МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет	*Кафедра*
вычислительной техники	«Вычислительная техника»

Специальность: 09.05.01 «Применение и эксплуатация автоматизированных систем специального назначения»

Специализация: «Эксплуатация вычислительных машин, комплексов, систем и сетей специального назначения»

ДИПЛОМНЫЙ ПРОЕКТ

на тему

АВТОМАТИЗИРОВАННАЯ СИСТЕМА УЧЕТА РЕМОНТА И ХРАНЕНИЯ КОМПЬЮТЕРНОЙ ТЕХНИКИ В ВОИНСКОЙ ЧАСТИ

Студент _____ 10.06.2025 Аширбакиева Р.М.
(подпись, дата)

Руководитель _____ 23.06.2025 Зинкин С.А.
(подпись, дата)

Нормоконтролёр _____ 23.06.2025 А.С. Бычков
(подпись, дата)

Работа допущена к защите (протокол заседания кафедры от 23.06.2025 № 16)

Заведующий кафедрой _____ М.А. Митрохин
(подпись)

Рецензент главный специалист АО «НПП «Рубин» _____ А.В. Васильков

Работа защищена с отметкой удовлетворительно (протокол заседания ГЭК от 23.06.25 № 11)

Секретарь ГЭК _____ Е.И. Гурин
(подпись)

Пенза 2025

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

«Утверждаю»
Заведующий кафедрой
_____М.А. Митрохин
«06» 05 2025г.

ЗАДАНИЕ
НА ДИПЛОМНЫЙ ПРОЕКТ

1. Студент Аширбакиева Рузиля Миниуровна гр.20ВОЭ1 факультета вычислительной техники.

2.Тема работы: «Информационная автоматизированная система учета ремонта и хранения компьютерной техники в воинской части»

Тема утверждена приказом ПГУ № 601/о от "06 " мая 2025 г.

3. Руководитель бакалаврской работы д.т.н., профессор, каф. ВТ ПГУ Зинкин С.А.

4. Задание на работу (назначение разработки, исходные данные и т.п.)

Разработать приложение «Информационная автоматизированная система учета ремонта и хранения компьютерной техники в воинской части», позволяющее автоматизировать ручной процесс ремонта и хранения компьютерной техники в воинской части. Приложение должно обладать графическим и интуитивно понятным интерфейсом, для взаимодействия с пользователем. Система должна предоставлять возможность для взаимодействия с информацией о сотрудниках, подразделениях, компьютерной техники, ремонта компьютерной техники, комплектующих. Разрабатываемое приложение также должно выполнять задачи по формированию электронных таблиц.

5. Перечень подлежащих разработке вопросов

Требуется произвести анализ предметных областей: изучить существующие автоматизированные системы, выделить их достоинства и недостатки. Выбрать язык программирования, который подойдет для создания графического интерфейса. Определить архитектуру приложения и вспомогательные средства разработки. Изучить функционал библиотек для взаимодействия с файлами формата *xlsx*.

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1	Изучение предметной области	10	06.05.2025 – 09.05.2025	
2	Анализ требований	10	10.05.2025 – 13.05.2025	
3	Создание макета графического интерфейса	20	14.05.2025 – 18.05.2025	
4	Выбор технологий, применяемых при проектировании и разработке	10	19.05.2025 – 22.05.2025	
5	Реализация графического интерфейса	10	23.05.2025 – 27.05.2025	
6	Разработка основной части программы	10	28.05.2025	
7	Оформление пояснительной записки	30	29.05.2025 10.06.2025	

Дата выдачи задания "6" 05 2025 г.

Руководитель дипломного проекта 06.05.2025 Зинкин С.А.
(подпись, дата)

Задание к исполнению принял студент 06.05.2025 Аширбакиева Р.М.
(подпись, дата)

Работу к защите допустить
Декан факультета 23.06.2025 Фионова Л.Р.
(подпись, дата)

Реферат

Дипломный проект содержит 107 л., 48 рис., 12 табл., 11 источн. и 3 прил.

АВТОМАТИЗИРОВАННАЯ СИСТЕМА, ОБРАБОТКА СОБЫТИЙ, ЛОГИРОВАНИЕ, БАЗА ДАННЫХ, ПАРАЛЛЕЛЬНОЕ ПРОГРАММИРОВАНИЕ, C#, MYSQL

Цель работы – разработка информационной автоматизированной системы учета ремонта и хранения компьютерной техники в воинской части с целью упрощения процесса выполнения повседневных и управленческих задач.

Для достижения поставленной цели было разработано настольное приложение, отвечающее следующим поставленным задачам: работа в системе под учетной записью сотрудника, добавление, изменение и удаление записей о сотрудниках и подразделениях воинской части, о компьютерной техники воинской части и комплектующих для нее, которые в свою очередь хранятся на складе, о всех проведенных ремонтах различной компьютерной техники воинской части.

В результате проведенной работы разработана автоматизированная система с графическим интерфейсом на C# и базой данных на СУБД MySQL, что обеспечивает учет и контроль ремонтных работ и хранение компьютерной техники в воинской части.

Содержание

Перечень сокращений и обозначений 5
Введение 6
1 Информационная автоматизированная система учета ремонта и хранения компьютерной техники воинской части …… …….....7
1.1 Анализ предметной области 8
1.1.1 Актуальность разработки 8
1.1.2 Анализ существующих аналогов 9
1.1.3 Обоснование актуальности разработки программы 13
1.1.4 Постановка задачи 16
1.2 Выбор технологий и средств разработки приложения 17
1.2.1 Выбор языка программирования 18
1.2.2 Выбор СУБД 22
1.2.3 Графический клиент DbGate 23
1.2.4 Выбор среды разработки 24
1.2.5 Выбор графической библиотеки 24
1.3 Проектирование информационной системы 24
1.3.1 Проектирование приложения 18
1.3.2 Проектирование базы данных 22
1.3.2.1 Логическая модель 22
1.3.2.2 Физическая модель 22
1.3.2.3 Описание архитектуры 22
1.4 Разработка приложения 25
1.4.1 Реализация БД 38
1.4.2 Описание программы 39
1.4.3 Тестирование 40
2 Охрана труда 61
Заключение 50
Список используемых источников 51

Приложение А – UML – диграммы 67
Приложение Б – Листинг программы 58
Приложение В – Презентация 106

Перечень сокращений и обозначений

АС – автоматизированная система

БД – база данных

Десктопное приложение - программа, предназначенная для установки и запуска на персональном компьютере или ноутбуке

СУБД – система управления базами данных

FK (Foreign key) – внешний ключ, понятие теории реляционных баз данных, относящееся к ограничениям целостности базы данных

GUI (Graphical User Interface) – графический интерфейс пользователя

ID (Identifier) – уникальный номер объекта

IDE (Integrated development environment) – интегрированная среда разработки

PK (Primary key) – первичный ключ, понятие теории реляционных баз данных, относящееся к ограничениям целостности базы данных

SQL (Structured Query Language) – язык структурированных запросов

IT (информационные технологии) — это область, связанная с созданием, хранением, обработкой, защитой и передачей информации с помощью компьютеров и программного обеспечения.

Введение

В двадцать первом веке информационные технологии являются важной частью повседневной жизни человека. Они активно применяются в различных сферах деятельности и позволяют существенно ускорить выполнение задач, снизить количество ошибок и оптимизировать затраты ресурсов. Постоянное развитие и совершенствование технологий обеспечивает их гибкость и способность удовлетворять новые потребности пользователей и организаций.

В сфере работы воинских частей и других военных подразделений надежность и исправность компьютерной техники играют ключевую роль, так как от этого напрямую зависит эффективность выполнения служебных и административных обязанностей. Для организации хранения, учета и своевременного ремонта компьютерного оборудования необходимо использование современных методов автоматизации.

Одним из таких методов является внедрение информационной автоматизированной системы, которая позволяет систематизировать данные о компьютерной технике, комплектующих, сотрудниках, ответственностях и проведенных ремонтных работах. Это помогает не только поддерживать технику в исправном состоянии, но и планировать закупки, замену оборудования и своевременное обслуживание.

На сегодняшний день во многих воинских частях учет компьютерной техники и ремонтных работ по-прежнему ведется вручную или в разрозненных документах. Такой подход нередко приводит к потере данных, несвоевременному ремонту и увеличению расходов на содержание техники. Поэтому актуальной задачей является разработка автоматизированной системы, позволяющей организовать учет и контроль всего жизненного цикла компьютерной техники — от хранения и эксплуатации до ремонта и списания.

Выбор темы «Информационная автоматизированная система учета ремонта и хранения компьютерной техники в воинской части» обусловлен высокой актуальностью данного вопроса и необходимостью повышения эффективности работы подразделений за счет применения современных IT-решений.

Целью выпускной квалификационной работы является:

– исследование предметной области и анализ текущего состояния учета техники и ремонтов в воинской части;

– проектирование базы данных с учетом специфики воинских подразделений;

– разработка настольного приложения для удобного взаимодействия с базой данных;

– создание и настройка пользовательского интерфейса;

– тестирование системы на реальных сценариях учета;

– подготовка рекомендаций по внедрению и эксплуатации системы.

Разработка и внедрение данной системы позволит воинским частям значительно повысить контроль за техническим состоянием оборудования, сократить время и затраты на ремонтные работы, а также повысить общую надежность IT-инфраструктуры военного подразделения.

1 Информационная система учета ремонта и хранения компьютерной техники в воинской части

1.1 Анализ предметной области

1.1.1 Актуальность разработки

Известно, что эффективная работа любого подразделения, в том числе воинской части, напрямую зависит от исправности и надежного функционирования всей используемой техники и оборудования. В современных условиях компьютерная техника играет ключевую роль в выполнении оперативных, управленческих и административных задач воинской части. От бесперебойной работы компьютеров, серверов и периферийного оборудования зависит своевременность обработки информации, хранение важных данных и поддержание связей между подразделениями.

На практике учет компьютерной техники и контроль за её состоянием часто ведется вручную или с помощью разрозненных таблиц и бумажных журналов. Такой подход не только отнимает много времени у ответственных сотрудников, но и повышает риск возникновения ошибок при передаче или обновлении данных. Зачастую информация о ремонтах и замене комплектующих может теряться или обновляться несвоевременно, что приводит к несвоевременной починке и простоям оборудования.

Для решения данной проблемы разрабатываются информационные автоматизированные системы, которые берут на себя выполнение рутинных задач по хранению и обработке данных о техническом состоянии оборудования, ремонтах и истории обслуживания. С помощью такой системы можно в любой момент получить актуальную информацию о наличии и состоянии компьютерной техники, контролировать график технического обслуживания, своевременно заказывать запасные части и комплектующие, а также планировать бюджет на ремонт и обновление оборудования.

Автоматизированная система учета ремонта и хранения компьютерной техники в воинской части позволяет снизить нагрузку на персонал, сократить время на подготовку отчетности, минимизировать риск потери данных и повысить уровень ответственности за сохранность техники. Немаловажно и то, что использование такой системы способствует рациональному использованию материальных ресурсов и снижению расходов на содержание оборудования.

В то время как крупные военные подразделения могут позволить себе использование дорогостоящих комплексных решений, в небольших воинских частях и отдельных подразделениях часто отсутствует техническая возможность или финансовые ресурсы для внедрения подобных систем. По этой причине учет техники и ремонтные работы продолжают вестись традиционными методами, что не соответствует современным требованиям по эффективности и надежности хранения данных.

Решение, предложенное в данной выпускной квалификационной работе, направлено на устранение указанных проблем. Разработанная система имеет невысокую стоимость разработки и сопровождения, гибко настраивается под потребности конкретного подразделения, не требует высоких затрат на внедрение и обучение персонала. Все данные хранятся локально в базе данных MySQL, а работа с ними осуществляется через удобное настольное приложение на языке C#, что позволяет использовать систему даже при отсутствии подключения к сети Интернет.

Таким образом, создание и внедрение автоматизированной системы учета ремонта и хранения компьютерной техники в воинской части является актуальным и востребованным решением, которое позволит повысить дисциплину учета, сократить время на подготовку отчетности и повысить надежность работы IT-инфраструктуры воинской части в целом.

1.1.2 Анализ существующих аналогов

Аналогов системы, разработанных специально именно для учета ремонта и хранения компьютерной техники конкретно в воинской части, как готового тиражного продукта нет в общем доступе.

Это можно объяснить тем, что для военных структур часто разрабатываются внутренние учетные системы, которые являются закрытыми и не публикуются в открытом доступе по соображениям безопасности.

Рассмотрим возможности АС «Склад компьютерной техники».

Общий вид данной программы представлен на рисунке 1.

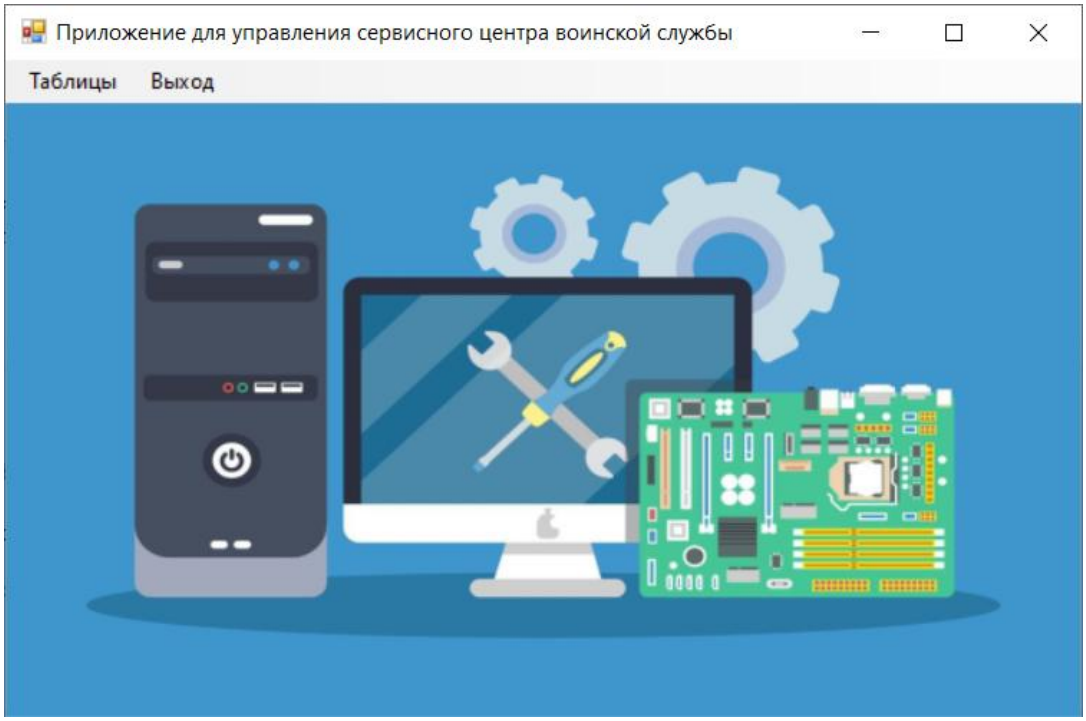


Рисунок 1 – Главный вид программы «Склад компьютерной техники»

Чтобы начать работу с программой, необходимо совершить вход (рисунок 2).

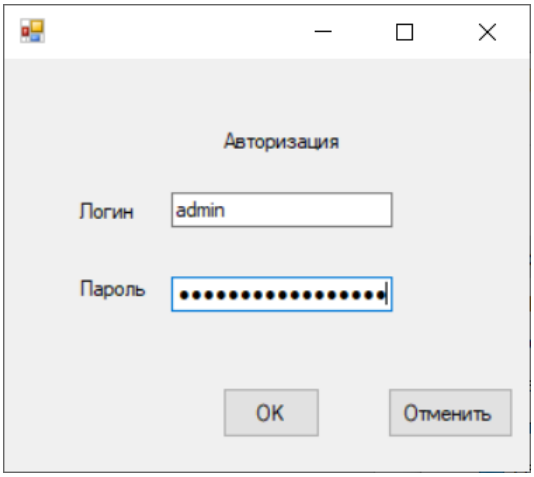


Рисунок 2 – Авторизация в системе

Весь функционал программы представлен в виде нескольких вкладок, каждая из которых выполняет определенные задачи.

На вкладке «Таблицы» доступна вся информация о таблицах склада компьютерной техники в воинской части (рисунок 3).

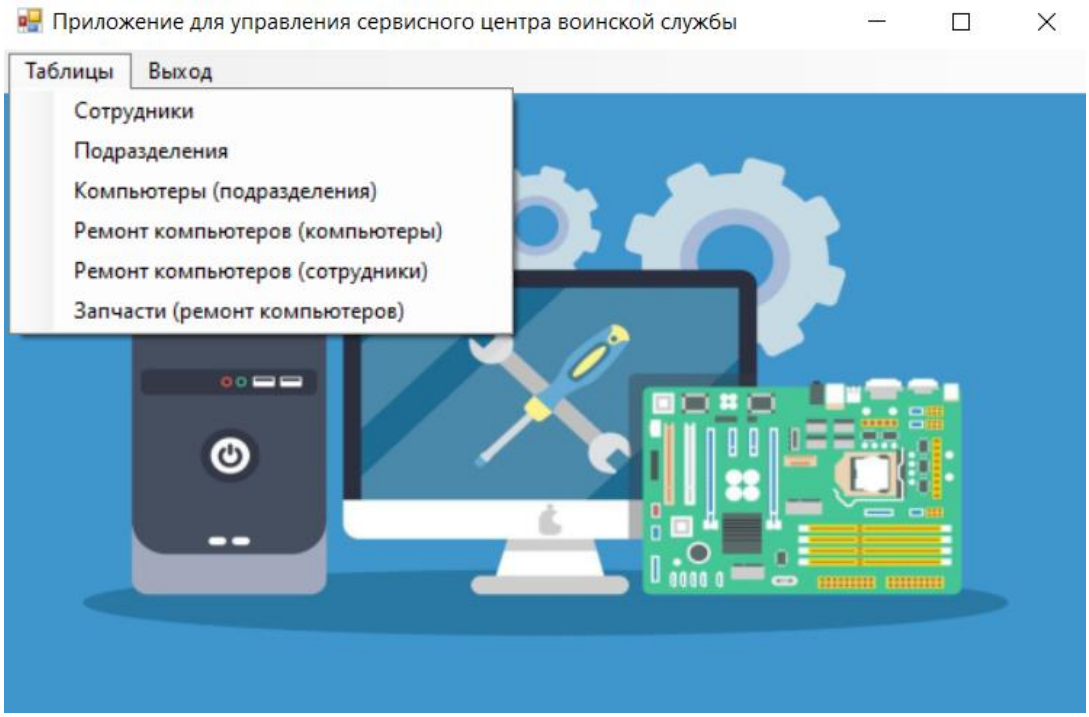


Рисунок 3 – Вкладка «Таблицы»

Для того чтобы посмотреть информацию об определенной сущности, необходимо один раз кликнуть на нужную подвкладку, после чего откроется форма с таблицей или таблицами базы данных учета ремонта и хранения компьютерной техники в воинской части для просмотра, редактирования или удаления. Эти функции осуществляются при помощи нажатия на кнопки «Добавить», «Изменить», «Удалить» и «Вывести изменения». Все формы для отображения данных конкретных таблиц базы данных представлены на рисунках 4-7.

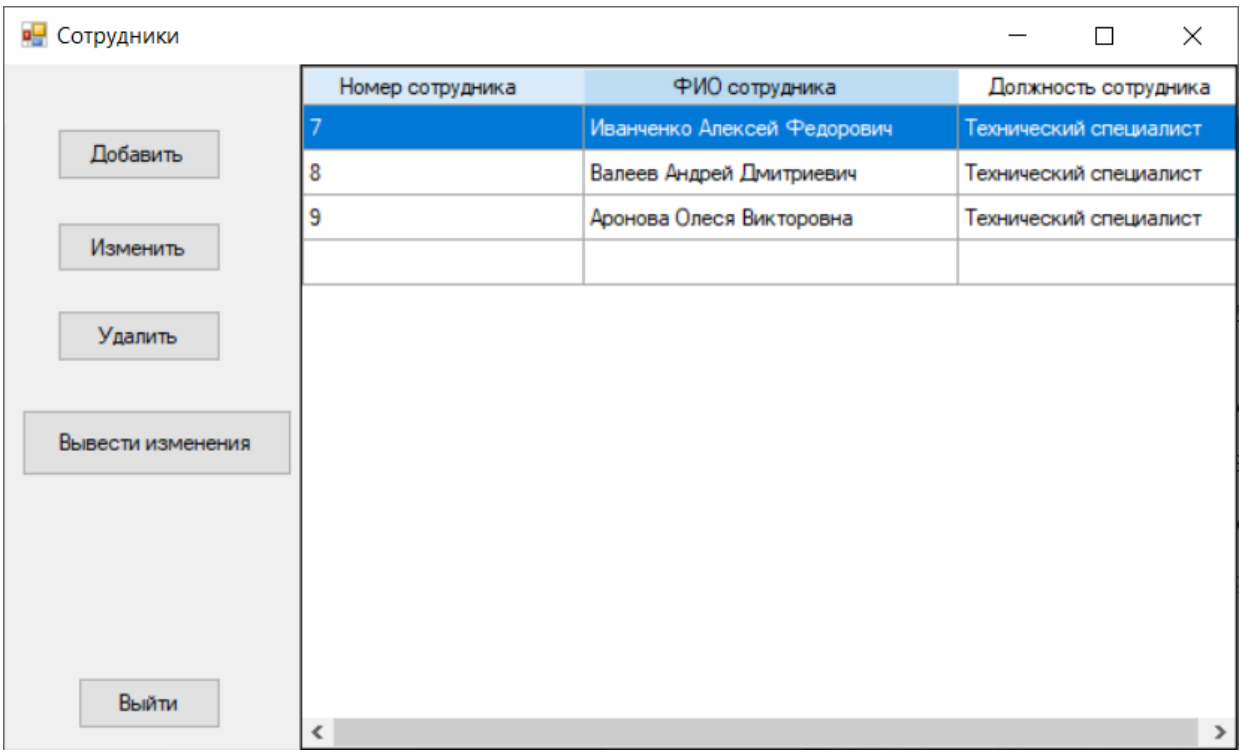


Рисунок 4 – Таблица «Сотрудники»

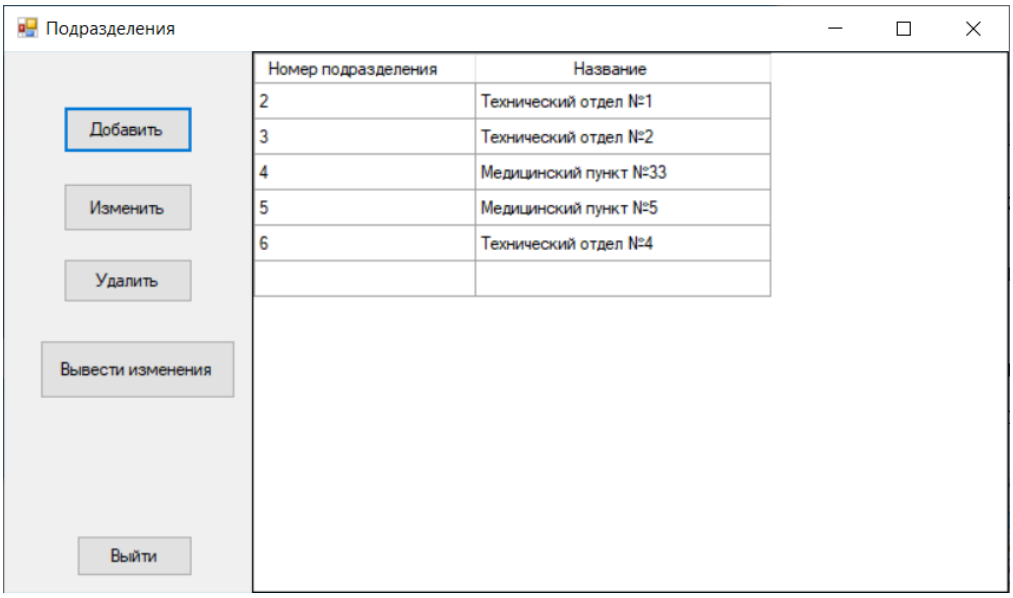


Рисунок 5 – Таблица «Подразделения»

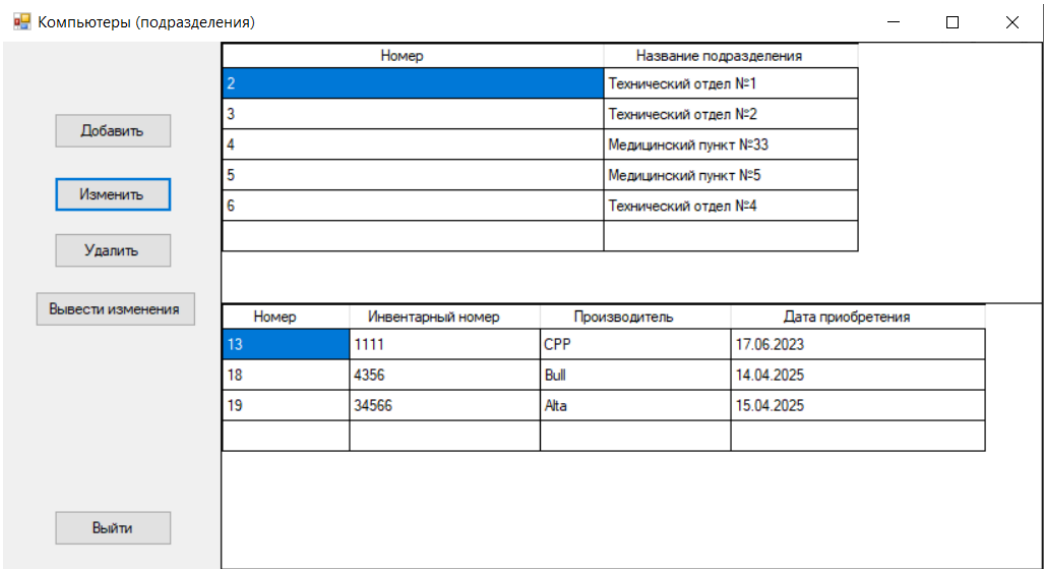


Рисунок 6 – Таблицы «Подразделения» и «Компьютеры»

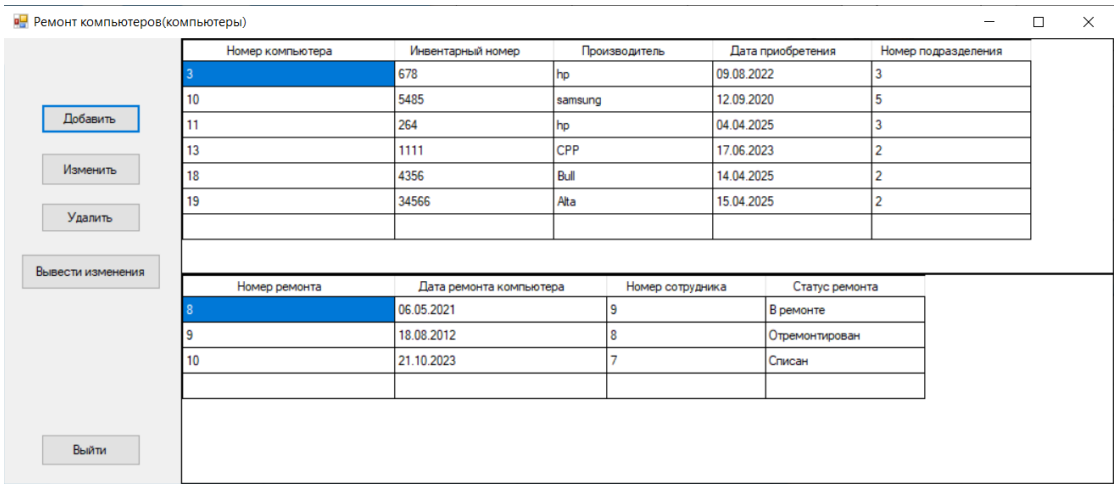


Рисунок 7 – Таблицы «Компьютеры» и «Ремонт компьютеров»

Программа обладает следующими преимуществами:

– программа обладает простой установкой и не требует много ресурсов системы;

– эта система хорошо подходит для склада воинской части, так как имеет весь необходимый функционал;

– присутствует возможность формирования заявок на списание компьютерной техники и комплектующих.

Из минусов данной автоматизированной системы можно выделить следующее:

– приложение является однопользовательским, из-за чего функционал не разграничен;

– устаревший интерфейс приложения;

– программа распространяется на коммерческой основе;

– отсутствует ведение истории действий сотрудников.

1.1.3 Обоснование актуальности разработки программы

Работа является актуальной, поскольку в современных условиях стремительного развития IT-инфраструктуры в военных учреждениях особое значение приобретает организация эффективного учёта, обслуживания и своевременного ремонта компьютерной техники. Существующие решения либо универсальны и не адаптированы под специфику воинских подразделений, либо требуют значительных затрат на внедрение. В этом контексте представленная система является востребованной, практически ориентированной и своевременной.

1.1.4 Постановка задачи

Требуется разработать десктопное приложение, обеспечивающее автоматизацию процессов учёта, обслуживания и хранения компьютерной техники, эксплуатируемой в рамках воинской части. Реализуемая информационная система должна удовлетворять следующим функциональным требованиям:

– регистрация новой единицы техники (внесение сведений о названии, серийном номере, дате приобретения, производителе);

– редактирование сведений об уже зарегистрированной технике;

– удаление техники при её списании;

– ведение справочника подразделений (отделов, рот, служб);

– распределение техники по подразделениям;

– оформление заявки на ремонт с указанием даты, подразделения и ответственного лица;

– ведение журнала выполненных ремонтов с фиксацией даты, исполнителя, использованных материалов и результата;

– возможность прикрепления ремонтной ведомости или акта технического состояния.

Реализация данной системы позволит значительно повысить прозрачность и управляемость процессов обслуживания компьютерной техники и обеспечить точное представление командованию о текущем состоянии компьютерной инфраструктуры воинской части.

На рисунке 8 изображена диаграмма вариантов использования приложения.

Разрабатываемая автоматизированная система должна соответствовать следующим техническим требованиям:

– удобный и интуитивно понятный интерфейс для работы с программой;

– расположение графических элементов должно предоставлять пользователю быстрое ориентирование в интерфейсе;

– разработанная программа является десктопным решением;

– наличие возможности сохранения данных локально, пока отсутствует подключения к серверу;

– операционная система: Windows 7 и более поздние версии;

– оперативная память: не менее 512 МБ ОЗУ;

– незанятное дисковое пространство: не менее 1,5 ГБ ПЗУ;

– процессор: Intel Pentium III 800 MHz;

– разрешение экрана: 1024 x 728.

На рисунке 8 изображена диаграмма вариантов использования приложения.

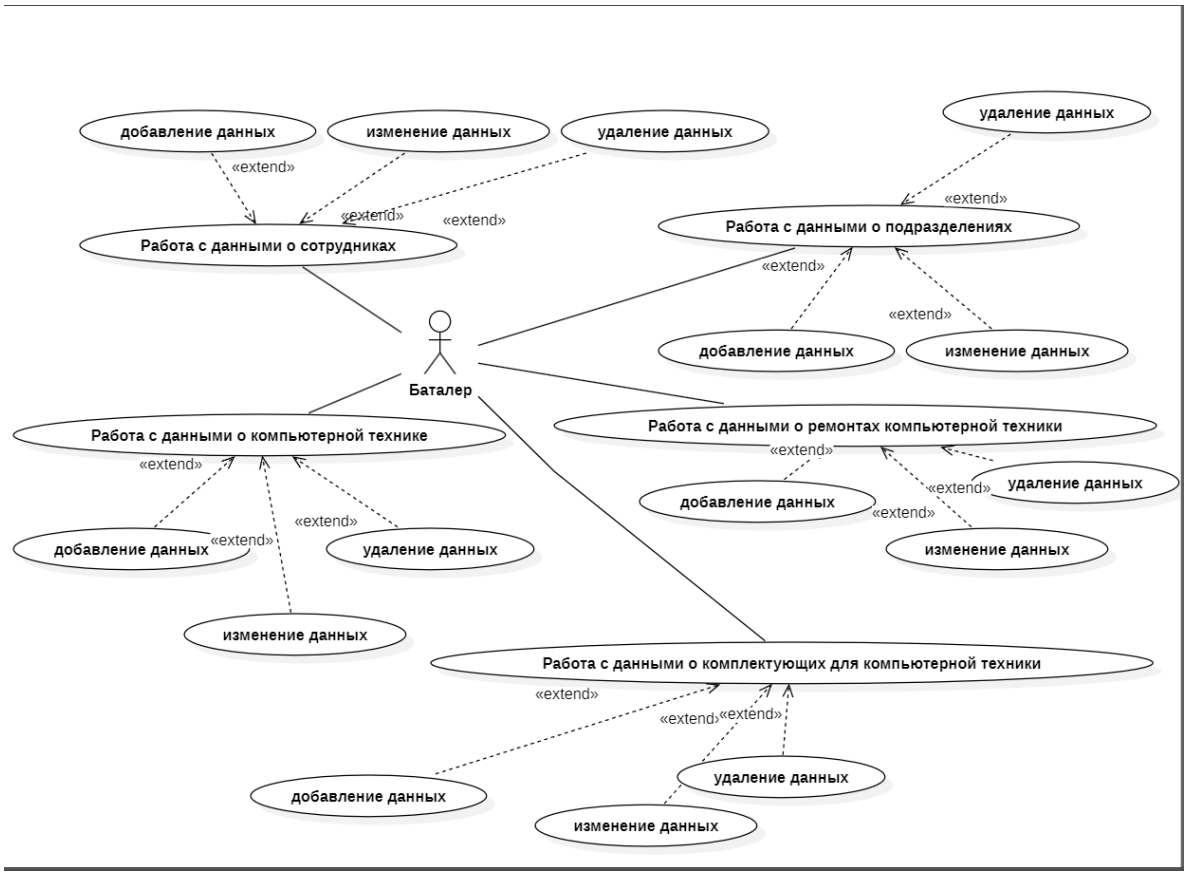


Рисунок 8 – Диаграмма вариантов использования

1.2 Выбор технологий и средств разработки приложения

В данном разделе будут описаны технологии и средства разработки, использованные при разработке программы по автоматизации процессов учёта, обслуживания и хранения компьютерной техники, эксплуатируемой в рамках воинской части.

1.2.1 Выбор языка программирования

Язык программирования C++.

C++ — это мощный компилируемый язык программирования общего назначения, разработанный как расширение языка C, с акцентом на объектно-ориентированный подход и высокую производительность. Он сочетает в себе низкоуровневый контроль над ресурсами и высокоуровневые конструкции, что делает его универсальным инструментом как для системного программирования, так и для разработки прикладного программного обеспечения.

Особенность C++ заключается в его способности предоставлять прямой доступ к памяти, управлять ресурсами вручную, а также использовать объектно-ориентированные парадигмы (инкапсуляция, наследование, полиморфизм), что позволяет создавать масштабируемые и структурированные проекты. Эти качества делают язык востребованным в таких областях, как разработка операционных систем, драйверов, игровых движков и сложных программных платформ.

Но минусами являются отсутствие встроенной поддержки GUI и проблемы с кроссплатформенностью.

Язык программирования Python.

Python — это высокоуровневый язык программирования общего назначения, ориентированный на читаемость кода и простоту синтаксиса. Язык активно используется в самых разных сферах — от веб-разработки и анализа данных до машинного обучения и автоматизации. Его популярность во многом обусловлена огромным числом готовых библиотек и активным сообществом, которое постоянно расширяет функциональные возможности Python.

Python предоставляет целый ряд инструментов. Среди наиболее известных — Tkinter (входит в стандартную библиотеку), PyQt (Python-обёртка над фреймворком Qt), wxPython, а также Kivy, ориентированный на мульти-тач интерфейсы и кроссплатформенность. Эти фреймворки позволяют создавать окна, кнопки, текстовые поля, выпадающие списки и другие элементы интерфейса без глубокого погружения в низкоуровневое программирование.

К сожалению, у Python низкая производительность и отсутствует строгая типизация.

Язык программирования Java.

Java — это объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (позже приобретённой Oracle) в середине 1990-х годов, как средство создания переносимого, надёжного и безопасного программного обеспечения. Он был задуман с учётом принципа «напиши один раз — запускай везде», что означает возможность выполнения одного и того же программного кода на различных операционных системах без необходимости модификации.

Эта переносимость достигается за счёт использования виртуальной машины Java (JVM), которая интерпретирует байт-код, полученный после компиляции исходного кода, и обеспечивает его выполнение независимо от платформы.

Для создания графического интерфейса в Java применяется библиотека Swing, предоставляющая широкие возможности для создания различных элементов интерфейса.

Swing предоставляет широкий спектр компонентов — от простых кнопок и текстовых полей до более сложных элементов, таких как таблицы, деревья или вкладки. Несмотря на свою зрелость, Swing сохраняет актуальность благодаря гибкости и возможности создания кроссплатформенных интерфейсов. Более современным и активно развиваемым решением является JavaFX — набор инструментов и API, ориентированных на создание более визуально насыщенных и интерактивных интерфейсов с поддержкой CSS-стилей, анимации и мультимедийных элементов.

Кроме графических возможностей, Java предлагает широкий набор стандартных библиотек для работы с файлами, сетями, базами данных и другими системными ресурсами. Благодаря этому, создание полноценных десктопных приложений — будь то текстовый редактор, калькулятор, органайзер или даже полноценная бизнес-система — становится возможным без необходимости прибегать к сторонним языкам или технологиям. Недостатком является объемный и «шумный» синтаксис, это увеличивает объем работы и замедляет разработку.

Язык программирования C#.

C# — это современный объектно-ориентированный язык программирования, разработанный компанией Microsoft в начале 2000-х годов как часть платформы .NET. Он был создан с целью объединить в себе лучшие идеи предыдущих языков — таких как C++, Java и Delphi — при этом сделать синтаксис более понятным, строгим и удобным для промышленной разработки.

C# отличается высокой степенью интеграции с экосистемой Windows и активно используется для создания приложений различной сложности — от простых утилит до масштабных корпоративных решений.

Одной из ключевых областей применения языка является разработка настольных приложений с графическим интерфейсом пользователя. C# предлагает несколько подходов к созданию таких программ, включая использование библиотек Windows Forms и Windows Presentation Foundation (WPF). Windows Forms — это зрелая и простая в освоении технология, позволяющая создавать формы, окна, кнопки, поля ввода и другие элементы GUI с минимальными усилиями. Она подходит для быстрого прототипирования и построения классических десктопных решений, характерных для операционной системы Windows. С другой стороны, WPF предоставляет более гибкие и современные инструменты для построения интерфейсов, включая поддержку XAML-разметки, привязку данных, анимацию и шаблоны визуализации, что делает её более предпочтительной для проектов с высоким уровнем визуальной сложности.

После сравнения возможностей одних из популярных языков для разработки приложения был выбран C#, так как Использование таких технологий, как Windows Forms и WPF, позволяет быстро разрабатывать функциональные и стабильные приложения с визуальной составляющей.

1.2.2 Выбор СУБД

СУБД PostgreSQL.

PostgreSQL — это объектно-реляционная система управления базами данных с открытым исходным кодом, известная своей надёжностью и соответствием стандартам SQL.

Достоинства: поддержка сложных запросов и транзакций, расширяемость, высокая надёжность, широкие возможности по настройке, активное сообщество разработчиков.

У PostgreSQL более сложная настройка по сравнению с другими СУБД, может требовать больше ресурсов при высокой нагрузке, не всегда оптимальна для простых или небольших проектов.

СУБД MySQL.

MySQL была выбрана в качестве СУБД для приложения. MySQL — это популярная реляционная СУБД, ориентированная на высокую скорость обработки данных и широкую совместимость с веб-технологиями.

Достоинства: высокая производительность, лёгкость в освоении, поддержка репликации, масштабируемость, широкая интеграция с PHP и другими языками.

СУБД SQLite.

SQLite — это встроенная легковесная СУБД, работающая без отдельного сервера и подходящая для небольших приложений и мобильных устройств.

Достоинства: минимальный размер, не требует установки, простота использования, высокая скорость при работе с локальными данными, кроссплатформенность.

Недостатками данной СУБД являются: ограниченная поддержка расширенных SQL-возможностей, проблемы с точностью при работе с транзакциями (в старых версиях).

1.2.3 Графический клиент DbGate

DbGate — это современный, кроссплатформенный инструмент для работы с базами данных, разработанный с открытым исходным кодом. Он представляет собой лёгкий, визуальный редактор баз данных, предназначенный для администрирования, редактирования и выполнения SQL-запросов к различным СУБД.

DbGate поддерживает популярные системы управления базами данных, включая PostgreSQL, MySQL, MariaDB, SQLite, Microsoft SQL Server, а также MongoDB — что делает его универсальным решением для разработчиков, аналитиков и администраторов. Интерфейс программы позволяет легко подключаться к базам, просматривать таблицы, редактировать записи, создавать схемы, выполнять экспорт и импорт данных.

1.2.4 Выбор среды разработки

Среда разработки Microsoft Visual Studio.

Microsoft Visual Studio — это полнофункциональная интегрированная среда разработки, предназначенная для создания приложений на C# и других языках, входящих в экосистему .NET.

Достоинства: мощный инструментарий для отладки, визуальное проектирование интерфейса, встроенный конструктор форм, поддержка различных шаблонов проектов, удобная интеграция с базами данных, система контроля версий и поддержка плагинов.

Среда разработки Visual Studio Code.

Visual Studio Code — это лёгкий и расширяемый текстовый редактор с поддержкой множества языков программирования, включая C# при наличии соответствующих расширений.

Достоинства: кроссплатформенность, высокая скорость работы, гибкость настройки, широкий выбор расширений, встроенный терминал, интеграция с Git и поддержка IntelliSense.

1.2.5 Выбор графической библиотеки

Графическая библиотека Windows Forms.

Windows Forms (WinForms) — это классическая библиотека для построения графических пользовательских интерфейсов в приложениях на C#, включённая в состав .NET Framework и .NET Core.

Достоинства: простота освоения, визуальный редактор в Visual Studio, высокая скорость разработки, хорошая интеграция с Windows, подходит для небольших и средних приложений.

Графическая библиотека WPF.

WPF (Windows Presentation Foundation) — это более современная библиотека для создания графического интерфейса с использованием декларативной разметки XAML и гибкой архитектуры.

Достоинства: поддержка привязки данных, шаблонов, анимаций и стилей, удобное разделение логики и интерфейса (MVVM), расширенные возможности дизайна, лучше подходит для сложных и визуально насыщенных приложений.

Выбор был сделан в пользу Windows Forms, так как эта графическая библиотека имеет Быстрая и простая разработка интерфейса, низкий порог вхождения, стабильная работа на всех версиях Windows, подходит для простых и средних проектов, удобный визуальный редактор в Visual Studio, не требует изучения XAML и паттерна MVVM.

1.3 Проектирование информационной системы

1.3.1 Архитектура приложения

Одноарендная (Single-Tenant) архитектура представляет собой модель, при которой каждому клиенту (арендатору) выделяется отдельный экземпляр приложения и соответствующая инфраструктура.

Основные характеристики одноарендной архитектуры включают:

- Изоляция данных. Все данные, настройки и конфигурации каждого арендатора полностью отделены друг от друга.

- Выделенные ресурсы. Это обеспечивает стабильную производительность и высокую доступность, поскольку работа одного арендатора не влияет на ресурсы других.

- Гибкость в настройке. Архитектура позволяет легко адаптировать систему под индивидуальные потребности конкретного клиента.

- Повышенная безопасность. Благодаря изоляции данных обеспечивается надежная защита информации каждого арендатора от доступа со стороны других.

Одноарендная архитектура преимущественно используется в проектах с высокими требованиями к безопасности, кастомизации и соответствию нормативным требованиям.

К основным недостаткам данного подхода относится более высокая стоимость эксплуатации, обусловленная необходимостью отдельного масштабирования инфраструктуры для каждого арендатора.

В связи с этим, одноарендные решения чаще всего выбирают крупные компании, финансовые учреждения и организации с особыми требованиями к безопасности и персонализации.

Вся работа с приложением будет осуществляться в виде диалога с пользователем. Диалоговая система состоит из трех ключевых компонентов (рисунок 30):

– клиентская часть, которая обеспечивает пользовательский интерфейс для взаимодействия с системой, включая кнопки, поля ввода, выпадающие списки, таблицы, чекбоксы и др.;

– серверная часть, которая обрабатывает данные, отправленные пользователем, выполняет необходимые алгоритмы и задачи;

– база данных, где хранятся данные с определенными правилами и принципами управления данными.

Система управления базами данных (СУБД) представляет собой набор программных инструментов, которые позволяют создавать, управлять и администрировать базы данных. Другими словами, СУБД предоставляет возможности для организации, контроля и администрирования данных в базах данных.

Рисунок 9 – Диаграмма развертывания приложения

1.3.2 Проектирование базы данных

1.3.2.1 Логическая модель

Логическая модель базы данных приведена на рисунке 10 и содержит три сущности.

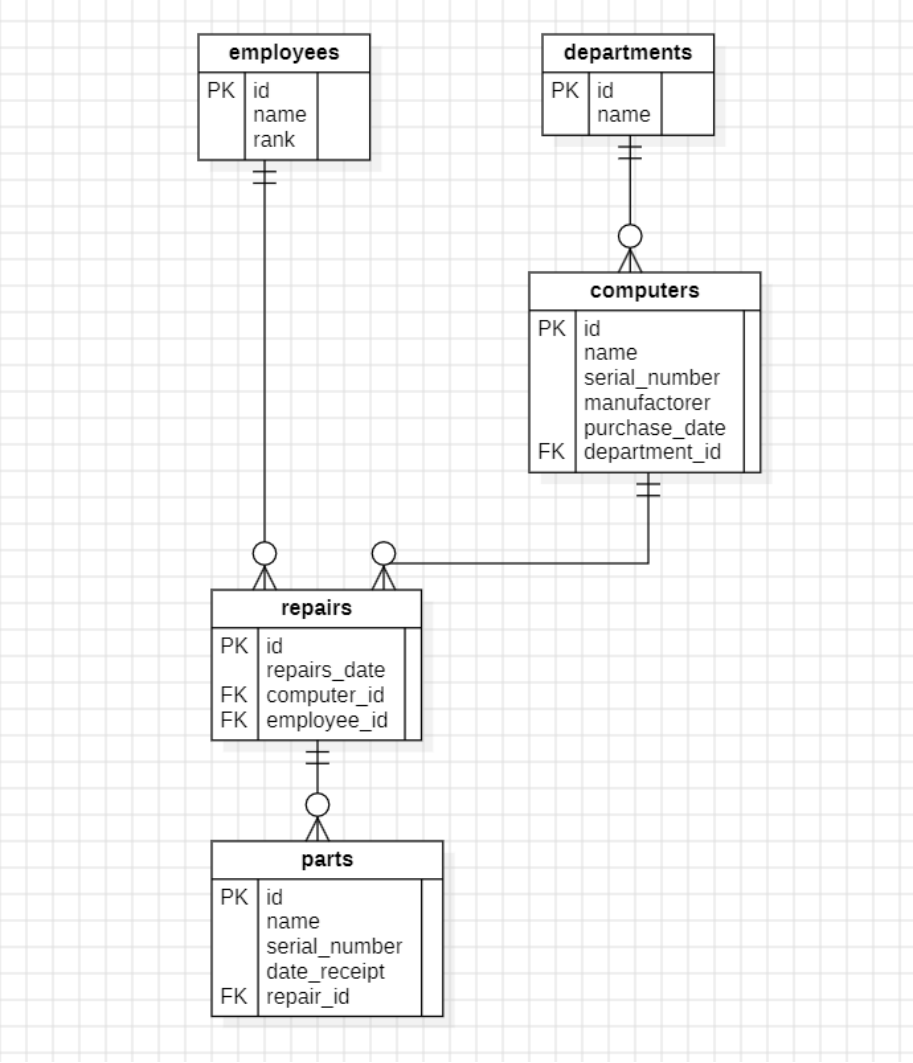


Рисунок 10 – Логическая модель данных

Описание атрибутов сущностей логической модели приведено в таблицах 1 – 5.

Таблица 1 – Описание атрибутов сущности «employees»

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
id	ID сотрудника	PK	
name	ФИО сотрудника		
rank	Должность сотрудника		

Таблица 2 – Описание атрибутов сущности «deaprtments»

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
id	ID подразделения	PK	
name	Название подразделения		

Таблица 3 - Описание атрибутов сущности «computers»

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
id	ID компьютерной техники	PK	
name	Название компьютерной техники		
serial_number	Серийный номер		
manufactorer	производитель		
purchase_date	Дата приобретения		
department_id	ID подразделения		FK

Таблица 4 - Описание атрибутов сущности «repairs»

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
id	ID ремонта	PK	
repairs_date	Дата ремонта		
computer_id	ID компьютерной техники		FK
employee_id	ID сотрудника		FK

Таблица 5 - Описание атрибутов сущности «parts»

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
id	ID комплектующего	PK	
name	Название комплектующего		
serial_number	Серийный номер		
date_receipt	Дата приобретения		
repair_id	ID ремонта		FK

Процесс нормализации – это последовательный процесс приведения базы данных к эталонному виду, то есть переход от одной нормальной формы к следующей.

Отношение находится в 1 НФ, если все строки таблицы различные и все атрибуты являются атомарными.

Отношение находится во 2 НФ, если оно находится в 1 НФ и каждый не ключевой атрибут таблицы функционально полно зависит от первичного ключа.

Отношение находится в 3 НФ, если оно находится во 2 НФ и между не ключевыми атрибутами не существует транзитивных зависимостей.

Все атрибуты трех сущностей атомарны и находятся в 3 НФ. Для подтверждения данного фактора необходимо рассмотреть каждую сущность:

- сущность «employees» имеет ключевой атрибут «id». Атрибуты «name», «rank» данной сущности являются простыми и неделимыми;
- сущность «departments» имеет ключевой атрибут «id». Атрибут «name» данной сущности является простым и неделимым;
- сущность «computers» имеет ключевой атрибут «id». Атрибуты «name», «serial_number», «manufactorer», «purchase_date» данной сущности являются простыми и неделимыми;
- сущность «repairs» имеет ключевой атрибут «id». Атрибут «repairs_date» данной сущности являются простыми и неделимыми;
- сущность «parts» имеет ключевой атрибут «id». Атрибуты «name», «serial_number», «date_receipt», «repair_id» данной сущности являются простыми и неделимыми.

Таким образом, была разработана логическая модель данных.

Процесс нормализации – это последовательный процесс приведения базы данных к эталонному виду, то есть переход от одной нормальной формы к следующей.

Отношение находится в 1 НФ, если все строки таблицы различные и все атрибуты являются атомарными.

Отношение находится во 2 НФ, если оно находится в 1 НФ и каждый не ключевой атрибут таблицы функционально полно зависит от первичного ключа.

Отношение находится в 3 НФ, если оно находится во 2 НФ и между не ключевыми атрибутами не существует транзитивных зависимостей.

Все атрибуты трех сущностей атомарны и находятся в 3 НФ. Для подтверждения данного фактора необходимо рассмотреть каждую сущность:

- сущность «employees» имеет ключевой атрибут «id». Атрибуты «name», «rank» данной сущности являются простыми и неделимыми;
- сущность «departments» имеет ключевой атрибут «id». Атрибут «name» данной сущности является простым и неделимым;
- сущность «computers» имеет ключевой атрибут «id». Атрибуты «name», «serial_number», «manufactorer», «purchase_date» данной сущности являются простыми и неделимыми;
- сущность «repairs» имеет ключевой атрибут «id». Атрибут «repairs_date» данной сущности являются простыми и неделимыми;
- сущность «parts» имеет ключевой атрибут «id». Атрибуты «name», «serial_number», «date_receipt», «repair_id» данной сущности являются простыми и неделимыми.

Таким образом, была разработана логическая модель данных.

1.3.3.2 Физическая модель

На основании разработанной логической модели данных была построена физическая модель данных. Физическая модель данных разработана для СУБД MySQL и представлена на рисунке 11.

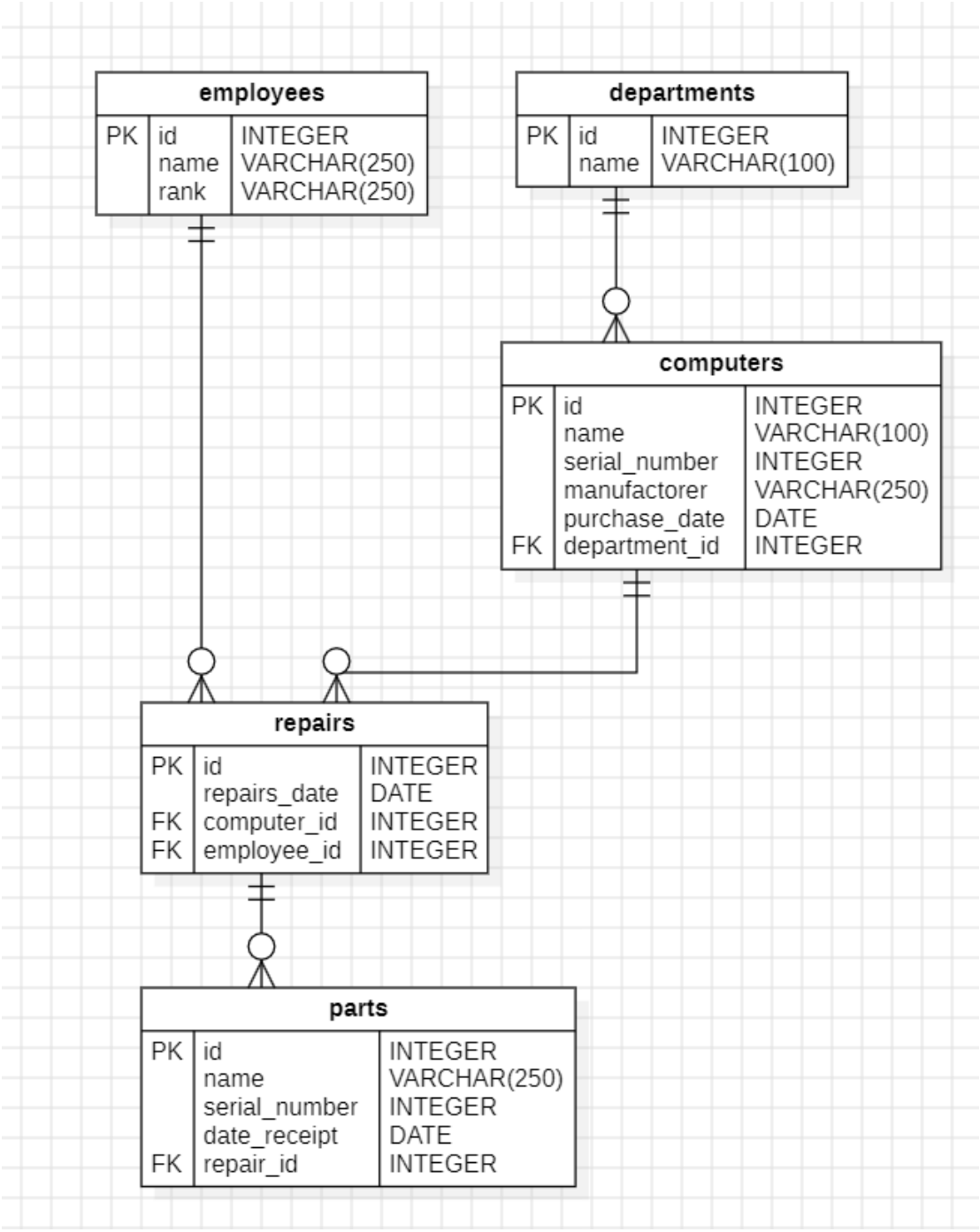


Рисунок 11 – Физическая модель данных

Физическая модель данных содержит три таблицы. Описание таблиц физической модели приведено в таблицах 6 - 10.

Таблица 6 – Описание полей таблицы «employees»

Имя атрибута	Физическое имя	Нулевое значение	Ключевое поле	Тип данных
id	id	Не допустимо	PK	INTEGER
name	name	Не допустимо		VARCHAR(250)
rank	rank	Не допустимо		VARCHAR(250)

Таблица 7 – Описание полей таблицы «departments»

Имя атрибута	Физическое имя	Нулевое значение	Ключевое поле	Тип данных
id	id	Не допустимо	PK	INTEGER
name	name	Не допустимо		VARCHAR(100)

Таблица 8 – Описание полей таблицы «computers»

Имя атрибута	Физическое имя	Нулевое значение	Ключевое поле	Тип данных
id	id	Не допустимо	PK	INTEGER
name	name	Не допустимо		VARCHAR(100)
serial_number	rank	Не допустимо		INTEGER
manufactorer	manufactorer	Не допустимо		VARCHAR(250)
purchase_date	purchase_date	Не допустимо		DATE
department_id	department_id	Не допустимо	FK	INTEGER

Таблица 9 – Описание полей таблицы «repairs»

Имя атрибута	Физическое имя	Нулевое значение	Ключевое поле	Тип данных
id	id	Не допустимо	PK	INTEGER
repairs_date	repairs_date	Не допустимо		DATE
computer_id	computer_id	Не допустимо	FK	INTEGER
employee_id	employee_id	Не допустимо	FK	INTEGER

Таблица 10 – Описание полей таблицы «parts»

Имя атрибута	Физическое имя	Нулевое значение	Ключевое поле	Тип данных
id	id	Не допустимо	PK	INTEGER
name	name	Не допустимо		VARCHAR(250)
serial_number	serial_number	Не допустимо		INTEGER
date_receipt	date_receipt	Не допустимо		DATE
repair_id	repair_id	Не допустимо	FK	INTEGER

Таким образом, была разработана физическая модель данных для СУБД MySQL.

1.3.2.4 Описание архитектуры

Программа реализована с помощью языка программирования C# в среде разработки Visual Studio 2022 и имеет модульную структуру, представленную на рисунке 12.

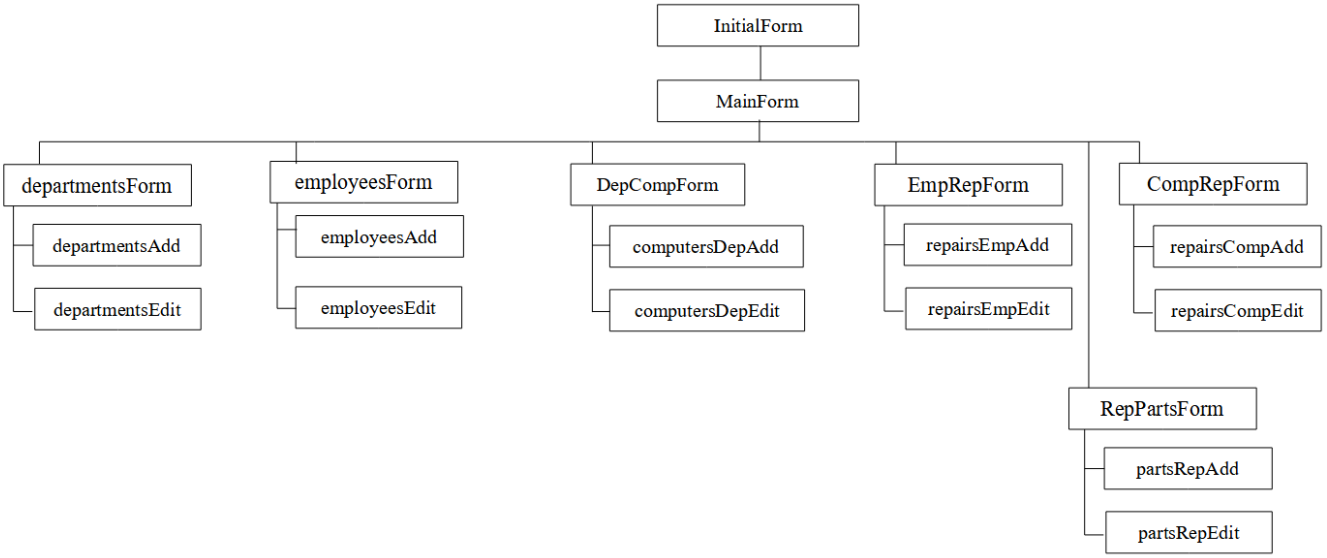


Рисунок 12 — Структура модулей программы

Вызов модулей приложения производится в соответствии с рисунком 4. При запуске на выполнение запускается модуль для авторизации пользователя InitialForm. Модуль авторизации связан с главным модулем приложения MainForm.

С помощью главного модуля MainForm можно вызвать модули:

- departmentsForm (для просмотра, редактирования информации о подразделениях воинской части);

- employeesForm (для просмотра, редактирования информации о продвижении сотрудниках воинской части);

- DepCompForm (для просмотра информации о подразделениях воинской части и просмотра, редактирования информации о компьютерной технике);

- EmpRepForm (для просмотра информации о сотрудниках воинской части и просмотра, редактирования информации о ремонте компьютерной техники);

- RepPartsForm (для просмотра информации о ремонте компьютерной техники и просмотра, редактирования информации о комплектующих для компьютерной техники);

- CompRepForm (для просмотра информации о компьютерной технике и просмотра, редактирования информации о ремонте компьютерной техники).

С помощью модуля departmentsForm можно вызвать модули:

- departmentsAdd (для добавления информации о подразделениях воинской части);

- departmentsEdit (для изменения информации о подразделениях воинской части).

С помощью модуля employeesForm можно вызвать модули:

- employeesAdd (для добавления информации о сотрудниках);

- employeesEdit (для изменения информации о сотрудниках).

С помощью модуля DepCompForm можно вызвать модули:

- computersDepAdd (для добавления информации о компьютерной технике);

- computersDepEdit (для изменения информации о компьютерной технике).

С помощью модуля EmpRepForm можно вызвать модули:

- repairsEmpAdd (для добавления информации о ремонте компьютерной техники);

- repairsEmpEdit (для изменения информации о ремонте компьютерной техники).

С помощью модуля RepPartsForm можно вызвать модули:

- partsRepAdd (для добавления информации о комплектующих для компьютерной техники);

- partsRepEdit (для изменения информации о комплектующих для компьютерной техники);

С помощью модуля CompRepForm можно вызвать модули:

- repairsCompAdd (для добавления информации о ремонте компьютерной техники);

- repairsCompEdit (для изменения информации о ремонте компьютерной техники).

Описание модулей представлено в таблице 11.

Таблица 11 – Описание модулей программы

Название модуля	Описание модуля
InitialForm	Модуль для отображения формы авторизации.
MainForm	Модуль для отображения главной формы приложения.
departmentsForm	Отображает форму приложения с таблицей «Подразделения»
departmentsAdd	Отображает форму добавления записей в таблицу «Подразделения»
departmentsEdit	Отображает форму изменения данных в таблице «Подразделения».
employeesForm	Отображает форму приложения с таблицей «Сотрудники»
employeesAdd	Отображает форму добавления записей в таблицу «Сотрудники»
employeesEdit	Отображает форму изменения данных в таблице «Сотрудники».
DepCompForm	Отображает форму приложения с таблицами «Подразделения» и «Компьютерная техника»
computersDepAdd	Отображает форму добавления записей в таблицу «Компьютерная техника»
computersDepEdit	Отображает форму изменения данных в таблице «Компьютерная техника».
EmpRepForm	Отображает форму приложения с таблицами «Сотрудники» и «Ремонт»
repairsEmpAdd	Отображает форму добавления записей в таблицу «Ремонт»
repairsEmpEdit	Отображает форму изменения данных в таблице «Ремонт».
employeesEdit	Отображает форму изменения данных в таблице «Сотрудники».

Название модуля	Описание модуля
CompRepForm	Отображает форму приложения с таблицами «Компьютерная техника» и «Ремонт»
repairsCompAdd	Отображает форму добавления записей в таблицу «Ремонт»
repairsCompEdit	Отображает форму добавления записей в таблицу «Ремонт».
RepPartsForm	Отображает форму приложения с таблицами «Ремонт» и «Комплектующие»
partsRepAdd	Отображает форму добавления записей в таблицу «Комплектующие»
partsRepEdit	Отображает форму добавления записей в таблицу «Комплектующие».

1.4 Разработка приложения

1.4.1 Реализация БД

Перед началом реализации БД её необходимо создать с помощью SQL-запроса:

```
CREATE DATABASE MilCompp;
```

Затем создадим внутри БД необходимые таблицы. В данном разделе будет описан процесс создания и работа с основными таблицами БД.

Для создания таблицы, которая будет хранить информацию о сотрудниках, напишем следующий запрос:

```
CREATE TABLE employees (
  id int NOT NULL AUTO_INCREMENT,
  name varchar(250) NOT NULL,
  rank varchar(250) NOT NULL,
  PRIMARY KEY (id)
)  ENGINE=InnoDB  AUTO_INCREMENT=11  DEFAULT  CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci
```

Для создания таблицы, которая будет хранить информацию о подразделениях воинской части, напишем следующий запрос:

```
CREATE TABLE departments (
  id int NOT NULL AUTO_INCREMENT,
  name varchar(100) NOT NULL,
  PRIMARY KEY (id)
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci
```

Для создания таблицы, которая будет хранить данные о компьютерной технике, напишем следующий запрос:

```
CREATE TABLE computers (
  id int NOT NULL AUTO_INCREMENT,
  name varchar(70) DEFAULT NULL,
  serial_number int NOT NULL,
  manufactorer varchar(250) NOT NULL,
  purchase_date date NOT NULL,
  department_id int NOT NULL,
  PRIMARY KEY (id),
  KEY dep_id (department_id),
  CONSTRAINT dep_id FOREIGN KEY (department_id) REFERENCES
departments (id) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=23 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci
```

Для создания таблицы, которая будет хранить сведения о ремонтах компьютерной техники, напишем следующий запрос:

```
CREATE TABLE repairs (
  id int NOT NULL AUTO_INCREMENT,
  repairs_date date NOT NULL,
  computer_id int NOT NULL,
  employee_id int NOT NULL,
  statud enum('В ремонте','Отремонтирован','Списан') NOT
NULL,
  PRIMARY KEY (id),
  KEY comp_id (computer_id),
  KEY emp_id1 (employee_id),
  CONSTRAINT comp_id FOREIGN KEY (computer_id) REFERENCES
computers (id) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT emp_id1 FOREIGN KEY (employee_id) REFERENCES
employees (id) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=12 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci
```

Для создания таблицы, которая будет хранить информацию о комплектующих, напишем следующий запрос:

```
CREATE TABLE parts (
  id int NOT NULL AUTO_INCREMENT,
  name varchar(250) DEFAULT NULL,
  serial_number int NOT NULL,
  date_receipt date DEFAULT NULL,
  repair_id int NOT NULL,
  PRIMARY KEY (id),
  KEY rep_id (repair_id),
  CONSTRAINT rep_id FOREIGN KEY (repair_id) REFERENCES repairs (id) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

Для добавления данных в таблицу использовался оператор INSERT. Например, следующий запрос добавляет в таблицу departments информацию о подразделении в воинской части:

```
INSERT INTO departments (name) VALUES ('Технический отдел №10');
```

Для добавления данных в таблицу использовался оператор UPDATE. Например, следующий запрос изменяет в таблице departments сведения о подразделении с id = 6:

```
UPDATE departments
SET name ='Технический отдел №14'
WHERE (departments.id = 6);
```

Для удаления данных из таблицы использовался оператор DELETE. Например, следующий запрос удаляет в таблице employees сведения о сотруднике с id = 11:

```
DELETE FROM employees
WHERE (employees.id = 11);
```

1.4.2 Описание программы

Файл MilComp.sln — решение Visual Studio 2022, объединяющее все компоненты программы.

Папка MilComp содержит исходный код приложения для учёта хранения и ремонта компьютерной техники в воинской части.

В папке MilComp/MilComp находятся формы и файлы логики приложения:

- InitialForm.cs — форма авторизации пользователя приложения;

- MainForm.cs — главная форма программы с вкладками Таблицы и Выход;

- employeesForm.cs — форма, в которой отображается таблица «Сотрудники» с помощью метода FillDGV():

```
public void FillDGV()
        {
            command = new
MySql.Data.MySqlClient.MySqlCommand("SELECT * FROM employees",
conn);
            adapter = new
MySql.Data.MySqlClient.MySqlDataAdapter(command);
            DataTable table = new DataTable();
            adapter.Fill(table);
            dg1.DataSource = table;
            dg1.ReadOnly = true;
            dg1.SelectionMode =
DataGridViewSelectionMode.FullRowSelect;
            dg1.RowHeadersVisible = false;
            dg1.Columns["id"].HeaderText = "Номер сотрудника";
            dg1.Columns["name"].HeaderText = "ФИО сотрудника";
            dg1.Columns["rank"].HeaderText = "Должность
сотрудника";
            dg1.Columns["id"].Width = 150;
            dg1.Columns["name"].Width = 200;
            dg1.Columns["rank"].Width = 150;

            foreach (DataGridViewColumn column in dg1.Columns)
            {
                column.HeaderCell.Style.Alignment =
DataGridViewContentAlignment.MiddleCenter;
            }
}
```

Подобные методы описаны во всех формах для отображения записей таблиц базы данных.

- CompRepForm.cs — форма, реализующая отображение таблиц компьютерной техники и ее ремонта;

- computersDepAdd.cs — форма для добавления или редактирования данных о компьютерах относительно подразделений воинской части;

Файл App.config — конфигурация приложения, включая строку подключения к базе данных MySQL.

Папка vs содержит служебные файлы Visual Studio (индексы, настройки) — в работе программы не участвуют.

Папка packages — содержит установленные NuGet-пакеты для работы с MySQL и другими библиотеками.

Программа реализует следующие функции:

- подключение к базе данных MySQL с таблицами: сотрудники, подразделения, компьютерная техника, ремонт компьютеров, комплектующие;
- регистрация новой компьютерной техники и комплектующих для нее;
- учёт текущего состояния компьютерной техники;
- добавление нового сотрудника, подразделения воинской части в базу данных;
- назначение ответственного за ремонт компьютерной техники сотрудника и запись деталей ее ремонта;
- добавление, редактирование и удаление записей через формы Windows Forms;
- визуальный интерфейс на базе стандартных компонентов WinForms.

1.4.3 Тестирование

Главная форма программы изображена на рисунке 13. Чтобы начать работу с программой, необходимо совершить вход (рисунок 14).

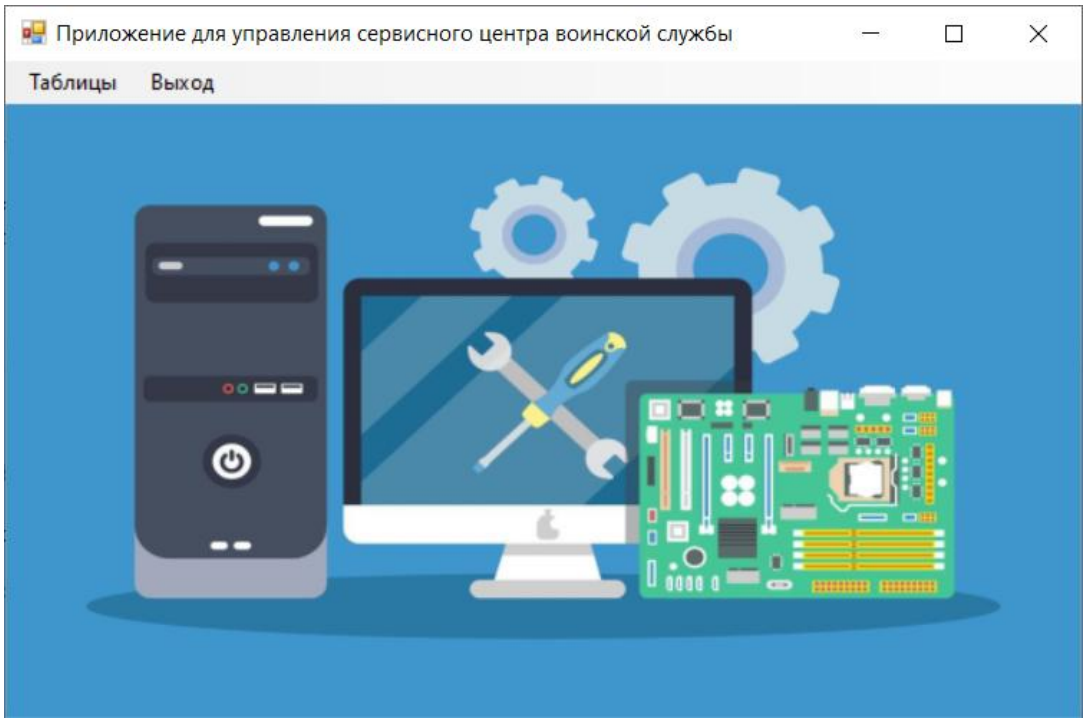


Рисунок 13 – Главный вид программы «Склад компьютерной техники»

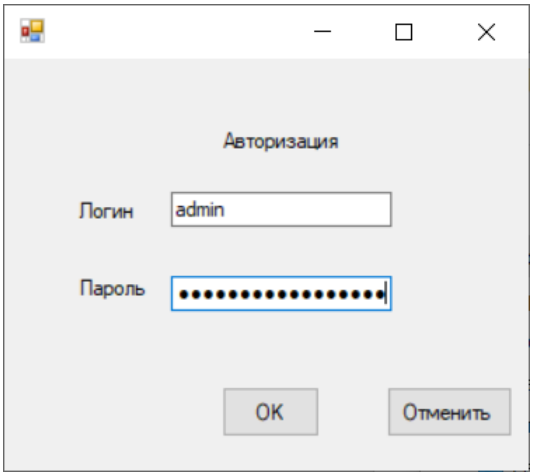


Рисунок 14 – Авторизация в системе

Весь функционал главной формы представлен в виде нескольких вкладок, каждая из которых выполняет определенные задачи.

На вкладке «Таблицы» доступна вся информация о таблицах склада компьютерной техники в воинской части (рисунок 15).

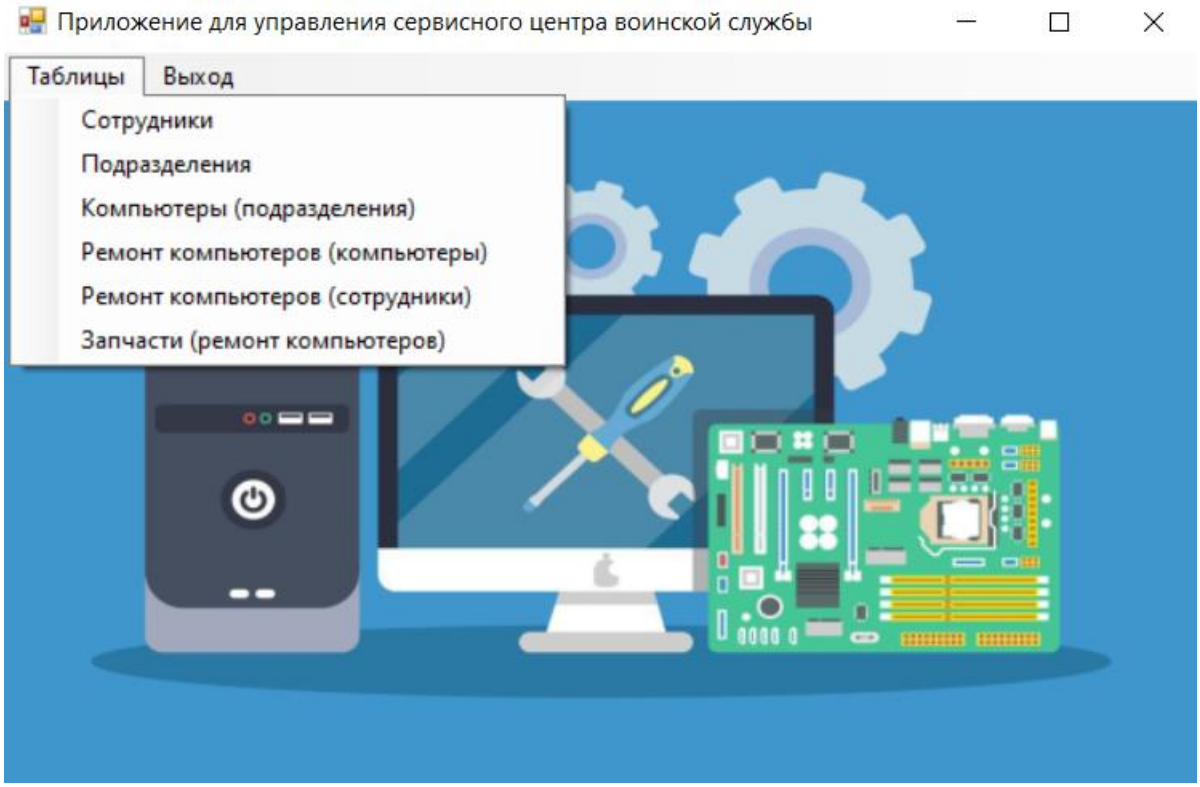


Рисунок 15 – Вкладка «Таблицы»

Для того чтобы посмотреть информацию о сотрудниках, таких как ФИО сотрудника и его должность, а также добавить, изменить или удалить сотрудника, необходимо один раз кликнуть на нужного сотрудника, после чего он станет для просмотра, редактирования или удаления (рисунок 16). С помощью кнопки «Добавить» в систему вносятся сведения о новом сотруднике, а с помощью кнопки «Вывести изменения» происходит обновление данных.

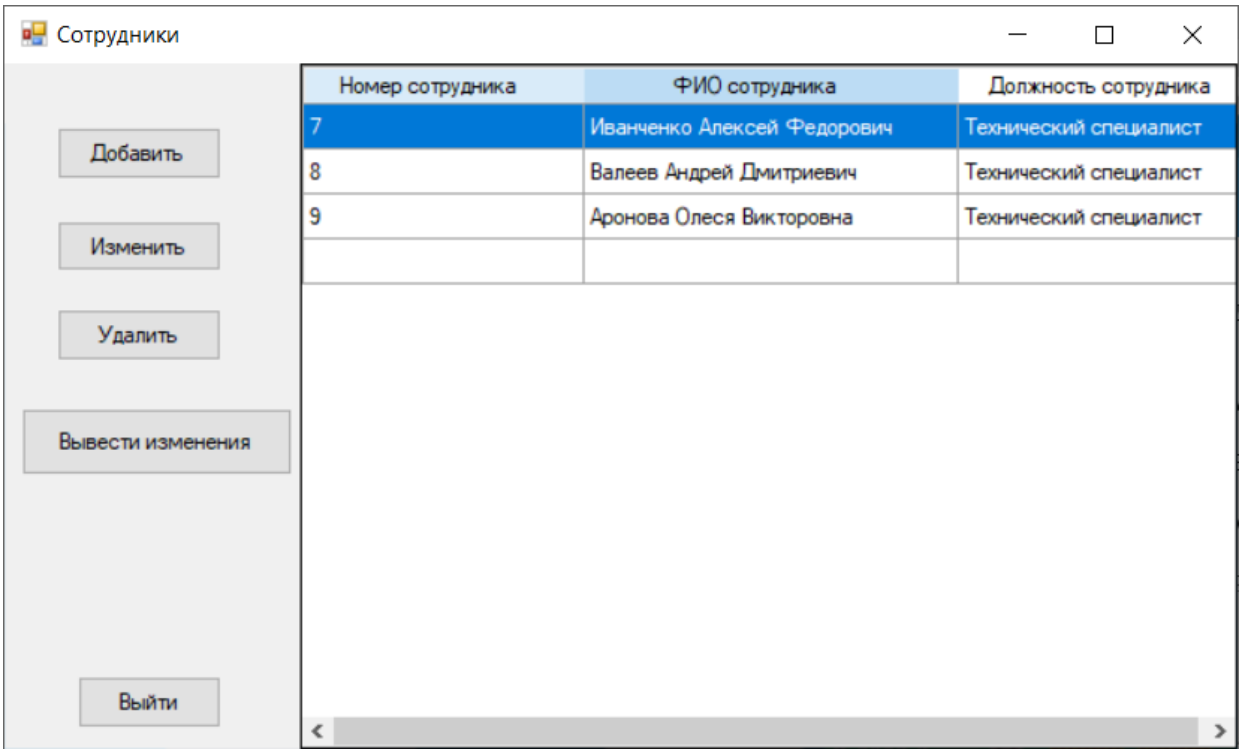


Рисунок 16 – Таблица «Сотрудники»

Для добавления записи в таблицу «Сотрудники» нужно нажать на кнопку «Добавить». После этого откроется форма для добавления записи. Чтобы добавить запись в таблицу «Сотрудники», нужно заполнить необходимые поля: ФИО сотрудника и его должность и нажать на кнопку «ОК» (рисунок 17). Результат добавления записи представлен на рисунке 18.

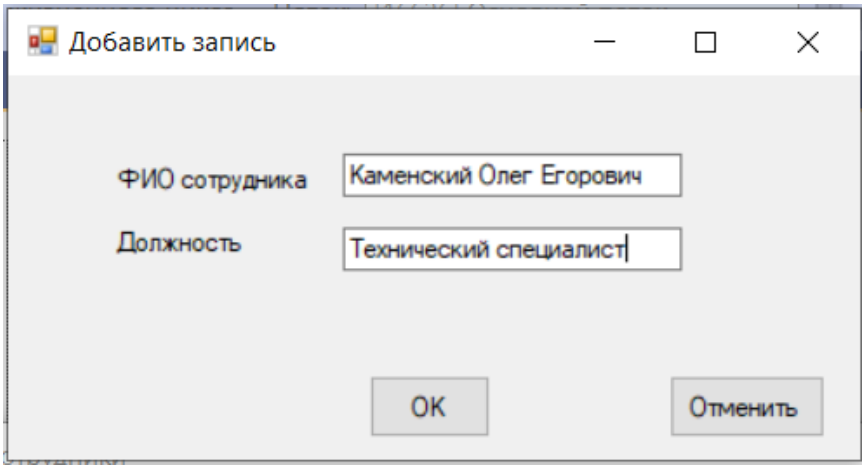


Рисунок 17 – Добавление данных в таблицу «Сотрудники»

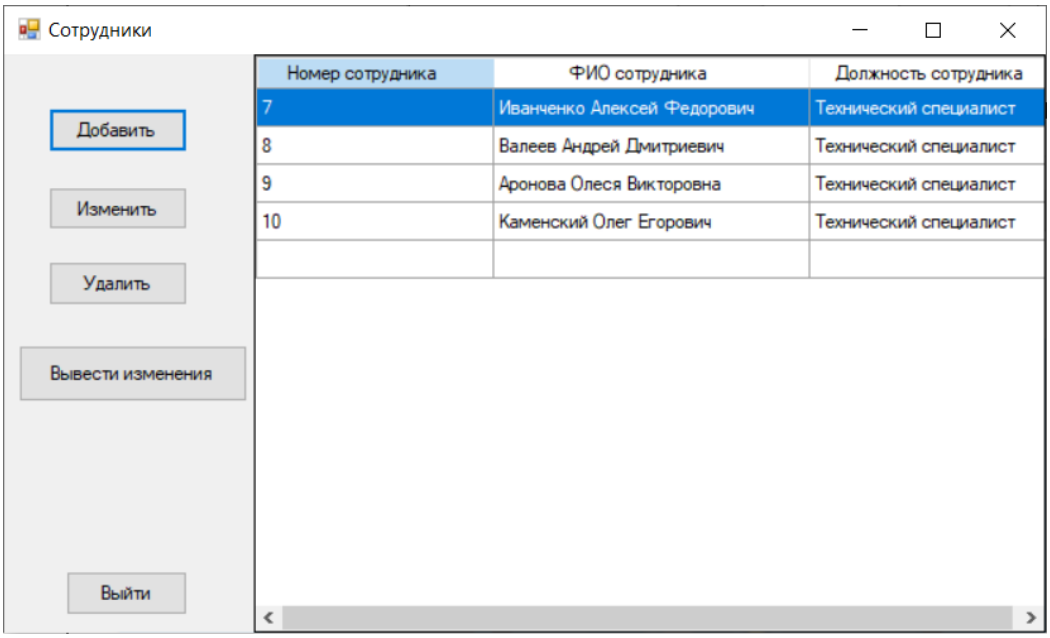


Рисунок 18 – Результат добавления данных в таблицу «Сотрудники»

Для изменения записи в таблице «Сотрудники» нужно выбрать запись и нажать на кнопку «Изменить». Например надо выбрать запись с номером 7. Далее открывается форма изменения этой записи в таблице. (рисунок 19). Затем происходит изменение данных полей выбранной записи. После этого нужно нажать на кнопку «ОК» (рисунок 20). Результат изменения записи с номером 7 в таблице «Сотрудники» представлен на рисунке 21.

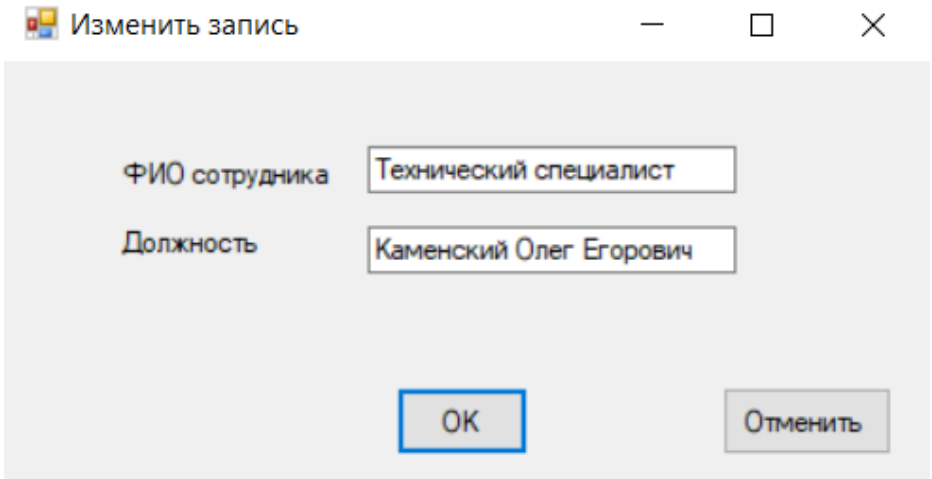


Рисунок 19 – Окно изменения записи в таблице «Сотрудники»

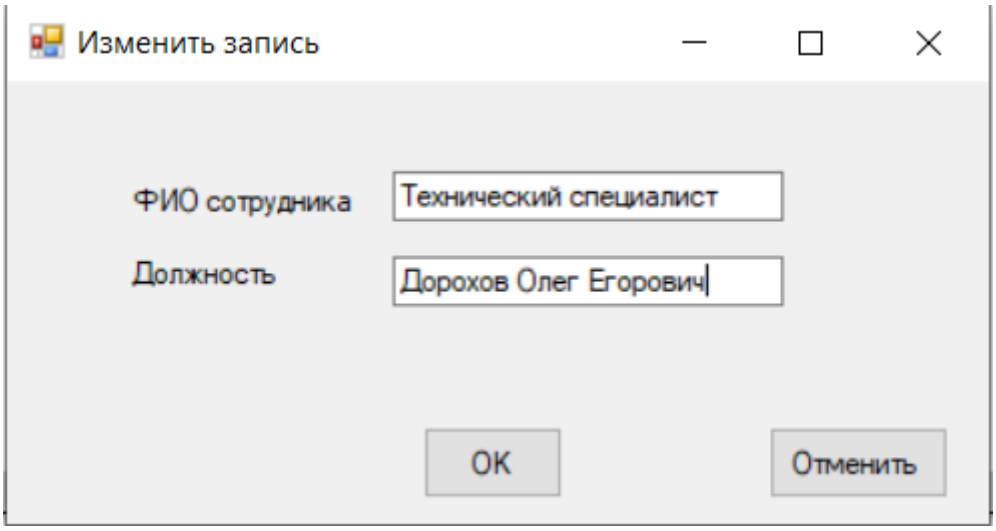


Рисунок 20 – Изменение записи в таблице «Сотрудники»

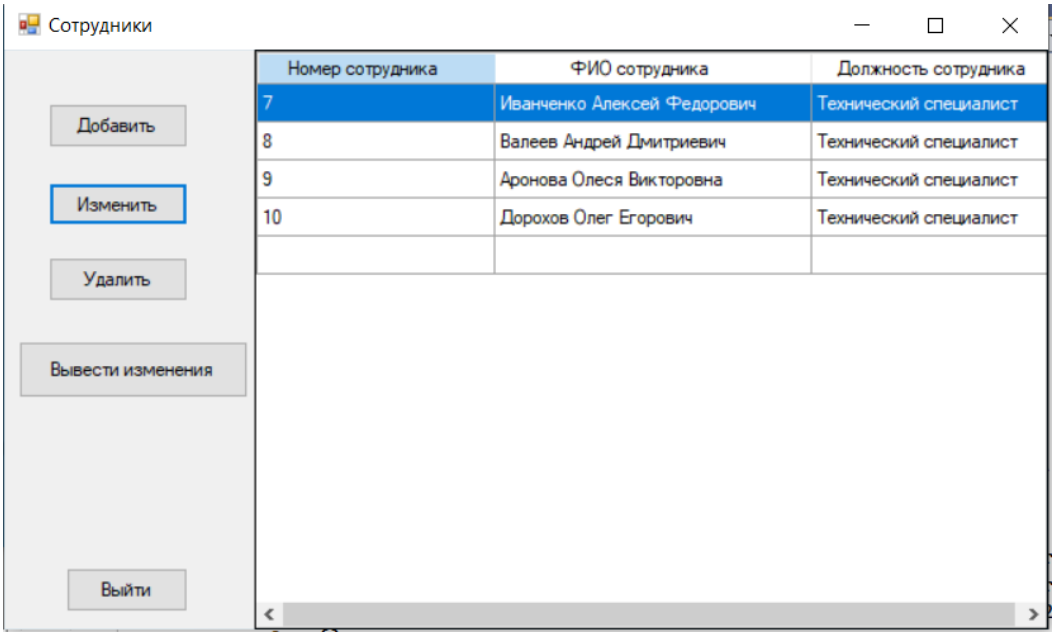


Рисунок 21 – Результат изменения записи в таблице «Сотрудники»

Для удаления записи в таблице «Сотрудники» надо выбрать запись для удаления, нажать кнопку «Удалить». После этого появится окно удаления данных (рисунок 22), на котором нужно нажать «ОК». Результат удаления записи представлен на рисунке 23.

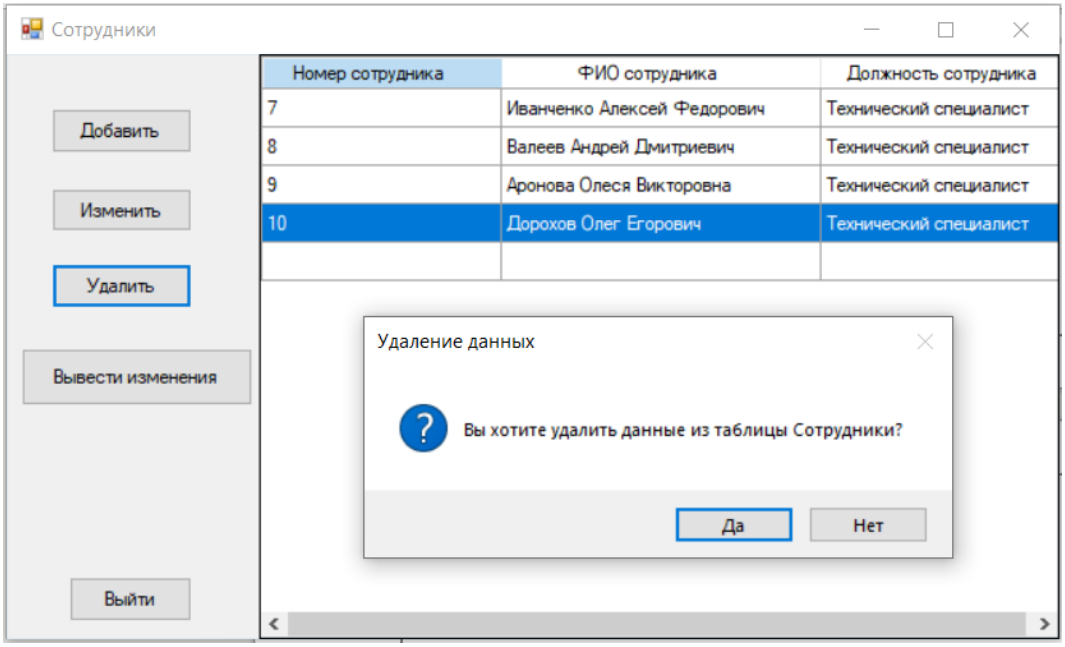


Рисунок 22 – Удаление записи в таблице «Сотрудники»

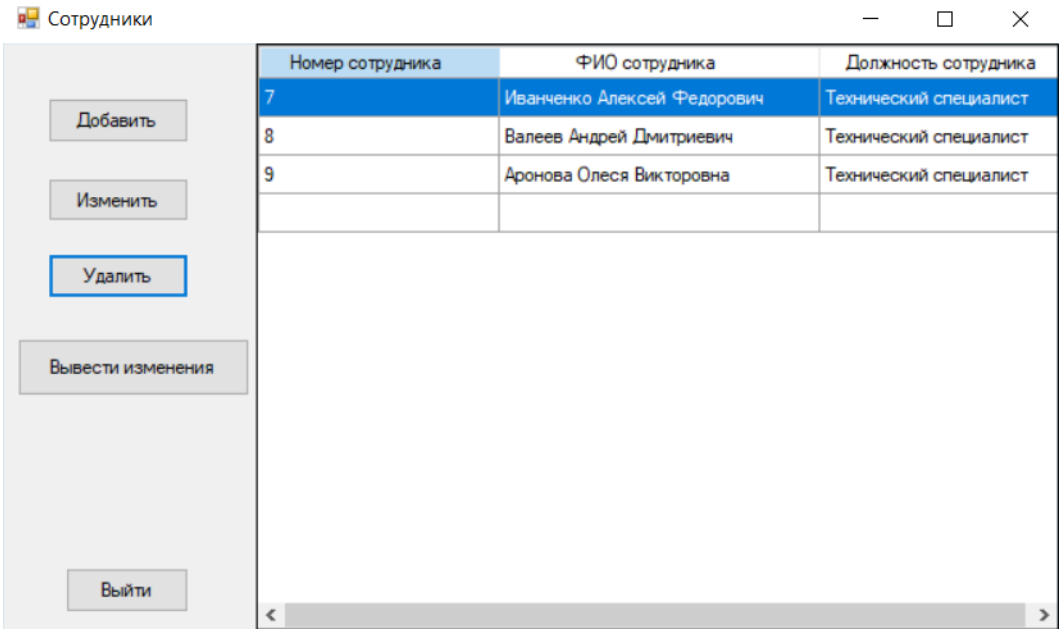


Рисунок 23 – Результат удаления записи в таблице «Сотрудники»

В таблице «Подразделения» хранятся данные о подразделениях воинской части (рисунок 24).

Управление меню происходит с помощью кнопок «Добавить», «Изменить», «Удалить», расположенных слева от таблицы «Подразделения». А также можно обновить данные в таблице, нажав на кнопку «Вывести изменения».

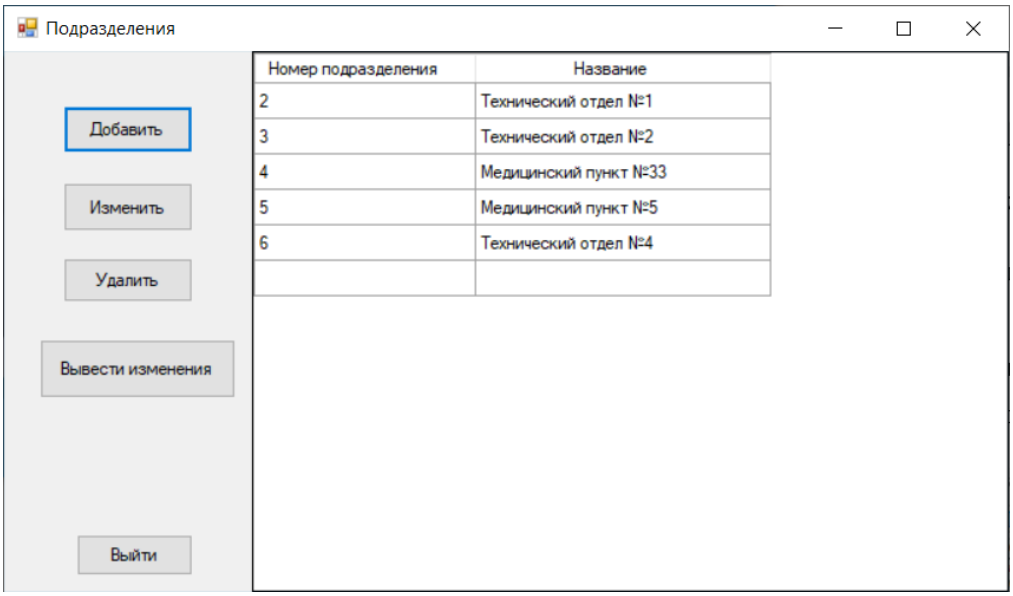


Рисунок 24 – Таблица «Подразделения»

Для добавления записи в таблицу «Подразделения» нужно нажать на кнопку «Добавить». После этого откроется форма для добавления записи. Чтобы добавить запись в таблицу «Подразделение», нужно заполнить поле названия подразделения и нажать на кнопку «ОК» (рисунок 25). Результат добавления записи представлен на рисунке 26.

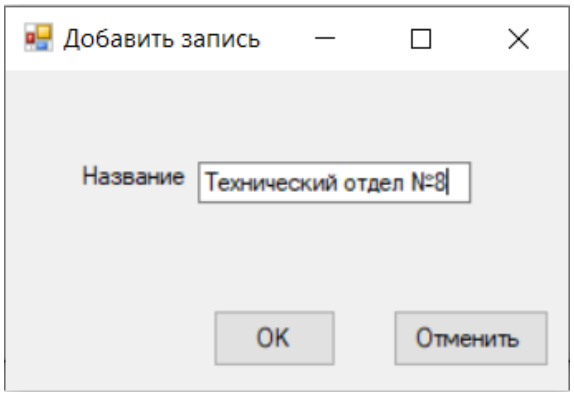


Рисунок 25 – Добавление записи в таблицу «Подразделения»

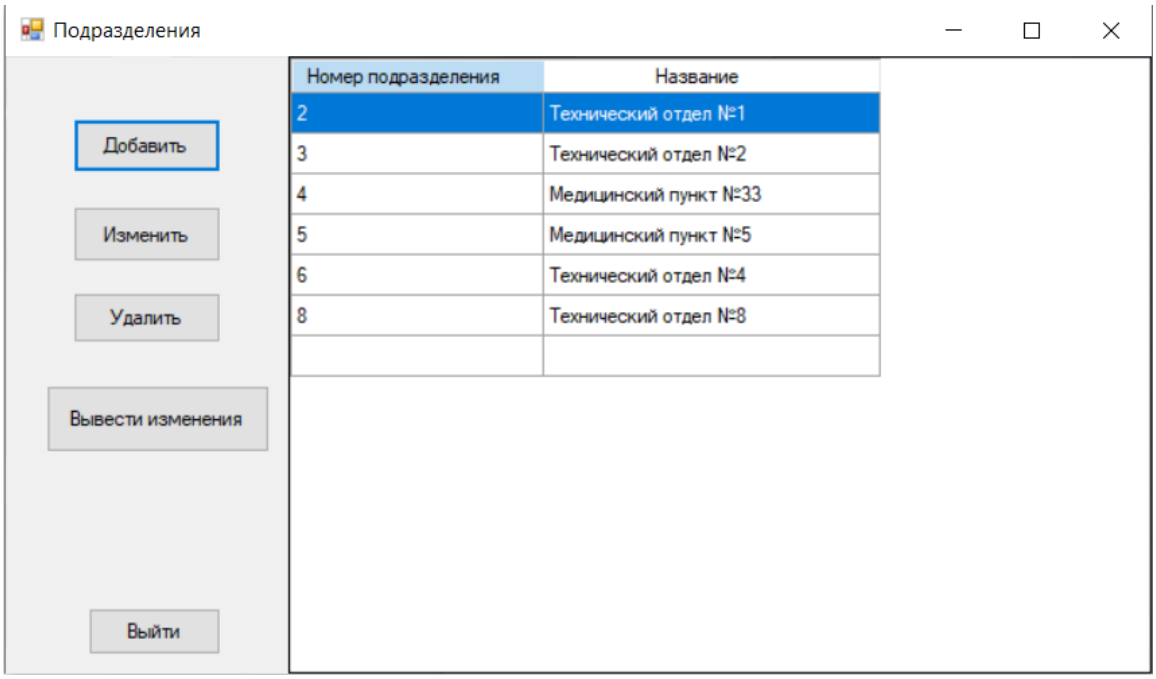


Рисунок 26 – Результат добавления данных в таблицу «Подразделения»

Для добавления записи в таблицу «Подразделения» нужно нажать на кнопку «Добавить». После этого откроется форма для добавления записи. Чтобы добавить запись в таблицу «Подразделения», нужно заполнить поле названия подразделения и нажать на кнопку «ОК» (рисунок 27). Результат добавления записи представлен на рисунке 28.

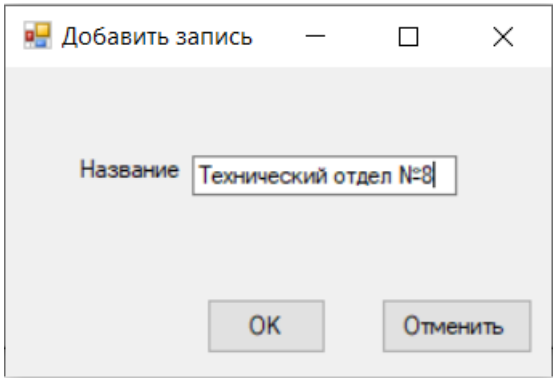


Рисунок 27 – Добавление записи в таблицу «Подразделения»

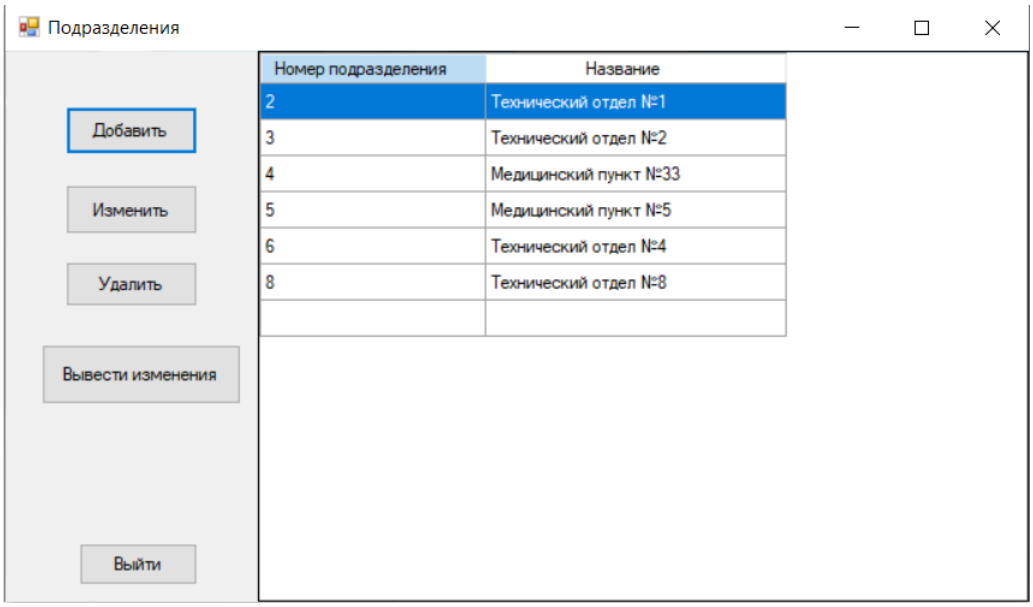


Рисунок 28 – Результат добавления данных в таблицу «Подразделения»

Для изменения записи в таблице «Подразделения» нужно выбрать запись и нажать на кнопку «Изменить». Например надо выбрать запись с номером 8. Далее открывается форма изменения этой записи в таблице. (рисунок 29). Затем происходит изменение данных полей выбранной записи. После этого нужно нажать на кнопку «ОК» (рисунок 30). Результат изменения записи с номером 8 в таблице «Сотрудники» представлен на рисунке 31.

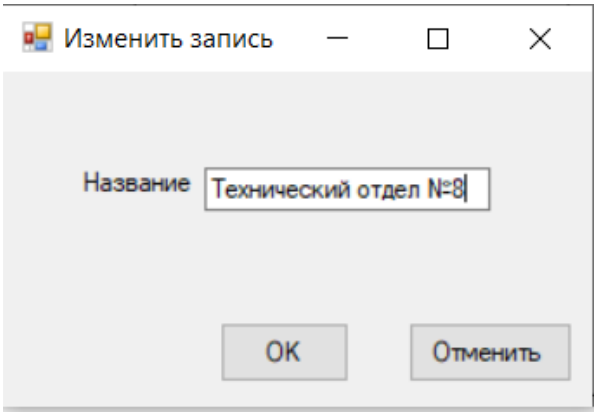


Рисунок 29 – Окно изменения записи в таблице «Подразделения»

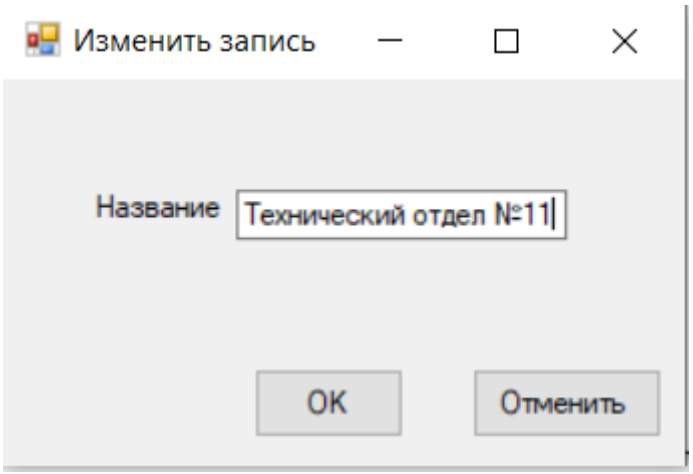


Рисунок 30 – Изменение записи в таблице «Подразделения»

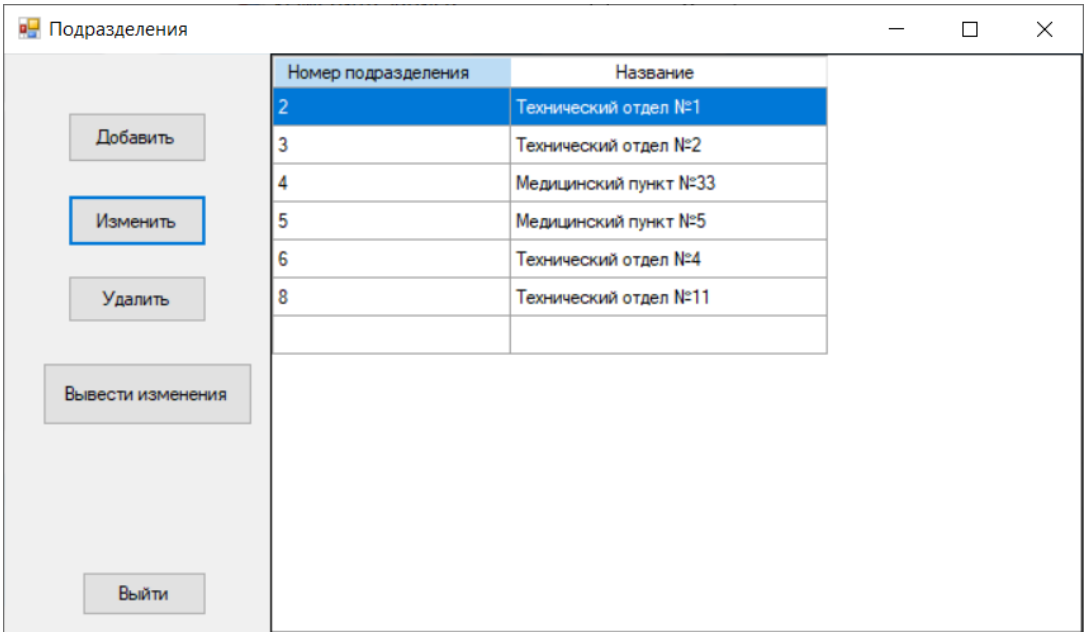


Рисунок 31 – Результат изменения записи в таблице «Подразделения»

Для удаления записи в таблице «Подразделения» надо выбрать запись для удаления, нажать кнопку «Удалить». После этого появится окно удаления данных (рисунок 32), на котором нужно нажать «ОК». Результат удаления записи представлен на рисунке 33.

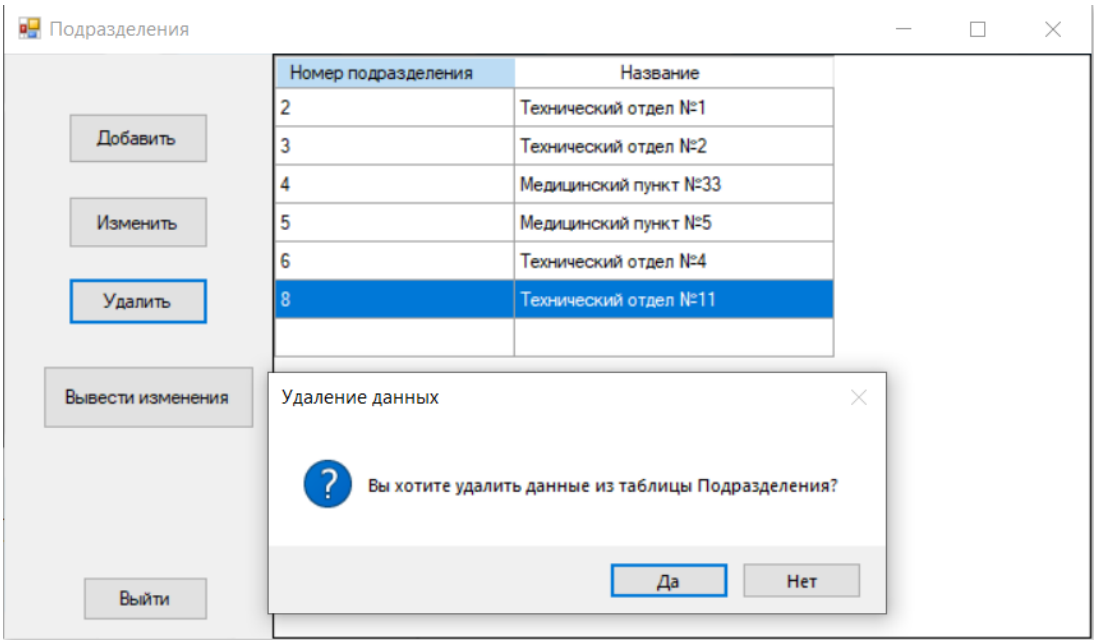


Рисунок 32 – Удаление записи в таблице «Подразделения»

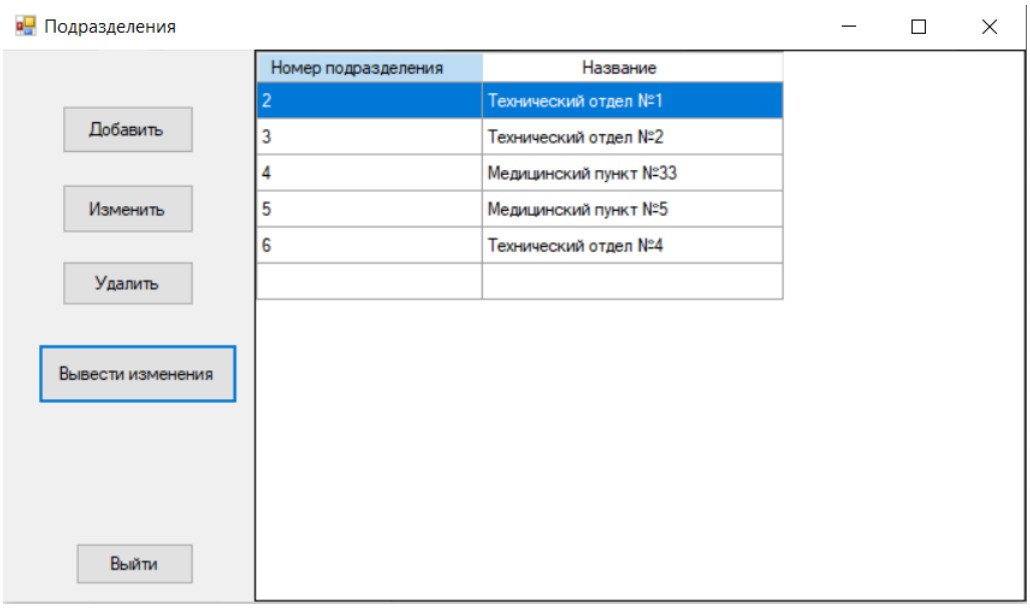


Рисунок 33 – Результат удаления записи в таблице «Подразделения»

Перейдя на форму с таблицами «Подразделения» и «Компьютеры», можно увидеть информацию компьютерах, которые находятся в определенном подразделении: номер компьютера, его инвентарный номер, производителя компьютера и дату приобретения компьютера (рисунок 34). В данной таблице можно просматривать данные о компьютерах, которые находятся в определенном подразделении воинской части. Также можно редактировать данные таблицы и обновлять их с помощью кнопки «Вывести изменения».

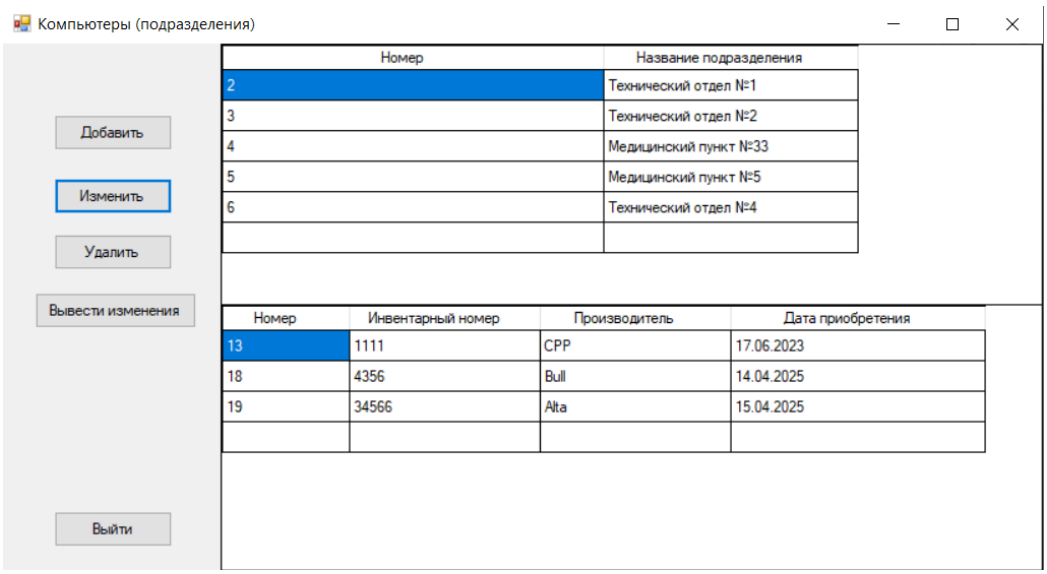


Рисунок 34 – Таблицы «Подразделения» и «Компьютеры»

Для добавления записи в таблицу «Компьютеры» нужно нажать на кнопку «Добавить». После этого откроется форма для добавления записи. Чтобы добавить запись в таблицу «Компьютеры», нужно заполнить необходимые поля: серийный номер, производителя и дату приобретения компьютера, а потом нажать на кнопку «ОК» (рисунок 35). Результат добавления записи представлен на рисунке 36.

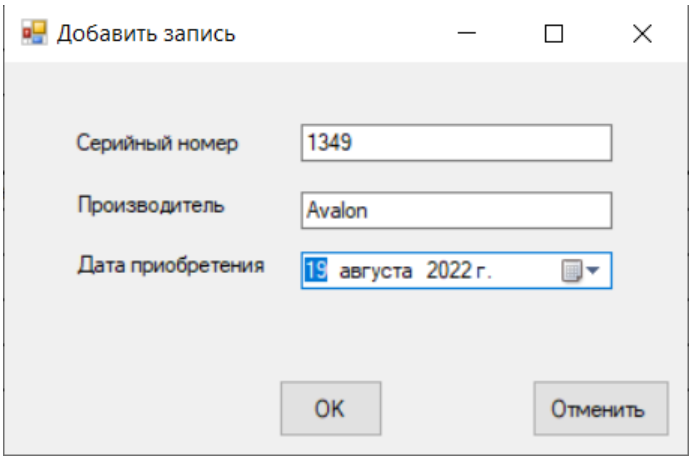


Рисунок 35 – Добавление записи в таблицу «Компьютеры»

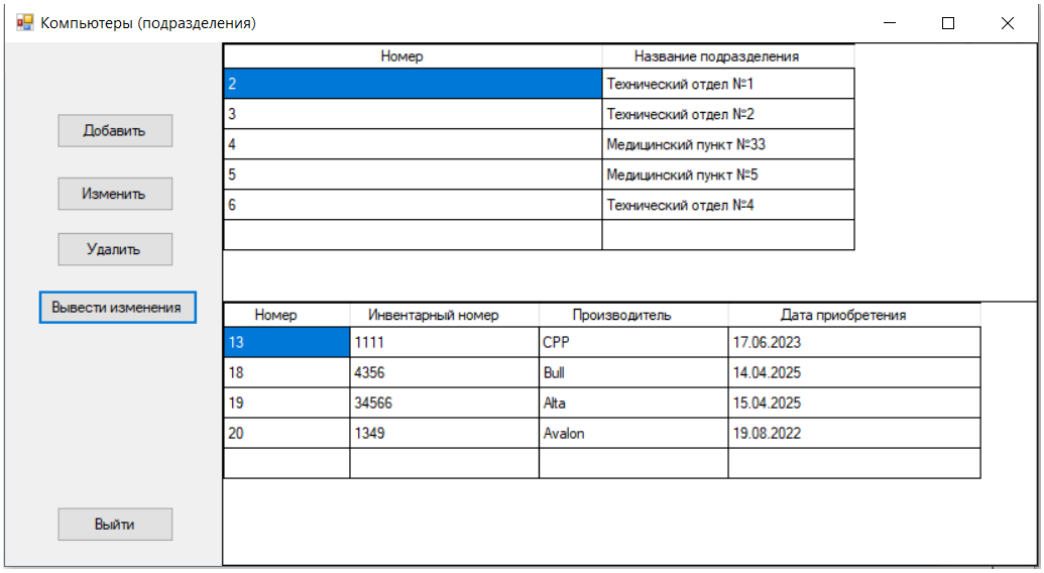


Рисунок 36 – Результат добавления данных в таблицу «Компьютеры»

Для изменения записи в таблице «Компьютеры» нужно выбрать запись и нажать на кнопку «Изменить». Например надо выбрать запись с номером 20. Далее открывается форма изменения этой записи в таблице. (рисунок 37). Затем происходит изменение данных полей выбранной записи. После этого нужно нажать на кнопку «ОК» (рисунок 38). Результат изменения записи с номером 20 в таблице «Компьютеры» представлен на рисунке 39.

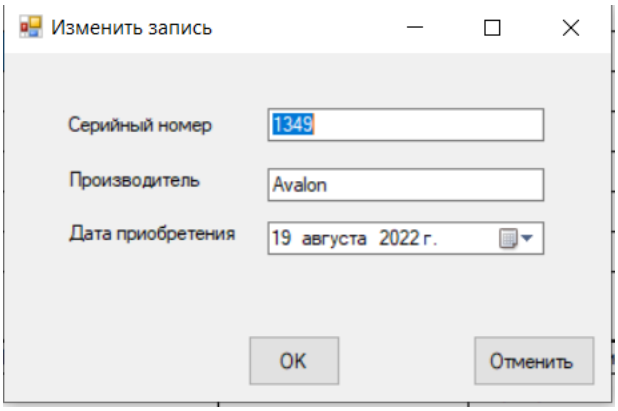


Рисунок 37 – Окно изменения записи в таблице «Компьютеры»

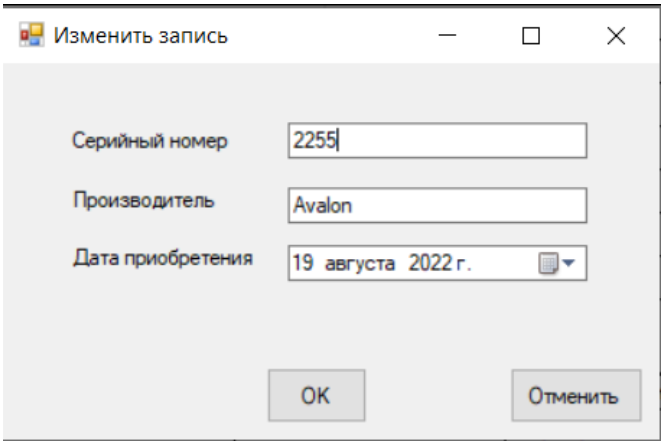


Рисунок 38 – Изменение записи в таблице «Компьютеры»

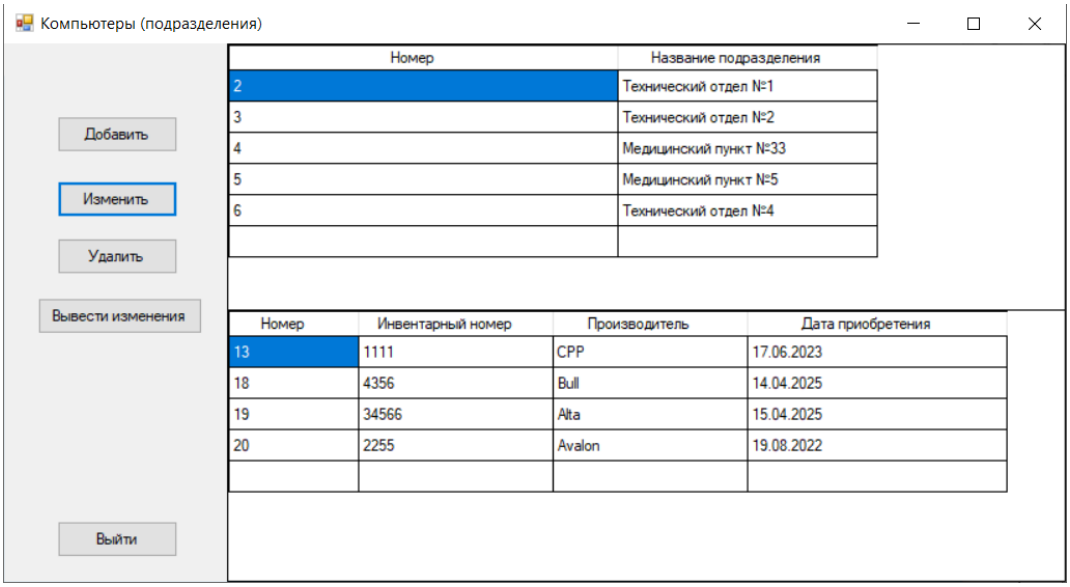


Рисунок 39 – Результат изменения записи в таблице «Компьютеры»

Для удаления записи в таблице «Компьютеры» надо выбрать запись для удаления, нажать кнопку «Удалить». После этого появится окно удаления данных (рисунок 40), на котором нужно нажать «ОК». Результат удаления записи представлен на рисунке 41.

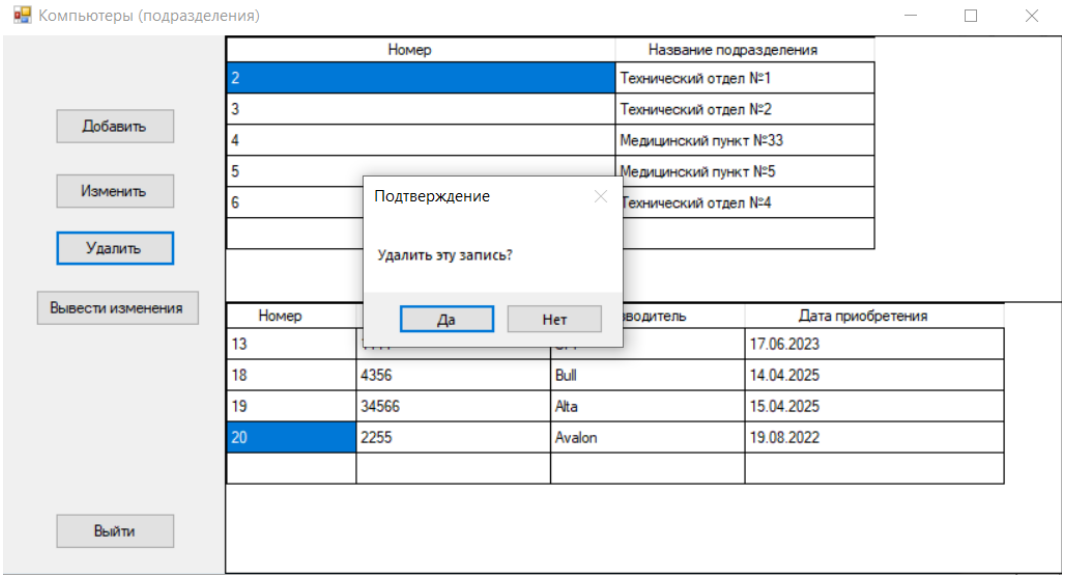


Рисунок 40 – Удаление записи в таблице «Компьютеры»

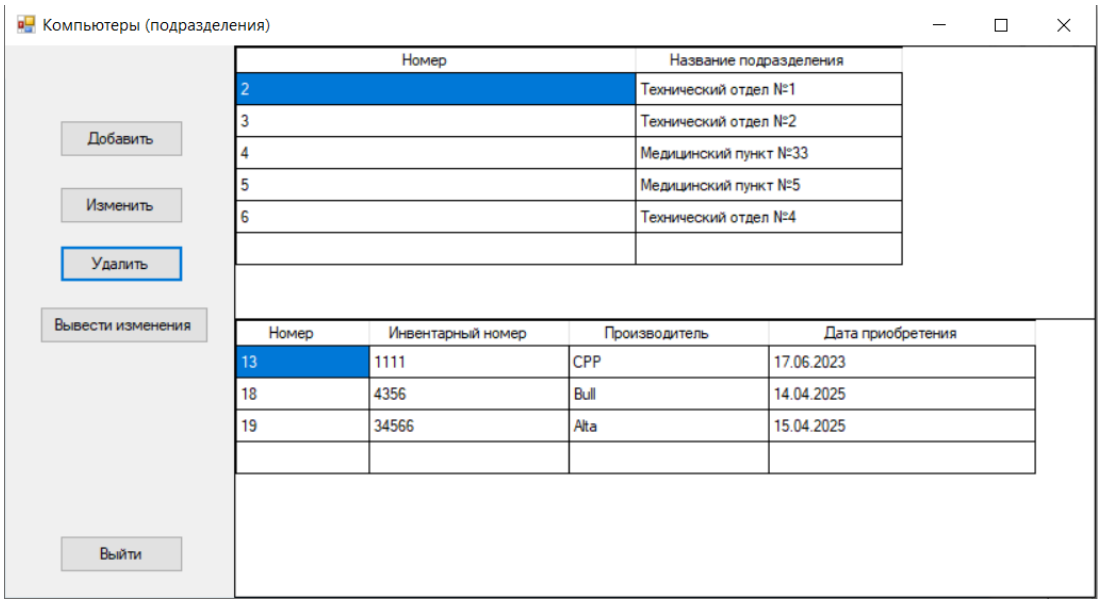


Рисунок 41 – Результат удаления записи в таблице «Компьютеры»

Перейдя на форму с таблицами «Компьютеры» и «Ремонт компьютеров», можно увидеть информацию ремонтах конкретного компьютера: дата ремонта, номер сотрудника, статус ремонта (рисунок 42). В данной таблице можно просматривать данные о ремонтах конкретного компьютера. Также можно редактировать данные таблицы и обновлять их с помощью кнопки «Вывести изменения».

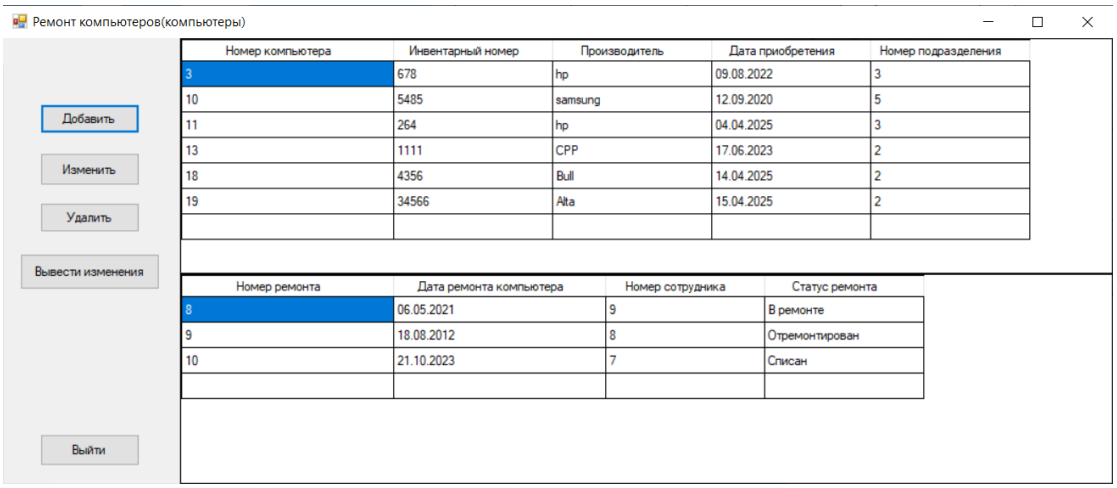


Рисунок 42 – Таблицы «Компьютеры» и «Ремонт компьютеров»

Для добавления записи в таблицу «Ремонт компьютеров» нужно нажать на кнопку «Добавить». После этого откроется форма для добавления записи. Чтобы добавить запись в таблицу «Ремонт компьютеров», нужно заполнить необходимые поля: дату ремонта компьютера, номер сотрудника, статус ремонта, а потом нажать на кнопку «ОК» (рисунок 43). Результат добавления записи представлен на рисунке 44.

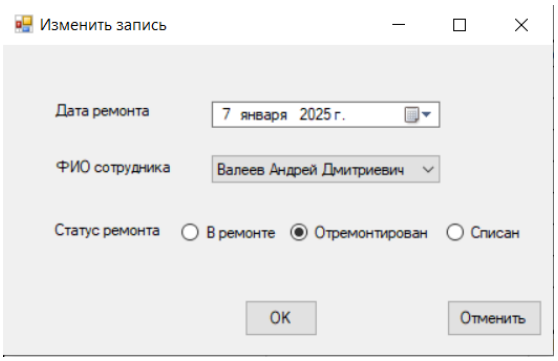


Рисунок 43 – Добавление записи в таблицу «Ремонт компьютеров»

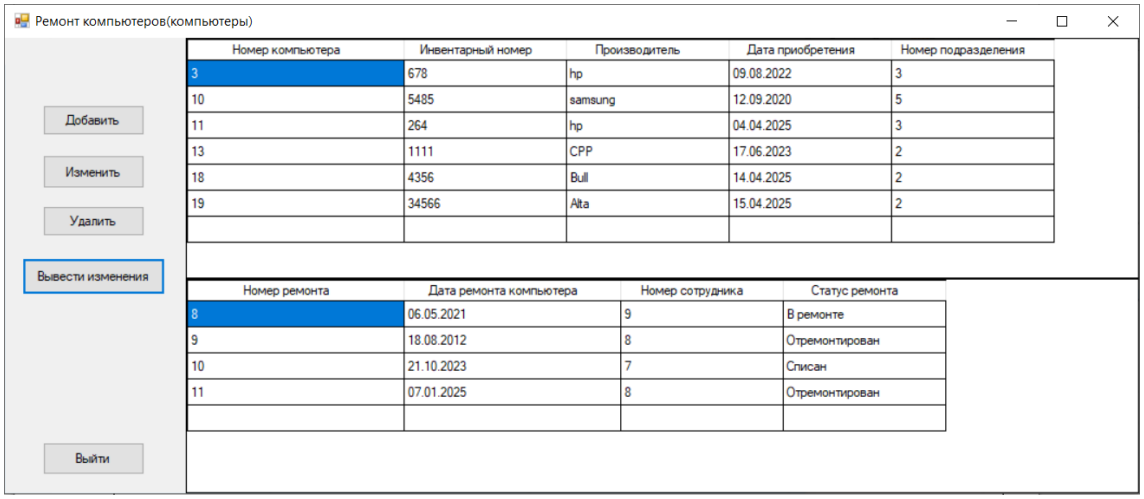


Рисунок 44 – Результат добавления данных в таблицу «Ремонт компьютеров»

Для изменения записи в таблице «Ремонт компьютеров» нужно выбрать запись и нажать на кнопку «Изменить». Например надо выбрать запись с номером 11. Далее открывается форма изменения этой записи в таблице. (рисунок 45). Затем происходит изменение данных полей выбранной записи. После этого нужно нажать на кнопку «ОК» (рисунок 46). Результат изменения записи с номером 11 в таблице «Ремонт компьютеров» представлен на рисунке 47.

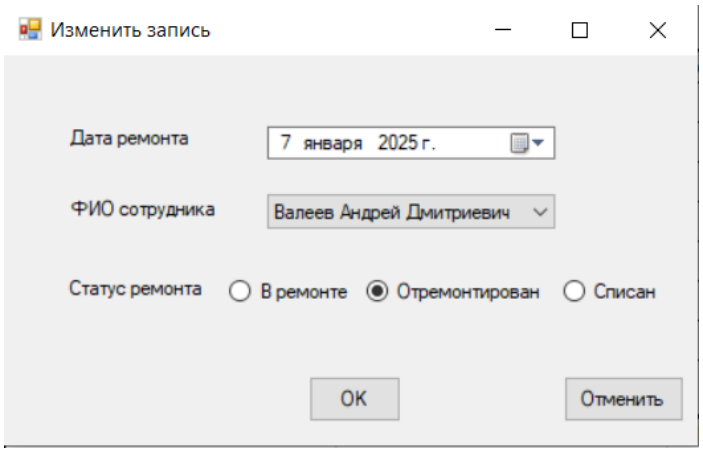


Рисунок 45 – Окно изменения записи в таблице «Ремонт компьютеров»

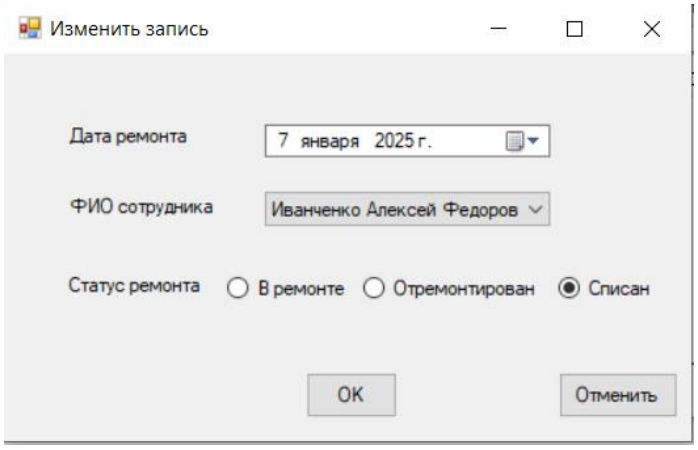


Рисунок 46 – Изменение записи в таблице «Ремонт компьютеров»

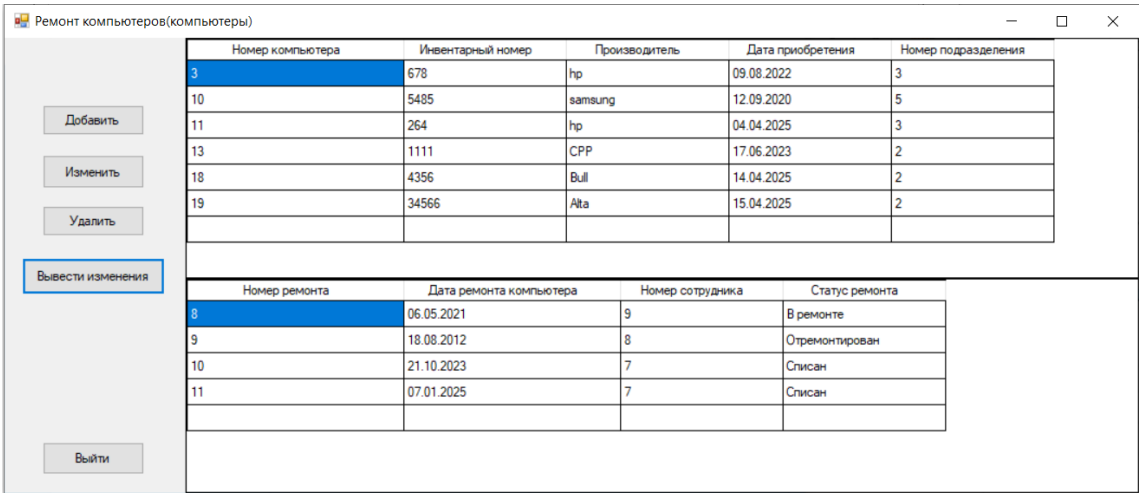


Рисунок 47 – Результат изменения записи в таблице «Ремонт компьютеров»

2 Охрана труда

Современное законодательство в области охраны труда при работе с электронно-вычислительными машинами (ЭВМ) выдвигает следующие требования:

Помещение, в котором осуществляется работа с ПК, должно соответствовать следующим требованиям:

– освещение должно соответствовать стандартам, а в случае отсутствия естественного освещения должны быть предоставлены расчеты, доказывающие безопасность работы без него;

– оконные проемы должны иметь регулируемые устройства, такие как жалюзи или занавеси;

– компьютеры не должны размещаться в местах, где есть риск попадания влаги;

– рабочие места с компьютерами должны быть оборудованы защитным заземлением;

– в помещениях с компьютерами следует проводить ежедневную влажную уборку и проветривание каждый час;

– рекомендуемая температура воздуха в помещении с компьютерами составляет +19°С - +21°С;

– рабочий стол должен быть спроектирован таким образом, чтобы обеспечить оптимальное расположение используемого оборудования в соответствии с выполняемой работой, разрешается использование различных типов рабочих столов, соответствующих современным стандартам эргономики;

– в помещениях с компьютерами следует проводить ежедневную влажную уборку и проветривание каждый час.

Рабочие места с персональными компьютерами должны быть организованы следующим образом:

– при организации рабочих мест с компьютерами необходимо учитывать, что расстояние между рабочими столами, на которых установлены видеомониторы, должно быть не менее 2,0 метров в направлении между ними и не менее 1,2 метра между боковыми поверхностями видеомониторов;

– размещение компьютеров должно предусматривать расстояние не менее 1 метра от источников тепла. Следует избегать прямого попадания солнечного света на них;

– запрещается размещать рабочие места с компьютерами рядом с силовыми кабелями, высоковольтными трансформаторами и другим оборудованием, способным создавать помехи в работе компьютеров;

– при установке компьютеров следует избегать прохождения интерфейсных и электропитания кабелей по полу. В случае необходимости, их следует закрыть защитными коробками;

– экран видеомонитора должен находиться от глаз пользователя на расстоянии 600-700 мм, но не ближе 500 мм с учетом размеров алфавитно-цифровых знаков и символов;

– системный блок и монитор следует разместить так, чтобы расстояние от вентиляционных отверстий этих устройств до стены (или стенки мебели) составляло не менее 10 сантиметров;

– искусственное освещение в помещениях для эксплуатации компьютеров должно осуществляться системой общего равномерного освещения. Освещенность на поверхности рабочего стола в зоне размещения ПК должна быть 300-500 люкс. Освещение не должно создавать бликов на поверхности экрана. Освещенность поверхности экрана не должна быть более 300 люкс.

В случае работы на компьютере в течение 8-часовой смены следует предусматривать перерывы в соответствии с категорией работ:

- для работников I категории перерывы устанавливаются через 2 часа после начала смены и через 2 часа после обеденного перерыва продолжительностью 15 минут каждый;

- работники II категории должны иметь перерыв через 2 часа после начала смены и через 1,5 - 2,0 часа после обеденного перерыва продолжительностью 15 минут каждый или продолжительностью 10 минут через каждый час работы;

- для работников III категории рекомендуется устанавливать перерыв через 1,5 - 2,0 часа после начала смены и также через 1,5 - 2,0 часа после обеденного перерыва продолжительностью 20 минут каждый или продолжительностью 15 минут через каждый час работы.

При 12-часовой смене регламентированные перерывы должны быть предусмотрены в первые 8 часов работы так же, как и при 8-часовой смене. В течение последних 4 часов работы, независимо от категории и вида работ, необходимо предусматривать перерыв каждый час продолжительностью 15 минут (таблица 12).

Таблица 12 – Время регламентированных перерывов в зависимости от категории работы

Категория работы	Уровень нагрузки за рабочую смену при видах работ с компьютером			Суммарное время регламентированных перерывов, мин.	
	группа А, количество знаков	группа Б, количество знаков	группа В, час.	при 8-часовой смене	при 12-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

Заключение

В ходе выполнения дипломного проекта была изучена предметная область, а именно организация учёта и контроля ремонта и хранения компьютерной техники в воинской части. На основе анализа текущих процессов был составлен план автоматизации учёта и сформулировано техническое задание на разработку базы данных и программного приложения для управления этой базой.

В качестве системы управления базами данных была выбрана СУБД MySQL, обеспечивающая надёжное хранение информации и доступ к ней. Для реализации прикладного программного обеспечения использован язык программирования C#, что позволило создать удобный и понятный пользовательский интерфейс на основе технологии Windows Forms.

Разработанное приложение полностью соответствует требованиям технического задания: реализованы функции добавления, редактирования и удаления информации о компьютерной технике, учёта ремонтов, регистрации сотрудников и учёта использования комплектующих. Все операции обеспечиваются безопасным доступом к базе данных MySQL.

В экономическом разделе дипломного проекта была произведена оценка затрат на разработку и внедрение системы учёта, а также обоснована её эффективность для повышения качества обслуживания техники и оптимизации работы сотрудников.

В разделе по безопасности жизнедеятельности рассмотрены основные факторы, влияющие на безопасность работы оператора программного обеспечения, и даны рекомендации по организации рабочего места и соблюдению норм охраны труда.

Задание на дипломное проектирование выполнено в полном объеме.

Список использованных источников

1. Сиддхартха Рао. Освой самостоятельно C++ за 21 день. 7-е издание. – М.: Издательство «Вильямс», 2013. – 685 с.

2. Доусон М. Программируем на Python. –СПб.: Питер, 2014. – 416 с.

3. Хорстманн, Кей С., Корнелл, Гари. Java – Библиотека профессионала, том 1. Основы. 9-е изд.: Пер. с англ. – М: ООО «И.Д. Вильямс», 2014. – 864 с.

4. Лафоре Р. Структуры данных и алгоритмы в Java. 2-е изд. – СПб: Питер, 2013. – 704 с.

5. Рогов Е. В. PostgreSQL 16 изнутри. – М.: ДМК Пресс, 2024. – 664 с.

6. Официальная документация Apache NetBeans 21. [Электронный ресурс] / Режим доступа: https://netbeans.apache.org/front/main/about/, свободный (дата обращения 17.04.2024).

7. Официальная документация по среде IntelliJ IDEA. [Электронный ресурс] / Режим доступа: https://www.jetbrains.com/help/idea/javadocs.html/, свободный (дата обращения 10.05.2024).

8. Иван Портянкин. Swing: Эффективные пользовательские интерфейсы. – М.: Издательство «Лори», 2011. – 591 с.

9. Официальная документация JavaFX. [Электронный ресурс] / Режим доступа: https://www.oracle.com/java/technologies/javase/javafx-docs.html/, свободный (дата обращения 15.05.2024).

10. Официальная документация Introducing JSON. [Электронный ресурс] / Режим доступа: https://www.json.org/json-en.html/, свободный (дата обращения 07.05.2024).

11. Описание шаблона проектирования MVC. [Электронный ресурс] / Режим доступа: https://pub.dev/documentation/mvc_pattern/latest/, свободный (дата обращения 11.05.2024).

12. Инструкция по охране труда при работе с ЭВМ. [Электронный ресурс] / Режим доступа: https://kpfu.ru/portal/docs/F_1086113514/1815.pdf/, свободный (дата обращения 12.05.24).

Приложение А

(Обязательное)

Информационная автоматизированная система учета ремонта и хранения компьютерной техники в воинской части

UML диаграммы

Баталер
Работа с данными о подразделениях
Работа с данными о сотрудниках
Работа с данными о компьютерной технике
Работа с данными о ремонтах компьютерной техники
Работа с данными о комплектующих для компьютерной техники
удаление данных
изменение данных
добавление данных
«extend»

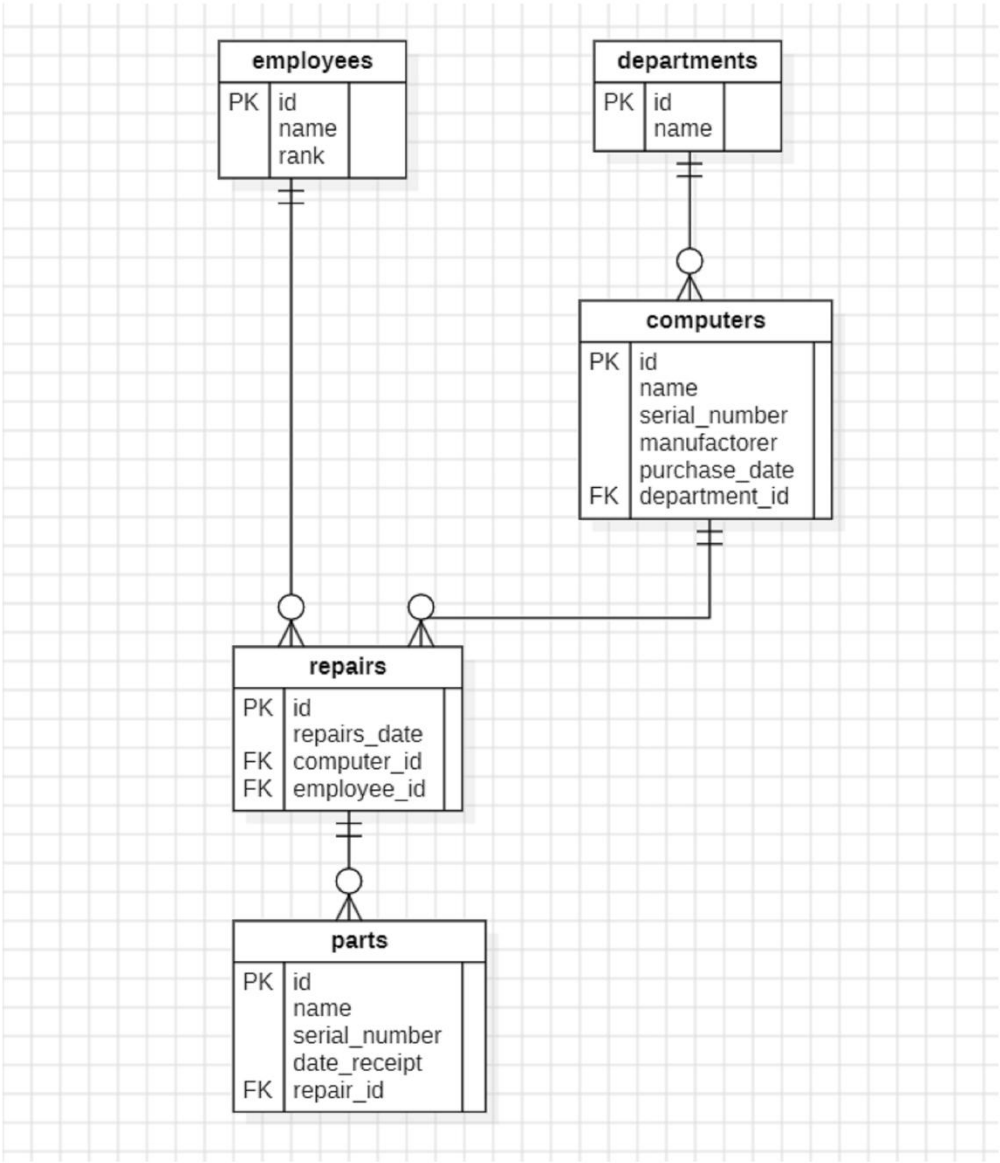
ПГУ 109050101.001 ПЗ

Информационная автоматизированная система учета личного состава воинской части. Пояснительная записка

Стадия | Лист | Листов
105

ФВТ каф. ВТ гр. 208031

Формат А1



ПГУ 1090501.01.001 ПЗ

Разраб. выполнил Р.Н.
Провер. Дунаев С.А.
Н. контр. Дунаев А.С.
Зав.каф. Петрова Л.А.

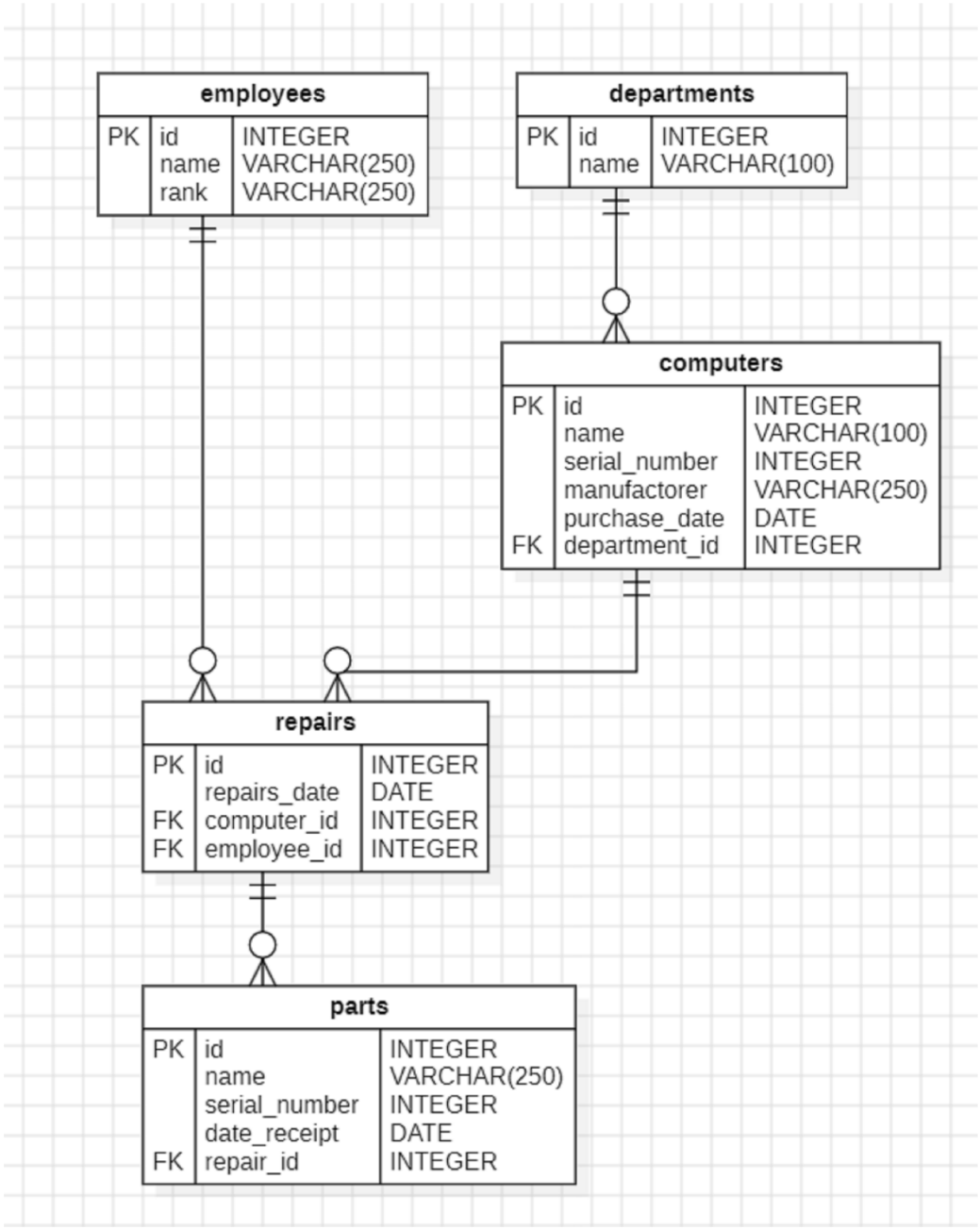
Информационная автоматизированная система учета ремонта и замены комплектующих персонального компьютера

Стадия | Лист | Листов
105

ФВТ каф. ВТ
гр. 20В031

Формат А1

Копирование документа разрешено только с разрешения автора работы. При цитировании ссылка на источник обязательна. Не для коммерческого использования



Копирование документа без разрешения запрещено. Все права защищены
Не для коммерческого использования

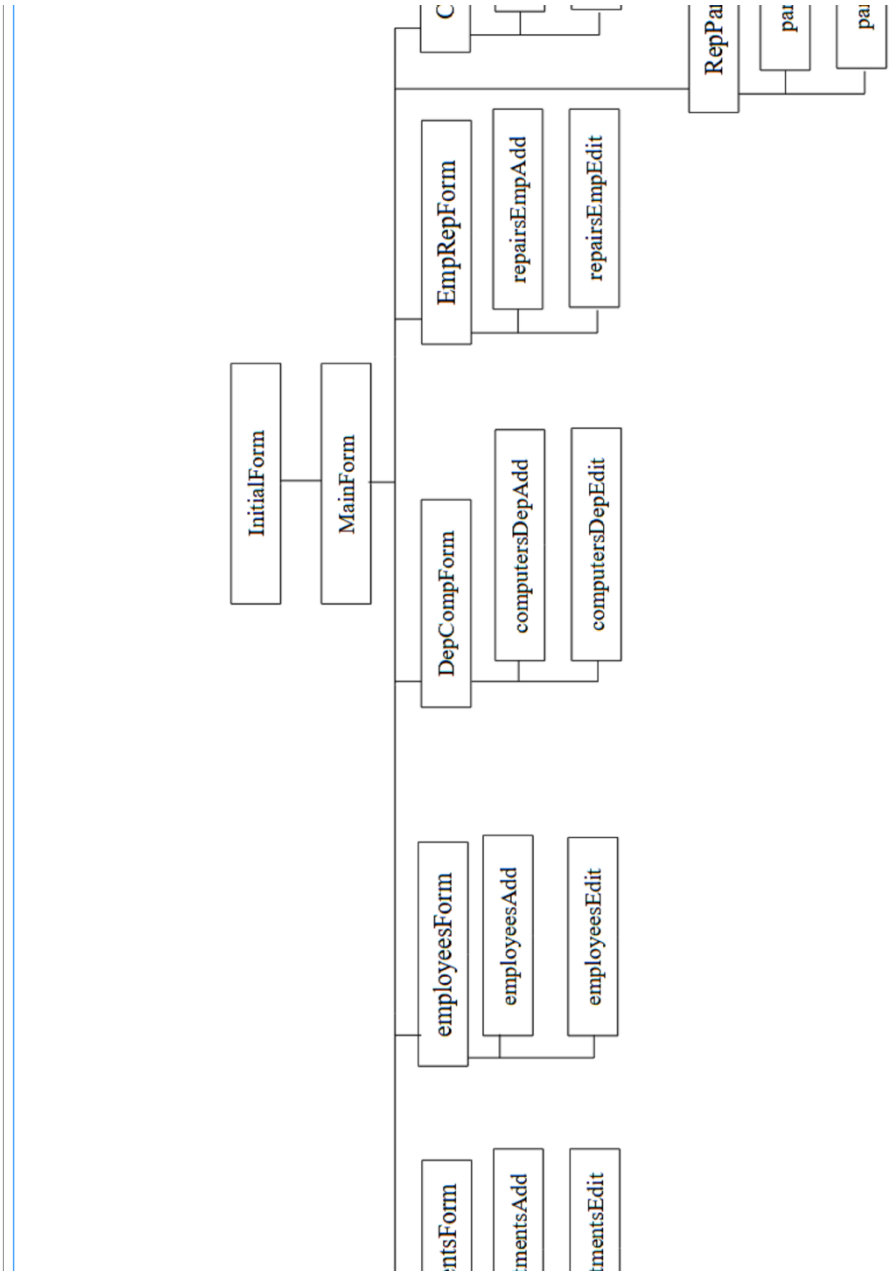
ПГУ 1.090501.01.001 ПЗ

Информационная автоматизированная система учета ремонта и оценки комплектной техники в Военной части Приволжского округа

Лит. | Лист | Листов
105

ФВТ, каф. ВТ
гр. 20В031

Формат А1



Приложение Б

(Обязательное)

Информационная автоматизированная система учета ремонта и хранения компьютерной техники в воинской части

Листинг программы

Файл InitialForm.cs:

```
using System;
using System.Windows.Forms;

namespace MilComp
{
    public partial class InitialForm : Form
    {
        private const string CorrectLogin = "admin";
        private const string CorrectPassword =
"fjirWEU258!@DFKiewiy";
        public InitialForm()
        {
            InitializeComponent();
        }

        private void buOK_Click(object sender, EventArgs e)
        {
            if (txtUsername.Text == CorrectLogin &&
txtPassword.Text == CorrectPassword)
            {
                MainForm fmMain = new MainForm();
                fmMain.Show();

                this.Hide();
            }
            else
            {
                MessageBox.Show("Неправильный логин или
пароль!", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }

        private void buCancel_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }
    }
}
```

Файл MainForm.cs:

```
using System;
using System.Windows.Forms;

namespace MilComp
{
    public partial class MainForm : Form
```

```
    {
        public MainForm()
        {
            InitializeComponent();
        }


        private void выходToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            Application.Exit();
        }

        private void сотрудникиToolStripMenuItem_Click(object
sender, EventArgs e)
        {
            employeesForm employeesForm = new employeesForm();
            employeesForm.Show();
        }

        private void подразделенияToolStripMenuItem_Click(object
sender, EventArgs e)
        {
            departmentsForm departmentsForm = new
departmentsForm();
            departmentsForm.Show();
        }

        private void сотрудникиКToolStripMenuItem_Click(object
sender, EventArgs e)
        {
            DepCompForm depCompForm = new DepCompForm();
            depCompForm.Show();
        }

        private void
компьютерыРемонтКомпьютеровToolStripMenuItem_Click(object
sender, EventArgs e)
        {
            CompRepForm empCompForm = new CompRepForm();
            empCompForm.Show();
        }

        private void pictureBox1_Click(object sender, EventArgs
e)
        {

        }
```

```
        private void
запчастиремонтКомпьютеровToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            RepPartsForm repPartsForm = new RepPartsForm();
            repPartsForm.Show();
        }
    }
}
```

Файл employeesForm.cs:

```
using System;
using System.Data;
using System.Windows.Forms;

namespace MilComp
{
    public partial class employeesForm : Form
    {
        public employeesForm()
        {
            InitializeComponent();
        }

        public void FillDGV()
        {
            command = new
MySql.Data.MySqlClient.MySqlCommand("SELECT * FROM employees",
conn);
            adapter = new
MySql.Data.MySqlClient.MySqlDataAdapter(command);
            DataTable table = new DataTable();
            adapter.Fill(table);
            dg1.DataSource = table;
            dg1.ReadOnly = true;
            dg1.SelectionMode =
DataGridViewSelectionMode.FullRowSelect;
            dg1.RowHeadersVisible = false;
            dg1.Columns["id"].HeaderText = "Номер сотрудника";
            dg1.Columns["name"].HeaderText = "ФИО сотрудника";
            dg1.Columns["rank"].HeaderText = "Должность
сотрудника";
            dg1.Columns["id"].Width = 150;
            dg1.Columns["name"].Width = 200;
            dg1.Columns["rank"].Width = 150;

            foreach (DataGridViewColumn column in dg1.Columns)
            {
                column.HeaderCell.Style.Alignment =
DataGridViewContentAlignment.MiddleCenter;
```

```
            }

        }

        private void buExit_Click(object sender, EventArgs e)
        {
            if (conn != null && conn.State !=
ConnectionState.Open)
            {
                conn.Close();
                conn.Dispose();
            }

            this.Close();
        }

        private void buAdd_Click(object sender, EventArgs e)
        {
            employeesAdd addForm = new employeesAdd();

            if (addForm.ShowDialog() == DialogResult.OK )
            {
                FillDGV();
            }
        }

        private void employeesForm_Load(object sender, EventArgs
e)
        {
            FillDGV();
        }

        private void buRefresh_Click(object sender, EventArgs e)
        {
            FillDGV();
        }

        private void buDelete_Click(object sender, EventArgs e)
        {
            DialogResult res;
            res = MessageBox.Show("Вы хотите удалить данные из
таблицы Сотрудники?", "Удаление данных",
MessageBoxButtons.YesNo, MessageBoxIcon.Question);
            if (res == DialogResult.Yes)
            {
                string idLocRemv =
dg1.SelectedRows[0].Cells[0].Value.ToString();
                string removeVolCred = "DELETE FROM employees
WHERE id = " + idLocRemv;
                using (command = new
MySql.Data.MySqlClient.MySqlCommand(removeVolCred, conn))
```

```
                {
                    conn.Open();
                    command.ExecuteNonQuery();
                    conn.Close();
                }
                FillDGV();
            }
        }

        private void buEdit_Click(object sender, EventArgs e)
        {
            if (dg1.SelectedRows.Count > 0)
            {
                DataGridViewRow selectedRow =
dg1.SelectedRows[0];

                employeesEdit employeesEdit = new employeesEdit(

Convert.ToInt32(selectedRow.Cells["id"].Value),
                    selectedRow.Cells["rank"].Value.ToString(),
                    selectedRow.Cells["name"].Value.ToString()
                    );
                if (employeesEdit.ShowDialog() ==
DialogResult.OK)
                {
                    FillDGV();
                }
            }
            else
            {
                MessageBox.Show("Не выбрана ни одна строка
таблицы Сотрудники!");
            }
        }
    }
}
```

Файл employeesAdd.cs:

```
using System;
using System.Windows.Forms;

namespace MilComp
{
    public partial class employeesAdd : Form
    {
        public employeesAdd()
        {
            InitializeComponent();
        }
```

```
        private void buCancel_Click(object sender, EventArgs e)
        {
            DialogResult res;
            res = MessageBox.Show("Вы хотите закрыть окно
добавления данных в таблицу Сотрудники?", "Выйти",
MessageBoxButtons.YesNo, MessageBoxIcon.Question);
            if (res == DialogResult.Yes)
            {
                this.Close();
            }
            else
            {
                this.Show();
            }
        }

        private void buOK_Click(object sender, EventArgs e)
        {
            using (command = new
MySql.Data.MySqlClient.MySqlCommand("INSERT INTO `employees`
(`name`, `rank`) VALUES (@name,@rank);", conn))
            {
                DialogResult res;
                res = MessageBox.Show("Вы хотите добавить данные
в таблицу Сотрудники?", "Добавление данных",
MessageBoxButtons.YesNo, MessageBoxIcon.Question);
                if (res == DialogResult.Yes)
                {
                    command.Parameters.Add("@name",
MySql.Data.MySqlClient.MySqlDbType.VarChar, 255).Value =
txtName.Text;
                    command.Parameters.Add("@rank",
MySql.Data.MySqlClient.MySqlDbType.VarChar, 50).Value =
txtRank.Text;
                    conn.Open();
                    command.ExecuteNonQuery();
                    conn.Close();
                    this.Close();
                }
                else
                {
                    this.Show();
                }
            }
        }

        private void CallFunc(object sender)
        {
            if (sender is employeesForm form)
            {
                form.FillDGV();
```

```
            }
        }
    }

}
```

Файл employeesEdit.cs:

```
using System;
using System.Windows.Forms;

namespace MilComp
{
    public partial class employeesEdit : Form
    {
        public int employee_id { get; set; }
        public string employee_name { get; set; }
        public string employee_rank { get; set; }
        public employeesEdit(int emp_id, string emp_name, string
emp_rank)
        {
            InitializeComponent();
            employee_id = emp_id;
            txtName.Text = emp_name;
            txtRank.Text = emp_rank;
        }

        private void buOK_Click(object sender, EventArgs e)
        {
            string query = "UPDATE `employees` SET `name`=@name,
`rank`=@rank WHERE (`employees`.`id` = @id);";
            using (command = new
MySql.Data.MySqlClient.MySqlCommand(query, conn))
            {
                command.Parameters.Add("@name",
MySql.Data.MySqlClient.MySqlDbType.VarChar, 255).Value =
txtName.Text;
                command.Parameters.Add("@rank",
MySql.Data.MySqlClient.MySqlDbType.VarChar, 255).Value =
txtRank.Text;
                command.Parameters.Add("@id",
MySql.Data.MySqlClient.MySqlDbType.Int32).Value = employee_id;
                conn.Open();
                command.ExecuteNonQuery();
                conn.Close();
            }
            this.Close();
        }
    }
```

Файл departmentsForm.cs:

```
using System;
using System.Data;
using System.Windows.Forms;

namespace MilComp
{
    public partial class departmentsForm : Form
    {
        public departmentsForm()
        {
            InitializeComponent();
        }

        public void FillDGV()
        {
            command = new
MySql.Data.MySqlClient.MySqlCommand("SELECT * FROM departments",
conn);
            adapter = new
MySql.Data.MySqlClient.MySqlDataAdapter(command);
            DataTable table = new DataTable();
            adapter.Fill(table);
            dg1.DataSource = table;
            dg1.ReadOnly = true;
            dg1.SelectionMode =
DataGridViewSelectionMode.FullRowSelect;
            dg1.RowHeadersVisible = false;
            dg1.Columns["id"].HeaderText = "Номер
подразделения";
            dg1.Columns["name"].HeaderText = "Название";
            dg1.Columns["id"].Width = 150;
            dg1.Columns["name"].Width = 200;

            foreach (DataGridViewColumn column in dg1.Columns)
            {
                column.HeaderCell.Style.Alignment =
DataGridViewContentAlignment.MiddleCenter;
            }

        }

        private void departmentsForm_Load(object sender,
EventArgs e)
        {

            FillDGV();
        }

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

```
        {
            FillDGV();
        }

        private void buExit_Click(object sender, EventArgs e)
        {
            if (conn != null && conn.State !=
ConnectionState.Open)
            {
                conn.Close();
                conn.Dispose();
            }

            this.Close();
        }

        private void buDelete_Click(object sender, EventArgs e)
        {
            DialogResult res;
            res = MessageBox.Show("Вы хотите удалить данные из
таблицы Подразделения?", "Удаление данных",
MessageBoxButtons.YesNo, MessageBoxIcon.Question);
            if (res == DialogResult.Yes)
            {
                string idLocRemv =
dg1.SelectedRows[0].Cells[0].Value.ToString();
                string removeVolCred = "DELETE FROM departments
WHERE id = " + idLocRemv;
                using (command = new
MySql.Data.MySqlClient.MySqlCommand(removeVolCred, conn))
                {
                    conn.Open();
                    command.ExecuteNonQuery();
                    conn.Close();
                }
            }
        }

        private void buAdd_Click(object sender, EventArgs e)
        {
            departmentsAdd departmentsAdd = new
departmentsAdd();
            if (departmentsAdd.ShowDialog() == DialogResult.OK)
            {
                FillDGV();
            }
        }

        private void buEdit_Click(object sender, EventArgs e)
        {
            if (dg1.SelectedRows.Count > 0)
```

```
            {
                DataGridViewRow selectedRow =
dg1.SelectedRows[0];

                departmentsEdit departmentsEdit = new
departmentsEdit(

Convert.ToInt32(selectedRow.Cells["id"].Value),
                    selectedRow.Cells["name"].Value.ToString()
                    );
                if (departmentsEdit.ShowDialog() ==
DialogResult.OK)
                {
                    FillDGV();
                }
            }
            else
            {
                MessageBox.Show("Не выбрана ни одна строка
таблицы Подразделения!");
            }
        }
    }
}
```

Файл departmentsAdd.cs:

```
using System;
using System.Windows.Forms;

namespace MilComp
{
    public partial class departmentsAdd : Form
    {
        public departmentsAdd()
        {
            InitializeComponent();
        }

        private void buCancel_Click(object sender, EventArgs e)
        {
            DialogResult res;
            res = MessageBox.Show("Вы хотите закрыть окно
добавления данных в таблицу Подразделения?", "Выйти",
MessageBoxButtons.YesNo, MessageBoxIcon.Question);
            if (res == DialogResult.Yes)
            {
                this.Close();
            }
            else
            {
```

```
                this.Show();
            }
        }

        private void buOK_Click(object sender, EventArgs e)
        {
            using (command = new
MySql.Data.MySqlClient.MySqlCommand("INSERT INTO
departments(name) VALUES (@name);", conn))
            {
                DialogResult res;
                res = MessageBox.Show("Вы хотите добавить данные
в таблицу Подразделения?", "Добавление данных",
MessageBoxButtons.YesNo, MessageBoxIcon.Question);
                if (res == DialogResult.Yes)
                {
                    command.Parameters.Add("@name",
MySql.Data.MySqlClient.MySqlDbType.VarChar, 255).Value =
txtName.Text;
                    conn.Open();
                    command.ExecuteNonQuery();
                    conn.Close();
                    this.Close();
                }
                else
                {
                    this.Show();
                }
            }
        }
    }
}
```

Файл departmentsEdit.cs:

```
using System;
using System.Windows.Forms;

namespace MilComp
{
    public partial class departmentsEdit : Form
    {
        public int depart_id { get; set; }
        public string depart_name { get; set; }
        public departmentsEdit(int dep_id, string dep_name)
        {
            InitializeComponent();
            depart_id = dep_id;
            txtName.Text = dep_name;

        }
```

```
        private void buOK_Click(object sender, EventArgs e)
        {
            string query = "UPDATE `departments` SET
`name`=@name WHERE (`departments`.`id` = @id)";
            using (command = new
MySql.Data.MySqlClient.MySqlCommand(query, conn))
            {
                command.Parameters.Add("@name",
MySql.Data.MySqlClient.MySqlDbType.VarChar, 255).Value =
txtName.Text;
                command.Parameters.Add("@id",
MySql.Data.MySqlClient.MySqlDbType.Int32).Value = depart_id;
                conn.Open();
                command.ExecuteNonQuery();
                conn.Close();
            }
            this.Close();
        }
    }
}
```

Файл DepCompForm.cs:

```
using MySql.Data.MySqlClient;
using System;
using System.Data;
using System.Drawing;
using System.Windows.Forms;

namespace MilComp
{
    public partial class DepCompForm : Form
    {
        private BindingSource masterBindingSource = new
BindingSource();
        private BindingSource detailsBindingSource = new
BindingSource();
        private DataSet data;
        public DepCompForm()
        {
            InitializeComponent();
        }

        private void GetData()
        {
            string masterQuery = "SELECT * FROM departments";
            string detailQuery = "SELECT * FROM computers";
```

```
            masterDataAdapter = new
MySqlDataAdapter(masterQuery, mycon);
            detailsDataAdapter = new
MySqlDataAdapter(detailQuery, mycon);

            DataSet ds = new DataSet();

            masterDataAdapter.Fill(ds, "departments");
            detailsDataAdapter.Fill(ds, "computers");

            DataColumn parentColumn =
ds.Tables["departments"].Columns["id"];
            DataColumn childColumn =
ds.Tables["computers"].Columns["department_id"];

            DataRelation relation = new
DataRelation("ПодразделенияКомпьютеры", parentColumn,
childColumn);
            ds.Relations.Add(relation);

            masterBindingSource = new BindingSource();
            detailsBindingSource = new BindingSource();

            masterBindingSource.DataSource = ds;
            masterBindingSource.DataMember = "departments";

            detailsBindingSource.DataSource =
masterBindingSource;
            detailsBindingSource.DataMember =
"ПодразделенияКомпьютеры";

            dg1.DataSource = masterBindingSource;
            dg2.DataSource = detailsBindingSource;
        }

        public void FillData()
        {
            GetData();
            dg1.RowHeadersVisible = false;
            dg2.RowHeadersVisible = false;
            dg1.Columns["id"].HeaderText = "Номер";
            dg1.Columns["name"].HeaderText = "Название
подразделения";
            dg1.Columns["id"].Width = 200;
            dg1.Columns["name"].Width = 200;
            dg2.Columns["id"].HeaderText = "Номер";
            dg2.Columns["serial_number"].HeaderText =
"Инвентарный номер";
            dg2.Columns["manufactorer"].HeaderText =
"Производитель";
```

```
            dg2.Columns["purchase_date"].HeaderText = "Дата
приобретения";
            dg2.Columns["department_id"].HeaderText = "Номер
подразделения";
            dg1.Columns["id"].Width = 300;
            dg2.Columns["serial_number"].Width = 150;
            dg2.Columns["manufactorer"].Width = 150;
            dg2.Columns["purchase_date"].Width = 200;
            dg2.Columns["department_id"].Width = 200;
            dg1.ReadOnly = false;
            dg2.ReadOnly = false;
            foreach (DataGridViewColumn column in dg1.Columns)
            {
                column.HeaderCell.Style.Alignment =
DataGridViewContentAlignment.MiddleCenter;
            }
            foreach (DataGridViewColumn column in dg2.Columns)
            {
                column.HeaderCell.Style.Alignment =
DataGridViewContentAlignment.MiddleCenter;
            }
            dg1.BorderStyle = BorderStyle.FixedSingle;
            dg2.BorderStyle = BorderStyle.FixedSingle;
            dg1.ColumnHeadersBorderStyle =
DataGridViewHeaderBorderStyle.Single;
            dg2.ColumnHeadersBorderStyle =
DataGridViewHeaderBorderStyle.Single;
            dg1.GridColor = Color.Black;
            dg2.GridColor = Color.Black;
            dg2.Columns["department_id"].Visible = false;
            bn1.DataSource = masterBindingSource;
            bn2.DataSource = detailsBindingSource;
        }

        private void DepCompForm_Load(object sender, EventArgs
e)
        {
            FillData();
        }

        private void buRefresh_Click(object sender, EventArgs e)
        {
            FillData();
        }

        private void buDelete_Click(object sender, EventArgs e)
        {

           if (dg2.CurrentCell != null)
             {
                int rowIndex = dg2.CurrentCell.RowIndex;
```

```
                DataGridViewRow selectedRow =
dg2.Rows[rowIndex];
                int idd =
Convert.ToInt32(selectedRow.Cells[0].Value);

                var confirmResult = MessageBox.Show("Удалить эту
запись?", "Подтверждение", MessageBoxButtons.YesNo);
                if (confirmResult == DialogResult.Yes)
                {
                    DeleteRecordFromDatabase(idd);
                }

              }

        }

        private void DeleteRecordFromDatabase(int id)
        {
            mycon.Open();
            string query = "DELETE FROM `computers` WHERE
(`computers`.`id` = @id);";

            using (command = new MySqlCommand(query, mycon))
            {
                command.Parameters.AddWithValue("@id", id);
                command.ExecuteNonQuery();
            }
            mycon.Close();
            FillData();
        }

        private void buAdd_Click(object sender, EventArgs e)
        {
           if (dg1.SelectedCells.Count > 0)
            {
                DataGridViewCell selectedCell =
dg1.SelectedCells[0];
                DataGridViewRow selectedRow =
selectedCell.OwningRow;
                int firstFielValue =
Convert.ToInt32(selectedRow.Cells[0].Value);
                computersDepAdd computersDepAdd = new
computersDepAdd(firstFielValue);
                computersDepAdd.Show();
            }
            else
            {
                MessageBox.Show("Не выбрана ни одна запись
главной таблицы Подразделения для вставки записи в подчиненную
таблицу Компьютеры!");
            }
```

```
        }

        private void button1_Click(object sender, EventArgs e)
        {
            if (mycon != null && mycon.State !=
ConnectionState.Open)
            {
                mycon.Close();
                mycon.Dispose();
            }

            this.Close();
        }

        private void buEdit_Click(object sender, EventArgs e)
        {

            if (dg2.CurrentCell != null)
            {
                int rowIndex = dg2.CurrentCell.RowIndex;
                var onecellvalue =
dg2.Rows[rowIndex].Cells[0].Value;
                var twocellvalue =
dg2.Rows[rowIndex].Cells[1].Value;
                var threecellvalue =
dg2.Rows[rowIndex].Cells[2].Value;
                var fourcellvalue =
dg2.Rows[rowIndex].Cells[3].Value;
                var fivecellvalue =
dg2.Rows[rowIndex].Cells[4].Value;
                int onevalue = Convert.ToInt32(onecellvalue);
                int twovalue = Convert.ToInt32(twocellvalue);
                string threevalue =
Convert.ToString(threecellvalue);
                DateTime fourvalue =
Convert.ToDateTime(fourcellvalue);
                int fivevalue = Convert.ToInt32(fivecellvalue);
                computersDepEdit computersDepEdit = new
computersDepEdit(onevalue, twovalue, threevalue, fourvalue,
fivevalue);
                if (computersDepEdit.ShowDialog() ==
DialogResult.OK)
                {
                    FillData();
                }
            }
            else
            {
                MessageBox.Show("Не выбрана ни одна строка
таблицы Компьютеры!");
```

```
            }
        }
    }
    }
```

Файл computersDepAdd.cs:

```
                using System;
using System.Windows.Forms;

namespace MilComp
{
    public partial class computersDepAdd : Form
    {
        public int depart_id { get; set; }
        public computersDepAdd(int dep_id)
        {
            InitializeComponent();
            depart_id = dep_id;
        }

        private void buOK_Click(object sender, EventArgs e)
        {
            using (command = new
MySql.Data.MySqlClient.MySqlCommand("INSERT INTO `computers`
(`serial_number`, `manufactorer`, `purchase_date`,
`department_id`) VALUES (@serial_number, @manufactorer,
@purchase_date, @department_id)", conn))
            {
                DialogResult res;
                res = MessageBox.Show("Вы хотите добавить данные
в таблицу Компьютеры?", "Добавление данных",
MessageBoxButtons.YesNo, MessageBoxIcon.Question);
                if (res == DialogResult.Yes)
                {

                    command.Parameters.Add("@serial_number",
MySql.Data.MySqlClient.MySqlDbType.VarChar, 50).Value =
txtIN.Text;
                    command.Parameters.Add("@manufactorer",
MySql.Data.MySqlClient.MySqlDbType.VarChar, 50).Value =
txtManuf.Text;
                    command.Parameters.Add("@purchase_date",
MySql.Data.MySqlClient.MySqlDbType.Date).Value = PD.Value;
                    command.Parameters.Add("@department_id",
MySql.Data.MySqlClient.MySqlDbType.Int32).Value = depart_id;
                    conn.Open();
                    command.ExecuteNonQuery();
                    conn.Close();
                    this.Close();
```

```
                }
                else
                {
                    this.Show();
                }
            }
        }

        private void buCancel_Click(object sender, EventArgs e)
        {
            DialogResult res;
            res = MessageBox.Show("Вы хотите закрыть окно
добавления данных в таблицу Подразделения?", "Выйти",
MessageBoxButtons.YesNo, MessageBoxIcon.Question);
            if (res == DialogResult.Yes)
            {
                this.Close();
            }
            else
            {
                this.Show();
            }
        }
    }
}
```

Файл computersDepEdit.cs:

```
                using System;
using System.Windows.Forms;

namespace MilComp
{
    public partial class computersDepEdit : Form
    {
        public int computerid { get; set; }
        public int serialnumber { get; set; }
        public string manufactorer { get; set; }
        public DateTime purchasedate { get; set; }
        public int departid { get; set; }

        public computersDepEdit(int ID, int serialnum, string
manuf, DateTime purchdate, int depid)
        {
            InitializeComponent();
            computerid = ID;
            txtIN.Text = serialnum.ToString();
            txtManuf.Text = manuf;
            PD.Value = purchdate;
            departid = depid;
```

```
        }

        private void buOK_Click(object sender, EventArgs e)
        {
            conn.Open();
            string query = "UPDATE `computers` SET
`serial_number`=@serial_number, `manufactorer`=@manufactorer,
`purchase_date`=@purchase_date, `department_id`=@department_id
WHERE (`computers`.`id` = @id);";
            using (command = new
MySql.Data.MySqlClient.MySqlCommand(query, conn))
            {
                command.Parameters.Add("@serial_number",
MySql.Data.MySqlClient.MySqlDbType.Int32).Value =
int.Parse(txtIN.Text);
                command.Parameters.Add("@manufactorer",
MySql.Data.MySqlClient.MySqlDbType.VarChar, 250).Value =
txtManuf.Text;
                command.Parameters.Add("@purchase_date",
MySql.Data.MySqlClient.MySqlDbType.Date).Value = PD.Value;
                command.Parameters.Add("@id",
MySql.Data.MySqlClient.MySqlDbType.Int32).Value = computerid;
                command.Parameters.Add("@department_id",
MySql.Data.MySqlClient.MySqlDbType.Int32).Value = departid;
                command.ExecuteNonQuery();
                conn.Close();
            }
            this.Close();
        }

        private void buCancel_Click(object sender, EventArgs e)
        {
            DialogResult res;
            res = MessageBox.Show("Вы хотите закрыть окно
добавления данных в таблицу Компьютеры?", "Выйти",
MessageBoxButtons.YesNo, MessageBoxIcon.Question);
            if (res == DialogResult.Yes)
            {
                this.Close();
            }
            else
            {
                this.Show();
            }
        }
    }
}
```

Файл CompRepForm.cs:

```
using System;
using System.Data;
using System.Drawing;
using System.Windows.Forms;
using MySql.Data.MySqlClient;

namespace MilComp
{
    public partial class CompRepForm : Form
    {

        private BindingSource masterBindingSource = new
BindingSource();
        private BindingSource detailsBindingSource = new
BindingSource();
        private DataSet data;
        public CompRepForm()
        {
            InitializeComponent();
        }

        private void GetData()
        {
            string masterQuery = "SELECT * FROM computers";
            string detailQuery = "SELECT * FROM repairs";

            masterDataAdapter = new
MySqlDataAdapter(masterQuery, mycon);
            detailsDataAdapter = new
MySqlDataAdapter(detailQuery, mycon);

            DataSet ds = new DataSet();

            masterDataAdapter.Fill(ds, "computers");
            detailsDataAdapter.Fill(ds, "repairs");

            DataColumn parentColumn =
ds.Tables["computers"].Columns["id"];
            DataColumn childColumn =
ds.Tables["repairs"].Columns["computer_id"];

            DataRelation relation = new
DataRelation("КомпьютерыРемонт", parentColumn, childColumn);
            ds.Relations.Add(relation);

            masterBindingSource = new BindingSource();
            detailsBindingSource = new BindingSource();

            masterBindingSource.DataSource = ds;
            masterBindingSource.DataMember = "computers";
```

```
            detailsBindingSource.DataSource =
masterBindingSource;
            detailsBindingSource.DataMember =
"КомпьютерыРемонт";

            dg1.DataSource = masterBindingSource;
            dg2.DataSource = detailsBindingSource;
        }

        public void FillData()
        {
            GetData();
            dg1.RowHeadersVisible = false;
            dg2.RowHeadersVisible = false;
            dg1.Columns["id"].HeaderText = "Номер компьютера";
            dg1.Columns["serial_number"].HeaderText =
"Инвентарный номер";
            dg1.Columns["manufactorer"].HeaderText =
"Производитель";
            dg1.Columns["purchase_date"].HeaderText = "Дата
приобретения";
            dg1.Columns["department_id"].HeaderText = "Номер
подразделения";
            dg1.Columns["id"].Width = 200;
            dg1.Columns["serial_number"].Width = 150;
            dg1.Columns["manufactorer"].Width = 150;
            dg1.Columns["purchase_date"].Width = 150;
            dg1.Columns["department_id"].Width = 150;
            dg2.Columns["id"].HeaderText = "Номер ремонта";
            dg2.Columns["repairs_date"].HeaderText = "Дата
ремонта компьютера";
            dg2.Columns["computer_id"].HeaderText = "Номер
компьютера";
            dg2.Columns["employee_id"].HeaderText = "Номер
сотрудника";
            dg2.Columns["statud"].HeaderText = "Статус ремонта";
            dg2.Columns["id"].Width=200;
            dg2.Columns["repairs_date"].Width =200;
            dg2.Columns["computer_id"].Width = 150;
            dg2.Columns["employee_id"].Width = 150;
            dg2.Columns["statud"].Width = 150;
            dg1.ReadOnly = false;
            dg2.ReadOnly = false;
            foreach (DataGridViewColumn column in dg1.Columns)
            {
                column.HeaderCell.Style.Alignment =
DataGridViewContentAlignment.MiddleCenter;
            }
            foreach (DataGridViewColumn column in dg2.Columns)
            {
```

```
                column.HeaderCell.Style.Alignment =
DataGridViewContentAlignment.MiddleCenter;
            }
            dg1.BorderStyle = BorderStyle.FixedSingle;
            dg2.BorderStyle = BorderStyle.FixedSingle;
            dg1.ColumnHeadersBorderStyle =
DataGridViewHeaderBorderStyle.Single;
            dg2.ColumnHeadersBorderStyle =
DataGridViewHeaderBorderStyle.Single;
            dg1.GridColor = Color.Black;
            dg2.GridColor = Color.Black;
            dg2.Columns["computer_id"].Visible = false;
            bn1.DataSource = masterBindingSource;
            bn2.DataSource = detailsBindingSource;
        }

        private void buExit_Click(object sender, EventArgs e)
        {
            if (mycon != null && mycon.State !=
ConnectionState.Open)
            {
                mycon.Close();
                mycon.Dispose();
            }

            this.Close();
        }

        private void EmpCompForm_Load(object sender, EventArgs
e)
        {
            FillData();
        }

        private void buRefresh_Click(object sender, EventArgs e)
        {
            FillData();
        }

        private void buAdd_Click(object sender, EventArgs e)
        {
            if (dg1.SelectedCells.Count > 0)
            {
                DataGridViewCell selectedCell =
dg1.SelectedCells[0];
                DataGridViewRow selectedRow =
selectedCell.OwningRow;
                int firstFielValue =
Convert.ToInt32(selectedRow.Cells[0].Value);
                repairsCompAdd repairsCompAdd = new
repairsCompAdd(firstFielValue);
```

```
                repairsCompAdd.Show();
            }
            else
            {
                MessageBox.Show("Не выбрана ни одна запись
главной таблицы Компьютеры для вставки записи в подчиненную
таблицу Ремонт компьютеров!");
            }
        }

        private void buEdit_Click(object sender, EventArgs e)
        {
            int rowIndex = dg2.CurrentCell.RowIndex;
            var onecellvalue =
dg2.Rows[rowIndex].Cells[0].Value;
            var twocellvalue =
dg2.Rows[rowIndex].Cells[1].Value;
            var threecellvalue =
dg2.Rows[rowIndex].Cells[2].Value;
            var fourcellvalue =
dg2.Rows[rowIndex].Cells[3].Value;
            var fivecellvalue =
dg2.Rows[rowIndex].Cells[4].Value;
            int onevalue = Convert.ToInt32(onecellvalue);
            DateTime twovalue =
Convert.ToDateTime(twocellvalue);
            int threevalue = Convert.ToInt32(threecellvalue);
            int fourvalue = Convert.ToInt32(fourcellvalue);
            string fivevalue = Convert.ToString(fivecellvalue);
            repairsCokpEdit repairsCokpEdit = new
repairsCokpEdit(onevalue, twovalue, threevalue, fourvalue,
fivevalue);
            repairsCokpEdit.Show();
        }
    }
}
```

Файл repairsCompAdd.cs:

```
using MySql.Data.MySqlClient;
using System;
using System.Data;
using System.Windows.Forms;

namespace MilComp
{
    public partial class repairsCompAdd : Form
```

```
    {
        public int comp_id { get; set; }
        public repairsCompAdd(int c_id)
        {
            InitializeComponent();
            comp_id = c_id;
        }


        private void repairsCompAdd_Load(object sender,
EventArgs e)
        {

            conn.Open();
            string query1 = "SELECT name FROM employees;";


            using (adapter = new MySqlDataAdapter(query1, conn))
            {
                DataTable dataTable = new DataTable();
                adapter.Fill(dataTable);

                foreach (DataRow row in dataTable.Rows)
                {
                    cb1.Items.Add(row["name"].ToString());
                }
            }
        }

        private string[] ParseEnumValues(string enumString)
        {
            try
            {
                enumString = enumString.Substring(5,
enumString.Length - 6);

                string[] values = enumString.Split(new[] { "','"
}, StringSplitOptions.None);
                for (int i = 0; i < values.Length; i++)
                {
                    values[i] = values[i].Trim('\'');
                }
                return values;
            }
            catch
            {
                return new string[0];
            }
        }

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

```
        {
            if ((cb1.SelectedItem != null))
            {
                string selectedName =
cb1.SelectedItem?.ToString();
                string queryName = "SELECT id FROM employees
WHERE name = @dopname LIMIT 1";
                int empid;
                using (cmd1 = new
MySql.Data.MySqlClient.MySqlCommand(queryName, conn))
                {
                    cmd1.Parameters.Add("@dopname",
MySql.Data.MySqlClient.MySqlDbType.VarChar, 250).Value =
selectedName;
                    object result = cmd1.ExecuteScalar();
                    empid = Convert.ToInt32(result); //
employees_id
                }


                string mainquery = "INSERT INTO `repairs`
(`repairs_date`, `computer_id`, `employee_id`, `statud`) VALUES
(@repairs_date, @computer_id, @employee_id, @statud);";
                using (cmd = new
MySql.Data.MySqlClient.MySqlCommand(mainquery, conn))
                {
                    string numStatud;

                    if (rb1.Checked)
                    {
                        numStatud = "В ремонте";
                        cmd.Parameters.Add("@statud",
MySql.Data.MySqlClient.MySqlDbType.Enum).Value = numStatud;
                    }
                    else if (rb2.Checked)
                    {
                        numStatud = "Отремонтирован";
                        cmd.Parameters.Add("@statud",
MySql.Data.MySqlClient.MySqlDbType.Enum).Value = numStatud;
                    }
                    else if (rb3.Checked)
                    {
                        numStatud = "Списан";
                        cmd.Parameters.Add("@statud",
MySql.Data.MySqlClient.MySqlDbType.Enum).Value = numStatud;
                    }


                    cmd.Parameters.Add("@repairs_date",
MySql.Data.MySqlClient.MySqlDbType.Date).Value = DP.Value;
```

```
                    cmd.Parameters.Add("@computer_id",
MySql.Data.MySqlClient.MySqlDbType.Int32).Value = comp_id;
                    cmd.Parameters.Add("@employee_id",
MySql.Data.MySqlClient.MySqlDbType.Int32).Value = empid;
                    cmd.ExecuteNonQuery();
                    conn.Close();
                    this.Close();
                }



            }
            else
            {
                MessageBox.Show("Не все значения выбраны!");
            }
        }
    }
}
```

Файл repairsCompEdit.cs:

```
using System;
using System.Data;
using System.Windows.Forms;
using MySql.Data.MySqlClient;

namespace MilComp
{
    public partial class repairsCokpEdit : Form
    {
        public int rep_id { get; set; }
        public DateTime rep_date { get; set; }
        public int comp_id { get; set; }
        public int emp_id { get; set; }
        public string stat { get; set; }
        public repairsCokpEdit(int repid, DateTime repdate, int
compid, int empid, string status)
        {
            InitializeComponent();
            rep_id = repid;
            rep_date = repdate;
            comp_id = compid;
            emp_id = empid;
            stat = status;

        }

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

```
        {
            conn.Open();
            string query1 = "SELECT name FROM employees;";

            DT.Value = rep_date; //repairs_date

            using (adapter = new MySqlDataAdapter(query1, conn))
            {
                DataTable dataTable = new DataTable();
                adapter.Fill(dataTable);

                foreach (DataRow row in dataTable.Rows)
                {
                    cb1.Items.Add(row["name"].ToString());
                }
            }
            string query2 = "SELECT name FROM employees WHERE id
= @id";
            cmd = new MySqlCommand(query2, conn);
            cmd.Parameters.Add("@id", MySqlDbType.Int32).Value =
emp_id;
            object result = cmd.ExecuteScalar();
            string str = Convert.ToString(result);
            cb1.SelectedItem = str; // employees_id

            if (stat == "В ремонте") //statud
            {
                rb1.Checked = true;
            }
            else if (stat == "Отремонтирован")
            {
                rb2.Checked = true;
            }
            else if(stat == "Списан")
            {
                rb3.Checked = true;
            }

        }

        private void buOK_Click(object sender, EventArgs e)
        {
            if ((cb1.SelectedItem != null))
            {
                string selectedName =
cb1.SelectedItem?.ToString();
                string queryName = "SELECT id FROM employees
WHERE name = @dopname LIMIT 1";
                int emplid;
                using (cmd1 = new
MySql.Data.MySqlClient.MySqlCommand(queryName, conn))
```

```
                {
                    cmd1.Parameters.Add("@dopname",
MySql.Data.MySqlClient.MySqlDbType.VarChar, 250).Value =
selectedName;
                    object result = cmd1.ExecuteScalar();
                    emplid = Convert.ToInt32(result); //
employees_id
                }


                string mainquery = "UPDATE `repairs` SET
`repairs_date`=@repairs_date, `computer_id`=@computer_id,
`employee_id`=@employee_id, `statud`=@statud WHERE
(`repairs`.`id` = @id);";
                using (cmd = new
MySql.Data.MySqlClient.MySqlCommand(mainquery, conn))
                {
                    string numStatud;

                    if (rb1.Checked)
                    {
                        numStatud = "В ремонте";
                        cmd.Parameters.Add("@statud",
MySql.Data.MySqlClient.MySqlDbType.Enum).Value = numStatud;
                    }
                    else if (rb2.Checked)
                    {
                        numStatud = "Отремонтирован";
                        cmd.Parameters.Add("@statud",
MySql.Data.MySqlClient.MySqlDbType.Enum).Value = numStatud;
                    }
                    else if (rb3.Checked)
                    {
                        numStatud = "Списан";
                        cmd.Parameters.Add("@statud",
MySql.Data.MySqlClient.MySqlDbType.Enum).Value = numStatud;
                    }

                    cmd.Parameters.Add("@id",
MySql.Data.MySqlClient.MySqlDbType.Int32).Value = rep_id;
                    cmd.Parameters.Add("@repairs_date",
MySql.Data.MySqlClient.MySqlDbType.Date).Value = DT.Value;
                    cmd.Parameters.Add("@computer_id",
MySql.Data.MySqlClient.MySqlDbType.Int32).Value = comp_id;
                    cmd.Parameters.Add("@employee_id",
MySql.Data.MySqlClient.MySqlDbType.Int32).Value = emplid;
                    cmd.ExecuteNonQuery();
                    conn.Close();
                    this.Close();
                }
```

```
            }
            else
            {
                MessageBox.Show("Не все значения выбраны!");
            }
        }
    }
}
```

Файл RepPartsForm.cs:

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

namespace MilComp
{
    public partial class RepPartsForm : Form
    {
        private BindingSource masterBindingSource = new
BindingSource();
        private BindingSource detailsBindingSource = new
BindingSource();
        private DataSet data;
        public RepPartsForm()
        {
            InitializeComponent();
        }

        private void GetData()
        {
            string masterQuery = "SELECT * FROM repairs";
            string detailQuery = "SELECT * FROM parts";

            masterDataAdapter = new
MySqlDataAdapter(masterQuery, mycon);
            detailsDataAdapter = new
MySqlDataAdapter(detailQuery, mycon);

            DataSet ds = new DataSet();
```

```
            masterDataAdapter.Fill(ds, "repairs");
            detailsDataAdapter.Fill(ds, "parts");

            DataColumn parentColumn =
ds.Tables["repairs"].Columns["id"];
            DataColumn childColumn =
ds.Tables["parts"].Columns["repair_id"];

            DataRelation relation = new
DataRelation("РемонтЗапчасти", parentColumn, childColumn);
            ds.Relations.Add(relation);

            masterBindingSource = new BindingSource();
            detailsBindingSource = new BindingSource();

            masterBindingSource.DataSource = ds;
            masterBindingSource.DataMember = "repairs";

            detailsBindingSource.DataSource =
masterBindingSource;
            detailsBindingSource.DataMember = "РемонтЗапчасти";

            dg1.DataSource = masterBindingSource;
            dg2.DataSource = detailsBindingSource;
        }


        public void FillData()
        {
            GetData();
            dg1.Columns["id"].HeaderText = "Номер ремонта";
            dg1.Columns["repairs_date"].HeaderText = "Дата
ремонта компьютера";
            dg1.Columns["computer_id"].HeaderText = "Номер
компьютера";
            dg1.Columns["employee_id"].HeaderText = "Номер
сотрудника";
            dg1.Columns["statud"].HeaderText = "Статус ремонта";
            dg1.Columns["id"].Width=200;
            dg1.Columns["repairs_date"].Width =200;
            dg1.Columns["computer_id"].Width = 150;
            dg1.Columns["employee_id"].Width = 150;
            dg1.Columns["statud"].Width = 150;
            dg1.RowHeadersVisible = false;
            dg2.RowHeadersVisible = false;
            dg2.Columns["id"].HeaderText = "Номер запчасти";
            dg2.Columns["name"].HeaderText = "Название
запчасти";
            dg2.Columns["serial_number"].HeaderText = "Серийный
номер";
```

```
            dg2.Columns["date_receipt"].HeaderText = "Дата
приобретения";
            dg2.Columns["repair_id"].HeaderText = "Номер
ремонта";
            dg2.Columns["id"].Width = 200;
            dg2.Columns["name"].Width = 150;
            dg2.Columns["serial_number"].Width = 150;
            dg2.Columns["date_receipt"].Width = 150;
            dg2.Columns["repair_id"].Width = 150;
            dg1.ReadOnly = false;
            dg2.ReadOnly = false;
            foreach (DataGridViewColumn column in dg1.Columns)
            {
                column.HeaderCell.Style.Alignment =
DataGridViewContentAlignment.MiddleCenter;
            }
            foreach (DataGridViewColumn column in dg2.Columns)
            {
                column.HeaderCell.Style.Alignment =
DataGridViewContentAlignment.MiddleCenter;
            }
            dg1.BorderStyle = BorderStyle.FixedSingle;
            dg2.BorderStyle = BorderStyle.FixedSingle;
            dg1.ColumnHeadersBorderStyle =
DataGridViewHeaderBorderStyle.Single;
            dg2.ColumnHeadersBorderStyle =
DataGridViewHeaderBorderStyle.Single;
            dg1.GridColor = Color.Black;
            dg2.GridColor = Color.Black;
            dg2.Columns["repair_id"].Visible = false;
            bn1.DataSource = masterBindingSource;
            bn2.DataSource = detailsBindingSource;
        }

        private void RepPartsForm_Load(object sender, EventArgs
e)
        {

            FillData();
        }

        private void buExit_Click(object sender, EventArgs e)
        {
            if (mycon != null && mycon.State !=
ConnectionState.Open)
            {
                mycon.Close();
                mycon.Dispose();
            }

            this.Close();
```

```
        }

        private void buRefresh_Click(object sender, EventArgs e)
        {
            FillData();
        }

        private void buDelete_Click(object sender, EventArgs e)
        {
            if (dg2.CurrentCell != null)
            {
                int rowIndex = dg2.CurrentCell.RowIndex;
                DataGridViewRow selectedRow =
dg2.Rows[rowIndex];
                int idd =
Convert.ToInt32(selectedRow.Cells[0].Value);

                var confirmResult = MessageBox.Show("Удалить эту
запись?", "Подтверждение", MessageBoxButtons.YesNo);
                if (confirmResult == DialogResult.Yes)
                {
                    DeleteRecordFromDatabase(idd);
                }

            }
            else
            {
                MessageBox.Show("Вы не выбрали ни одной записи
таблицы Запчасти");
            }
        }

        private void DeleteRecordFromDatabase(int id)
        {
            mycon.Open();
            string query = "DELETE FROM `parts` WHERE
(`parts`.`id` = @id);";

            using (command = new MySqlCommand(query, mycon))
            {
                command.Parameters.AddWithValue("@id", id);
                command.ExecuteNonQuery();
            }
            mycon.Close();
            FillData();
        }
    }
}
```

Приложение В

(Обязательное)

Автоматизированная система планирования учебной нагрузки преподавателей

Презентация