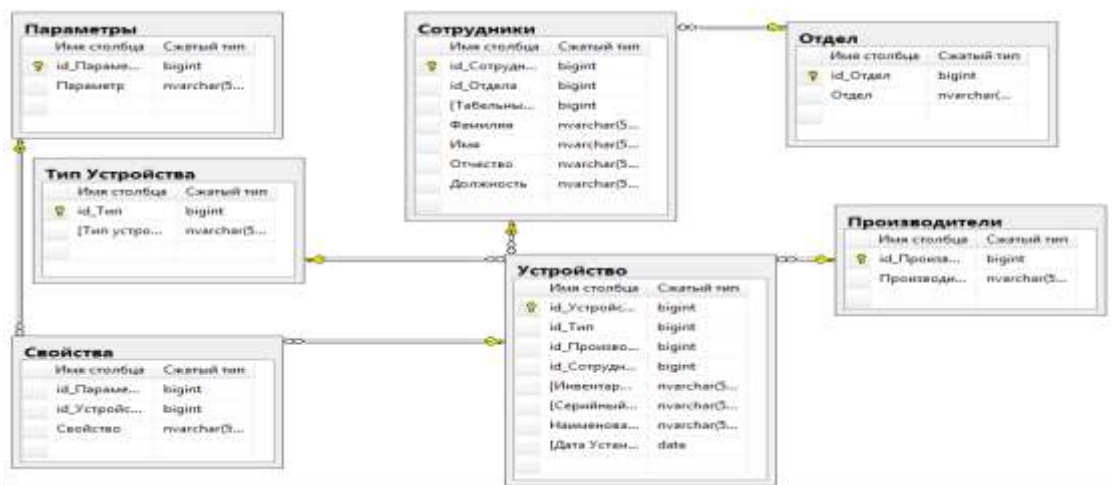


Рисунок 2.1 – Схема структуры базы данных

2.4 Проектирование форм

Поскольку данная программа является приложением Windows, то для ее реализации будем использовать стандартный подход при написании программ под Windows с использованием графического интерфейса.

Программа состоит из 7 форм (включая окно «О программе»). В формах программы расположены компоненты доступа к данным базы MySQL, выборка из которой производится посредством SQL-запросов. Так, отдельно создадим форму с визуальными компонентами, для различных визуальных компонент сопоставляем процедуры-обработчики. Далее, как только происходит событие на каком-то из визуальных компонент, (щелчок мыши, нажатие на кнопку или пункт меню), операционная система посылает приложению соответствующее сообщение, и запускается соответствующий обработчик именно для этого события. Тогда решение задачи можно разбить на 3 этапа:

1. Разработка визуального интерфейса пользователя, на основе экранных форм среды VSC# и визуальных компонент, предназначенных для ввода исходных данных, запуска действий по обработке исходных данных, вывода результата обработки, вывода результатов на экранную форму.

2. Разработка обработчиков событий от элементов экранных форм, разработка алгоритмов и программных кодов действий обработки текстового файла согласно заданию.

3. Подготовка тестовых исходных данных и отладка программы.

Для проектируемого приложения определим количество и назначение форм. Список форм приведен в таблице 2.8.

Таблица 2.8 – Список форм приложения

Имя формы	Заголовок формы	Назначение
Start_Form	Автоматизация	Окно авторизации
Form1_Form	Учет техники на предприятии	Главное окно программы, где отображаются все данные из базы данных
Form2_Form	Добавление/редактирование	Окно добавления и редактирования информации о устройствах и сотрудниках
Form3_Form	Осуществить поиск по:	Окно для просмотра и поиска информации по любому критерию
Form4_Form	Свойства	Окно просмотра и редактирования свойств
Form5_Form	Отчетность	Окно просмотра и вывода на печать отчетов
Form7_Form	Настройка	Окно просмотра и редактирования программы

2.5 Структура программного комплекса

Программный продукт реализован средствами VisualStudio на языке программирования C#.

Архитектура базы данных.

В данном программном комплексе используется архитектура базы данных клиент–сервер. Клиент формирует и отправляет запрос удаленному серверу, на котором размещена база данных. Запрос формируется на языке SQL, после чего удаленный сервер направляет серверу базы данных.

Приведем описание используемых методов и описание их назначения.

GetLoginPassword () – чтение и запись логина и пароля из файла;

newSqlConnection () – создание нового подключения к MS SQL Server;

newSqlCommand() – создание нового запроса в MS SQL Server;

messageBox.Show() – вывод сообщения;

toString() – нахождение текстового представления;

dispose() – выполняет определяемые приложением задачи, связанные с удалением, высвобождением или сбросом неуправляемых ресурсов;

newDataSet() – инициализация нового экземпляра SQL.DataSet;

clear() – удаление всех строк в таблицах элемента DataSet;
open() – открытие подключения к MS SQL Server;
close() – закрытие подключения к MS SQL Server.

2.6 Описание программного интерфейса

При запуске приложения открывается форма авторизации пользователей.

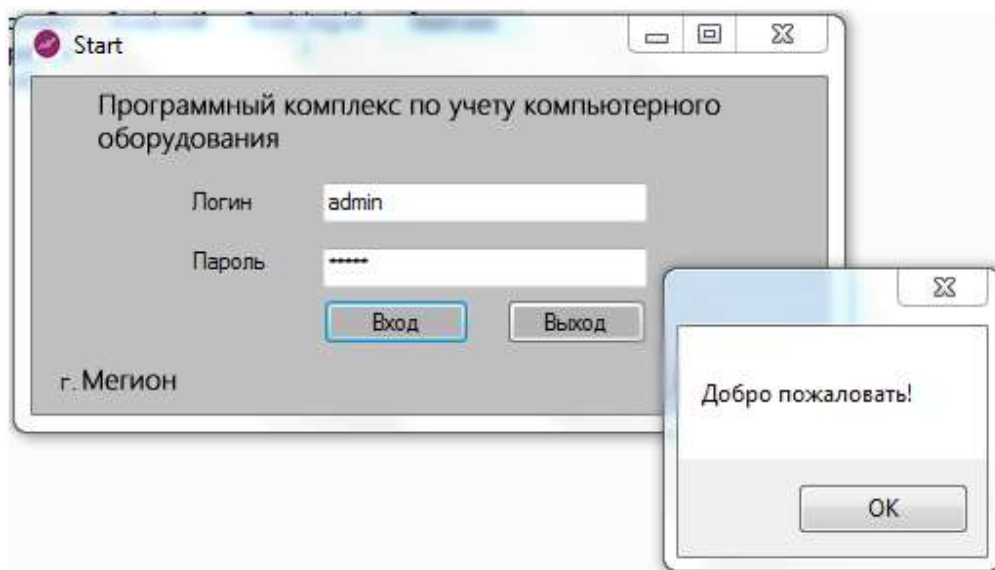


Рисунок 2.2 – Авторизация пользователя

Тип устройства	Производитель	Ответственный	Инвентарный номер	Серийный номер	Наименование	Дата Установки
PC	ViewSonic	Бобков	65455210	81559610	Ноутбук	18.01.2017
Сканер	ViewSonic	Иванов	525252	636363	ывавы	20.01.2017
Сканер	ViewSonic	Бобков	1111	1111	сканер	14.12.2017
Принтер	ViewSonic	Романов	155455	54554	Херох	09.01.2018
Монитор	Asus	Иванов	65545545	135486	Asus VM215	05.12.2017
Мышка	Genius	Юрченко	Клавиатура	665514	Genius Gf 12	05.12.2017
Клавиатура	Genius	Шишков	458798	87	Bloody	20.12.2017
Процессор	Intel	Юрченко	214587	87	Intel COre i5 8700	09.01.2018
Процессор	Intel	Шишков	Процессор	985547	Intel Core 2 Duo	05.12.2017

Рисунок 2.3 – Основное окно программы

После авторизации становится доступной главная форма приложения с информацией о работниках, зарегистрированных в системе; а также, информация об оборудовании. Можно редактировать каждое поле. Для некоторых полей, для которых предусмотрены вспомогательные таблицы в базе данных, при редактировании открывается автоматический список, представлено на рисунке 2.4.

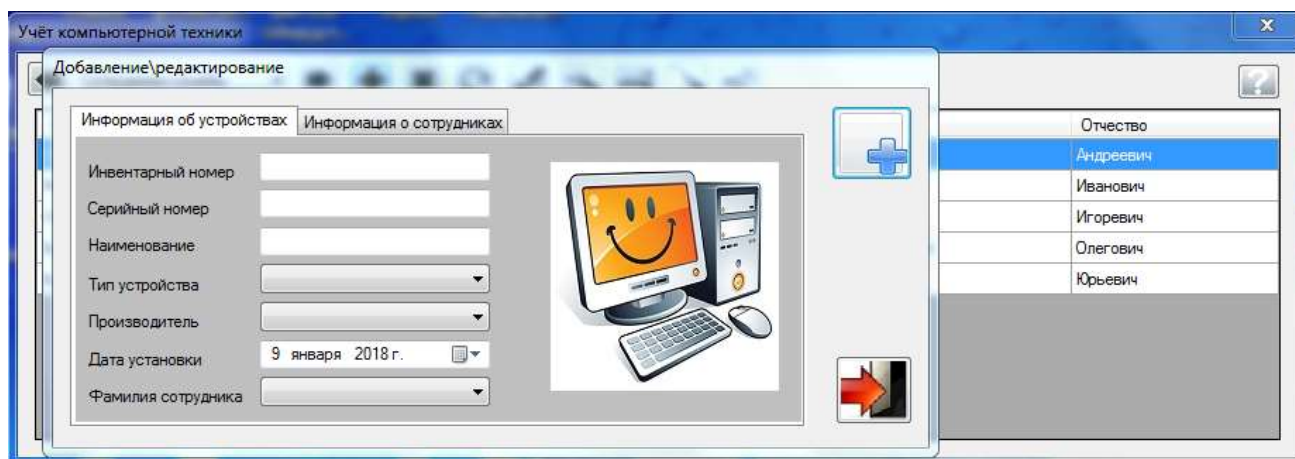


Рисунок 2.4 – Редактирование/добавление записей

Для добавления, удаления, редактирования записи необходимо выбрать нужное поле из таблицы и нажать определённую кнопку.

Также для сокращения времени поиска по определенному параметру предусмотрен поиск. Для этого нужно на панели инструментов нажать кнопку «Поиск» (рисунок 2.5) и выбрать необходимый критерий. Далее по результатам этого самого поиска данные можно вывести данные, как в само приложение, так в печать (рисунок 2.6).

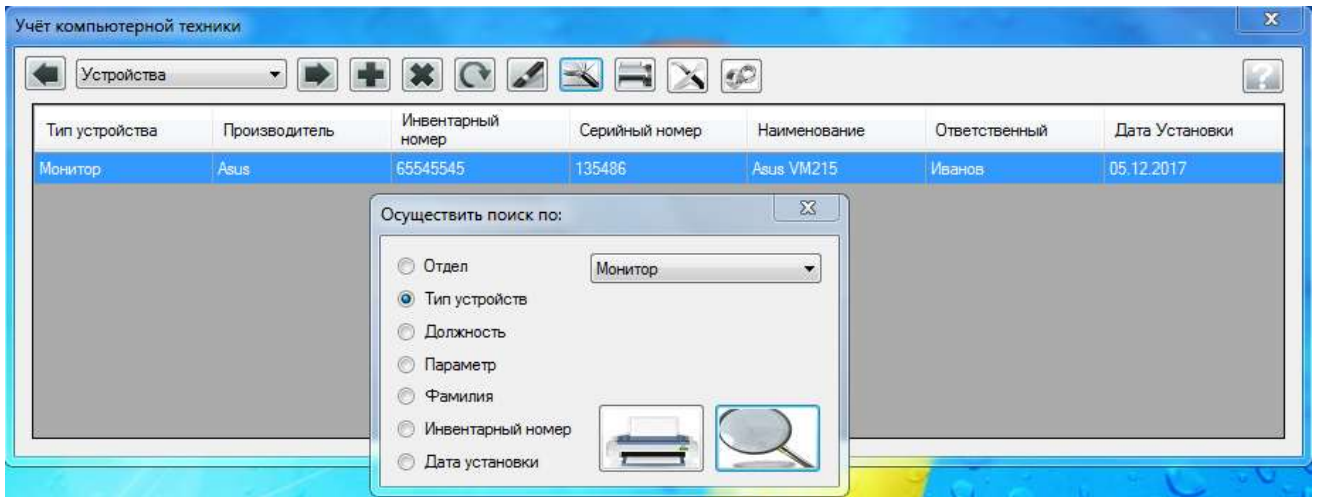


Рисунок 2.5 – Поиск по типу устройства

№ п/п	Тип устройства	Производитель	Ответственный	Инвентарный номер	Серийный номер	Наименование	Дата Установки
1	PC	ViewSonic	Бобков	65455210	81559610	Ноутбук	18.01.2017г.
2	Сканер	ViewSonic	Иванов	525252	636363	Сканер	20.01.2017г.
3	Сканер	ViewSonic	Бобков	1111	1111	Сканер	14.12.2017г.
4	Принтер	ViewSonic	Романов	155455	54554	Xerox	09.01.2018г.
5	Монитор	Asus	Иванов	65545545	135486	Asus VM215	05.12.2017г.
6	Мышка	Genius	Юрченко	Клавиатура	665514	Genius Gf 12	05.12.2017г.
7	Клавиатура	Genius	Шинцов	458798	87	Bloody	20.12.2017г.
8	Процессор	Intel	Юрченко	214587	87	Intel Core i5 8700	09.01.2018г.
9	Процессор	Intel	Шинцов	Процессор	985547	Intel Core 2 Duo	05.12.2017г.

Рисунок 2.6 – Вывод отчета по поиску

2.7 Сообщения оператору

В процессе работы программы пользователю могут выдаваться сообщения, представленные в таблице 2.9.

Таблица 2.9 – Сообщения оператору

Ошибка (сообщение)	Причина возникновения	Способ устранить
Логин и пароль введены не верно	Сотрудник неправильно ввел логин или пароль.	Повторить попытку еще раз, используя свой логин или пароль
Введите значение	Некоторые поля являются пустыми либо содержат информацию недопустимую для данного поля.	Заполнить поля, проверить наличие неверных символов.
Сначала выберите запись для редактирования/удаления	При нажатии на кнопку не выделена ни одна из записей в таблицах	Выберете необходимое для редактирования/удаления поле

Выводы по разделу два:

В данном разделе были показано проектирование базы данных, представлена инфологическая модель. Показаны этапы проектирования форм программного комплекса. Также описан программный интерфейс.

3 ОРГАНИЗАЦИОННО-ЭКОНОМИЧЕСКИЙ РАЗДЕЛ

3.1 Постановка задачи

При разработке программного комплекса огромную роль играют организация производства и управление производством. Целью данной выпускной квалификационной работы является разработка программного комплекса по учету компьютерного оборудования.

3.2 Расчет затрат на создание графического приложения

Создание графического приложения производит один инженер-программист с зарплатой 30000 руб./мес.

Часовая ставка заработной платы вычисляется по формуле (1):

$$Z_{\text{час}} = \frac{\text{Окладвмесяц}}{\text{Фондрабочеговремени}} = \frac{30000}{160} = 188 \text{ руб.} \quad (1)$$

Размер основной заработной платы за i -тый этап определяется как часовая ставка, умноженная на время в часах, необходимое для выполнения i -того этапа рассчитывается по формуле (2):

$$Z_{\text{этап}} = Z_{\text{час}} * t_{\text{вып}} \quad (2)$$

Размер основной заработной платы при восьмичасовом рабочем дне, требуемых на разработку, вычисляется по формуле (3):

$$Z_{\text{осн}} = \sum_i Z_{i\text{-тый_этап}} = \sum_i (Z_{\text{час}} \cdot t_{\text{вып}}), \quad (3)$$

где $t_{вып}$ – время в часах, необходимое для выполнения i -того этапа

В таблице 3.1 приведен расчет трудоемкости, длительности разработки и размера заработной платы.

Таблица 3.1 – Расчет длительности разработки графического приложения и размера заработной платы

Этапы	Виды работ	Исполнители		Часовая ставка, руб.	Длительность выполнения, часы	Размер заработной платы, руб.
		Количество	Должность			
Начальный	Формирование требований к программному комплексу	1	Программист	188	12	2256
Подготовительный	Подбор информации используемой для программного комплекса	1	Программист	188	24	4512
Внешнее проектирование	Разработка архитектуры программного комплекса и его интерфейса	1	Программист	188	24	4512
Основной	Написание кода	1	Программист	188	56	10528
Заключительный	Отладка и исправление	1	Программист	188	24	4512
	Итоговое тестирование	1	Программист	188	8	1504
Всего					148	27824
Дополнительная зарплата (10%)						2782,4
Итого						30606,4

3.3 Определение себестоимости приложения

В себестоимость включаются следующие статьи затрат:

- основная зарплата;
- дополнительная зарплата;
- отчисления на социальные взносы (30%);

- накладные расходы;
- прочие прямые расходы.

Основная и дополнительная зарплата были определены выше. К дополнительной зарплате относятся выплаты, предусмотренные законодательством (оплата отпусков, доплаты за выслугу лет и т.п.). В научных организациях дополнительная зарплата составляет 10-12% от основной (в коммерческих 10 – 30%). Таким образом, сумма отчислений во внебюджетный фонд:

$$O_{\text{страх}} = 30606,4 \cdot 0,3 = 9181,92 \text{ руб.}$$

Затраты на амортизацию оборудования проводятся за период их использования, т.е. за период внедрения и создания дополнений к программному обеспечению.

Денежное выражение амортизации является амортизационным отчислением, которое входит в текущие затраты.

Величина амортизационных отчислений определяется на основе норм амортизации.

Норма амортизации – это установленный размер амортизационных отчислений на полное восстановление, выраженное в %. Норма амортизации устанавливается на основе экономически целесообразного срока службы и должна обеспечить возмещение износа основных средств к моменту возможного их морального и физического износа и создать экономическую основу для замены.

$$A = \frac{30000}{5} \frac{148}{2000} = 444 \text{ руб.}$$

К прочим прямым расходам относятся расходы на обслуживание компьютера и плата за потребляемую электроэнергию. По данным документов энергосбытовой компании – акционерное общество «Тюменская энергосбытовая компания», стоимость 1 кВт-час электроэнергии – 3 руб., потребляемая ЭВМ мощность 0,23 кВт, получим затраты на электроэнергию в размере:

$$P_{\text{проч}} = 148 \cdot 3 \cdot 0,23 = 102,12 \text{ руб.}$$

Рассчитанные величины сведем в таблицу 3.2:

Таблица 3.2 – Себестоимость разработки программного комплекса

Статья расходов	Сумма, руб.	Удельный вес, %
Основная зарплата	27824	55,34
Дополнительная зарплата	2782,4	14,51
Отчисления во внебюджетные фонды (30%)	9181,92	23,62
Амортизация	444	4,7
Прочие прямые расходы	102,12	1,83
Итого	40334,44	100,00

3.4 Экономическая эффективность разработки

Без использования приложения разработанного в рамках выпускной квалификационной работы происходит нерациональная трата рабочего времени сотрудников библиотеки, деятельность которых связана с учетом большого количества оборудования, что серьезно замедляют работу сотрудников. Используя программный комплекс, программист рациональнее использует трудовые ресурсы.

Разрабатываемый программный комплекс предназначен для облегчения работы программиста, поэтому дополнительных рабочих мест, а соответственно и увеличения заработной платы не планируется. Соответственно, дополнительных затрат на оплаты труда не будет.

Для нахождения средней стоимости часа $Z_{\text{ч}}$, необходимо разделить среднюю заработанную плату программиста за день (960 рублей) на количество рабочих часов (8 часов).

$$Z_{\text{ч}} = 960 / 8 = 120 \text{ руб.}$$

При использовании ручного способа на работу с документами, составление отчетов, различных списков тратиться $t_{n1} = 3$ час., а с помощью программы работа с документами должна сократиться примерно до $t_{n2} = 1$ час.

Экономия времени составит:

$$t_{n1} - t_{n2} = 3 \text{ час.} - 1 \text{ час.} = 2 \text{ час. в день для одного рабочего места}$$

За месяц экономия времени одного рабочего места составит:

$T_{pm1} = 26 * 2 \text{ час} = 52 \text{ час.}$ С учетом, что средняя заработная плата специалиста, который будет пользоваться разработанным ПО, составляет 120 руб/час, поэтому экономия на заработной плате за 1 год составит: $12 * 26 * 2 \text{ ч} * 120 \text{ руб/час} = 74880 \text{ руб.}$

В результате экономия составляет 74880 руб.

Выводы по разделу три:

Если МБУ «ЦБС» по г. Мегиону задействует в своей практике разработанный программный комплекс, то экономия составит 74880 рублей. Программный комплекс снизит нерациональную трату рабочего времени сотрудников.

4 БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ

Охрана труда – система сохранения жизни и здоровья работников, включающая в себя социально-экономические, организационные, технические, гигиенические и врачебно-профилактические мероприятия, обеспечивающие безопасность, сохранение здоровья и трудоспособности человека в процессе труда. В связи с научно-техническим прогрессом условия труда стали наиболее напряженными, активными и много затратными как в умственном, так и в эмоциональном плане. Это потребовало комплексного решения проблем эргономики, гигиены и организации труда, регламентации режимов труда и отдыха [13].

Охрана здоровья рабочих, обеспечение безопасности критерий труда, ликвидация болезней и производственного травматизма считается одной из основных проблем человеческого общества. Обращается внимание на необходимость широкого внедрения современных форм научной организации труда, сведения к минимальному количеству ручного, малоквалифицированного труда, создания обстановки, исключающей профессиональные заболевания и производственный травматизм [18].

4.1 Организация рабочего места программиста

Рабочее место – пространство, где сотрудник должен находиться в течение рабочего дня и где он выполняет работу в условиях и режиме, предусмотренных нормативно-технической документацией.

Проектирование рабочих мест относится к числу важных проблем эргономического проектирования в области вычислительной техники. Правильная организация рабочего места, имеет возможность понизить либо свести на нет огромную часть опасных и вредоносных факторов, воздействующих на пользователя [18]. При правильном соблюдении организации рабочего места производительность труда инженера растет с 8 до 20 процентов.

Основными элементами рабочего места программиста считается письменный стол и кресло. Основным рабочим положением считается сидячее положение. Рабочее место для выполнения заданий в сидячем положении организуется в соответствии с ГОСТ.

Конструкция рабочего места и размещения всех его частей обязано соответствовать психологическим, антропометрическим и физическим требованиям. При организации рабочего места программиста должны быть соблюдены требования [18]:

1. оптимальное размещение оборудования, входящего в состав рабочего места;
2. достаточное рабочее пространство, позволяющее осуществлять все необходимые движения и перемещения;
3. необходимо естественное и искусственное освещение для выполнения поставленных задач;
4. уровень акустического шума не должен превышать допустимого значения.

Выполнение трудовых операций «часто» и «очень часто» должно быть обеспечено в пределах зоны легкой досягаемости и оптимальной зоны моторного поля [18].

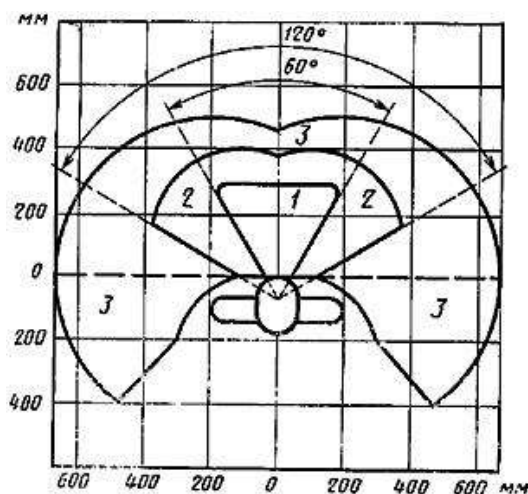


Рисунок 4.1 – Зоны для выполнения ручных операций и размещения органов управления

Где 1 – зона для размещения наиболее важных и очень часто используемых органов управления (оптимальная зона моторного поля);

2 – зона для размещения часто используемых органов управления (зона легкой досягаемости моторного поля);

3 – зона для размещения редко используемых органов управления (зона досягаемости моторного поля).

Характеристики рабочего места выбираются в соответствии с антропометрическими чертами. При применении данных в расчетах надлежит исходить из максимальных антропометрических характеристик.

При работе в сидячем положении рекомендуются следующие характеристики рабочего места:

1. ширина не менее 700 мм;
2. глубина не менее 400 мм;
3. высота рабочей поверхности стола над полом 700-750 мм.

Оптимальными размерами стола являются [18]:

1. высота 710 мм;
2. длина стола 1300 мм;
3. ширина стола 650 мм.

Поверхность для письма должна иметь не менее 400 мм в глубину и не менее 600 мм в ширину.

Под рабочей поверхностью должно быть предусмотрено пространство для ног:

1. высота не менее 600 мм;
2. ширина не менее 500 мм;
3. глубина не менее 400 мм.

Принципиальным составляющим рабочего места считается кресло. Конструкция кресла должна обеспечивать поддержание рациональной рабочей позы при работе, позволять видоизменять позу с целью понижения статического напряжения мышц спины и шейно-плечевой области для предостережения

развития утомления. Вид кресла нужно избирать с учетом роста пользователя, характера и длительности работы. Кресло обязано быть подъемно-поворотным, регулируемым по высоте и углам наклона сиденья и спинки, а еще расстоянию спинки от переднего края сиденья, в этом случаи регулировка каждого параметра должна быть независимой, легко осуществляемой и обладать надежной фиксацией. Поверхность сиденья, спинки и остальных частей кресла должна быть полумягкой, с нескользящим, слабо электризующимся и воздухопроницаемым покрытием, обеспечивающим простую очистку от загрязнений [13].

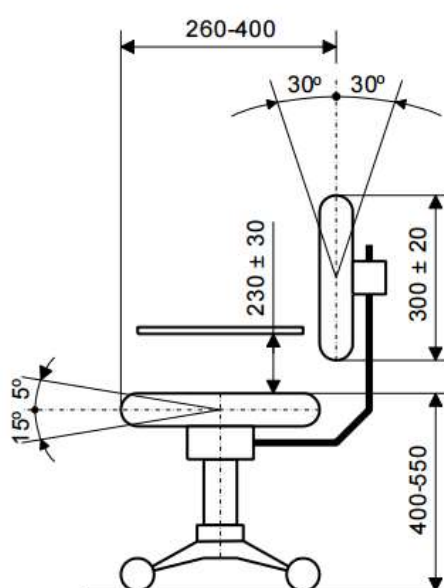


Рисунок 4.2 – Основные размеры рабочего стула

Принципиальным фактором считается также рациональное размещение на рабочем месте документации, канцелярских принадлежностей, что должно помочь занять работающему комфортную рабочую позу, лучшие экономичные движения и минимальные траектории движения работающего и предмета труда на предоставленном рабочем месте.

Создание комфортных условий работы и грамотное оформление рабочего места на предприятии имеет большое значение, как для облегчения работы, так и для увеличения его привлекательности, позитивно оказывающей большое

влияние на продуктивность труда. Расцветка помещений и мебели должна способствовать созданию комфортных условий для визуального восприятия, положительного настроения. В служебных рабочих помещениях, в которых выполняется одинаковая умственная работа, требующая огромного сосредоточения и большого нервного напряжения, расцветка должна быть спокойных тонов – малонасыщенные цвета холодного зеленого либо голубого цветов.

При разработке оптимальных условий труда программиста необходимо учитывать освещенность, шум и микроклимат.

4.2 Микроклимат рабочего места

Микроклимат рабочего помещения – климат внутренней среды здания, который определяется действующими на организм человека сочетаниями температуры, влажности и скорости движения воздуха.

Характеристики микроклимата могут меняться в широких пределах, в то время как важным условием жизнедеятельности человека считается поддержание постоянства температуры тела благодаря свойству терморегуляции, то имеется возможность организма регулировать ответную реакцию тепла в окружающую среду. Главный принцип нормирования микроклимата - создание комфортных условий для теплообмена тела человека с находящейся вокруг средой. В санитарных нормах поставлены величины характеристик микроклимата, создающие благоприятные условия.

Таблица 4.1 – Оптимальные нормы микроклимата для помещений с ПЭВМ

Температура, °С	Относительная влажность, %	Абсолютная влажность, г/м ³	Скорость движения воздуха, м/с
19	62	10	< 0,1
20	58	10	< 0,1
21	55	10	< 0,1

Для обеспечения благоприятных критериев используются организационные способы и технические средства. К организационным способам относятся: рациональная организация проведения работ в зависимости от суток и времени года, также организация правильного чередования отдыха и труда. Технические средства включают кондиционирование воздуха, отопительную систему, вентиляцию.

4.3 Освещенность рабочего места

Программист работает в помещениях, поэтому освещение на рабочих местах должно приближаться к благоприятным условиям зрительного солнечного освещения. Рациональное освещение рабочего места считается одним из важнейших причин, оказывающий большое влияние на эффективность трудовой деятельности человека, предупреждающих профессиональные заболевания и травматизм.

Требования в помещениях предприятий к освещенности, в том месте где установлены компьютеры, таковы: при выполнении зрительных работ высокой точности общая освещенность должна составлять 300 лк, а комбинированная – 750 лк; аналогичные требования при выполнении работ средней точности – 200 и 300 лк соответственно. При этом источники света и монитор должны быть расположены таким образом, чтобы не создавать бликов на поверхности монитора. Кроме этого, все поле зрения программиста должно быть освещено достаточно равномерно, так как это основное гигиеническое требование [13].

По СанПиН 2.6.1.2748-10, помещение с ЭВМ обязаны обладать и искусственным и естественным освещением. При этом дополнительное искусственное освещение применяется как в светлое, так и в темное время суток. Искусственное освещение в помещениях где работают с ЭВМ обязано исполняться системой общего равномерного освещения. В качестве источников искусственного освещения обычно используются люминесцентные лампы, которые попарно объединяются в светильники, которые должны располагаться над рабочими поверхностями равномерно.

Ограничить отраженную блескость на рабочих поверхностях можно за счет верного выбора светильников и расположения рабочих мест по отношению к источникам искусственного и естественного освещения, при этом яркость бликов на экране не должна превышать 40 кд/кВ.м и насыщенность потока, при изменении системы отраженного освещения, не должна превышать 200 кд/кв.м. Для мониторов на ЭЛТ частота обновления изображения должна быть не менее 75 Гц при всех режимах разрешения экрана, гарантируемых нормативной документацией определенный вид монитора и не менее 60 Гц для мониторов на плоских дискретных экранах (жидкокристаллических, плазменных и т.п.).

4.4 Нормирование шума

Шум ухудшает условия труда, оказывая вредоносное воздействие на организм человека. При продолжительном действии шума с человеком могут происходить следующие действия: снижается острота слуха, зрения, понижается внимание, увеличивается кровяное давление. Сильный продолжительный шум может стать причиной многофункциональных изменений нервной и сердечнососудистой систем.

Согласно [15] характеристикой постоянного шума на рабочих местах считаются среднеквадратичные значения давлений в октавных полосах частот со среднегеометрическими обычными частотами: 63, 125, 250, 500, 1000, 2000, 4000 и 8000 Гц. В стандарте указаны значения максимально возможных значений

шума на рабочих местах. Для помещений конструкторских бюро, расчетчиков и разработчиков программного обеспечения уровни шума не должны превосходить: 71, 61, 54, 49, 45, 42, 40, 38 дБ. Эта совокупность нормативных значений звукового давления называется предельным спектром.

На рабочем месте программиста источниками шума, являются технические средства: вентиляционное оборудование, компьютер, принтер, а также внешний шум. Они издают достаточно незначительный шум, поэтому в помещении достаточно использовать звукопоглощение. Убавление шума, проникающего в помещение снаружи, достигается уплотнением по периметру притворов дверей и окон. Звукопоглощение считается достаточно эффективным мероприятием по убавлению шума. Наиболее выраженными звукопоглощающими качествами владеют волокнисто-пористые материалы которые были использованы. К таковым материалам относятся те, коэффициент звукопоглощения которых не ниже 0.2. Наибольшее звукопоглощение станет достигнуто при облицовке не менее 60% единой площади ограждающих поверхностей помещения.

4.5 Электробезопасность

Электробезопасность – система организационных и технических мероприятий, и средств, обеспечивающих охрану людей от вредоносного и опасного воздействия электрического тока, электромагнитного поля и статического электроэнергии.

Опасное и вредоносное воздействие на людей электрического тока, электрической дуги и электромагнитных полей появляются в виде электро-травм и профессиональных заболеваний. Нормы на допустимые токи и напряжения прикосновения в электроустановках должны устанавливаться в соответствии со стандартом [16]. Максимально возможные значения напряжений токов представлены ниже.

Таблица 4.2 – Значения предельно допустимого напряжения и тока

Режим (вид электроустановки)	Вид тока, частота		
	50 Гц		
	U, В	I, мА	Продолжительность
нормальный	2	0,3	не более 10 мин
производственные электроустановки (напряжением до 1000В, с глухо заземленной или изолированной нейтралью)	36	6,0	>1с
аварийный режим (бытовые электроустановки напряжением до 1000 В)	12	2,0	>1с

Для обеспечения защиты от случайного прикосновения к токоведущим частям необходимо применять следующие способы и средства:

1. защитные оболочки;
2. безопасное расположение токоведущих частей;
3. изоляция токоведущих частей;
4. предупредительная сигнализация, блокировка, знаки безопасности.

Компьютеры относятся к электроустановкам напряжением до 1000 В. Для обеспечения защиты от поражения электрическим током, применяют защитное зануление. Сопротивление такого зануления не должно превышать 4 Ом.

4.6 Излучения ПЭВМ

ПЭВМ являются источниками широкополостных электромагнитных излучений:

- мягкого рентгеновского;
- ультрафиолетового 200-400 нм;
- видимого 400-750 нм;
- ближнего ИК 750-2000 нм;
- радиочастотного диапазона 3кГц;

- электростатических полей.

Экспозиционная мощность дозы рентгеновского излучения в любой точке пространства на расстоянии 5 см от поверхности ПЭВМ не должна превышать $7,74 \cdot 10^{-12}$ А/КГ, что соответствует эквивалентной дозе 0,1 мБэр/ч или 100 мкр/ч, согласно [17]. Ультрафиолетовое излучение в диапазоне 200-315 нм не должно превышать 10 мкВт/м², излучение в диапазоне 315-400 нм и видимом диапазоне 400-750 нм - 0,1 Вт/м², в ближнем ИК-диапазоне - 2000нм - 1мм-4 Вт/м². Уровни напряженности электростатического поля не должны превышать 15 кВ/м.

В целях предосторожности следует обязательно использовать защитные экраны, а также рекомендуется ограничивать продолжительность работы с экраном ВДТ, не размещать их концентрированно в рабочей зоне и выключать их, если на них не работают.

Наряду с этим нужно устанавливать в помещении с ВДТ ионизаторы воздуха, чаще проветривать помещение и хотя бы один раз в течение рабочей смены очищать экран от пыли.

4.7 Анализ опасных и вредных производственных факторов

При работе с электроприбором может проявиться ряд опасных и вредных факторов, к числу которых относятся:

- электрические факторы, обусловленные:
- повышенным уровнем электромагнитных излучений;
- повышенным уровнем статического электричества;
- повышенной напряженностью электрического поля;
- повышенной напряженностью магнитного поля.

Электрические факторы появляются из-за электрических сетей, электроприборов, оборудования с электроприводом и т. п. Данные факторы могут повлечь за собой:

- механические травмы;
- психические травмы;

- ожоги.

Психофизиологические факторы, обусловленные статическими физическими перегрузками:

- продолжительная работа с дисплеем;
- работа в неудобной позе.

Психофизиологические факторы, обусловленные нервно-психическими перегрузками:

- умственным перенапряжением;
- перенапряжением анализаторов;
- монотонностью труда.

Психофизиологические факторы являются причиной:

- радикулита;
- миозита;
- психических расстройств;
- усталости и др.

Для предотвращения возникновения перечисленных опасностей необходимо соблюдать меры безопасности при работе с электроприбором.

4.8 Требования к пожарной безопасности

Источником возгорания на рабочем месте может возникнуть короткое замыкание, электрические схемы персонального компьютера, прибора электропитания, сильно нагревающиеся узлы приборов. Потому на 100 м площади таковых помещений, оснащенных компьютерной техникой обязан располагаться в количестве от одного огнетушителя углекислого вида. Все работники организаций обязаны допускаться к работе только после прохождения противопожарного инструктажа, а при изменении особенности работы проходить дополнительное обучение предупреждению и тушению возникающих пожаров. Начальники организаций или индивидуальные предприниматели имеют право назначать лиц, которые по занимаемой должности или по характеру

выполняемых работ должны выполнять соответствующие правила пожарной безопасности. Муниципальные органы в пределах своей компетенции реализуют меры пожарной безопасности в подведомственных организациях и на соответствующих территориях, оказывают важное содействие пожарной охране [18]. Во всех производственных, административных, складских и вспомогательных помещениях на видных местах должны быть вывешены таблички с указанием номера телефона вызова пожарной охраны. В любой организации распорядительным документом должны быть определены и оборудованы места для курения, определен распорядок обесточивания электрооборудования в случае пожара и по окончании рабочего дня, действия работников при обнаружении пожара.

В случае пожара для тушения электронно-вычислительной техники нужно использовать углекислотные и газовые огнетушители. Их достоинством является сохранность электронного оборудования и высокая эффективность тушения пожара.

4.9 Рекомендации по организации режима труда и отдыха

В таблице 4.3 представлено время регламентированных перерывов в зависимости от длительности рабочей смены, вида и категории трудовой деятельности с ВДТ и ПЭВМ.

Таблица 4.3 – Время регламентированных перерывов

Категория работы с ВДТ или ПЭВМ	Уровень нагрузки за рабочую смену при видах работ с ВДТ			Суммарное время регламентированных перерывов, мин.	
	группа А, количество знаков	группа Б, количество знаков	группа В, час.	при 8-ми часовой смене	при 12-ти часовой смене
I	до 20000	до 15000	до 2,0	30	70
II	до 40000	до 30000	до 4,0	50	90
III	до 60000	до 40000	до 6,0	70	120

Выводы по разделу четыре:

В данном разделе выпускной квалификационной работы были изложены требования к рабочему месту программиста. Созданные условия должны обеспечивать комфортную работу. На основании изученной литературы по данной проблеме были указаны оптимальные размеры рабочего стола и кресла, рабочей поверхности, а также проведен выбор системы и расчет оптимального освещения производственного помещения. Соблюдение условий, определяющих оптимальную организацию рабочего места программиста, позволит сохранить хорошую работоспособность в течение всего рабочего дня, повысит производительность труда программиста.

ЗАКЛЮЧЕНИЕ

Цель данной выпускной квалификационной работы – разработка программного комплекса по учету компьютерного оборудования.

Исходя из цели и поставленных задач, была определена структура данной выпускной квалификационной работы, в ходе которой был произведён анализ деятельности учреждения, анализ алгоритмов для создания комплекса, проведен анализ инструментария, представлена программная реализация для написания и описан результат работы программного комплекса.

Результатом решения поставленных задач является создание программного комплекса по учету компьютерного оборудования, который способен сократить временные и человеческие ресурсы.

В начале выполнения выпускной квалификационной работы были изложены теоретические основы построения систем учета.

Рассмотрены известные аналогичные разработки – программы учета компьютерной техники, кратко описана их функциональность и приведены иллюстрации их интерфейса. Система-прототип описана со всеми видами обеспечения: организационное, информационное, техническое и программно-алгоритмическое.

В конце аналитического была сформулирована проблема на этом объекте. Существующую систему учета компьютерной и оргтехники предприятия образует информация на бумажных носителях и частично в файлах MS Excel. Действующая система учета оборудования хаотична, форма и структура файлов не унифицированы. Таким образом, она не способна в полной мере справиться с теми объемами работы, которые существуют в данное время.

Рассмотренная в выпускной квалификационной работе проблема в рамках определённых критериев и ограничений была разбита на две частные задачи: на создание базы данных, необходимой для работы системы, и создания программного комплекса по учету компьютерного оборудования.

В аналитической части была выполнена постановка задачи построения базы данных и постановка задачи построения программного комплекса.

В ходе проектирования базы данных были пройдены все этапы проектирования: описание информационных объектов предметной области, проектирование инфологической модели предметной области, логическое и физическое проектирование базы данных. В качестве СУБД была выбрана MySQL.

Для разработки была выбрана концепция Windows-приложения на основе клиент-серверной архитектуры. В качестве среды разработки была выбрана среда VisualStudio.

В результате была создан программный комплекс, представляющая клиентское приложение для учета оборудования, планирования профилактических работ, исходя из данных по предыдущим профилактическим работам. Программа отвечает основным заявленным требованиям и реализует интуитивно понятный интерфейс.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Астахова, И.Ф. SQL в примерах и задачах: учебное пособие/ И.Ф. Астахова, А.П. Толстобров, В.М. Мельников – Минск: Новое знание, 2002. – 176 с.
2. Иванова, Г.С. Технология программирования: учебник/ Г.С. Иванова – М.: КНОРУС, 2011. – 336 с.
3. Орлова, И.В. Экономико-математические методы и модели: компьютерное моделирование: Учебное пособие / И.В. Орлова. - М.: Вузовский учебник, НИЦ ИНФРА-М, 2013. - 389 с.
4. Учет компьютеров, оборудования и IT инфраструктуры. – <http://it-invent.ru/> [дата обращения – 21.10.2017]
5. Павловская, Т.А. С#. Программирование на языке высокого уровня. Учебник для вузов. / Т.А. Павловская — СПб.: Питер, 2014. — 432 с.: ил.
6. Орлов, С.А. Теория и практика языков программирования: Учебник для вузов. Стандарт 3-го поколения. / С.А. Орлов – СПб.: Питер, 2014. – 688 с.: ил.
7. Иванова, Г.С. Объектно- ориентированное программирование: учебник для вузов / Г.С.Иванова, Т.Н.Ничушкина, Е.К.Пугачев. – М.: Изд-во МГТУ им.Н.Э.Баумана, 2004.-320с.: ил.- (Серия Информатика в техническом университете).
8. Павловская, Т.А. С/С ++. Программирование на языке высокого уровня: учебник для вузов /Т.А.Павловская. – СПб.: Питер, 2009.- 461с.: ил.
9. Барсегян А.А., Куприянов М.С., Степаненко В.В., Холод И.И. Методы и модели анализа данных: OLAP и Data Mining – СПб.: БХВ-Петербург, 2004. – 336 с.: ил.
10. Обеспечение высокой доступности систем на основе MySQL / Пер. с англ. — М. : Издательство «Русская редакция» ; СПб. : БХВ-Петербург, 2012. –

624 .: ил. Скляренко, В.К. Экономика предприятия: Учебное пособие / В.К. Скляренко, В.М. Прудников. – 2-е изд. - М.: НИЦ ИНФРА-М, 2014. - 192 с.

11. Скляренко, В.К. Экономика предприятия: Учебное пособие / В.К. Скляренко, В.М. Прудников. – 2-е изд. - М.: НИЦ ИНФРА-М, 2014. - 192 с.

12. ГОСТ 12.2.032.78 ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования – <http://www.consultant.ru/cons/cgi/online.cgi?req=doc;base=STR;n=6342#0> [дата обращения – 25.11.2017]

13. СанПиН 2.2.2/2.4.1340-03. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы. – http://www.consultant.ru/document/cons_doc_LAW_42836/ [дата обращения – 25.11.2017]

14. ГОСТ 12.1.003-88. Шум. Общие требования безопасности. – <http://docs.cntd.ru/document/5200291> [дата обращения – 25.11.2017]

15. ГОСТ 12.1.003-83. Система стандартов безопасности труда (ССБТ). – <http://docs.cntd.ru/document/5200291> [дата обращения – 25.11.2017]

16. ГОСТ 12.1.038-82 ССБТ И – 1.04.88. Электробезопасность. Предельно допустимые уровни напряжений прикосновения и токов. – <http://docs.cntd.ru/document/5200313> [Дата обращения – 25.11.2017]

17. СанПиН 2.6.1.2748-10. Гигиенические требования по обеспечению радиационной безопасности при работе с источниками неиспользуемого рентгеновского излучения. – <http://docs.cntd.ru/document/902243228> [дата обращения – 26.11.2017]

18. ГОСТ 12.1.004-91 ССБТ. Пожарная безопасность. Общие требования. – <http://docs.cntd.ru/document/9051953> [дата обращения – 26.11.2017]

19. СанПиН 2.2.2.542-96. Гигиенические требования к видеодисплейным терминалам, персональным электронно-вычислительным машинам и организация работ. – <http://docs.cntd.ru/document/5200235> [дата обращения – 27.11.2017]

20. Методические рекомендации по подготовке и оформлению выпускной квалификационной работы (проекта) для технических направлений подготовки 09.03.01 Информатика и вычислительная техника, 09.03.04 Программная инженерия, 12.03.01 Приборостроение, 23.03.01 Технология транспортных процессов / сост. Л.Н.Буйлушкина. - Нижневартовск, 2017. – 35 с.

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А. ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУ ПРОГРАММНОГО КОМПЛЕКСА ПО УЧЕТУ КОМПЬЮТЕРНОГО ОБОРУДОВАНИЯ

1. Наименование разработки

Программный комплекс по учету компьютерного оборудования.

2. Основание для разработки

Разработка ведется на основании задания выпускной квалификационной работы по направлению 09.03.01 Информатика и вычислительная техника.

3. Исполнитель

Исполнитель – Романов Глеб Игоревич.

4. Назначение и цель разработки

Разрабатываемый программный комплекс предназначен для хранения информации о компьютерного оборудования предприятия. Программный комплекс должен позволять специалисту быстро и легко осуществлять поиск и ввод новой информации о компьютерном оборудовании предприятия, хранить информации обо всем компьютерном оборудовании на предприятии.

5. Содержание работы

5.1. Задачи, подлежащие решению

5.1.1 Анализ предметной области и создание ее формального описания.

5.1.2 Определение функций, выполняемых системой.

5.1.3 Выбор основной технологии и средств разработки и реализации программных модулей.

5.1.4 Проектирование базы данных.

5.1.5 Составление структуры программного комплекса.

5.1.6 Разработка интерфейса взаимодействия программы с пользователем.

5.1.7 Разработка алгоритма работы программы.

5.1.8 Оформление документации.

5.2 Требования к программному продукту

5.2.1 Требования к функциональным характеристикам:

Программный комплекс должен выполнять следующие функции:

– Содержать макет приложения, с помощью которого работник сможет вести учет компьютерного оборудования.

– Содержать разделы общей работы системы.

– Осуществлять поиск оборудования.

– Содержать список всего оборудования на предприятии.

5.3 Требования к составу программных компонентов

Для функционирования данной системы необходимо, чтобы на компьютере(сервере) были установлены следующие программные продукты:

– Microsoft SQL Server 2012 или более высокой редакции.

5.4 Требования к архитектуре системы

Система должна включать в себя:

- приложение, производящее работу с базой данных и реализующую интерфейс с пользователем;
- базу данных для хранения информации о оборудовании, пользователях;

5.5 Требования к базе данных

Приложение использует СУБД MySQL для хранения следующей информации:

- сотрудники;
- устройства;
- отделы;
- производители устройств;
- свойства устройств;
- типы устройств;
- параметры приложения.

5.6 Требования к входным и выходным данным

5.6.1 Входные данные

Входными данными является информация о сотрудниках, персональных компьютерах и комплектующих и даты установки оборудования.

5.6.2 Выходные данные

Выходными данным является информация о оборудовании, найденная по запросу специалиста.

5.7 Требования к составу и характеристикам технических средств

Приложение должно функционировать на ЭВМ следующей конфигурации:

- процессор семейства Intel Core i5 и выше;
- оперативная память не менее 4 Гбайт;
- дисковая подсистема со свободным дисковым пространством;
- не менее 20 Гбайт;
- цветной монитор с поддержкой SVGA–режимов;
- стандартная русифицированная клавиатура;
- манипулятор мышь.

5.7.1 Требования к лингвистическому обеспечению

Интерфейс пользователя должен быть реализован на русском языке.

6. Требования к документации

Для приема программного продукта должны быть предоставлены следующие документы:

- 7.1. Техническое задание.
- 7.2. Пояснительная записка.

7. Порядок приема

Прием и контроль программного продукта проводится приемочной комиссией, в состав которой входят представители заказчика и исполнитель, в течение трех рабочих дней после завершения работ.

Дополнительные условия

Данное техническое задание может уточняться и изменяться в установленном порядке.

ПРИЛОЖЕНИЕ Б. КОМПАКТ-ДИСК

Содержание:

1. Пояснительная записка к выпускной квалификационной работе.
2. Разработанный программный продукт.
3. Презентация.

ПРИЛОЖЕНИЕ В. ИСХОДНЫЙ КОД ПРОГРАММНОГО КОМПЛЕКСА ПО УЧЕТУ КОМПЬЮТЕРНОГО ОБОРУДОВАНИЯ

Листинг «Формы 1» файл «Form1.cs»

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.IO;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WindowsFormsApplication5
{
    public partial class Form1 : Form
    {
        private Microsoft.Office.Interop.Excel.Application
excel;

        public Form1()
        {
            try
            {
                data = new DataSet();
                InitializeComponent();
                tool = new ToolTip();
                tool.SetToolTip(button1, "Добавить");
                this.Controls.Add(button1);
                tool.SetToolTip(button_del, "Удалить");
                this.Controls.Add(button_del);
                tool.SetToolTip(button3, "Редактировать");
                this.Controls.Add(button3);
                tool.SetToolTip(button5, "Следующая таблица");
                this.Controls.Add(button5);
                tool.SetToolTip(button6, "Предыдущая
таблица");

                this.Controls.Add(button6);
                tool.SetToolTip(button2, "Обновить");
                this.Controls.Add(button2);
                tool.SetToolTip(button4, "Поиск");
                this.Controls.Add(button4);
                tool.SetToolTip(button_print, "Печать");
            }
        }
    }
}
```

```

        this.Controls.Add(button_print);
        tool.SetToolTip(button7, "О программе");
        this.Controls.Add(button7);
        tool.SetToolTip(button8, "Настройка");
        this.Controls.Add(button8);

        StreamReader      str      =      new
StreamReader("config.txt", Encoding.GetEncoding(1251));
        Speak.connect = str.ReadLine();

        SqlConnection      connection      =      new
SqlConnection(Speak.connect);
        Speak.grid = dataGridView1;

        //      ****для
представлений адаптеры
        string command = "SELECT * FROM View_2";
        Sql_adapter_view = new SqlDataAdapter(command,
connection);
        Sql_adapter_view.Fill(data,      "Сотрудники
службы");

        command = "Select * From View_1";
        Sql_adapter_view_1      =      new
SqlDataAdapter(command, connection);
        Sql_adapter_view_1.Fill(data, "Устройства");

        //
****табличные адаптеры
        command = "Select * From Отдел";
        Sql_adapter_1 = new SqlDataAdapter(command,
connection.ConnectionString);
        Sql_adapter_1.Fill(data, "Отдел");

        command = "Select * From Производители";
        Sql_adapter_2 = new SqlDataAdapter(command,
connection.ConnectionString);
        Sql_adapter_2.Fill(data, "Производители");

        command = "Select * From [Тип Устройства]";
        Sql_adapter_3 = new SqlDataAdapter(command,
connection.ConnectionString);
        Sql_adapter_3.Fill(data, "Тип Устройства");

        command = "Select * From Свойства";
        Sql_adapter_4 = new SqlDataAdapter(command,
connection.ConnectionString);
        Sql_adapter_4.Fill(data, "Свойства");

        command = "Select * From Параметры";

```

```

        Sql_adapter_5 = new SqlDataAdapter(command,
connection.ConnectionString);
        Sql_adapter_5.Fill(data, "Параметры");

        command = "Select * From Сотрудники";
        Sql_adapter_6 = new SqlDataAdapter(command,
connection.ConnectionString);
        Sql_adapter_6.Fill(data, "Сотрудники");

        command = "Select * From Устройство";
        Sql_adapter_7 = new SqlDataAdapter(command,
connection.ConnectionString);
        Sql_adapter_7.Fill(data, "Устройство");

        comboBox1.Items.Add("Сотрудники службы");
        comboBox1.Items.Add("Устройства");
        comboBox1.SelectedIndex = 0;
        dataGridView1.DataSource =
data.Tables[comboBox1.Text];
    }
    catch
    {
        MessageBox.Show("Ошибка установления связи",
"SQL ERROR",
        MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    }
}

private void comboBox1_SelectedIndexChanged(object
sender, EventArgs e)
{
    dataGridView1.DataSource =
data.Tables[comboBox1.Text];
    if (comboBox1.Text == "Сотрудники службы")
        dataGridView1.Columns[0].Visible = false;
    else if (comboBox1.Text == "Устройства")
        dataGridView1.Columns[0].Visible = false;
}

private void Form1_Load(object sender, EventArgs e)
{
}

private void button1_Click(object sender, EventArgs e)
{
    Speak.flag = 100;
    Speak.dat = data;
}

```

```

        Form2 add = new Form2(this);
        add.ShowDialog();
        button2_Click(sender, e);
    }

    public void button2_Click(object sender, EventArgs e)
    {
        data.Clear();
        Sql_adapter_view.Fill(data, "Сотрудники службы");
        Sql_adapter_view_1.Fill(data, "Устройства");
        Sql_adapter_1.Fill(data, "Отдел");
        Sql_adapter_2.Fill(data, "Производители");
        Sql_adapter_3.Fill(data, "Тип Устройства");
        Sql_adapter_4.Fill(data, "Свойства");
        Sql_adapter_5.Fill(data, "Параметры");
        Sql_adapter_6.Fill(data, "Сотрудники");
        Sql_adapter_7.Fill(data, "Устройство");

        comboBox1.SelectedIndex = 1;
    }

    private void button3_Click(object sender, EventArgs e)
    {
        try
        {
            if (dataGridView1.CurrentRow.Index > -1)
            {
                Speak.mydata =
data.Tables[comboBox1.Text];
                Speak.dat = data;
                Speak.Number_strl =
dataGridView1.CurrentRow.Index;
                Speak.flag = comboBox1.SelectedIndex;
                Form2 add = new Form2(this);
                add.ShowDialog();
                button2_Click(sender, e);
            }
        }
        catch
        {
            MessageBox.Show("Выберите запись для
редактирования", "Редактирование",
                MessageBoxButtons.OK,
                MessageBoxIcon.Information);
        }
    }

    private void button_del_Click(object sender, EventArgs
e)
    {

```

```

        try
        {
            if (dataGridView1.CurrentRow == null)
                MessageBox.Show("Сначала выберите запись для удаления", "Удаление",
                    MessageBoxButtons.OK,
                    MessageBoxIcon.Information);
            else if (dataGridView1.CurrentRow.Index > -1)
            {
                if (comboBox1.Text == "Сотрудники службы")
                {
                    if (MessageBox.Show("Вы действительно
                    хотите удалить выбранную запись? Данная запись будет навсегда
                    удалена из БД.", "Удаление", MessageBoxButtons.YesNo) ==
                    DialogResult.Yes)
                    {
                        SqlConnection connection = new
                        SqlConnection(Speak.connect);
                        SqlCommand myCmd = new
                        SqlCommand();
                        connection.Open();
                        myCmd.Connection = connection;
                        myCmd.CommandType =
                        CommandType.StoredProcedure;
                        myCmd.CommandText =
                        "del_record_sotr";

                        myCmd.Parameters.AddWithValue("@id",
                        data.Tables[comboBox1.Text].Rows[dataGridView1.CurrentRow.Index][0])
                        ;
                        myCmd.ExecuteNonQuery();
                        connection.Close();
                        button2_Click(sender, e);
                        MessageBox.Show("Удаление записи
                        выполнено.", "Удаление", MessageBoxButtons.OK,
                        MessageBoxIcon.Information);
                    }
                    else
                    {
                        MessageBox.Show("Удаление записи
                        отменено.", "Удаление", MessageBoxButtons.OK,
                        MessageBoxIcon.Information);
                    }
                }
                else if (comboBox1.Text == "Устройства")
                {
                    if (MessageBox.Show("Вы действительно
                    хотите удалить выбранную запись? Данная запись будет навсегда
                    удалена из БД.", "Удаление", MessageBoxButtons.YesNo) ==
                    DialogResult.Yes)
                    {
                        SqlConnection connection = new
                        SqlConnection(Speak.connect);

```

```

SqlCommand myCmd = new
SqlCommand();
connection.Open();
myCmd.Connection = connection;
myCmd.CommandType =
CommandType.StoredProcedure;
myCmd.CommandText =
"del_record_yst";
myCmd.Parameters.AddWithValue("@id",
data.Tables[comboBox1.Text].Rows[dataGridView1.CurrentRow.Index][0])
;
myCmd.ExecuteNonQuery();
connection.Close();
button2_Click(sender, e);
MessageBox.Show("Удаление записи
выполнено.", "Удаление", MessageBoxButtons.OK,
MessageBoxIcon.Information);
}
else
{
MessageBox.Show("Удаление записи
отменено.", "Удаление", MessageBoxButtons.OK,
MessageBoxIcon.Information);
}
}
}
catch
{
MessageBox.Show("Удаление записи не было
выполнено.", "Удаление", MessageBoxButtons.OK,
MessageBoxIcon.Error);
}
}

private void button5_Click(object sender, EventArgs e)
{
switch (comboBox1.SelectedIndex)
{
case -1:
{
dataGridView1.DataSource =
data.Tables[0];
dataGridView1.Columns[0].Visible =
false;
comboBox1.SelectedIndex = 0;
break;
}
case 0:
{

```



```

data.Tables[1];
false;
dataGridView1.DataSource =
dataGridView1.Columns[1].Visible =
comboBox1.SelectedIndex++;
break;
}
}
private void button6_Click(object sender, EventArgs e)
{
switch (comboBox1.SelectedIndex)
{
case 1:
{
dataGridView1.DataSource =
dataGridView1.Columns[0].Visible =
comboBox1.SelectedIndex--;
break;
}
case 0:
{
dataGridView1.DataSource =
dataGridView1.Columns[0].Visible =
break;
}
}
}
private void button4_Click(object sender, EventArgs e)
{
Speak.dat = data;
Form3 find = new Form3();
find.ShowDialog();
button2_Click(sender, e);
}
private void button_print_Click(object sender,
EventArgs e)
{
Speak.dat = data;
Form5 print = new Form5();
print.ShowDialog();
button2_Click(sender, e);
if (Speak.flag_print == 1)
{

```

```

        dataGridView1.DataSource =
data.Tables["Устройства"];
        dataGridView1.Columns[0].Visible = false;
        excel = new
Microsoft.Office.Interop.Excel.Application();
        excel.Visible = true;
        excel.Workbooks.Add(1);

        Microsoft.Office.Interop.Excel.Worksheet
excel_sheet = excel.Worksheets.get_Item(1);
        excel_sheet.get_Range("A2").Value = "№ п/п";
        excel_sheet.get_Range("B2").Value = "Тип
устройства";
        excel_sheet.get_Range("E2").Value =
"Производитель";
        excel_sheet.get_Range("G2").Value =
"Ответственный";
        excel_sheet.get_Range("I2").Value =
"Инвентарный номер";
        excel_sheet.get_Range("L2").Value = "Серийный
номер";
        excel_sheet.get_Range("O2").Value =
"Наименование";
        excel_sheet.get_Range("Q2").Value = "Дата
Установки";
        for (int i = 0; i < dataGridView1.Rows.Count;
i++)
        {
            excel_sheet.get_Range("A" + (i +
4).ToString()).Value = (i + 1).ToString();
            excel_sheet.get_Range("B" + (i +
4).ToString()).Value =
dataGridView1.Rows[i].Cells[1].Value.ToString();
            excel_sheet.get_Range("E" + (i +
4).ToString()).Value =
dataGridView1.Rows[i].Cells[2].Value.ToString();
            excel_sheet.get_Range("G" + (i +
4).ToString()).Value =
dataGridView1.Rows[i].Cells[3].Value.ToString();
            excel_sheet.get_Range("I" + (i +
4).ToString()).Value =
dataGridView1.Rows[i].Cells[4].Value.ToString();
            excel_sheet.get_Range("L" + (i +
4).ToString()).Value =
dataGridView1.Rows[i].Cells[5].Value.ToString();
            excel_sheet.get_Range("O" + (i +
4).ToString()).Value =
dataGridView1.Rows[i].Cells[6].Value.ToString();
            excel_sheet.get_Range("Q" + (i +
4).ToString()).Value =
dataGridView1.Rows[i].Cells[7].Value.ToString().Substring(0, 10) +
"Г.";
        }

```

```

    }
}
else if (Speak.flag_print == 3)
{
    SqlConnection connection = new
SqlConnection(Speak.connect);
    SqlCommand myCmd = new SqlCommand();
    connection.Open();
    myCmd.Connection = connection;
    myCmd.CommandType =
CommandType.StoredProcedure;
    myCmd.Parameters.AddWithValue("@value",
Speak.parametr_str);
    myCmd.CommandText = "get_by_otdel";
    myCmd.ExecuteNonQuery();
    Sql_zapros_1 = new SqlDataAdapter(myCmd);
    Sql_zapros_1.Fill(data, "Byotdel");

    dataGridView1.DataSource =
data.Tables["Byotdel"];
    excel = new
Microsoft.Office.Interop.Excel.Application();
    excel.Visible = true;
    excel.Workbooks.Add(1);

    Microsoft.Office.Interop.Excel.Worksheet
excel_sheet = excel.Worksheets.get_Item(1);
    excel_sheet.get_Range("A2").Value = "№
п/п";
    excel_sheet.get_Range("B2").Value =
"Отдел";
    excel_sheet.get_Range("E2").Value =
"Табельный номер";
    excel_sheet.get_Range("H2").Value =
"Фамилия";
    excel_sheet.get_Range("J2").Value = "Имя";
    excel_sheet.get_Range("L2").Value =
"Отчетство";
    excel_sheet.get_Range("N2").Value =
"Должность";
    for (int i = 0; i <
dataGridView1.Rows.Count; i++)
    {
        excel_sheet.get_Range("A" + (i +
4).ToString()).Value = (i + 1).ToString();
        excel_sheet.get_Range("B" + (i +
4).ToString()).Value =
dataGridView1.Rows[i].Cells[0].Value.ToString();
        excel_sheet.get_Range("E" + (i +
4).ToString()).Value =
dataGridView1.Rows[i].Cells[1].Value.ToString();
    }
}
}

```

```

                                excel_sheet.get_Range("H" + (i +
4).ToString()).Value =
dataGridView1.Rows[i].Cells[2].Value.ToString();
                                excel_sheet.get_Range("J" + (i +
4).ToString()).Value =
dataGridView1.Rows[i].Cells[3].Value.ToString();
                                excel_sheet.get_Range("L" + (i +
4).ToString()).Value =
dataGridView1.Rows[i].Cells[4].Value.ToString();
                                excel_sheet.get_Range("N" + (i +
4).ToString()).Value =
dataGridView1.Rows[i].Cells[5].Value.ToString();
                                }
                                }
                                else if (Speak.flag_print == 2)
                                {
                                        SqlConnection connection = new
SqlConnection(Speak.connect);
                                        SqlCommand myCmd = new SqlCommand();
                                        connection.Open();
                                        myCmd.Connection = connection;
                                        myCmd.CommandType =
CommandType.StoredProcedure;
                                        myCmd.Parameters.AddWithValue("@value",
Speak.parametr_str);
                                        myCmd.CommandText = "get_by_type";
                                        myCmd.ExecuteNonQuery();
                                        Sql_zapros_1 = new SqlDataAdapter(myCmd);
                                        Sql_zapros_1.Fill(data, "Bytype");

                                        dataGridView1.DataSource =
data.Tables["Bytype"];
                                        excel = new
Microsoft.Office.Interop.Excel.Application();
                                        excel.Visible = true;
                                        excel.Workbooks.Add(1);

                                        Microsoft.Office.Interop.Excel.Worksheet
excel_sheet = excel.Worksheets.get_Item(1);
                                        excel_sheet.get_Range("A2").Value = "№ п/п";
                                        excel_sheet.get_Range("B2").Value = "Тип
устройства";
                                        excel_sheet.get_Range("E2").Value =
"Производитель";
                                        excel_sheet.get_Range("G2").Value =
"Инвентарный номер";
                                        excel_sheet.get_Range("J2").Value = "Серийный
номер";
                                        excel_sheet.get_Range("L2").Value =
"Наименование";
                                        excel_sheet.get_Range("O2").Value =
"Ответственный";

```

```

        excel_sheet.get_Range("Q2").Value = "Дата
Установки";
        for (int i = 0; i < dataGridView1.Rows.Count;
i++)
        {
            excel_sheet.get_Range("A" + (i +
4).ToString()).Value = (i + 1).ToString();
            excel_sheet.get_Range("B" + (i +
4).ToString()).Value
            dataGridView1.Rows[i].Cells[0].Value.ToString();
            excel_sheet.get_Range("E" + (i +
4).ToString()).Value
            dataGridView1.Rows[i].Cells[1].Value.ToString();
            excel_sheet.get_Range("G" + (i +
4).ToString()).Value
            dataGridView1.Rows[i].Cells[2].Value.ToString();
            excel_sheet.get_Range("J" + (i +
4).ToString()).Value
            dataGridView1.Rows[i].Cells[3].Value.ToString();
            excel_sheet.get_Range("L" + (i +
4).ToString()).Value
            dataGridView1.Rows[i].Cells[4].Value.ToString();
            excel_sheet.get_Range("O" + (i +
4).ToString()).Value
            dataGridView1.Rows[i].Cells[5].Value.ToString();
            excel_sheet.get_Range("Q" + (i +
4).ToString()).Value
            dataGridView1.Rows[i].Cells[6].Value.ToString().Substring(0, 10) +
"Г.";
        }
    }

    private void button7_Click(object sender, EventArgs e)
    {
        MessageBox.Show("Данная программ была разработана
в учебных целях.Разработчик студент группы НвФл 526 Романов Глеб
Игоревич", "О программе", MessageBoxButtons.OK,
MessageBoxIcon.Information);
    }
    private void button8_Click(object sender, EventArgs e)
    {
        Speak.dat = data;
        Настройка add_static = new Настройка(this);
        add_static.ShowDialog();
        button2_Click(sender, e);
    }
    public void restForm4Data(int id)
    {
        SqlConnection connection = new
SqlConnection(Speak.connect);
        SqlCommand myCmd = new SqlCommand();

```

```

        connection.Open();
        myCmd.Connection = connection;
        myCmd.CommandType = CommandType.StoredProcedure;
        myCmd.Parameters.AddWithValue("@id", id);
        myCmd.CommandText = "get_sv_vo";
        myCmd.ExecuteNonQuery();
        Sql_zapros_1 = new SqlDataAdapter(myCmd);
        Sql_zapros_1.Fill(data, "Свойств");
    }
private void button9_Click_1(object sender, EventArgs
e)
    {
        try
        {
            if (comboBox1.Text == "Устройства")
            {
                if (dataGridView1.CurrentRow.Index > -1)
                {
restForm4Data (Int32.Parse (data.Tables [comboBox1.Text] .Rows [dataGridV
iew1.CurrentRow.Index] [0] .ToString ())) ;
                    Speak.mydata =
data.Tables [comboBox1.Text] ;
                    Speak.dat = data;
                    Speak.Number_str1 =
dataGridView1.CurrentRow.Index;
                    Speak.idd =
Int32.Parse (data.Tables [comboBox1.Text] .Rows [dataGridView1.CurrentRo
w.Index] [0] .ToString ());
                    Speak.flag = comboBox1.SelectedIndex;
                    Form4 sv = new Form4 (this);
                    sv.ShowDialog ();
                    button2_Click (sender, e);
                }
            }
        }
        catch
        {
            MessageBox.Show ("Выберите запись для
редактирования", "Редактирование",
                MessageBoxButtons.OK,
                MessageBoxIcon.Information);
        }
    }
}
}

```

Листинг «Формы 2» файл «Form2.cs»

```
using System;
```

```

using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WindowsFormsApplication5
{
    public partial class Form2 : Form
    {
        Form1 obnov;

        public Form2(Form1 obn)
        {
            obnov = obn;
            InitializeComponent();
            tool = new ToolTip();
            tool.SetToolTip(button1, "Добавить");
            this.Controls.Add(button1);
            tool.SetToolTip(button2, "Редактировать");
            this.Controls.Add(button2);
            tool.SetToolTip(button_отмена, "Отменить и
            выйти");

            this.Controls.Add(button_отмена);
            button2.Hide();
            Speak.idd = 1;
            button1.Show();
            tabPage1.Show();
            tabPage4.Show();
            comboBox5.DataSource = Speak.dat.Tables["Отдел"];
            comboBox5.ValueMember = "Отдел";

            comboBox1.DataSource =
            Speak.dat.Tables["Сотрудники"];
            comboBox1.ValueMember = "Фамилия";

            comboBox3.DataSource =
            Speak.dat.Tables["Производители"];
            comboBox3.ValueMember = "Производитель";

            comboBox2.DataSource =
            Speak.dat.Tables["Тип_Устройства"];
            comboBox2.ValueMember = "Тип устройства";

            if (Speak.flag == 0)
            {

```

```

        button2.Show();
        button1.Hide();
        tabPage1.Parent = null;
        textBox7.Text =
Speak.mydata.Rows[Speak.Number_str1][4].ToString();
        textBox6.Text =
Speak.mydata.Rows[Speak.Number_str1][5].ToString();
        textBox5.Text =
Speak.mydata.Rows[Speak.Number_str1][6].ToString();
        textBox8.Text =
Speak.mydata.Rows[Speak.Number_str1][2].ToString();
        textBox9.Text =
Speak.mydata.Rows[Speak.Number_str1][3].ToString();
        comboBox5.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDown;
        comboBox5.Text =
Speak.mydata.Rows[Speak.Number_str1][1].ToString();
        comboBox5.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDownList;
        Speak.flag = 2;
    }
    else if (Speak.flag == 1)
    {
        button2.Show();
        button1.Hide();
        tabPage4.Parent = null;
        textBox1.Text =
Speak.mydata.Rows[Speak.Number_str1][4].ToString();
        textBox2.Text =
Speak.mydata.Rows[Speak.Number_str1][5].ToString();
        textBox3.Text =
Speak.mydata.Rows[Speak.Number_str1][6].ToString();
        comboBox2.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDown;
        comboBox2.Text =
Speak.mydata.Rows[Speak.Number_str1][1].ToString();
        comboBox2.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDownList;
        comboBox3.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDown;
        comboBox3.Text =
Speak.mydata.Rows[Speak.Number_str1][2].ToString();
        comboBox3.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDownList;
        comboBox1.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDown;
        comboBox1.Text =
Speak.mydata.Rows[Speak.Number_str1][3].ToString();
        comboBox1.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDownList;
        dateTimePicker1.Value =
Convert.ToDateTime(Speak.mydata.Rows[Speak.Number_str1][7]);

```



```

        Speak.flag = 3;
    }
    else
    {
        comboBox1.SelectedIndex = -1;
        comboBox2.SelectedIndex = -1;
        comboBox3.SelectedIndex = -1;
        comboBox5.SelectedIndex = -1;
    }
}

private void button1_Click(object sender, EventArgs e)
//добавление
{
    if (MessageBox.Show("Подтвердите добавление новой
записи?", "Добавление", MessageBoxButtons.YesNo) ==
DialogResult.Yes)
    {
        try
        {
            SqlConnection connection = new
SqlConnection(Speak.connect);
            SqlCommand myCmd = new SqlCommand();
            connection.Open();
            myCmd.Connection = connection;
            if (tabControl1.SelectedIndex == 0)
            {
                if ((textBox1.Text.ToString().Length
== 0) || (textBox2.Text.ToString().Length == 0)
                ||
(textBox3.Text.ToString().Length == 0) ||
(comboBox1.Text.ToString().Length == 0)
                ||
(comboBox1.Text.ToString().Length == 0) ||
(comboBox3.Text.ToString().Length == 0))
                    MessageBox.Show("Введите
значение", "Добавление", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                else
                {
                    myCmd.Connection = connection;
                    myCmd.CommandType =
CommandType.StoredProcedure;
                    myCmd.CommandText =
"add_record_yst";
                    myCmd.Parameters.AddWithValue("@i", textBox1.Text);
                    myCmd.Parameters.AddWithValue("@s", textBox2.Text);
                    myCmd.Parameters.AddWithValue("@n", textBox3.Text);
                }
            }
        }
    }
}

```

```

        myCmd.Parameters.AddWithValue("@t",
comboBox2.Text.ToString());

myCmd.Parameters.AddWithValue("@p", comboBox3.Text.ToString());

myCmd.Parameters.AddWithValue("@d", dateTimePicker1.Value.Date);

myCmd.Parameters.AddWithValue("@f", comboBox1.Text.ToString());
        myCmd.ExecuteNonQuery();
        connection.Close();
        MessageBox.Show("Добавление записи
выполнено.", "Добавление", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        this.Close();
    }
}
else if (tabControl1.SelectedIndex == 1)
{
    if ((textBox5.Text.ToString().Length
== 0) || (textBox6.Text.ToString().Length == 0)
||
(textBox7.Text.ToString().Length == 0) ||
(textBox8.Text.ToString().Length == 0)
||
(textBox9.Text.ToString().Length == 0) ||
(comboBox5.Text.ToString().Length == 0))
        MessageBox.Show("Введите
значение", "Добавление", MessageBoxButtons.OK,
MessageBoxIcon.Error);
    else
    {
        myCmd.Connection = connection;
        myCmd.CommandType =
CommandType.StoredProcedure;
        myCmd.CommandText =
"add_record_sotr";
        myCmd.Parameters.AddWithValue("@f",
textBox7.Text);

myCmd.Parameters.AddWithValue("@n", textBox6.Text);

myCmd.Parameters.AddWithValue("@s", textBox5.Text);

myCmd.Parameters.AddWithValue("@t", Int32.Parse(textBox8.Text));

myCmd.Parameters.AddWithValue("@d", textBox9.Text);

myCmd.Parameters.AddWithValue("@o", comboBox5.Text);
        myCmd.ExecuteNonQuery();
        connection.Close();
    }
}

```

```

        MessageBox.Show("Добавление записи
выполнено.", "Добавление", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        this.Close();
    }
}
}
catch
{
    MessageBox.Show("Добавление записи не было
выполнено. Возможно были введены некорректные данные.",
"Добавление", MessageBoxButtons.OK,
MessageBoxIcon.Error);
}
}

private void button_edit_Click(object sender,
EventArgs e) //////////////редактирование
{
    if (MessageBox.Show("Подтвердите редактирование
записи?", "Редактирование", MessageBoxButtons.YesNo) ==
DialogResult.Yes)
    {
        try
        {
            SqlConnection connection = new
SqlConnection(Speak.connect);
            SqlCommand myCmd = new SqlCommand();
            connection.Open();
            myCmd.Connection = connection;
            if (Speak.flag == 3)
            {
                if ((textBox1.Text.ToString()).Length
== 0) || (textBox2.Text.ToString()).Length == 0)
                ||
                (textBox3.Text.ToString()).Length == 0) ||
                (comboBox1.Text.ToString()).Length == 0)
                ||
                (comboBox1.Text.ToString()).Length == 0) ||
                (comboBox3.Text.ToString()).Length == 0))
                MessageBox.Show("Введите
значение", "Добавление", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            else
            {
                myCmd.Connection = connection;
                myCmd.CommandType =
CommandType.StoredProcedure;

```

```

myCmd.CommandText =
"up_record_yst";

myCmd.Parameters.AddWithValue("@id",
Speak.mydata.Rows[Speak.Number_str1][0]);

myCmd.Parameters.AddWithValue("@i", textBox1.Text);
myCmd.Parameters.AddWithValue("@s", textBox2.Text);
myCmd.Parameters.AddWithValue("@n", textBox3.Text);
myCmd.Parameters.AddWithValue("@t", comboBox2.Text.ToString());
myCmd.Parameters.AddWithValue("@p", comboBox3.Text.ToString());
myCmd.Parameters.AddWithValue("@d", dateTimePicker1.Value.Date);
myCmd.Parameters.AddWithValue("@f", comboBox1.Text.ToString());
myCmd.ExecuteNonQuery();
connection.Close();
MessageBox.Show("Добавление записи
выполнено.", "Добавление", MessageBoxButtons.OK,
MessageBoxIcon.Information);
this.Close();
}
}
else if (Speak.flag == 2)
{
if ((textBox5.Text.ToString().Length
== 0) || (textBox6.Text.ToString().Length == 0)
||
(textBox7.Text.ToString().Length == 0) ||
(textBox8.Text.ToString().Length == 0)
||
(textBox9.Text.ToString().Length == 0) ||
(comboBox5.Text.ToString().Length == 0))
MessageBox.Show("Введите
значение", "Добавление", MessageBoxButtons.OK,
MessageBoxIcon.Error);
else
{
myCmd.Connection = connection;
myCmd.CommandType =
CommandType.StoredProcedure;
myCmd.CommandText =
"up_record_sotr";

myCmd.Parameters.AddWithValue("@id",
Speak.mydata.Rows[Speak.Number_str1][0]);

myCmd.Parameters.AddWithValue("@f", textBox7.Text);

```

```

myCmd.Parameters.AddWithValue("@n", textBox6.Text);

myCmd.Parameters.AddWithValue("@s", textBox5.Text);

myCmd.Parameters.AddWithValue("@t", Int32.Parse(textBox8.Text));

myCmd.Parameters.AddWithValue("@d", textBox9.Text);

myCmd.Parameters.AddWithValue("@o", comboBox5.Text);
myCmd.ExecuteNonQuery();
connection.Close();
MessageBox.Show("Добавление записи
выполнено.", "Добавление", MessageBoxButtons.OK,
MessageBoxIcon.Information);
this.Close();
}
}
}
catch
{
    MessageBox.Show("Редактирование записи не
было выполнено. Возможно были введены некорректные данные.",
"Редактирование",
MessageBoxButtons.OK,
MessageBoxIcon.Error);
}
}
private void button_отмена_Click(object sender,
EventArgs e)
{
    if (MessageBox.Show("Отменить и выйти?", "Выход",
MessageBoxButtons.YesNo) == DialogResult.Yes)
        this.Close();
}
private void textBox7_KeyPress(object sender,
KeyPressEventArgs e)
{
    if (!Char.IsLetter(e.KeyChar) && e.KeyChar !=
Convert.ToChar(8))
        e.Handled = true;
}
private void textBox6_KeyPress(object sender,
KeyPressEventArgs e)
{
    if (!Char.IsLetter(e.KeyChar) && e.KeyChar !=
Convert.ToChar(8))
        e.Handled = true;
}

```

```

        private void textBox5_KeyPress(object sender,
KeyPressEventArgs e)
        {
            if (!Char.IsLetter(e.KeyChar) && e.KeyChar !=
Convert.ToChar(8))
                e.Handled = true;
        }
    }
}

```

Листинг «Формы 3» файл «Form3.cs»

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WindowsFormsApplication5
{
    public partial class Form3 : Form
    {
        private Microsoft.Office.Interop.Excel.Application
excel;

        public Form3()
        {
            InitializeComponent();
            comboBox2.Hide();
            textBox1.Hide();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            SqlConnection connection = new
SqlConnection(Speak.connect);
            SqlCommand myCmd = new SqlCommand();
            connection.Open();
            myCmd.Connection = connection;
            myCmd.CommandType =
CommandType.StoredProcedure;
            if (radioButton1.Checked == true)
            {
                myCmd.Parameters.AddWithValue("@value",
comboBox2.Text);
            }
        }
    }
}

```

```

        if
(Speak.dat.Tables["get_by_otdel"]!=null)
Speak.dat.Tables["get_by_otdel"].Clear();
        myCmd.CommandText = "get_by_otdel";
        myCmd.ExecuteNonQuery();
        Sql_adapter_find = new
SqlDataAdapter(myCmd);
        Sql_adapter_find.Fill(Speak.dat,
"get_by_otdel");
        Speak.grid.DataSource =
Speak.dat.Tables["get_by_otdel"];
        textBox1.Clear();
    }
    else if (radioButton2.Checked == true)
    {
        myCmd.Parameters.AddWithValue("@value",
comboBox2.Text);
        if (Speak.dat.Tables["get_by_type"] !=
null)
Speak.dat.Tables["get_by_type"].Clear();
        myCmd.CommandText = "get_by_type";
        myCmd.ExecuteNonQuery();
        Sql_adapter_find = new
SqlDataAdapter(myCmd);
        Sql_adapter_find.Fill(Speak.dat,
"get_by_type");
        Speak.grid.DataSource =
Speak.dat.Tables["get_by_type"];
        textBox1.Clear();
    }
    else if (radioButton3.Checked == true)
    {
        if (textBox1.Text.ToString().Length != 0)
        {
myCmd.Parameters.AddWithValue("@value", textBox1.Text);
            if (Speak.dat.Tables["get_by_dolg"] !=
null)
Speak.dat.Tables["get_by_dolg"].Clear();
            myCmd.CommandText = "get_by_dolg";
            myCmd.ExecuteNonQuery();
            Sql_adapter_find = new
SqlDataAdapter(myCmd);
            Sql_adapter_find.Fill(Speak.dat,
"get_by_dolg");
            Speak.grid.DataSource =
Speak.dat.Tables["get_by_dolg"];
            textBox1.Clear();
        }
    }

```

```

else
{
    MessageBox.Show("Введите ключевое
слово поиска", "Поиск", MessageBoxButtons.OK,
MessageBoxIcon.Information);
}
}
else if (radioButton8.Checked == true)
{
    if (textBox1.Text.ToString().Length != 0)
    {
myCmd.Parameters.AddWithValue("@value", textBox1.Text);
        if (Speak.dat.Tables["get_by_fam"] !=
null)
Speak.dat.Tables["get_by_fam"].Clear();
        myCmd.CommandText = "get_by_fam";
        myCmd.ExecuteNonQuery();
        Sql_adapter_find = new
SqlDataAdapter(myCmd);
        Sql_adapter_find.Fill(Speak.dat,
"get_by_fam");
        Speak.grid.DataSource =
Speak.dat.Tables["get_by_fam"];
        textBox1.Clear();
    }
}
else
{
    MessageBox.Show("Введите ключевое
слово поиска", "Поиск", MessageBoxButtons.OK,
MessageBoxIcon.Information);
}
}
else if (radioButton7.Checked == true)
{
    if (textBox1.Text.ToString().Length != 0)
    {
myCmd.Parameters.AddWithValue("@value", Int32.Parse(textBox1.Text));
        if (Speak.dat.Tables["get_by_inv"] !=
null)
Speak.dat.Tables["get_by_inv"].Clear();
        myCmd.CommandText = "get_by_inv";
        myCmd.ExecuteNonQuery();
        Sql_adapter_find = new
SqlDataAdapter(myCmd);
        Sql_adapter_find.Fill(Speak.dat,
"get_by_inv");
        Speak.grid.DataSource =
Speak.dat.Tables["get_by_inv"];
    }
}
}

```



```

        textBox1.Clear();
    }
    else
    {
        MessageBox.Show("Введите ключевое
слово поиска", "Поиск", MessageBoxButtons.OK,
MessageBoxIcon.Information);
    }
}
else if (radioButton6.Checked == true)
{
myCmd.Parameters.AddWithValue("@value", dateTimePicker3.Value.Date);
if (Speak.dat.Tables["get_by_date"] !=
null)
Speak.dat.Tables["get_by_date"].Clear();
myCmd.CommandText = "get_by_date";
myCmd.ExecuteNonQuery();
Sql_adapter_find = new
SqlDataAdapter(myCmd);
Sql_adapter_find.Fill(Speak.dat,
"get_by_date");
Speak.grid.DataSource =
Speak.dat.Tables["get_by_date"];
}
else if (radioButton4.Checked == true)
{
if (textBox1.Text.ToString().Length != 0)
{
myCmd.Parameters.AddWithValue("@value2", textBox1.Text);
myCmd.Parameters.AddWithValue("@value1", comboBox2.Text);
if (Speak.dat.Tables["get_by_par"] !=
null)
Speak.dat.Tables["get_by_par"].Clear();
myCmd.CommandText = "get_by_par";
myCmd.ExecuteNonQuery();
Sql_adapter_find = new
SqlDataAdapter(myCmd);
Sql_adapter_find.Fill(Speak.dat,
"get_by_par");
Speak.grid.DataSource =
Speak.dat.Tables["get_by_par"];
textBox1.Clear();
}
else
{

```

```

        MessageBox.Show("Введите ключевое
слово поиска", "Поиск", MessageBoxButtons.OK,
MessageBoxIcon.Information);
    }
}

private void radioButton1_CheckedChanged(object
sender, EventArgs e)
{
    textBox1.Hide();
    comboBox2.Show();
    dateTimePicker3.Hide();
    comboBox2.DataSource = Speak.dat.Tables["Отдел"];
    comboBox2.ValueMember = "Отдел";
}

private void radioButton2_CheckedChanged(object
sender, EventArgs e)
{
    textBox1.Hide();
    comboBox2.Show();
    dateTimePicker3.Hide();
    comboBox2.DataSource =
Speak.dat.Tables["Тип_Устройства"];
    comboBox2.ValueMember = "Тип устройства";
}

private void radioButton4_CheckedChanged(object
sender, EventArgs e)
{
    textBox1.Show();
    comboBox2.Show();
    dateTimePicker3.Hide();
    comboBox2.DataSource =
Speak.dat.Tables["Параметры"];
    comboBox2.ValueMember = "Параметр";
}

private void radioButton8_CheckedChanged(object
sender, EventArgs e)
{
    textBox1.Show();
    comboBox2.Hide();
    dateTimePicker3.Hide();
}

private void radioButton7_CheckedChanged(object
sender, EventArgs e)
{

```

```

        textBox1.Show();
        comboBox2.Hide();
        dateTimePicker3.Hide();
    }

    private void radioButton6_CheckedChanged(object sender, EventArgs e)
    {
        dateTimePicker3.Show();
        textBox1.Hide();
        comboBox2.Hide();
    }

    private void radioButton3_CheckedChanged(object sender, EventArgs e)
    {
        textBox1.Show();
        comboBox2.Hide();
        dateTimePicker3.Hide();
    }

    private void button2_Click(object sender, EventArgs e)
    {
        if (radioButton1.Checked == true)
        {
            if (Speak.dat.Tables["get_by_otdel"] != null){
                excel = new
Microsoft.Office.Interop.Excel.Application();
                excel.Visible = true;
                excel.Workbooks.Add(1);

                Microsoft.Office.Interop.Excel.Worksheet
excel_sheet = excel.Worksheets.get_Item(1);
                excel_sheet.get_Range("A2").Value = "№
п/п";
                excel_sheet.get_Range("B2").Value =
"Отдел";
                excel_sheet.get_Range("E2").Value =
"Табельный номер";
                excel_sheet.get_Range("H2").Value =
"Фамилия";
                excel_sheet.get_Range("J2").Value = "Имя";
                excel_sheet.get_Range("L2").Value =
"Отчетство";
                excel_sheet.get_Range("N2").Value =
"Должность";
                for (int i = 0; i < Speak.grid.Rows.Count;
i++)
                {
                    excel_sheet.get_Range("A" + (i +
4).ToString()).Value = (i + 1).ToString();

```

```

        excel_sheet.get_Range("B" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[0].Value.ToString();
        excel_sheet.get_Range("E" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[1].Value.ToString();
        excel_sheet.get_Range("H" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[2].Value.ToString();
        excel_sheet.get_Range("J" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[3].Value.ToString();
        excel_sheet.get_Range("L" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[4].Value.ToString();
        excel_sheet.get_Range("N" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[5].Value.ToString();
    }
}
else if (radioButton2.Checked == true)
{
    if (Speak.dat.Tables["get_by_type"] != null)
    {
        excel = new
Microsoft.Office.Interop.Excel.Application();
        excel.Visible = true;
        excel.Workbooks.Add(1);

        Microsoft.Office.Interop.Excel.Worksheet
excel_sheet = excel.Worksheets.get_Item(1);
        excel_sheet.get_Range("A2").Value = "№
п/п";
        excel_sheet.get_Range("B2").Value = "Тип
устройства";
        excel_sheet.get_Range("E2").Value =
"Производитель";
        excel_sheet.get_Range("G2").Value =
"Инвентарный номер";
        excel_sheet.get_Range("J2").Value =
"Серийный номер";
        excel_sheet.get_Range("L2").Value =
"Наименование";
        excel_sheet.get_Range("O2").Value =
"Ответственный";
        excel_sheet.get_Range("Q2").Value = "Дата
Установки";
        for (int i = 0; i < Speak.grid.Rows.Count;
i++)
        {
            excel_sheet.get_Range("A" + (i +
4).ToString()).Value = (i + 1).ToString();
            excel_sheet.get_Range("B" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[0].Value.ToString();
            excel_sheet.get_Range("E" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[1].Value.ToString();

```

```

        excel_sheet.get_Range("G" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[2].Value.ToString();
        excel_sheet.get_Range("J" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[3].Value.ToString();
        excel_sheet.get_Range("L" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[4].Value.ToString();
        excel_sheet.get_Range("O" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[5].Value.ToString();
        excel_sheet.get_Range("Q" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[6].Value.ToString().Substring(0, 10) +
"Г.";
    }
}
else if (radioButton3.Checked == true)
{
    if (Speak.dat.Tables["get_by_dolg"] != null)
    {
        excel = new
Microsoft.Office.Interop.Excel.Application();
        excel.Visible = true;
        excel.Workbooks.Add(1);

        Microsoft.Office.Interop.Excel.Worksheet
excel_sheet = excel.Worksheets.get_Item(1);
        excel_sheet.get_Range("A2").Value = "№
п/п";
        excel_sheet.get_Range("B2").Value =
"Отдел";
        excel_sheet.get_Range("E2").Value =
"Табельный номер";
        excel_sheet.get_Range("H2").Value =
"Фамилия";
        excel_sheet.get_Range("J2").Value = "Имя";
        excel_sheet.get_Range("L2").Value =
"Отчетство";
        excel_sheet.get_Range("N2").Value =
"Должность";
        for (int i = 0; i < Speak.grid.Rows.Count;
i++)
        {
            excel_sheet.get_Range("A" + (i +
4).ToString()).Value = (i + 1).ToString();
            excel_sheet.get_Range("B" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[0].Value.ToString();
            excel_sheet.get_Range("E" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[1].Value.ToString();
            excel_sheet.get_Range("H" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[2].Value.ToString();
            excel_sheet.get_Range("J" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[3].Value.ToString();

```

```

        excel_sheet.get_Range("L" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[4].Value.ToString();
        excel_sheet.get_Range("N" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[5].Value.ToString();
    }
}

}
else if (radioButton8.Checked == true)
{

    if (Speak.dat.Tables["get_by_fam"] != null)
    {
        excel = new
Microsoft.Office.Interop.Excel.Application();
        excel.Visible = true;
        excel.Workbooks.Add(1);

        Microsoft.Office.Interop.Excel.Worksheet
excel_sheet = excel.Worksheets.get_Item(1);
        excel_sheet.get_Range("A2").Value = "№
п/п";
        excel_sheet.get_Range("B2").Value =
"Отдел";
        excel_sheet.get_Range("E2").Value =
"Табельный номер";
        excel_sheet.get_Range("H2").Value =
"Фамилия";
        excel_sheet.get_Range("J2").Value = "Имя";
        excel_sheet.get_Range("L2").Value =
"Отчетство";
        excel_sheet.get_Range("N2").Value =
"Должность";
        for (int i = 0; i < Speak.grid.Rows.Count;
i++)
        {
            excel_sheet.get_Range("A" + (i +
4).ToString()).Value = (i + 1).ToString();
            excel_sheet.get_Range("B" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[0].Value.ToString();
            excel_sheet.get_Range("E" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[1].Value.ToString();
            excel_sheet.get_Range("H" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[2].Value.ToString();
            excel_sheet.get_Range("J" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[3].Value.ToString();
            excel_sheet.get_Range("L" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[4].Value.ToString();
            excel_sheet.get_Range("N" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[5].Value.ToString();

```

```

    }
    }
}
else if (radioButton7.Checked == true)
{
    if (Speak.dat.Tables["get_by_inv"] != null)
    {
        excel = new
Microsoft.Office.Interop.Excel.Application();
        excel.Visible = true;
        excel.Workbooks.Add(1);

        Microsoft.Office.Interop.Excel.Worksheet
excel_sheet = excel.Worksheets.get_Item(1);
        excel_sheet.get_Range("A2").Value = "№
п/п";
        excel_sheet.get_Range("B2").Value = "Тип
устройства";
        excel_sheet.get_Range("E2").Value =
"Производитель";
        excel_sheet.get_Range("G2").Value =
"Ответственный";
        excel_sheet.get_Range("J2").Value =
"Инвентарный номер";
        excel_sheet.get_Range("M2").Value =
"Серийный номер";
        excel_sheet.get_Range("O2").Value =
"Наименование";
        excel_sheet.get_Range("Q2").Value = "Дата
Установки";
        for (int i = 0; i < Speak.grid.Rows.Count;
i++)
        {
            excel_sheet.get_Range("A" + (i +
4).ToString()).Value = (i + 1).ToString();
            excel_sheet.get_Range("B" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[0].Value.ToString();
            excel_sheet.get_Range("E" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[1].Value.ToString();
            excel_sheet.get_Range("G" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[2].Value.ToString();
            excel_sheet.get_Range("J" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[3].Value.ToString();
            excel_sheet.get_Range("M" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[4].Value.ToString();
            excel_sheet.get_Range("O" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[5].Value.ToString();
            excel_sheet.get_Range("Q" + (i +
4).ToString()).Value =
Speak.grid.Rows[i].Cells[6].Value.ToString().Substring(0, 10) +
"Г.";
        }
    }
}

```

```

    }
}
else if (radioButton6.Checked == true)
{
    if (Speak.dat.Tables["get_by_date"] != null)
    {
        excel = new
Microsoft.Office.Interop.Excel.Application();
        excel.Visible = true;
        excel.Workbooks.Add(1);

        Microsoft.Office.Interop.Excel.Worksheet
excel_sheet = excel.Worksheets.get_Item(1);
        excel_sheet.get_Range("A2").Value = "№
п/п";
        excel_sheet.get_Range("B2").Value = "Тип
устройства";
        excel_sheet.get_Range("E2").Value =
"Производитель";
        excel_sheet.get_Range("G2").Value =
"Ответственный";
        excel_sheet.get_Range("J2").Value =
"Инвентарный номер";
        excel_sheet.get_Range("M2").Value =
"Серийный номер";
        excel_sheet.get_Range("O2").Value =
"Наименование";
        excel_sheet.get_Range("Q2").Value = "Дата
Установки";
        for (int i = 0; i < Speak.grid.Rows.Count;
i++)
        {
            excel_sheet.get_Range("A" + (i +
4).ToString()).Value = (i + 1).ToString();
            excel_sheet.get_Range("B" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[0].Value.ToString();
            excel_sheet.get_Range("E" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[1].Value.ToString();
            excel_sheet.get_Range("G" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[2].Value.ToString();
            excel_sheet.get_Range("J" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[3].Value.ToString();
            excel_sheet.get_Range("M" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[4].Value.ToString();
            excel_sheet.get_Range("O" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[5].Value.ToString();
            excel_sheet.get_Range("Q" + (i +
4).ToString()).Value =
Speak.grid.Rows[i].Cells[6].Value.ToString().Substring(0, 10) +
"Г.";
        }
    }
}

```



```

    }
}
else if (radioButton4.Checked == true)
{
    if (Speak.dat.Tables["get_by_par"] != null)
    {
        excel = new
Microsoft.Office.Interop.Excel.Application();
        excel.Visible = true;
        excel.Workbooks.Add(1);

        Microsoft.Office.Interop.Excel.Worksheet
excel_sheet = excel.Worksheets.get_Item(1);
        excel_sheet.get_Range("A2").Value = "№
п/п";
        excel_sheet.get_Range("B2").Value = "Тип
устройства";
        excel_sheet.get_Range("E2").Value =
"Производитель";
        excel_sheet.get_Range("G2").Value =
"Инвентарный номер";
        excel_sheet.get_Range("J2").Value =
"Серийный номер";
        excel_sheet.get_Range("L2").Value =
"Наименование";
        excel_sheet.get_Range("O2").Value =
"Параметр";
        excel_sheet.get_Range("Q2").Value =
"Свойство";
        for (int i = 0; i < Speak.grid.Rows.Count;
i++)
        {
            excel_sheet.get_Range("A" + (i +
4).ToString()).Value = (i + 1).ToString();
            excel_sheet.get_Range("B" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[0].Value.ToString();
            excel_sheet.get_Range("E" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[1].Value.ToString();
            excel_sheet.get_Range("G" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[2].Value.ToString();
            excel_sheet.get_Range("J" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[3].Value.ToString();
            excel_sheet.get_Range("L" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[4].Value.ToString();
            excel_sheet.get_Range("O" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[5].Value.ToString();
            excel_sheet.get_Range("Q" + (i +
4).ToString()).Value = Speak.grid.Rows[i].Cells[6].Value.ToString();
        }
    }
}

```

```

        }
    }
}

```

Листинг «Формы 4» файл «Form4.cs»

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data.SqlClient;
using System.Data;
using System.Data.Sql;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace WindowsFormsApplication5
{
    public partial class Form4 : Form
    {
        Form1 obnov;
        public Form4(Form1 obn)
        {
            InitializeComponent();
            obnov = obn;
            tool = new ToolTip();
            tool.SetToolTip(button1, "Добавить");
            this.Controls.Add(button1);
            tool.SetToolTip(button_del, "Удалить");
            this.Controls.Add(button_del);
            tool.SetToolTip(button3, "Редактировать");
            this.Controls.Add(button3);
            tool.SetToolTip(button2, "Подтвердить");
            this.Controls.Add(button2);
            tool.SetToolTip(button5, "Выйти");
            this.Controls.Add(button5);
            button2.Hide();
            comboBox2.DataSource =
Speak.dat.Tables["Параметры"];
            comboBox2.ValueMember = "Параметр";
            dataGridView1.DataSource =
Speak.dat.Tables["Свойств"];
            dataGridView1.Columns[0].Visible = false;
            dataGridView1.Columns[1].Visible = false;
            comboBox2.SelectedIndex = -1;
        }

        private void button5_Click(object sender, EventArgs e)
        {

```

```

        if (MessageBox.Show("Вы действительно желаете
выйти?", "Выход", MessageBoxButtons.YesNo) == DialogResult.Yes)
            this.Close();
    }
    private void button2_Click(object sender, EventArgs e)
    {
        if (MessageBox.Show("Подтвердите редактирование
записи?", "Редактирование", MessageBoxButtons.YesNo) ==
DialogResult.Yes)
        {
            try
            {
                SqlConnection connection = new
SqlConnection(Speak.connect);
                SqlCommand myCmd = new SqlCommand();
                connection.Open();
                myCmd.Connection = connection;
                if ((textBox1.Text.ToString().Length == 0)
|| (comboBox2.Text.ToString().Length == 0))
                    MessageBox.Show("Введите значение",
"Добавление", MessageBoxButtons.OK, MessageBoxIcon.Error);
                else
                {
                    myCmd.Connection = connection;
                    myCmd.CommandType =
CommandType.StoredProcedure;
                    myCmd.CommandText = "up_record_sv";
                    myCmd.Parameters.AddWithValue("@id",
Speak.idd);
                    myCmd.Parameters.AddWithValue("@id1",
Speak.dat.Tables["Свойств"].Rows[dataGridView1.CurrentRow.Index][1])
;
                    myCmd.Parameters.AddWithValue("@name",
comboBox2.Text);
                    myCmd.Parameters.AddWithValue("@value", textBox1.Text);
                    SqlParameter par = new
SqlParameter("ret", SqlDbType.Int);
                    par.Direction =
ParameterDirection.ReturnValue;
                    myCmd.Parameters.Add(par);
                    myCmd.ExecuteNonQuery();
                    if (Convert.ToInt32(par.Value) == 1)
                        MessageBox.Show("Данная запись уже
существует в БД", "Добавление", MessageBoxButtons.OK,
MessageBoxIcon.Information);
                    else
                        MessageBox.Show("Добавление записи
выполнено.", "Добавление", MessageBoxButtons.OK,
MessageBoxIcon.Information);
                    connection.Close();
                    obnov.button2_Click(sender, e);
                }
            }
        }
    }
}

```



```

        comboBox2.SelectedIndex = -1;
        textBox1.Clear();
    }
    catch
    {
        MessageBox.Show("Удаление записи не было
выполнено.", "Удаление",
        MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    }
}
private void button3_Click(object sender, EventArgs e)
{
    if (dataGridView1.CurrentRow.Index > -1)
    {
        button2.Show();
        button3.Hide();
        button5.Hide();
        button_del.Hide();
        button1.Hide();
        textBox1.Text =
Speak.dat.Tables["Свойств"].Rows[dataGridView1.CurrentRow.Index][3].
ToString();
        comboBox2.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDown;
        comboBox2.Text =
Speak.dat.Tables["Свойств"].Rows[dataGridView1.CurrentRow.Index][2].
ToString();
        comboBox2.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDownList;
    }
}

private void button1_Click(object sender, EventArgs e)
{
    if (MessageBox.Show("Подтвердите добавление новой
записи?", "Добавление", MessageBoxButtons.YesNo) ==
DialogResult.Yes)
    {
        try
        {
            SqlConnection connection = new
SqlConnection(Speak.connect);
            SqlCommand myCmd = new SqlCommand();
            connection.Open();
            myCmd.Connection = connection;

            if ((textBox1.Text.ToString().Length == 0)
|| (comboBox2.Text.ToString().Length == 0))
                MessageBox.Show("Введите значение",
"Добавление", MessageBoxButtons.OK, MessageBoxIcon.Error);

```



```

using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WindowsFormsApplication5
{
    public partial class Form5 : Form
    {
        public Form5()
        {
            InitializeComponent();
            comboBox2.Hide();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            if (radioButton1.Checked == true)
            {
                Speak.flag_print = 1;
                this.Close();
            }
            else if (radioButton2.Checked == true)
            {
                Speak.flag_print = 2;
                Speak.parametr_str = comboBox2.Text;
                this.Close();
            }
            else if (radioButton3.Checked == true)
            {
                Speak.flag_print = 3;
                Speak.parametr_str = comboBox2.Text;
                this.Close();
            }
        }

        private void radioButton2_CheckedChanged(object
sender, EventArgs e)
        {
            comboBox2.Show();
            comboBox2.DataSource =
Speak.dat.Tables["Тип_Устройства"];
            comboBox2.ValueMember = "Тип устройства";
        }

        private void radioButton3_CheckedChanged(object
sender, EventArgs e)
        {

```

```

        comboBox2.Show();
        comboBox2.DataSource = Speak.dat.Tables["Отдел"];
        comboBox2.ValueMember = "Отдел";

    }

    private void radioButton1_CheckedChanged(object
sender, EventArgs e)
    {
        comboBox2.Hide();
    }
}
}

```

Листинг «Формы 6» файл «Form7.cs»

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data.SqlClient;
using System.Data;
using System.Data.Sql;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WindowsFormsApplication5
{
    public partial class Настройка : Form
    {
        Form1 obnov;
        public Настройка(Form1 obn)
        {
            obnov = obn;
            InitializeComponent();
            tool = new ToolTip();
            tool.SetToolTip(button1, "Добавить");
            this.Controls.Add(button1);
            tool.SetToolTip(button_del, "Удалить");
            this.Controls.Add(button_del);
            tool.SetToolTip(button3, "Редактировать");
            this.Controls.Add(button3);
            tool.SetToolTip(button2, "Подтвердить");
            this.Controls.Add(button2);
            tool.SetToolTip(button5, "Выйти");
            this.Controls.Add(button5);
            dataGridView1.DataSource =
Speak.dat.Tables["Отдел"];
            dataGridView2.DataSource =
Speak.dat.Tables["Производители"];
        }
    }
}

```



```

        dataGridView3.DataSource =
Speak.dat.Tables["Тип_Устройства"];
        dataGridView4.DataSource =
Speak.dat.Tables["Параметры"];
        dataGridView1.Columns[0].Visible = false;
        dataGridView2.Columns[0].Visible = false;
        dataGridView3.Columns[0].Visible = false;
        dataGridView4.Columns[0].Visible = false;
        button2.Hide();

    }

    private void button1_Click(object sender, EventArgs e)
    {
        if (MessageBox.Show("Подтвердите добавление новой
записи?", "Добавление", MessageBoxButtons.YesNo) ==
DialogResult.Yes)
        {
            try
            {
                SqlConnection connection = new
SqlConnection(Speak.connect);
                SqlCommand myCmd_1 = new SqlCommand();
                SqlCommand myCmd = new SqlCommand();
                connection.Open();
                myCmd.Connection = connection;
                if (tabControl5.SelectedIndex == 0)
                {
                    if (textBox1.Text.ToString().Length ==
0)
                        MessageBox.Show("Введите
значение", "Добавление", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                    else
                    {
                        myCmd.CommandType =
CommandType.StoredProcedure;
                        myCmd.CommandText =
"add_record_otdel";
                        myCmd.Parameters.AddWithValue("@value",
textBox1.Text.ToString());
                        SqlParameter par = new
SqlParameter("ret", SqlDbType.Int);
                        par.Direction =
ParameterDirection.ReturnValue;
                        myCmd.Parameters.Add(par);
                        myCmd.ExecuteNonQuery();
                        if (Convert.ToInt32(par.Value) ==
1)

```

```

        MessageBox.Show("Данная запись
уже существует в БД", "Добавление", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        else
        MessageBox.Show("Добавление
записи выполнено.", "Добавление", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        connection.Close();
        textBox1.Clear();
        obnov.button2_Click(sender, e);
    }
}
else if (tabControl5.SelectedIndex == 1)
{
    if (textBox2.Text.ToString().Length ==
0)
        MessageBox.Show("Введите
значение", "Добавление", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        else
        {
            myCmd.CommandType =
CommandType.StoredProcedure;
            myCmd.CommandText =
"add_record_proiz";
            myCmd.Parameters.AddWithValue("@value", textBox2.Text.ToString());
            SqlParameter par = new
SqlParameter("ret", SqlDbType.Int);
            par.Direction =
ParameterDirection.ReturnValue;
            myCmd.Parameters.Add(par);
            myCmd.ExecuteNonQuery();
            if (Convert.ToInt32(par.Value) ==
1)
                MessageBox.Show("Данная запись
уже существует в БД", "Добавление", MessageBoxButtons.OK,
MessageBoxIcon.Information);
                else
                MessageBox.Show("Добавление
записи выполнено.", "Добавление", MessageBoxButtons.OK,
MessageBoxIcon.Information);
                connection.Close();
                textBox2.Clear();
                obnov.button2_Click(sender, e);
            }
        }
else if (tabControl5.SelectedIndex == 2)
{
    if (textBox3.Text.ToString().Length ==
0)

```

```

        MessageBox.Show("Введите
значение", "Добавление", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        else
        {
            myCmd.CommandType =
CommandType.StoredProcedure;
            myCmd.CommandText =
"add_record_ty";
myCmd.Parameters.AddWithValue("@value", textBox3.Text.ToString());

            SqlParameter par = new
SqlParameter("ret", SqlDbType.Int);
            par.Direction =
ParameterDirection.ReturnValue;
            myCmd.Parameters.Add(par);
            myCmd.ExecuteNonQuery();
            if (Convert.ToInt32(par.Value) ==
1)
                MessageBox.Show("Данная запись
уже существует в БД", "Добавление", MessageBoxButtons.OK,
MessageBoxIcon.Information);
            else
                MessageBox.Show("Добавление
записи выполнено.", "Добавление", MessageBoxButtons.OK,
MessageBoxIcon.Information);
            connection.Close();
            textBox3.Clear();
            obnov.button2_Click(sender, e);
        }
    }
    else if (tabControl5.SelectedIndex == 3)
    {
        if (textBox4.Text.ToString().Length ==
0)
            MessageBox.Show("Введите
значение", "Добавление", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            else
            {
                myCmd.CommandType =
CommandType.StoredProcedure;
                myCmd.CommandText =
"add_record_par";
myCmd.Parameters.AddWithValue("@value", textBox4.Text.ToString());

                SqlParameter par = new
SqlParameter("ret", SqlDbType.Int);
                par.Direction =
ParameterDirection.ReturnValue;

```

```

        myCmd.Parameters.Add(par);
        myCmd.ExecuteNonQuery();
        if (Convert.ToInt32(par.Value) ==
1)
            MessageBox.Show("Данная запись
уже существует в БД", "Добавление", MessageBoxButtons.OK,
MessageBoxIcon.Information);
            else
                MessageBox.Show("Добавление
записи выполнено.", "Добавление", MessageBoxButtons.OK,
MessageBoxIcon.Information);
                connection.Close();
                textBox4.Clear();
                obnov.button2_Click(sender, e);
            }
        }
    }
    catch
    {
        MessageBox.Show("Добавление записи не было
выполнено. Возможно были введены некорректные данные.",
"Добавление",
        MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    }
}

private void button_del_Click(object sender, EventArgs
e)
{
    if (dataGridView1.CurrentRow.Index > -1)
    {
        try
        {
            if (dataGridView1.CurrentRow.Index > -1)
            {
                if (tabControl5.SelectedIndex == 0)
                {
                    SqlConnection connection = new
SqlConnection(Speak.connect);
                    SqlCommand myCmd = new
SqlCommand();
                    connection.Open();
                    myCmd.Connection = connection;
                    myCmd.CommandType =
CommandType.StoredProcedure;
                    myCmd.CommandText =
"del_record_otdel";
                    myCmd.Parameters.AddWithValue("@id",
Speak.dat.Tables["Отдел"].Rows[dataGridView1.CurrentRow.Index][0]);
                    myCmd.ExecuteNonQuery();
                }
            }
        }
    }
}

```

```

        MessageBox.Show("Удаление записи
выполнено.", "Удаление", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        connection.Close();
        obnov.button2_Click(sender, e);
    }
    else if (tabControl5.SelectedIndex ==
1)
    {
        SqlConnection connection = new
SqlConnection(Speak.connect);
        SqlCommand myCmd = new
SqlCommand();
        connection.Open();
        myCmd.Connection = connection;
        myCmd.CommandType =
CommandType.StoredProcedure;
        myCmd.CommandText =
"del_record_proiz";
        myCmd.Parameters.AddWithValue("@id",
Speak.dat.Tables["Производители"].Rows[dataGridView2.CurrentRow.Inde
x][0]);
        myCmd.ExecuteNonQuery();
        MessageBox.Show("Удаление записи
выполнено.", "Удаление", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        connection.Close();
        obnov.button2_Click(sender, e);
    }
    else if (tabControl5.SelectedIndex ==
2)
    {
        SqlConnection connection = new
SqlConnection(Speak.connect);
        SqlCommand myCmd = new
SqlCommand();
        connection.Open();
        myCmd.Connection = connection;
        myCmd.CommandType =
CommandType.StoredProcedure;
        myCmd.CommandText =
"del_record_ty";
        myCmd.Parameters.AddWithValue("@id",
Speak.dat.Tables["Тип_Устройства"].Rows[dataGridView3.CurrentRow.Inde
x][0]);
        myCmd.ExecuteNonQuery();
        MessageBox.Show("Удаление записи
выполнено.", "Удаление", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        connection.Close();
    }

```

```

        obnov.button2_Click(sender, e);
    }
    else if (tabControl5.SelectedIndex ==
3)
    {
        SqlConnection connection = new
SqlConnection(Speak.connect);
        SqlCommand myCmd = new
SqlCommand();
        connection.Open();
        myCmd.Connection = connection;
        myCmd.CommandType =
CommandType.StoredProcedure;
        myCmd.CommandText =
"del_record_par";
        myCmd.Parameters.AddWithValue("@id",
Speak.dat.Tables["Параметры"].Rows[dataGridView3.CurrentRow.Index][0
]);
        myCmd.ExecuteNonQuery();
        MessageBox.Show("Удаление записи
выполнено.", "Удаление", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        connection.Close();
        obnov.button2_Click(sender, e);
    }
}
catch
{
    MessageBox.Show("Удаление записи не было
выполнено.", "Удаление",
    MessageBoxButtons.OK,
    MessageBoxIcon.Error);
}
}

private void button3_Click(object sender, EventArgs e)
{
    if (dataGridView1.CurrentRow.Index > -1)
    {
        button2.Show();
        button3.Hide();
        button5.Hide();
        button_del.Hide();
        button1.Hide();
        if (tabControl5.SelectedIndex == 0)
            textBox1.Text =
Speak.dat.Tables["Отдел"].Rows[dataGridView1.CurrentRow.Index][1].To
String();
    }
}

```

```

        else if (tabControl5.SelectedIndex == 1)
            textBox2.Text =
Speak.dat.Tables["Производители"].Rows[dataGridView2.CurrentRow.Index][1].ToString();
        else if (tabControl5.SelectedIndex == 2)
            textBox3.Text =
Speak.dat.Tables["Тип_Устройства"].Rows[dataGridView3.CurrentRow.Index][1].ToString();
        else if (tabControl5.SelectedIndex == 3)
            textBox4.Text =
Speak.dat.Tables["Параметры"].Rows[dataGridView4.CurrentRow.Index][1].ToString();
    }
}

private void button2_Click(object sender, EventArgs e)
{
    if (MessageBox.Show("Подтвердите редактирование записи?", "Редактирование", MessageBoxButtons.YesNo) == DialogResult.Yes)
    {
        try
        {
            SqlConnection connection = new
SqlConnection(Speak.connect);
            SqlCommand myCmd_1 = new SqlCommand();
            SqlCommand myCmd = new SqlCommand();
            connection.Open();
            myCmd.Connection = connection;
            if (tabControl5.SelectedIndex == 0)
            {
                if (textBox1.Text.ToString().Length ==
0)
                    MessageBox.Show("Введите значение", "Редактирование", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                else
                {
                    myCmd.CommandType =
CommandType.StoredProcedure;
                    myCmd.CommandText =
"up_record_otdel";

                    myCmd.Parameters.AddWithValue("@id",
Speak.dat.Tables["Отдел"].Rows[dataGridView1.CurrentRow.Index][0]);
                    myCmd.Parameters.AddWithValue("@value", textBox1.Text.ToString());
                    SqlParameter par = new
SqlParameter("ret", SqlDbType.Int);
                    par.Direction =
ParameterDirection.ReturnValue;
                }
            }
        }
    }
}

```

```

myCmd.Parameters.Add(par);
myCmd.ExecuteNonQuery();
if (Convert.ToInt32(par.Value) ==
1)
    MessageBox.Show("Данная запись
уже существует в БД", "Редактирование", MessageBoxButtons.OK,
MessageBoxIcon.Information);
else
    MessageBox.Show("Редактирование записи выполнено.",
"Редактирование", MessageBoxButtons.OK, MessageBoxIcon.Information);
connection.Close();
textBox1.Clear();
obnov.button2_Click(sender, e);
}
}
else if (tabControl5.SelectedIndex == 1)
{
if (textBox2.Text.ToString().Length ==
0)
    MessageBox.Show("Введите
значение", "Редактирование", MessageBoxButtons.OK,
MessageBoxIcon.Error);
else
{
myCmd.CommandType =
CommandType.StoredProcedure;
myCmd.CommandText =
"up_record_proiz";
myCmd.Parameters.AddWithValue("@id",
Speak.dat.Tables["Производители"].Rows[dataGridView2.CurrentRow.Inde
x][0]);
myCmd.Parameters.AddWithValue("@value", textBox2.Text.ToString());
SqlParameter par = new
SqlParameter("ret", SqlDbType.Int);
par.Direction =
ParameterDirection.ReturnValue;
myCmd.Parameters.Add(par);
myCmd.ExecuteNonQuery();
if (Convert.ToInt32(par.Value) ==
1)
    MessageBox.Show("Данная запись
уже существует в БД", "Редактирование", MessageBoxButtons.OK,
MessageBoxIcon.Information);
else
    MessageBox.Show("Редактирование записи выполнено.",
"Редактирование", MessageBoxButtons.OK, MessageBoxIcon.Information);
connection.Close();
textBox2.Clear();

```



```

                                obnov.button2_Click(sender, e);
                                }
                                }
                                else if (tabControl5.SelectedIndex == 2)
                                {
                                    if (textBox3.Text.ToString().Length ==
0)
                                        MessageBox.Show("Введите
значение", "Редактирование", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                                        else
                                        {
                                            myCmd.CommandType =
CommandType.StoredProcedure;
                                            myCmd.CommandText =
"up_record_ty";

myCmd.Parameters.AddWithValue("@id",
Speak.dat.Tables["Тип_Устройства"].Rows[dataGridView3.CurrentRow.Ind
ex][0]);

myCmd.Parameters.AddWithValue("@value", textBox3.Text.ToString());
                                            SqlParameter par = new
SqlParameter("ret", SqlDbType.Int);
                                            par.Direction =
ParameterDirection.ReturnValue;
                                            myCmd.Parameters.Add(par);
                                            myCmd.ExecuteNonQuery();
                                            if (Convert.ToInt32(par.Value) ==
1)
                                                MessageBox.Show("Данная запись
уже существует в БД", "Редактирование", MessageBoxButtons.OK,
MessageBoxIcon.Information);
                                                else

MessageBox.Show("Редактирование записи выполнено.",
"Редактирование", MessageBoxButtons.OK, MessageBoxIcon.Information);
                                                connection.Close();
                                                textBox3.Clear();
                                                obnov.button2_Click(sender, e);
                                            }
                                        }
                                else if (tabControl5.SelectedIndex == 3)
                                {
                                    if (textBox4.Text.ToString().Length ==
0)
                                        MessageBox.Show("Введите
значение", "Редактирование", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                                        else
                                        {

```

```

myCmd.CommandType =
CommandType.StoredProcedure;
myCmd.CommandText =
"up_record_par";

myCmd.Parameters.AddWithValue("@id",
Speak.dat.Tables["Параметры"].Rows[dataGridView4.CurrentRow.Index][0
]);

myCmd.Parameters.AddWithValue("@value", textBox4.Text.ToString());
SqlParameter par = new
SqlParameter("ret", SqlDbType.Int);
par.Direction =
ParameterDirection.ReturnValue;
myCmd.Parameters.Add(par);
myCmd.ExecuteNonQuery();
if (Convert.ToInt32(par.Value) ==
1)
    MessageBox.Show("Данная запись
уже существует в БД", "Редактирование", MessageBoxButtons.OK,
MessageBoxIcon.Information);
else
    MessageBox.Show("Редактирование записи выполнено.",
"Редактирование", MessageBoxButtons.OK, MessageBoxIcon.Information);
connection.Close();
textBox4.Clear();
obnov.button2_Click(sender, e);
}
}
catch
{
    MessageBox.Show("Редактирование записи не
было выполнено. Возможно были введены некорректные данные.",
"Редактирование",
    MessageBoxButtons.OK,
    MessageBoxIcon.Error);
}
button2.Hide();
button3.Show();
button5.Show();
button_del.Show();
button1.Show();
}
private void button5_Click(object sender, EventArgs e)
{
    if (MessageBox.Show("Вы действительно желаете
выйти?", "Выход", MessageBoxButtons.YesNo) == DialogResult.Yes)
        this.Close();
}

```

```
private void textBox1_KeyPress(object sender,
KeyPressEventArgs e)
{
}

private void textBox2_KeyPress(object sender,
KeyPressEventArgs e)
{
}

private void textBox3_KeyPress(object sender,
KeyPressEventArgs e)
{
}
}
}
```