



**ФЕДЕРАЛЬНОЕ АГЕНТСТВО МОРСКОГО И РЕЧНОГО ТРАНСПОРТА**

**Федеральное государственное бюджетное образовательное учреждение высшего образования  
«Государственный университет морского и речного флота имени адмирала С.О. Макарова»  
(ФГБОУ ВО «ГУМРФ имени адмирала С.О. Макарова»)  
Воронежский филиал**

Кафедра математики, информационных систем и технологий

Направление подготовки 09.03.02 Информационные системы и технологии

(код, наименование направления подготовки/специальности)

Форма обучения очная

«К ЗАЩИТЕ ДОПУЩЕН(А)»  
Заведующий кафедрой

(подпись)

Черняева С. Н.

(ФИО)

20

**Выпускная квалификационная работа**

Обучающегося Зализко Валерия Витальевича

(фамилия, имя, отчество)

Вид работы выпускная квалификационная работа бакалавра

(выпускная квалификационная работа бакалавра, специалиста, магистра)

**Пояснительная записка**

Тема Разработка приложения выбора и оценки стоимости оборудования для проектирования ЛВС (на примере ООО «Сфера Ай Ти»).

(полное название темы квалификационной работы, в соответствии с приказом об утверждении тематики ВКР)

Руководитель работы доцент, к.э.н., Скрипников О.А., 21.06.2024

(должность, подпись, фамилия, инициалы, дата)

Консультант \_\_\_\_\_

(при наличии)

(должность, подпись, фамилия, инициалы, дата)

Консультант \_\_\_\_\_

(должность, подпись, фамилия, инициалы, дата)

Обучающийся Зализко В.В., 21.06.2024

(подпись, фамилия, инициалы, дата)

Воронеж  
2024

ФЕДЕРАЛЬНОЕ АГЕНТСТВО МОРСКОГО И РЕЧНОГО ТРАНСПОРТА

Федеральное государственное бюджетное образовательное учреждение высшего образования  
«Государственный университет морского и речного флота имени адмирала С.О. Макарова»  
(ФГБОУ ВО «ГУМРФ имени адмирала С.О. Макарова»)  
Воронежский филиал

Кафедра математики, информационных систем и технологий

Направление подготовки 09.03.02 Информационные системы и технологии

(код, наименование направления подготовки/специальности)

Форма обучения очная

УТВЕРЖДАЮ  
Заведующий кафедрой

\_\_\_\_\_

(подпись)

Черняева С. Н.

(ФИО)

\_\_\_\_\_ 202

**Задание  
на выпускную квалификационную работу**

Вид работы ВКР бакалавра

(ВКР бакалавра, ВКР специалиста, ВКР магистра)

Обучающемуся Зализко Валерию Витальевичу

(фамилия, имя, отчество)

Тема Разработка приложения выбора и оценки стоимости оборудования для проектирования ЛВС (на примере ООО «Сфера Ай Ти»).

Утверждена приказом ректора университета от \_\_\_\_\_ 20\_\_\_\_, № \_\_\_\_\_

Срок сдачи законченной работы \_\_\_\_\_ 20\_\_\_\_

Исходные данные (или цель ВКР):

Разработать приложение для выбора и оценки стоимости оборудования при проектировании ЛВС (на примере ООО «Сфера Ай Ти»).

Перечень подлежащих исследованию, разработке, проектированию вопросов (краткое содержание ВКР):

*(актуальность темы, цели и задачи ВКР; аналитический обзор литературных источников; постановка задачи исследования, разработки, проектирования; содержание процедуры исследования, разработки, проектирования; обсуждение результатов; дополнительные вопросы, подлежащие разработке; заключение – выводы по работе в целом, оценка степени решения поставленных задач, практические рекомендации; и др.)*

– Введение. Актуальность выбранной темы, цель и задачи ВКР.

(наименование вопроса, раздела и его краткое содержание)

- 1. Исследовательский раздел. Анализ предметной области ВКР.  
(наименование вопроса, раздела и его краткое содержание)
- 1.1. История развития ЛВС. Анализ развития появления ЛВС.  
(наименование вопроса, раздела и его краткое содержание)
- 1.2. Классификация ЛВС. Анализ ключевых современных подходов построения ЛВС.  
(наименование вопроса, раздела и его краткое содержание)
- 1.3. Стандарты и типы, регламентирующие построение ЛВС. Анализ стандартов, регламентирующих построение ЛВС.  
(наименование вопроса, раздела и его краткое содержание)
- 1.4. Теоретические основы выбора оборудования для проектирования ЛВС. Анализ основ выбора оборудования для проектирования ЛВС.  
(наименование вопроса, раздела и его краткое содержание)
- 2. Проектный раздел. Планировка структур и задач. Разработка приложения. Получение общей оценочной стоимости проектирования ЛВС с использованием разработанного приложения на примере ООО «Сфера Ай Ти».  
(наименование вопроса, раздела и его краткое содержание)
- 2.1. Аналитическая часть. Построение структуры процесса проектирования ЛВС для понимания взаимосвязей между этапами. Составление технического задания к приложению, выбор среды разработки.  
(наименование вопроса, раздела и его краткое содержание)
- 2.2. Исследование организации ООО «Сфера Ай Ти». Исследования объекта. Подбор стандартов и выбор оборудования для проектирования ЛВС.  
(наименование вопроса, раздела и его краткое содержание)
- 2.3. Разработка приложения. Создание БД. Создание приложения.  
(наименование вопроса, раздела и его краткое содержание)
- 2.4. Оценка общей стоимости оборудования для проектирования ЛВС на примере ООО «Сфера Ай Ти». Получение оценочной стоимости проектирования ЛВС на основе исследования ООО «Сфера Ай Ти».  
(наименование вопроса, раздела и его краткое содержание)
- Заключение. Выводы по работе в целом. Оценка степени решения поставленных задач.  
(наименование вопроса, раздела и его краткое содержание)

Практические рекомендации. Использовать приложение для получения общей оценочной стоимости проектирования ЛВС.

Перечень графического материала (или презентационного материала):

1. Титульный лист.
2. Актуальность.
3. Цель и задачи ВКР.
4. Диаграмма декомпозиции.
5. Требования к приложению.
6. Используемые инструменты.
7. Исследование ООО «Сфера Ай Ти».
8. Схема построения ЛВС ООО «Сфера Ай Ти».
9. Результаты ВКР.

Консультанты по разделам ВКР (при наличии):

1. \_\_\_\_\_  
(наименование раздела, ученая степень, ученое звание и должность, ФИО консультанта)
2. \_\_\_\_\_  
(наименование раздела, ученая степень, ученое звание и должность, ФИО консультанта)
3. \_\_\_\_\_

(наименование раздела, ученая степень, ученое звание и должность, ФИО консультанта)

Дата выдачи задания: \_\_\_\_\_ 20\_\_\_\_\_

Задание согласовано и принято к исполнению: \_\_\_\_\_ 20\_\_\_\_\_

Руководитель ВКР: доцент, к.э.н., Скрипников Олег Александрович \_\_\_\_\_  
(должность, ученая степень, ученое звание, ФИО) (подпись)

Обучающийся: ИТ-4-1, Зализко Валерий Витальевич \_\_\_\_\_  
(учебная группа, ФИО) (подпись)

## Содержание

ВВЕДЕНИЕ .....	6
1. ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ .....	8
1.1. История развития ЛВС .....	8
1.2. Классификация ЛВС .....	9
1.3. Стандарты и типы, регламентирующие построение ЛВС .....	18
1.4. Теоретические основы выбора оборудования для проектирования ЛВС 25	
2. ПРОЕКТНЫЙ РАЗДЕЛ.....	37
2.1. Аналитическая часть.....	37
2.2. Исследование организации ООО «Сфера Ай Ти».....	45
2.3. Разработка приложения .....	48
2.4. Оценка общей стоимости оборудования для проектирования ЛВС на примере ООО «Сфера Ай Ти» .....	61
ЗАКЛЮЧЕНИЕ .....	67
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ .....	68
ПРИЛОЖЕНИЕ .....	70

## ВВЕДЕНИЕ

Выпускная квалификационная работа по теме «Разработка приложения выбора и оценки стоимости оборудования для проектирования ЛВС» содержит 105 страниц текстового документа, 57 рисунков, 3 таблицы, 3 диаграммы и 20 использованных источников. Во введении описана актуальность разработки приложения, сформулирована цель и задачи исследования. Первая глава содержит ответы на многие теоретические вопросы, определения. Она вся в целом посвящена локально вычислительным сетям. Вторая глава описывает рабочие процессы программы на примере построенных диаграмм. Поставлены технические задачи, по которому разработано ПО. Описан практический метод проектирования локальной вычислительной сети на примере ООО «Сфера Ай Ти» с выбором и оценкой стоимости необходимого оборудования. Заключение содержит основные выводы, полученные в ходе выполнения поставленных задач.

В современную эпоху, без сомнения, мы являемся свидетелями беспрецедентного развития и распространения сетевых технологий. Число людей, подключенных к глобальной сети интернет, неуклонно и стремительно растет с каждым днем, демонстрируя весьма впечатляющую динамику. Организации всех типов и размеров активно и целенаправленно внедряют локальные сетевые инфраструктуры как для внутреннего пользования, так и в целях многопользовательского выхода в сеть. Следует особо отметить, что даже государственная политика в сфере предоставления государственных услуг и информирования граждан теперь ориентирована на дистанционную работу с гражданами посредством всеобщей сети. Данные изменения, очевидно и неоспоримо не могли не сказаться на рынке труда и, как следствие, на сфере образования. В самом деле, потребность в квалифицированных IT-специалистах с хорошим уровнем знаний в данной области возросла. В связи с вышеперечисленным практически любое здание в современном мире, по всей видимости, нуждается в проектировании локальной вычислительной

сети. Организации вынуждены искать формы и модели планирования, которые обеспечивали бы максимальную эффективность принимаемых решений.

И хотя инженер в принципе может справиться в проектировании локальных вычислительных сетей без сторонних программ, но это имеет и два серьезных недостатка. Они заключаются в том, что поиск информации об оборудовании, характеристиках, актуальных ценах занимает чрезмерно большое количество времени, а при проектировании больших и объемных ЛВС у специалиста просто не хватает внимания, чтобы не пропустить какой-либо организационный момент. Подобные проектирования в современном мире, бесспорно, совершенно не эффективны.

В связи с этим главная цель данной бакалаврской работы – это снизить нагрузку инженеров и пользователей, проектирующих ЛВС за счет адаптивной системы с необходимым функционалом для проектирования больших и объемных локальных вычислительных сетей. Для этого необходимо выполнить следующие задачи:

- провести анализ предметной области;
- рассмотреть теоретические основы выбора оборудования для проектирования ЛВС;
- составить техническое задание к программе;
- определить инструментальные средства и технологии разработки;
- спроектировать и разработать программное обеспечение;
- выполнить краткий анализ выполненных задач.

# 1. ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ

## 1.1. История развития ЛВС

По своей сути локально вычислительная сеть в действительности представляет собой группу компьютеров и других устройств, которые, собственно, соединены между собой с целью совместного использования ресурсов и обмена данными в пределах ограниченной географической области, такой как, к примеру, дом, офис или кампус (рис.1).



Рисунок 1 – ЛВС

Безусловно, следует отметить, что локальные сети прошли длительную эволюцию в своем развитии, прежде чем достигли современного уровня качества и скорости передачи данных. Анализируя историю развития локальных вычислительных сетей, необходимо подчеркнуть, что она действительно берет свое начало в 1970-х годах, когда впервые появились технологии для объединения компьютеров в сеть в пределах одной организации или здания. На данном начальном этапе, как можно видеть, использовались такие протоколы и решения, как Ethernet, Token Ring и ARCnet. В 1980-х годах, как свидетельствуют соответствующие исследования, начался процесс стандартизации с появлением протоколов TCP/IP и модели OSI, что, в свою очередь, позволило создавать сети, совместимые с оборудованием и программным обеспечением различных производителей [1].

Важно отметить, что в 1990-х годах ЛВС получили широкое распространение благодаря главным образом снижению стоимости сетевого оборудования и появлению более быстрых технологий, таких как Fast Ethernet и Gigabit Ethernet, а также беспроводных сетей Wi-Fi. Согласно имеющимся данным, в 2000-х и 2010-х годах развитие ЛВС продолжилось с внедрением сверхбыстрых решений, таких как 10-Gigabit Ethernet и новых стандартов беспроводной связи, таких как Wi-Fi 6 [14].

В заключение следует подчеркнуть, что на сегодняшний день ЛВС, без сомнения, являются неотъемлемой частью ИТ-инфраструктуры большинства организаций, обеспечивая связь и передачу данных между различными устройствами и системами. Это, несомненно, свидетельствует о значительном прогрессе в области сетевых технологий и их ключевой роли в современном информационном обществе.

## 1.2. Классификация ЛВС

Современные подходы построения локально вычислительных сетей действительно классифицируют себя по топологии и методам доступа к среде. Выбор подхода фактически играет ключевую роль для определения архитектуры сети, методов передачи данных, скорости передачи данных и масштаба сети. Классификация, без сомнения, помогает определить элементы, необходимые для построения сети, правила и протоколы для обмена информацией между устройствами сети. Более того, классификация ЛВС помогает принять наиболее рациональное решение по выбору сетевых решений, связанных с организацией, мониторингом, обслуживанием и упорядочением сетевой структуры.

Рассмотрим классификацию сети по топологии. При проектировании ЛВС, как правило, рассматривают существующие топологии. По своей сути они определяют физическую структуру, масштаб, способ организации подключения устройств в сети, состоящий из конечных узлов сети

(компьютеры, маршрутизаторы) и сетевых ребер, которые, разумеется, отвечают за соединение между вершинами. Следует отметить, что существуют полно-связные и неполно-связные топологии. При полной связи конфигурации в действительности каждый компьютер связан со всеми остальными конечными узлами сети. Этот вариант логически прост, но, очевидно, оказывается усложненным и неэффективным. В самом деле, при таком построении ребер между вершинами каждый компьютер в сети должен иметь большое количество коммуникационных портов, достаточное для связи с каждым из остальных компьютеров сети. Фактически для каждой пары компьютеров должна быть выделена отдельная физическая линия связи для двусторонней передачи.

Полно-связные топологии, по всей видимости, в крупных сетях применяются очень редко. Чаще этот вид топологии используется в многомашинных комплексах или в сетях, объединяющих небольшое количество компьютеров. При неполной связи конфигурации, несомненно, все устройства соединены таким образом, что не каждое устройство связано с каждым другим. Вместо этого одно или несколько устройств, как представляется, имеют несколько соединений с другими устройствами, создавая структуру, схожую со смешанным режимом полно-связной и других топологий. При таком построении топологии следует признать, что компьютеру в сети необязательно иметь большое количество коммуникационных портов, что, безусловно, помогает оправдать затраты на поддержку сети за счет отказоустойчивости и эффективности сети.

Опишем основные существующие топологии локально вычислительной сети. Звездообразная топология конфигурация сети, в которой все узлы соединены с центральным узлом или концентратором (рис 2).

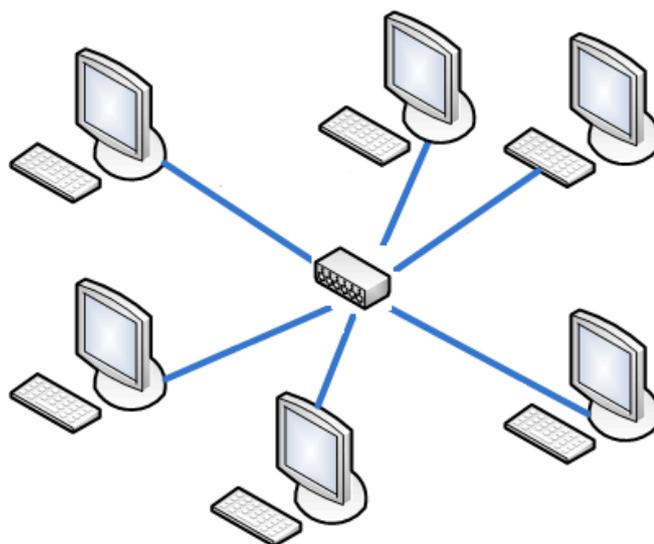


Рисунок 2 – Звездообразная топология

В центре этой топологии, собственно, находится один главный узел, к которому подключены все остальные узлы. Этот центральный узел, как правило, может быть концентратором, коммутатором или маршрутизатором и неограниченно расширяться в зависимости от потребностей. В данной топологии отсутствуют прямые соединения между узлами, ребра не связаны друг с другом напрямую. Каждый узел соединяется только с центральным узлом, и весь трафик между любыми двумя узлами должен проходить через этот центральный узел, что, разумеется, облегчает управление и мониторинг сети. Выход из строя любого отдельного узла звездообразной топологии действительно не влияет на остальную часть сети, поскольку все узлы независимы друг от друга, а благодаря центральному узлу можно контролировать и управлять трафиком, обеспечивая безопасность и производительность сети. Для добавления новых узлов в такую топологию, как представляется, потребуется лишь подключить маршрутизатор к центральному узлу. Звездообразная топология без промежуточных узлов, по всей видимости, может быть дорогостоящей и ограниченной с точки зрения производительности, если центральный узел не обладает достаточной пропускной способностью для обработки всего трафика в сети [3].

Шинная топология по своей сути представляет собой конфигурацию сети, в которой все узлы соединены с одним общим кабелем, называемым шиной (рис. 3). В этой топологии сигналы, передаваемые одним узлом, конечно, распространяются по всей шине и доступны для всех остальных узлов. Топология была одной из первых и наиболее распространенных топологий в ранних локальных сетях. Она проста в реализации, обладает относительно низкой стоимостью, имеется возможность подключения новых узлов посредством добавления их на шину. Однако при построении шинной топологии, как представляется, ограничена пропускная способность, поскольку все узлы делят одну общую среду передачи данных, если выйдет из строя один узел или произойдет повреждение кабеля, это может привести к сбою всей построенной сети. По мере развития технологий шинная топология, по всей видимости, постепенно была вытеснена более современными и эффективными топологиями, такими как звездообразная и древовидная, которые обеспечивают лучшую производительность, надежность и масштабируемость.

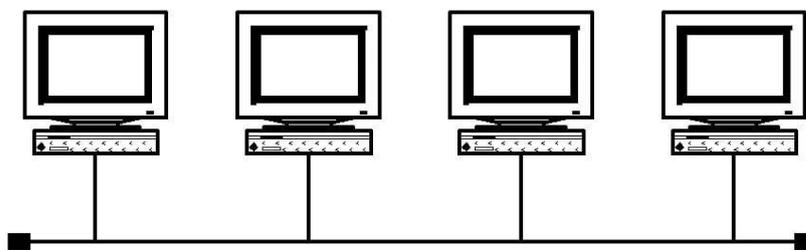


Рисунок 3 – Шинная топология

В действительности кольцевая топология, по сути дела, представляет собой конфигурацию сети, в которой все узлы соединены в замкнутое кольцо или петлю с помощью двунаправленных соединений (рис 4). Фактически в такой топологии каждое устройство подключено к двум соседним устройствам, образуя кольцо, передавая данные в одном направлении, зачастую используя метод передачи с маркером или жетоном, который

циркулирует по кольцу, определяя, какой узел может передавать данные. Несомненно, кольцевая топология обеспечивает распределенный характер доступа к среде передачи данных, а также избыточность соединений, что, собственно, повышает отказоустойчивость сети. Однако следует отметить, что выход из строя одного узла или соединения приводит к разрыву кольца и отключению всей сети до восстановления кольца. Раньше кольцевая топология использовалась в ранних сетях, таких как Token Ring, но с развитием технологий и появлением более эффективных решений, таких как коммутируемые сети Ethernet, ее применение снизилось. Тем не менее она по-прежнему находит применение в определенных специализированных сетевых средах, где требуется высокая отказоустойчивость и равномерное распределение нагрузки. Фактически передача данных в одном направлении способствует устойчивости сети, однако поломка или отключение одного устройства может нарушить работу всего кольца.

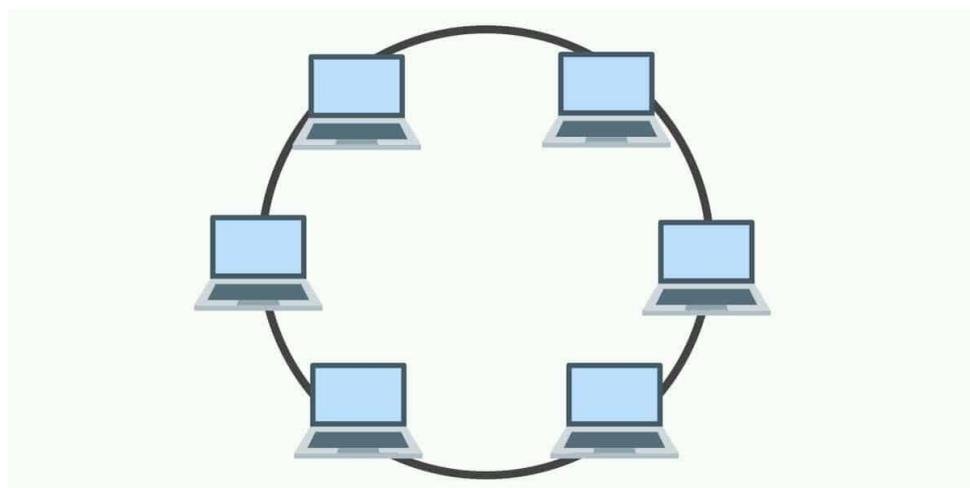


Рисунок 4 – Кольцевая топология

В настоящее время смешанная топология представляет собой комбинацию нескольких типов топологий в одной сети. По существу, в этом случае различные части сети или подсети построены с использованием разных топологических структур, таких как звезда, шина, кольцо или древовидная топология. Фактически эти отдельные сегменты соединяются вместе при помощи сетевых устройств, таких как концентраторы, коммутаторы или

маршрутизаторы, образуя единую гибридную сетевую инфраструктуру. Без всякого сомнения, смешанная топология позволяет объединить преимущества различных топологий и компенсировать их недостатки. В действительности, например, отдельные сегменты могут быть организованы по принципу звездообразной топологии для упрощенного управления и отказоустойчивости, в то время как межсегментные соединения выполняются с использованием топологии кольца или шины для обеспечения необходимой избыточности [5]. По сути, смешанная топология распространена в корпоративных и крупномасштабных сетях, где требуется высокая производительность, гибкость и масштабируемость, а также в сетях, построенных на базе различных технологий и стандартов, так как позволяет адаптировать сеть под конкретные потребности. Однако следует отметить, что построение данной конфигурации требует более сложной инфраструктуры и управления, что, собственно, повышает затраты на ее реализацию (рис 5).

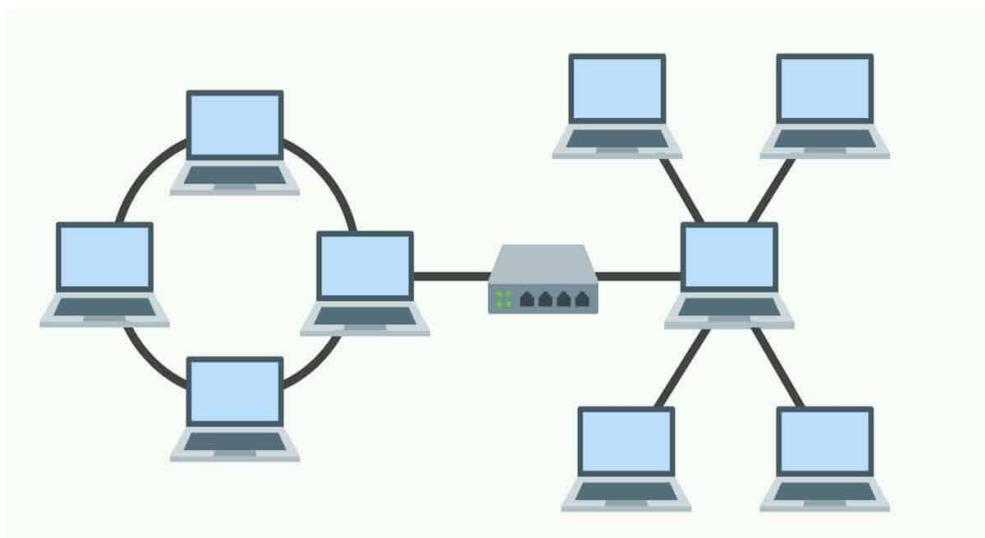


Рисунок 5 – Смешанная топология

В современном мире древовидная топология представляет собой иерархическую структуру сети, в которой узлы организованы в виде разветвленного дерева. По сути, в этой топологии есть один корневой узел, к которому подключаются несколько дочерних узлов или сегментов. Фактически каждый из этих дочерних узлов может иметь свои собственные

ответвления, образуя подсети. Данные передаются от корневого узла к конечным узлам по разветвленным путям древовидной структуры и обратно [12]. Вне сомнения, древовидная топология обеспечивает хорошую масштабируемость и управляемость, поскольку позволяет организовать сеть на несколько уровней. Она также обладает некоторой избыточностью за счет наличия нескольких путей передачи данных между узлами. Однако следует отметить, что сбой в корневом узле или в ключевых сегментах может привести к отказу значительной части сети. В действительности древовидная топология широко используется в современных корпоративных и провайдерских сетях, а также в сетях крупных зданий, объединяя преимущества централизованного управления и возможности масштабирования. По существу, такая топология сложна в установке и управлении, при поломке центральных устройств может повлечь за собой проблемы во всей сети (рис 6).

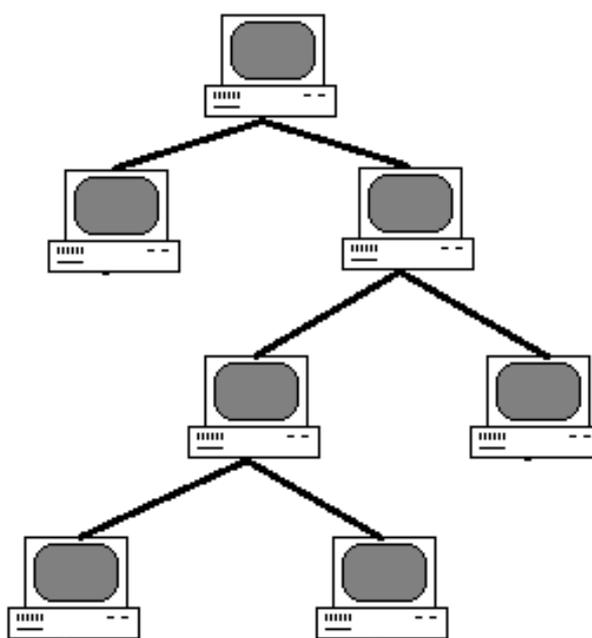


Рисунок 6 – Древовидная топология

Рассмотрим классификацию ЛВС по методу доступа к среде. В действительности Ethernet – технология, которая представляет собой широко используемые локальные сети. Фактически разработка технологии была предназначена для взаимодействия между участниками сети по протоколу,

который является общим сетевым языком. Несомненно, разработка также служила для заполнения технологического разрыва между низкоскоростными сетями большой протяженности и специализированными сетями масштаба небольшого помещения, в которых данные передаются с большими скоростями на очень коротких расстояниях [19]. Фактически технология Ethernet является одним из наиболее распространенных стандартов для локальных вычислительных сетей. Она обладает разделяемой средой передачи данных, в которой все узлы сети соединены общей средой, такой как коаксиальный кабель или витая пара. В действительности передача данных осуществляется посредством широковещательной рассылки пакетов всем узлам сети, адресация происходит на канальном уровне с использованием MAC-адресов для идентификации источника и получателя данных. Существует метод множественного доступа с контролем несущей и обнаружением коллизий CSMA/CD для предотвращения конфликтов при одновременной передаче данных несколькими узлами. Фактически данный сетевой протокол является протоколом для передачи потока, который работает на уровне управления доступом к среде (MAC). Он определяет или прослушивает, занят или нет общий канал для передачи и откладывает передачу до тех пор, пока канал не освободится. CSMA/CD известен как состязательный метод, поскольку сетевые компьютеры конкурируют между собой за право передавать данные. Среда Ethernet гибка и масштабируема за счет возможности построения древовидной топологии с коммутаторами, что повышает производительность сети. Технология поддерживает различных скоростей передачи данных от 10 Мбит/с до 100 Гбит/с в современных реализациях Ethernet проста в развертывании и использовании благодаря стандартизации на физическом и канальном уровнях модели OSI. Технология широко используется в офисах, учебных учреждениях, дата-центрах, общественных учреждениях и домашних сетях для обеспечения связи и обмена информацией.

Token Ring – технология построения локальных вычислительных сетей, разработанная компанией IBM в 1980-х годах. В данной сети компьютеры соединяются в кольцевую топологию, где данные передаются по кругу от одного узла к другому с помощью специального пакета, называемого токеном. Только тот узел, который владеет токеном, может передавать данные, что предотвращает коллизии и обеспечивает детерминированный доступ к среде передачи. Когда узел заканчивает передачу, он освобождает токен, позволяя следующему узлу в кольце начать передачу данных. Технология поддерживает скорости передачи данных 4 Мбит/с и 16 Мбит/с и использует экранированную витую пару или оптоволоконный кабель в качестве физической среды. Несмотря на свою надежность и детерминированность, Token Ring постепенно уступил место Ethernet из-за более высокой стоимости оборудования и меньшей гибкости в масштабировании сети. Тем не менее некоторые концепции, используемые в Token Ring, такие как детерминированный доступ к среде и отсутствие коллизий, нашли применение в других технологиях, таких как FDDI (Fiber Distributed Data Interface) и технологии промышленных сетей.

Wireless LAN (WLAN) – технология, позволяющая устройствам подключаться к локальной вычислительной сети без использования проводных соединений. Примечательно, что технология использует радиочастотные волны для передачи данных между устройствами и точками доступа, которые подключены к проводной сети. Несомненно, наиболее распространенным стандартом WLAN является IEEE 802.11, также известный как Wi-Fi. Устройства, поддерживающие Wi-Fi, такие как ноутбуки, смартфоны и планшеты, могут подключаться к сети в пределах определенного радиуса действия точки доступа [2]. Безусловно, данная сеть обеспечивает гибкость и мобильность для пользователей, позволяя им получать доступ к сети и Интернету без необходимости физического подключения к проводной инфраструктуре. Однако следует отметить, что беспроводные сети более восприимчивы к помехам и проблемам с безопасностью по сравнению с

проводными сетями. В связи с этим, для обеспечения конфиденциальности и целостности данных в WLAN используются различные методы шифрования, такие как WEP, WPA и WPA2. Кроме того, WLAN может работать в различных частотных диапазонах, таких как 2,4 ГГц и 5 ГГц, с разными скоростями передачи данных и радиусом действия. Важно подчеркнуть, что развитие технологий WLAN привело к появлению новых стандартов, таких как 802.11ac и 802.11ax (Wi-Fi 6) которые обеспечивают более высокие скорости передачи данных, улучшенное покрытие и повышенную эффективность использования спектра для поддержки большего количества устройств в сети.

Таким образом, классификация локальных сетей основывается на физической организации, типе используемой среды и размере сети, что определяет ее производительность, надежность и области применения. Выбор конкретной классификации зависит от потребностей конкретной сети и ее предполагаемого использования.

### 1.3. Стандарты и типы, регламентирующие построение ЛВС

Документация, регламентирующая построение локальных вычислительных сетей, представляет собой набор стандартов, спецификаций и руководств, которые определяют правила, рекомендации и лучшие практики для проектирования, внедрения и управления ЛВС. Эти документы разрабатываются различными организациями по стандартизации, такими как IEEE (Institute of Electrical and Electronics Engineers) и ISO (International Organization for Standardization), и TIA (Telecommunications Industry Association). Они охватывают различные аспекты ЛВС, включая физическую инфраструктуру (кабельные системы, коннекторы, шкафы и т.д.), сетевые протоколы и сервисы, производительность, безопасность и управление сетью. Некоторые из наиболее важных стандартов включают серию IEEE 802,

которая определяет спецификации для физического и канального уровней ЛВС [3].

Стандарты Ethernet, разработанные IEEE, охватывают широкий спектр аспектов, обеспечивающих эффективную и надежную работу локальных сетей на основе технологии Ethernet [14]. Эти стандарты определяют топологии сети, физические среды передачи данных, протоколы доступа к среде, форматы кадров данных, механизмы управления сетью, меры безопасности, показатели производительности и требования совместимости между устройствами различных производителей. Все эти элементы тщательно прописаны в стандартах, чтобы гарантировать структурированный и унифицированный подход к проектированию, внедрению и функционированию локальных сетей, обеспечивая их надежность, высокую производительность и совместимость на всех уровнях.

Fast Ethernet представляет собой общее название для комплекса взаимосвязанных стандартов передачи данных в компьютерных сетях на основе технологии Ethernet (таблица 1).

Таблица 1 – Комплекс стандартов Fast Ethernet

100 Мбит/с Ethernet (Fast Ethernet)	Стандарт	Год выхода	Тип	Скорость передачи (Mbps)	Максимальная длина прокладки в метрах	Тип кабеля
	IEEE 802.3u	1995	100Base-FX	99,999	одномод – два км. многомод – четыреста м.	оптоволоконный
			100Base-T	99,999	сто метров	UTP/STP cat 5
			100Base-T4	99,999	сто метров	UTP/STP cat $\geq 3$
			100Base-TX	99,999	сто метров	UTP/STP cat 5

Окончание таблицы 1

IEEE 802.12	1995	100Base-VG	99,999	сто метров	UTP cat 3,5
IEEE 802.3u	1998	100Base-T2	99,999	сто метров	UTP cat 3,5
TIA/EIA-785	2001	100Base-SX	99,999	триста метров	оптоволоконный
IEEE 802.3ah	2004	100Base-LX10	99,999	десять километров	
IEEE 802.3ah	2004	100Base-BX10	99,999	десять километров	

Основным отличием Fast Ethernet является значительно увеличенная по сравнению с исходной спецификацией Ethernet пропускная способность, достигающая скорости до 100 Мбит/с, в то время как первоначальный стандарт технологии обеспечивал скорость передачи данных лишь 10 Мбит/с. Столь существенное в десять раз повышение скорости передачи данных было достигнуто благодаря применению ряда инновационных технических решений, включая использование более совершенных методов кодирования и модуляции сигналов, а также оптимизацию протоколов доступа к среде передачи данных.

Gigabit Ethernet (GbE) представляет собой комплекс взаимосвязанных технологических решений, направленных на обеспечение передачи пакетов данных в сетях Ethernet со скоростью 1 гигабит в секунду (1 Гбит/с). Данный набор стандартов был всесторонне проработан и официально принят в документе IEEE 802.3 – 2005, что позволило произвести качественный скачок в развитии технологий Ethernet и удовлетворить растущие потребности в более высокой пропускной способности компьютерных сетей (таблица 2).

Таблица 2 – Комплекс стандартов Gigabit Ethernet

1000 Мбит/с (Gigabit Ethernet)	Стандарт	Год выхода	Тип	Скорость передачи (Mbps)	Максимальная длина прокладки в метрах	Тип кабеля
	IEEE 802.3z	1998	1000Base-CX	999,99	двадцать метров	UTP/STP cat 5,5e,6
			1000Base-LX	999,99	одномод – пять км. многомод – пятьсот м	оптоволоконный
			1000Base-SX	999,99	пятьсот метров	
	IEEE 802.3ab	1999	1000Base-T	999,99	сто метров	UTP/STP cat 5,5e,6,7
	TIA 854	2001	1000BASE-TX	999,99	сто метров	UTP/STP cat 6,7
	IEEE 802.3ah	2004	1000BASE-LX10	999,99	десять километров	оптоволоконный
	IEEE 802.3ah	2004	1000BASE-BX10	999,99	десять километров	оптоволоконный
	IEEE 802.3ap	2007	1000BASE-KX	999,99	десять метров	для объединительной платы

Продолжая тенденцию к дальнейшему повышению скоростей передачи данных, в 2006 году был представлен стандарт 10 Gigabit Ethernet. На тот момент он являлся новейшим и самым быстрым из существующих стандартов семейства Ethernet. Согласно проведенным научно-техническим разработкам, 10GbE передает пакеты с номинальной скоростью передачи данных 10 гигабит в секунду, что в десять раз превышает пропускную способность Gigabit Ethernet (таблица 3). Для обеспечения столь высокой скорости передачи были применены передовые методы кодирования, модуляции и множественного доступа.

Таблица 3 – Комплекс стандартов 10 Gigabit Ethernet

10 Гбит/с Ethernet (10 GbE)	Стандарт	Год выхода стандарта	Тип	Скорость передачи (Gbps)	Максимальная длина сегмента в метрах	Тип кабеля
	IEEE 802.3ae	2003	10GBASE-SR	9,999	одномод – один км. многомод – триста метров	оптоволоконный
		2003	10GBASE-LX4	9,999	одномод – один км. многомод – триста метров	
		2003	10GBASE-LR	9,999	десять километров	
		2003	10GBASE-ER	9,999	сорок километров	
		2003	10GBASE-SW	9,999	двадцать метров – сорок километров	
		2003	10GBASE-LW	9,999		
		2003	10GBASE-EW	9,999		
	IEEE 802.3ak	2004	10GBASE-CX4	9,999	пятнадцать метров	медный кабель CX4
	IEEE 802.3an	2006	10GBASE-T	9,999	сто метров	UTP/STP cat 6,6a,7
	IEEE 802.3aq	2006	10GBASE-LRM	9,999	двести метров	оптоволоконный
	IEEE 802.3ap	2007	10GBASE-KX4	9,999	один метр	для объединительной платы
	IEEE 802.3ap	2007	10GBASE-KR	9,999	один метр	
	IEEE 802.3av	2009	10GBASE-PR	9,999	двадцать километров	оптоволоконный

Стандарты Token Ring, разработанные IEEE, определяют требования к проектированию, внедрению и функционированию локальных вычислительных сетей, использующих технологию Token Ring. Основным стандартом – IEEE 802.5, который описывает физический и канальный уровни сети. На физическом уровне стандарт определяет топологию сети в виде кольца, где каждый узел соединен с двумя соседними узлами с помощью экранированной витой пары (STP) или оптоволоконного кабеля. Стандарт устанавливает требования к характеристикам кабеля коннекторам. На канальном уровне стандарт описывает механизм доступа к среде передачи данных с помощью циркулирующего маркера (токена). Узел может передавать данные только тогда, когда он получает свободный токен. После завершения передачи узел освобождает токен, позволяя другим узлам передавать данные. Стандарт определяет формат кадров данных, процедуры управления доступом к среде и обработки ошибок. Соблюдение требований стандартов Token Ring гарантирует совместимость оборудования от разных производителей, обеспечивает детерминированный доступ к среде передачи данных и позволяет создавать надежные и эффективные локальные вычислительные сети. Однако из-за более высокой стоимости оборудования и меньшей гибкости по сравнению с Ethernet, технология Token Ring постепенно вытесняется из современных сетевых инфраструктур.

Стандарты Wireless LAN, разработанные IEEE, определяют спецификации для проектирования, развертывания и функционирования беспроводных сетей, обеспечивая их эффективность, безопасность и совместимость. Эти стандарты охватывают широкий спектр аспектов, включая радиочастотные диапазоны и модуляцию сигналов, протоколы множественного доступа и управления трафиком, форматы кадров данных, механизмы аутентификации и шифрования, методы обеспечения качества обслуживания, поддержку мобильности и роуминга, а также требования к производительности, таким как пропускная способность, радиус действия и задержка. Стандарты определяют правила совместимости между различными

устройствами и технологиями, позволяя беспроводным сетям беспрепятственно интегрироваться с проводными сетями и обеспечивать бесперебойную связь для пользователей в различных средах.

Стандарты TIA/EIA (Telecommunications Industry Association/Electronic Industries Alliance) регламентируют требования к кабельной инфраструктуре, включая категории кабелей, схемы разводки и методы тестирования. Стандарт TIA/EIA-568 является одним из наиболее широко используемых стандартов для структурированных кабельных систем в коммерческих зданиях. Он определяет требования к проектированию и установке телекоммуникационных кабельных систем, включая медные и оптоволоконные кабели, коннекторы, кроссы, шкафы и другие компоненты. Стандарт охватывает физическую топологию сети, схемы разводки, требования к производительности и методы тестирования. Основная цель TIA/EIA-568 – обеспечить единообразие, совместимость и качество кабельной инфраструктуры независимо от производителя компонентов.

Стандарты QoS (Quality of Service) используются для оптимизации работы сети и обеспечения надежности передачи данных, что особенно важно для приложений с высокими требованиями к качеству обслуживания, таким как видеоконференции, онлайн-игры и потоковое видео. Основные компоненты стандартов QoS включают в себя:

- возможность определения различных уровней приоритета для различных типов трафика. Например, голосовой трафик может иметь более высокий приоритет, чем обычные данные;
- управление доступной пропускной способностью сети, чтобы обеспечить равномерное распределение ресурсов и предотвратить перегрузки сети;
- управление задержками и вариациями задержек в сети, чтобы обеспечить качественное воспроизведение аудио и видео контента;
- возможность обеспечить надежную и гарантированную доставку критически важных данных без потерь или задержек.

Следование этим стандартам и рекомендациям помогает обеспечить совместимость, надежность, масштабируемость и безопасность ЛВС, а также упрощает процесс проектирования, внедрения и управления сетью. ИТ-специалисты, ответственные за создание и поддержку ЛВС, должны хорошо разбираться в соответствующих стандартах и уметь применять их на практике для обеспечения эффективной работы сети.

#### 1.4. Теоретические основы выбора оборудования для проектирования ЛВС

Теоретические основы выбора оборудования для проектирования локальной вычислительной сети включают в себя комплексный анализ различных факторов, влияющих на эффективность и производительность будущей сети [7]. При проектировании необходимо учитывать:

- Топологию – топология сети определяет физическое и логическое расположение узлов и их взаимосвязи. Выбор топологии влияет на тип используемого оборудования, его количество и расположение.

- Масштабируемость – масштабируемость сети предполагает возможность её расширения с минимальными затратами. Оборудование должно поддерживать увеличение числа пользователей и устройств без значительной потери производительности.

- Производительность – производительность сети зависит от пропускной способности каналов связи, быстродействия сетевых устройств и эффективности протоколов. Важно выбирать оборудование с учетом текущих и перспективных требований к скорости передачи данных.

- Надежность – надежность и отказоустойчивость обеспечиваются за счет дублирования критических компонентов использования устройств с высоким показателем наработки на отказ и внедрения механизмов балансировки нагрузки.

– Безопасность – безопасность сети реализуется путем выбора оборудования с поддержкой современных протоколов шифрования, возможностью сегментации сети и наличием встроенных функций межсетевого экранирования.

– Совместимость – совместимость оборудования различных производителей и поколений критически важна для обеспечения бесперебойной работы сети. Необходимо учитывать поддерживаемые стандарты и протоколы.

– Управляемость – управляемость сетевых устройств позволяет централизованно настраивать, мониторить и оперативно реагировать на изменения в работе сети. Следует отдавать предпочтение оборудованию с развитыми средствами управления;

– Стоимость – стоимость владения включает не только первоначальные затраты на приобретение, но и расходы на эксплуатацию, обслуживание и модернизацию.

Основными компонентами для проектирования локальной сети является сетевое оборудование (маршрутизаторы, коммутаторы, точки доступа WiFi) и среда передачи данных (кабели или беспроводная связь). Локальная сеть обеспечивает совместный доступ к интернету, общим файлам, принтерам и другим ресурсам, а также позволяет осуществлять обмен данными между устройствами внутри сети. Сетевое оборудование исполняет роль в функционировании современных сетей и делится на два типа, активное и пассивное. Активным сетевым оборудованием считаются устройства, которые выполняют обработку и передачу данных в сети, участвующие в процессе передачи и управления данными в сети [6].

#### 1. Маршрутизаторы (рис 7):

– функции – обеспечивают маршрутизацию данных между различными сетями, принимая решения о передаче данных на основе IP-адресов;

- применение – используются для соединения различных сетей (например, локальной сети с интернетом).



Рисунок 7 – Маршрутизатор

## 2. Коммутаторы (рис 8):

- функции – передача данных внутри одной сети обычно в локально вычислительной сети путем управления трафиком на основе MAC-адресов;
- применение – позволяют устройствам в локальной сети обмениваться данными между собой.



Рисунок 8 – Коммутатор

## 3. Маршрутизаторы с межсетевым экраном (рис 7):

- функции – обеспечивают безопасность сети путем фильтрации и контроля трафика, обеспечивая защиту от несанкционированного доступа из внешних сетей;

- применение – используются для обеспечения безопасности сети, особенно при подключении к интернету.

#### 4. Беспроводные точки доступа (рис 9):

- функции – создание беспроводной сети, обеспечивая доступ к сети через Wi-Fi;

- применение – используются для предоставления беспроводного доступа к сети в офисах, общественных местах и домах.



Рисунок 9 – Беспроводная точки доступа

#### 5. Модемы (рис 10):

- функции – преобразование цифровых сигналов компьютера в аналоговые сигналы, необходимые для передачи данных по аналоговым линиям связи и наоборот;

- применение – используются для подключения к интернету через телефонные линии, кабельное или DSL соединение.



Рисунок 10 – Модем

#### 6. Шлюзы (рис 11):

- функции – предоставляют интерфейс для связи между различными типами сетей с разными протоколами;
- применение – используются для подключения сетей различных технологий, например, соединения локальной сети с облачными сервисами или служит выходом в другую сеть.



Рисунок 11 – Шлюз

Пассивным сетевым оборудованием называются компоненты, которые не участвуют напрямую в обработке и передаче данных, а служат для организации физической инфраструктуры сети необходимой для прокладки кабельных систем и монтажа сети.

#### 1. Медный кабель (витая пара) (рис 11):

- 1) UTP (негерметизированный медный кабель):
  - функции – передача данных, работа в условиях электромагнитного излучения, UTP–кабели обычно дешевле и легче в установке по сравнению с другими типами кабелей;
  - применение – используется для прокладки в офисных зданиях, коммерческих помещениях и домашних сетях.
- 2) FTP (экранированный медный кабель):
  - функции – обладает защитой от внешних электромагнитных помех, часто используется в условиях высокой электромагнитной активности;
  - применение – сети передачи данных для использования внутри зданий, системы связи высокого разрешения.
- 3) STP (экранированный витой кабель):
  - функции – имеет металлический экран для защиты от электромагнитных помех, обеспечивает более высокую защиту, чем UTP;
  - применение – промышленность, высокоскоростная передача данных.

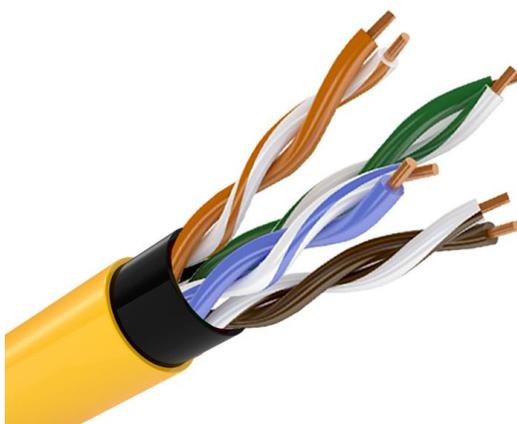


Рисунок 12 – Витая пара

2. Оптоволоконный кабель (рис 13):
  - 1) Одномодовый кабель:
    - функции – используется для передачи сигналов на большие расстояния, обеспечивает высокую пропускную способность;

– применение – трансокеанские и морские кабельные линии связи, наземные магистральные линии дальней связи, системы связи с высокой пропускной способностью, телекоммуникации и радиовещание [8].

2) Многомодовый кабель:

– функции – используется для передачи сигналов на более короткие расстояния в зданиях и помещениях;

– применение – 1-гигабитные приложения и расстояния до 550 м, Высокопроизводительные приложения, локальные сети с небольшой протяженностью [4].



Рисунок 13 – Оптоволоконный кабель

3. Коаксиальный кабель (рис 14):

– функции – используются для передачи высокочастотных сигналов и находят применение в звуковоспроизводящей, телевизионной, компьютерной и другой технике. Эта конструкция гарантирует, что сигналы, которые передаются по кабелю, остаются внутри и не подвергаются внешним электромагнитным помехам;

– применение – используются в разнообразных областях для передачи радиочастотных сигналов.



Рисунок 14 – Коаксиальный кабель

4. Разъемы и коннекторы (рис 15):

– функции – используются для соединения кабелей с активным оборудованием и другими компонентами;

– применение – используются в электронике и телекоммуникациях для обеспечения физического соединения между различными устройствами и компонентами.

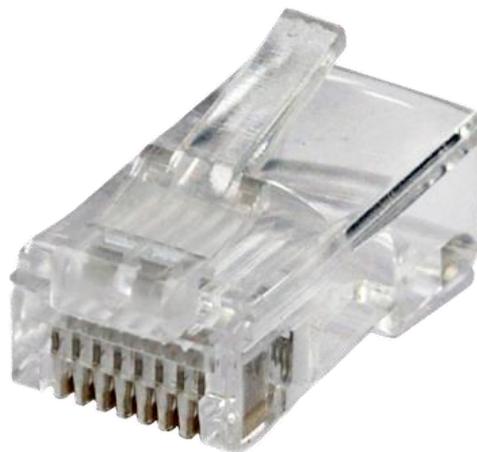


Рисунок 15 – Коннектор RJ-45

5. Патч-панели (рис 16):

- функции – служат для упорядоченного монтажа и организации кабельных соединений;
- применение – коммутация сети, структурированные кабельные системы, объединение сетевых соединений, управление кабелями.



Рисунок 16 – Патч-панели

6. Розетки и гнезда (рис 17):

- функции – предназначены для подключения конечных устройств (компьютеров, принтеров и т.д.) к сети;
- применение – различные типы и модели розеток позволяют подключить электроприборы различного типа и мощности. Кроме того, они обеспечивают безопасность электроустановок и их пользователей, предотвращая короткие замыкания и перегрузки.



Рисунок 17 – Сетевая розетка

7. Кроссовые стойки и шкафы (рис 18):

– функции – используются для размещения и организации активного и пассивного сетевого оборудования;

– применение – предназначены для упорядоченного соединения, размещения и управления различными кабелями и проводами в коммуникационных системах. Они обеспечивают объединение и распределение кабелей, упрощая управление кабельной инфраструктурой и облегчая доступ к проводам для обслуживания и диагностики.



Рисунок 18 – Серверный шкаф

8. Кабельные организаторы и лотки (рис 19):

– функции – облегчают прокладку и группировку кабелей внутри стоек и шкафов;

– применение – используются для аккуратной укладки и администрирования кабелей в различных технических помещениях, что облегчает процесс монтажа, ремонта и обслуживания кабельной инфраструктуры.



Рисунок 19 – Кабельные организаторы

9. Соединительные муфты и сплиттеры (рис 20):

– функции – служат для объединения или разветвления кабельных линий;

– применение – соединительные муфты используются для обеспечения безопасного соединения отдельных отрезков кабеля в процессе его монтажа или ремонта. Сплиттеры используются для разделения сигнала на несколько потоков или для объединения нескольких сигналов в один.



Рисунок 20 – Соединительная муфта

Все эти пассивные компоненты не принимают участия в обработке данных, лишь являются важными элементами для создания физической инфраструктуры компьютерной сети и обеспечения соединений между активными устройствами. Каждый вид прокладки ЛВС имеет свои преимущества и ограничения, и выбор определенного вида зависит от

требований к пропускной способности, длине прокладки, условиях эксплуатации и бюджета. Учет теоретических основ при выборе оборудования для ЛВС позволяет создать надежную, эффективную и адаптивную сетевую инфраструктуру, способную удовлетворить потребности и обеспечить потенциал для будущего развития.

## 2. ПРОЕКТНЫЙ РАЗДЕЛ

### 2.1. Аналитическая часть

До начала выполнения разработки приложения выбора и оценки стоимости оборудования для проектирования ЛВС составим техническое задание:

#### 1. Введение:

Разрабатываемое приложение поможет автоматизировать и упростить процесс выбора подходящего оборудования для ЛВС на основе требований и спецификаций проекта. Программное обеспечение поможет сэкономить время и усилия инженеров-проектировщиков. Приложение позволит быстро оценить стоимость необходимого оборудования и выбрать оптимальные компоненты с учетом бюджетных ограничений проекта. Это поможет эффективно распределять финансовые ресурсы. С его помощью пользователи смогут быстро и удобно подобрать необходимое оборудование, оценить его стоимость и сделать правильный выбор, исходя из требований и бюджета проекта.

Таким образом, разработка приложения для выбора и оценки стоимости оборудования ЛВС будет способствовать экономии времени и средств, обеспечению совместимости компонентов и повышению общей эффективности работы проектировщиков локальных вычислительных сетей.

#### 2. Требования:

##### 2.1. Функциональные требования:

- работа в офлайн режиме на персональном компьютере без необходимости подключения к сети интернет;
- определение рынка оборудования для ЛВС – необходимо внедрить адаптивную базу данных для выбора активного сетевого оборудования (маршрутизаторы, коммутаторы, устройства беспроводной связи) и

пассивного сетевого оборудования (сетевые шкафы, серверные стойки, кабели, патч-корды) – тип оборудования, бренд, поставщик, стоимость:

- необходимо внедрить ввод количества выбранного сетевого оборудования;
- нужно осуществить возможность поиска оборудования из БД для сокращения затрат рабочего времени;
- должна быть осуществлена возможность сравнения характеристик оборудования;
- необходимо внедрить алгоритм получения общей стоимости проектирования ЛВС после ввода данных исследования;
- следует минимизировать возможные допустимые ошибки пользователя за счёт интуитивно понятного интерфейса программы.

## 2.2. Нефункциональные требования:

- приложение должно обеспечивать быстрый отклик и оптимальное время отображения данных;
- защита данных и конфиденциальной информации;
- необходимо визуализировать интуитивно понятный и простой в использовании интерфейс;
- приложение должно работать на различных операционных системах (Windows, macOS, Linux).

## 3. Среда разработки:

Для хранения информации об оборудовании было принято решение использовать Microsoft SQL Server 2022. Она предоставляет комплексное решение для хранения, обработки и анализа данных в масштабах предприятия.

Эта версия СУБД отличается улучшенной производительностью, расширенными возможностями безопасности и интеграцией с облачными технологиями. SQL Server 2022 поддерживает работу как с реляционными, так и с нереляционными данными, включая графовые базы данных и большие данные.

Среди ключевых особенностей можно отметить поддержку интеллектуального анализа данных, машинного обучения и искусственного интеллекта непосредственно внутри базы данных. Система также предлагает улучшенные инструменты для обеспечения высокой доступности и аварийного восстановления. SQL Server 2022 хорошо интегрируется с другими продуктами Microsoft и предоставляет расширенные возможности для разработчиков, включая поддержку контейнеров и улучшенную совместимость с открытым исходным кодом. Эта СУБД разработана для обеспечения высокой производительности и масштабируемости, что делает ее подходящей для организаций различного размера - от небольших компаний до крупных корпораций.

Данные организованы в виде таблиц, состоящих из строк (записей) и столбцов (полей). графический интерфейс программы имеет следующие плюсы:

- использует язык запросов SQL;
- полностью совместим с операционной системой Windows;
- импорт и экспорт имеет широкий спектр возможностей, от таблиц Excel и текстовых файлов, до практически любой серверной СУБД [20].

По результатам тщательного анализа доступных инструментов разработки для реализации проекта была выбрана интегрированная среда разработки Microsoft Visual Studio 2022. Согласно современным исследованиям в области программной инженерии, данный продукт является передовым решением для разработки, предоставляющим обширный набор инструментов для проектирования и создания высокофункциональных приложений, которые способны удовлетворить самым взыскательным требованиям заказчиков [10]. Основываясь на критериях производительности и удобства использования, в качестве основного языка программирования был выбран C#. Выбранный язык представляет собой современный объектно-ориентированный язык программирования [9]. Он был создан как язык для разработки приложений в рамках платформы .NET и предназначен для

упрощения создания разнообразных программных продуктов, таких как мобильные веб-приложения, игры, программы для рабочих станций и серверов. C# поддерживает широкий набор современных концепций программирования, таких как обобщенное программирование, асинхронное программирование, безопасность типов и автоматическую сборку мусора [13].

#### 4. Приложения:

Для более полного понимания требований к функционалу программы воспользуемся методологией IDEF0. Методология IDEF0 (Integration Definition for Function Modeling) является широко используемым подходом к моделированию и анализу систем. Она предоставляет структурированный и систематический способ описания и визуализации функциональности системы, ее компонентов и взаимосвязей между ними. Основная цель IDEF0 - создать ясное и понятное представление о том, как работает система, какие функции она выполняет и как эти функции связаны между собой. Методология использует иерархическую декомпозицию, при которой система разбивается на подсистемы и функциональные блоки.

IDEF0 использует графическую нотацию, состоящую из блоков и стрелок. Каждый блок представляет собой функцию или активность системы, а стрелки отображают потоки данных, материалов или управляющих сигналов между блоками. Входные стрелки слева представляют данные или материалы, необходимые для выполнения функции, выходные стрелки справа - результаты или продукты функции. Управляющие стрелки сверху отображают условия или ограничения, влияющие на выполнение функции, а стрелки снизу - механизмы или ресурсы, используемые для ее реализации.

Применение методологии IDEF0 помогает достичь нескольких ключевых целей. Во-первых, она позволяет создать ясное и структурированное описание системы, понятное всем заинтересованным сторонам. Это облегчает коммуникацию между участниками проекта, включая заказчиков, аналитиков, разработчиков и других стейкхолдеров. Во-вторых, IDEF0 помогает выявить и проанализировать требования к системе,

определить ее границы и интерфейсы с внешней средой. Это позволяет избежать неоднозначностей и пробелов в понимании функциональности системы. Контекстная диаграмма и диаграмма декомпозиции являются ключевыми элементами методологии IDEF0 (Integration Definition for Function Modeling) широко используемой для моделирования и анализа систем. Они помогают структурировать и визуализировать сложные системы, облегчают коммуникацию между заинтересованными сторонами и служат основой для дальнейшей разработки и реализации системы.

Контекстная диаграмма представляет собой высокоуровневое описание системы и ее взаимодействия с внешней средой. Она отображает систему в виде единого блока, связанного с внешними сущностями (инженерами, ПО), которые взаимодействуют с системой. Стрелки, входящие в блок и выходящие из него, представляют собой потоки данных, материалов или управляющих сигналов между системой и внешними сущностями. Контекстная диаграмма позволяет очертить границы системы, определить ее основные входы и выходы, а также выявить внешние факторы, влияющие на ее функционирование.

Построим контекстную диаграмму (диаграмма 1) в веб-приложении draw.io [18]. Draw.io – бесплатный онлайн-сервис, который помогает создавать блок-схемы, прототипы, инфографику и диаграммы любого вида. Чаще всего его используют именно для построения диаграмм.

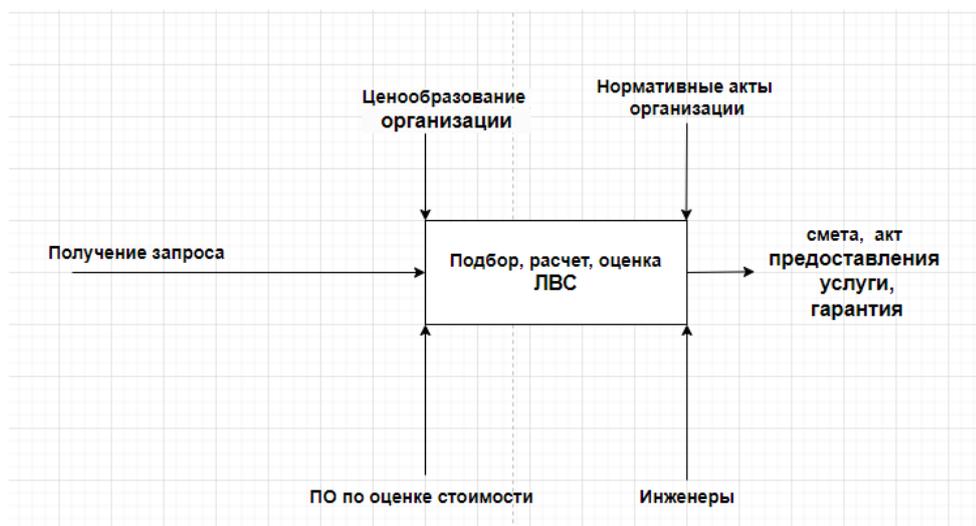


Диаграмма 1 – Контекстная диаграмма

Диаграмма декомпозиции, в свою очередь, представляет собой детализацию системы, отображенной на контекстной диаграмме. Она разбивает систему на подсистемы или функциональные блоки, каждый из которых выполняет определенную роль в рамках общей функциональности системы. Диаграмма декомпозиции строится по иерархическому принципу, где каждый блок может быть далее декомпозирован на более детальные уровни. Связи между блоками показывают потоки данных или управляющие сигналы, которыми они обмениваются. Выбранный тип диаграммы позволит визуализировать процесс проектирования ЛВС для дальнейшей разработки приложения по выбору и оценки стоимости оборудования, поможет определить входы и выходы каждого этапа, а также механизмы управления и исполнения.

Построим диаграмму декомпозиции (диаграмма 2) все в том же веб-приложении.

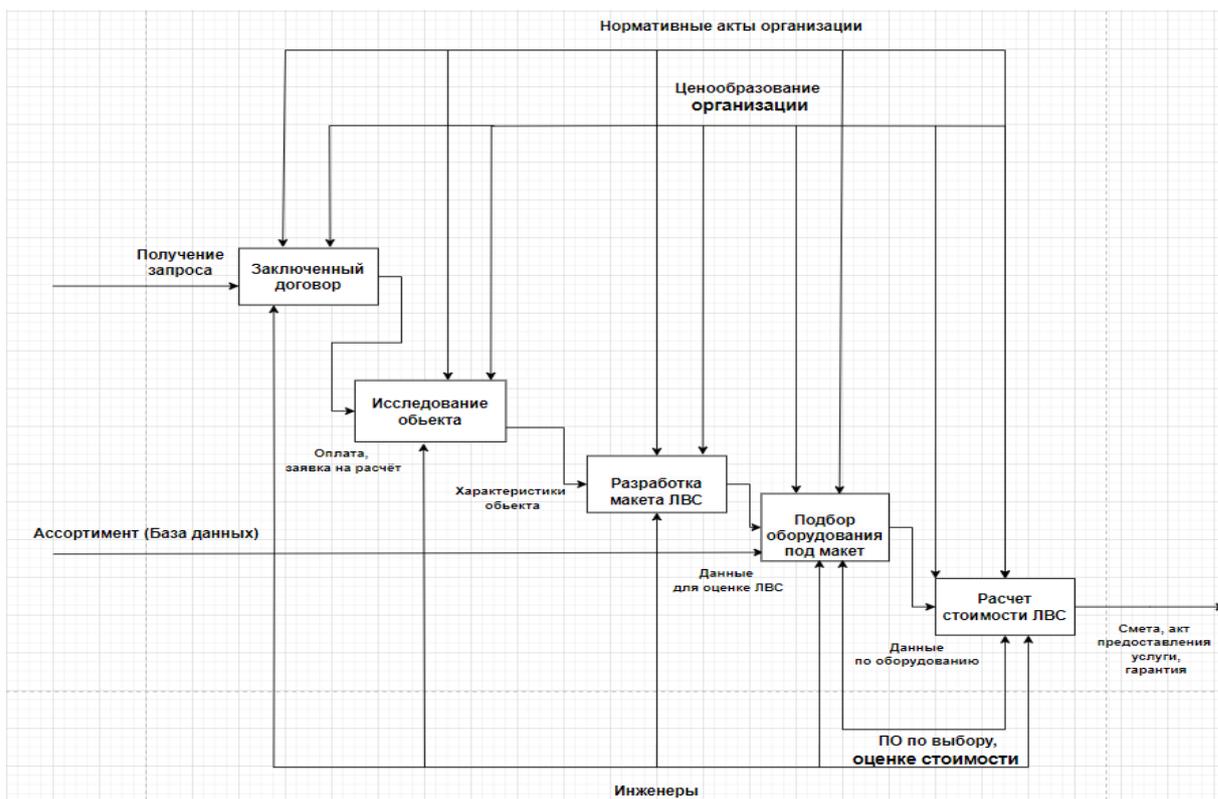


Диаграмма 2 – Декомпозиция

Построение диаграммы начинается с размещения блоков в порядке их логической последовательности слева направо. Каждый блок представляет собой прямоугольник с названием соответствующего процесса внутри.

Первым располагается блок «Заклучение договора». Его входами являются потребности заказчика и предварительные требования к сети. Выходом становится подписанный договор на проектирование ЛВС, оплата. Управляющими факторами выступают юридические нормы и внутренние регламенты компании. Механизмами служат инженеры по работе с клиентами.

Следующий блок – «Исследование объекта». Он принимает на вход информацию из договора и данные о существующей инфраструктуре заказчика. Выходом являются характеристики обследования: планы помещений, схемы размещения рабочих мест, требования к производительности и надежности сети. Управление осуществляется на основе методик обследования и стандартов информационной безопасности.

Механизмами выступают инженеры-проектировщики и инструменты измерения и диагностики.

Третьим располагается блок «Разработка макета ЛВС». Его входными данными служат результаты исследования объекта. На выходе формируется концептуальная схема сети с указанием топологии, предварительного расположения и типов устройств. Собранные данные инженер занесёт в разработанное приложение для дальнейшего выбора и оценки стоимости оборудования. Управляющими элементами являются принципы построения сетей и отраслевые рекомендации. Механизмы включают САПР и экспертные знания инженеров.

Далее следует блок «Подбор оборудования под макет». Он использует разработанный макет ЛВС в качестве входных данных. Выходом становится спецификация активного, пассивного сетевого оборудования с конкретными моделями и их характеристиками, оценочная стоимость проектирования ЛВС. Управление основывается на технических регламентах и совместимости компонентов. Механизмом служит разработанное программное обеспечение, которое содержит каталоги производителей, систему поиска и сравнения оборудования для проектирования ЛВС, и инженеры.

Завершающим является блок «Расчет стоимости ЛВС». Его вход - спецификация оборудования, а выход - детальная смета проекта. Управляющими факторами выступают ценовая политика поставщиков и приложение расчета стоимости внедрения оборудования. Механизм включает в себя разработанное приложение для составления финансовых расчетов.

Между блоками устанавливаются связи в виде стрелок, показывающие, как выходы одних процессов становятся входами для других. Например, договор из первого блока служит входом для исследования объекта, а макет ЛВС - для подбора оборудования в приложении.

Кроме того, на диаграмме отображаются обратные связи, когда результаты последующих этапов могут потребовать корректировки предыдущих. Так, расчет стоимости может привести к пересмотру выбора

оборудования, что отражается стрелкой, идущей от блока расчета стоимости к блоку подбора оборудования.

Полученная диаграмма декомпозиции наглядно демонстрирует структуру процесса проектирования ЛВС, обеспечивает понимание взаимосвязей между этапами и помогает в управлении проектом, выявляя критические точки и зависимости.

## 2.2. Исследование организации ООО «Сфера Ай Ти»

Объектом анализа для выбора и оценки стоимости оборудования для проектирования ЛВС была установлена организация ООО «Сфера Ай Ти».

Целью исследования является получение общей оценочной стоимости выбранного сетевого оборудования для проектирования ЛВС с помощью разработанного приложения. Для ясного представления о том, как работает организация, какие функции и требования она хочет реализовать необходимо выполнить следующее:

- провести исследование объекта;
- подобрать топологию сети;
- спроектировать логическую схему сети;
- собрать все необходимые значения для оценки и выбора

необходимого оборудования.

Произведем исследование для выбора и оценки стоимости оборудования на примере организации ООО «Сфера Ай Ти», оказывающую технические услуги. Организация строится по линейной структуре (диаграмма 3). Линейная организационная структура управления, по сути, характеризуется тем, что, безусловно, во главе каждого структурного подразделения находится руководитель-единоначальник, наделенный всеми полномочиями и фактически осуществляющий единоличное руководство подчиненными ему работниками и, несомненно, сосредоточивающий в своих руках все функции управления.

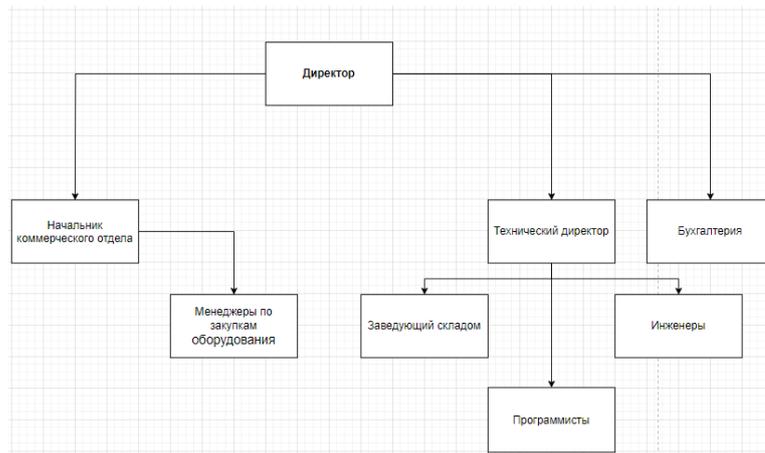


Диаграмма 3 – Линейная структура

В штате организации шестнадцать человек, размещенные на двух этажах. На втором этаже находится серверная, два отдела, в каждом четыре рабочих места, на каждый отдел выделен кабинет, три сетевых телефона. На первом этаже расположен кабинет директора, отдел коммерции, кабинет бухгалтеров. В отделе коммерции отдельный кабинет, четыре рабочих места, четыре сетевых телефона, в кабинете бухгалтеров три рабочих места, три сетевых телефона (рис. 21).

Далее, как ознакомимся с архитектурой здания и количеством рабочих мест, выберем топологию построения сети. Для организации, расположенной на двух этажах, наиболее подходящей будет топология сети «Звезда» [16]. Потому что в данной топологии вся сеть организована вокруг центрального узла все рабочие станции и другие сетевые устройства подключаются к этому центральному узлу, если одна из рабочих станций выйдет из строя, это не повлияет на работу остальной сети, так как каждая станция подключена независимо к центральному узлу. Благодаря централизованной структуре администрирование и мониторинг сети значительно упрощаются [17]. Если организация начнет расширяться выбранная топология легко масштабируется. Спроектируем сеть организации согласно выбранной топологии (рис. 21).

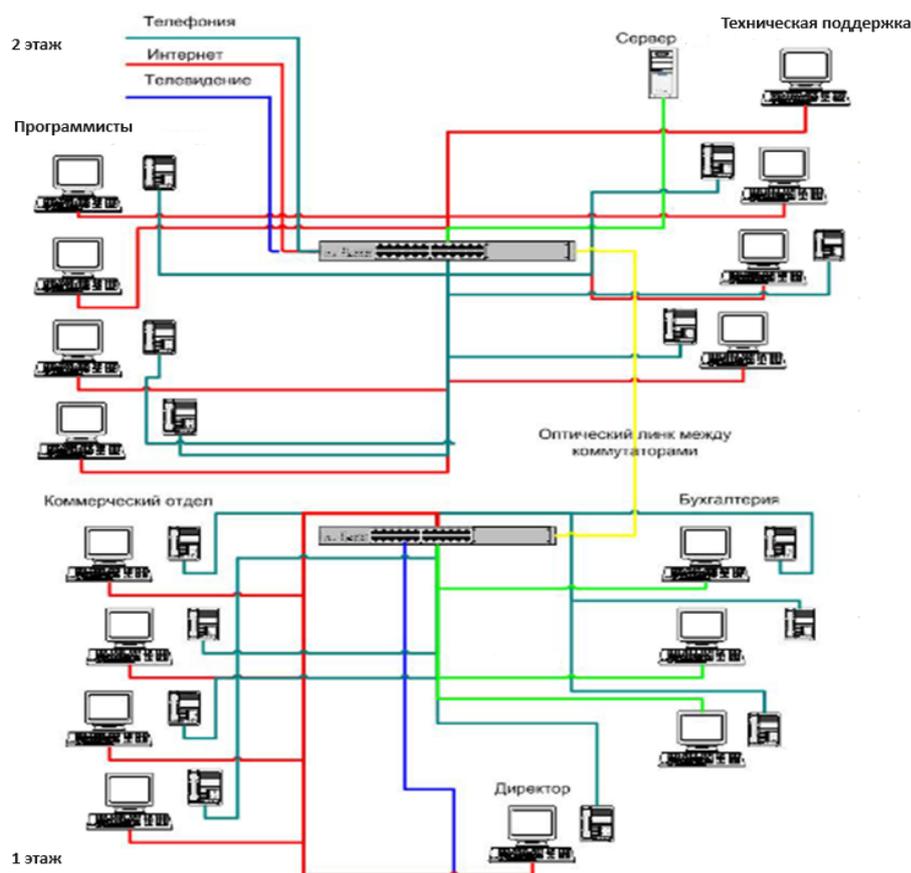


Рисунок 21 – Схема ЛВС организации

Для организации максимально бюджетной, но эффективной локально вычислительной сети, согласно спроектированной схеме исследования ООО «Сфера Ай Ти» понадобится два управляемых коммутатора на 24 порта. Активное сетевое оборудование выбрано с запасом по количеству портов на случай расширения организации. Два серверных шкафа небольшого размера для безопасного хранения коммутаторов. Понадобится сорок метров кабель-канала для упорядоченного монтажа и организации кабельных соединений. Шестнадцать сетевых розеток к каждому рабочему месту для безопасности электроустановок и их пользователей. Сетевые розетки предотвращают короткие замыкания и перегрузки ЛВС. Сто штук разъемов RJ-45 для обжима протянутого кабеля. Кабинеты расположены недалеко друг от друга поэтому принимаем решение прокинуть наружным способом прокладки негерметизированный медный кабель UTP, соответствующая стандарту IEEE

802.3. УТР–кабели обычно дешевле и легче в установке по сравнению с другими типами кабелей, поддерживают скорость передачи данных от 10 Мбит/с до 100 Гбит/с, передают поток даже в условиях электромагнитного излучения. По результатам замеров понадобится около двух ста метров кабеля. Для связи центрального коммутатора с коммутатором нижнего этажа, выберем оптоволокно – многомодовый кабель, который протянем соответствуя стандарту 802.3u. Он обеспечивает самый высокочастотный поток связи. Выбранный вариант прокладки – под полом. По результатам замеров понадобится около десяти метров многомод-кабеля.

### 2.3. Разработка приложения

Создадим базу данных в Microsoft SQL Server 2022. База данных создается в пару кликов, а для ее наполнения запишем SQL запрос, который поможет создать нужные таблицы (рис.22).

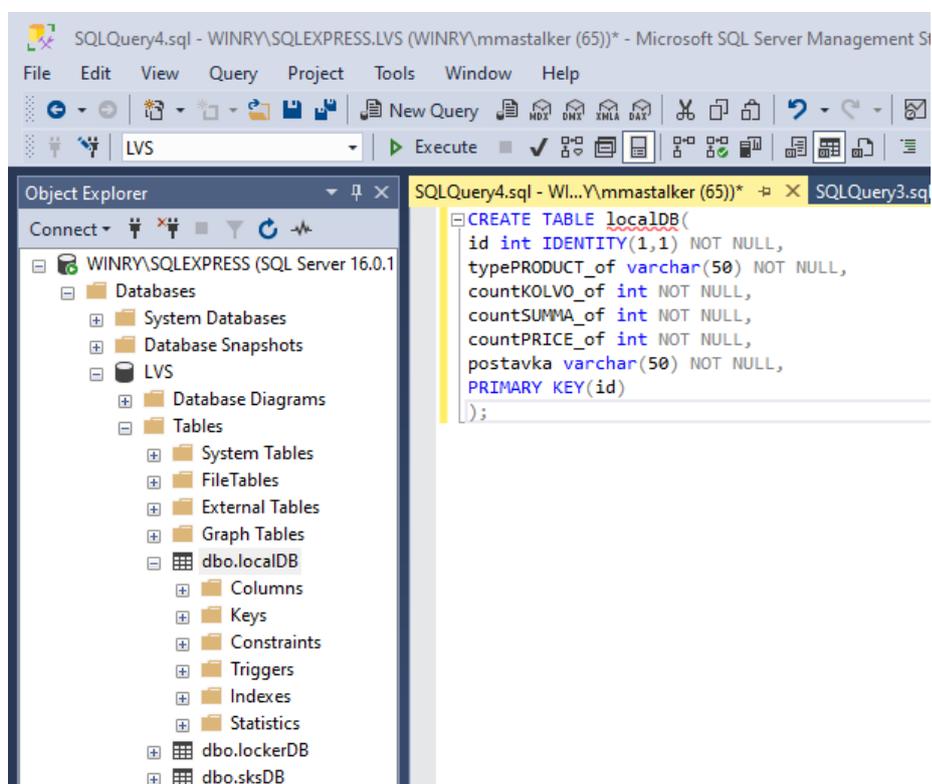


Рисунок 22 – Создание таблицы

Аналогичным образом создадим в базе данных еще две таблицы, добавим в них переменные. Были разработаны следующие таблицы:

– активное сетевое оборудование – содержит данные по оборудованию, которое предлагается пользователю для активной прокладки ЛВС (коммутаторы, роутеры и иные маршрутизаторы) (рис.23);

	ID активного сетевого оборудования	Тип маршрутизатора	Цена	Модель	Поставщик
▶	2	Беспроводная т...	7000	TP-Link EAP115	ModifiedNew
	3	Беспроводная т...	9000	MikroTik 69	ModifiedNew
	4	Коммутатор упр...	16000	Cisco SG200	ModifiedNew
	5	Беспроводная т...	9000	MikroTik 69	ModifiedNew
	6	Коммутатор неу...	1000	D-Link DAP-1360	ModifiedNew
	7	Коммутатор неу...	2500	D-Link DAP-1760	ModifiedNew
	8	Коммутатор упр...	7000	Cisco SG200	ModifiedNew
	9	Роутер	1000	Xiaomi Mi Wi-Fi R...	ModifiedNew
	10	Роутер	2000	TP-Link Archer C7	ModifiedNew

Рисунок 23 – Активное сетевое оборудование

– пассивное сетевое оборудование (рис.24) – содержит данные по оборудованию, которое предлагается пользователю для пассивной прокладки ЛВС (патч-панели, розетки, разъемы, кроссовые шкафы и стойки);

	ID пассивного сетевого оборудования	Тип пассивного сетевого оборудования	Цена	Модель	Поставщик
▶	1	Сетевой шкаф	3000	TP-Link	ModifiedNew
	2	Сетевой шкаф/...	6000	Keystone	ModifiedNew
	3	Сетевой шкаф/...	9000	5Bites	ModifiedNew
	4	Сетевой шкаф/...	12000	Keystone	ModifiedNew
	5	Сетевая розетка	300	TP-Link	ModifiedNew
	6	Серверные стой...	12000	5Bites	ModifiedNew
	7	Разъемы RJ-45...	1000	TP-Link	ModifiedNew
	8	Кабель канал/1м	500	Keystone	ModifiedNew
	9	Патч-панель/24...	1000	Cabeus	ModifiedNew
	10	Патч-панель/16...	650	Cabeus	ModifiedNew
	11	Патч-панель/64...	3500	5Bites	ModifiedNew

Рисунок 24 – Пассивное сетевое оборудование

– вариации прокладки ЛВС – содержит типы прокладки ЛВС (прокладка по воздуху, прокладка под полом, прокладка в стенах, наружная прокладка, беспроводная ЛВС) (рис.25).

ID прокладки ЛВС	Вариант прокладки ЛВС	Цена(руб)/1метр	Модель	Поставщик
2	Прокладка по в...	100	Витая пара-нез...	ModifiedNew
3	Прокладка по в...	150	Витая пара-экр...	ModifiedNew
4	Прокладка под ...	200	Витая пара-нез...	ModifiedNew
5	Прокладка под ...	250	Витая пара-экр...	ModifiedNew
6	Прокладка в ст...	200	Витая пара-нез...	ModifiedNew
7	Прокладка в ст...	250	Витая пара-экр...	ModifiedNew
8	Наружная прок...	50	Витая пара-нез...	ModifiedNew
9	Наружная прок...	75	Витая пара-экр...	ModifiedNew
10	Наружная прок...	50	Витая пара-нез...	ModifiedNew
11	Беспроводная ...	15	Wi-fi	ModifiedNew
12	Прокладка по в...	100	Оптическое вол...	ModifiedNew

Рисунок 25 – Вариации прокладки

Функция созданных таблиц – обеспечение сохранения данных, адаптация под поступление товара.

Для создания приложения в среде Visual Studio на языке C# выполним следующую последовательность Файл → Создать → Проект. В открывшемся окне осуществим выбор типа проекта: приложение Windows forms.(.NET Framework) (рис.26).

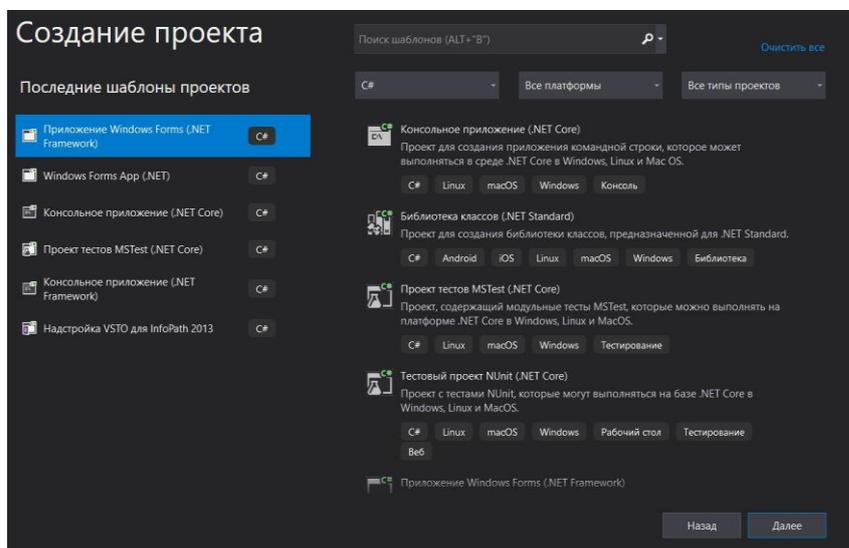


Рисунок 26 – Диалоговое окно

Когда форма будет добавлена, откроется окно Form1.cs[Design] (рис. 27).

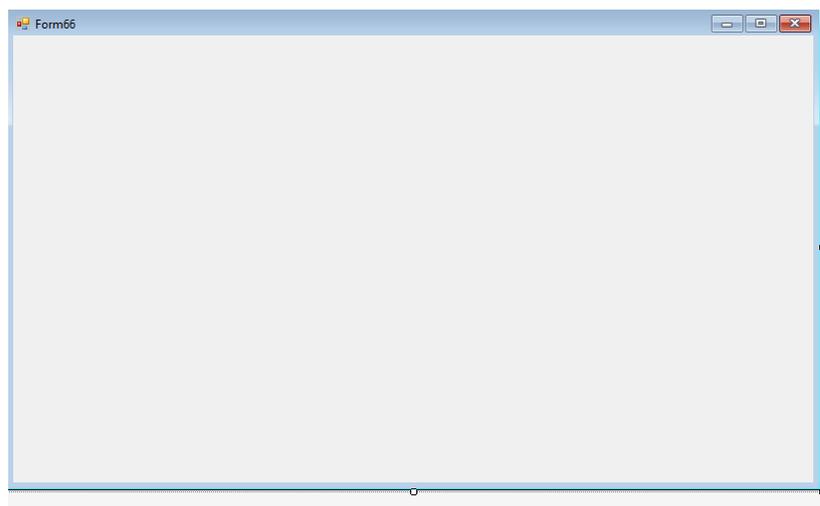


Рисунок 27 – Создание приложения

Для начала создадим подходящий дизайн, добавим компоненты Button, разработаем возможность перехода в другие окна программы (рис.28).



Рисунок 28 – Титульная форма приложения

Button – метафора кнопки в технике внешне схожа и выполняет аналогичные задачи. При нажатии на неё происходит программно-связанное с этим нажатием действие либо событие [15]. Рассмотрим возможности всех трёх кнопок, начиная снизу. С помощью кнопки «Выход из программы» разработана возможность выхода из программы (рис 29).



Рисунок 29 – Выход из программы

С помощью кнопки «Ассортимент оборудования» переходим в новое окно. В нем разработаем подключение к базе данных Microsoft SQL Server 2022 [15] (рис 30).

```
дизайн  Тест  Анализ  Средства  Расширения  Окно  Справка  Поиск -  WindowsFormsApplication1
- Пуск -
LVDataSet.xsd  Form1_passive.cs  rating2.cs  DB3.Designer.cs  DB3.cs [Конструктор]  Form3.Designer.cs
rating2.Designer.cs  Form1_passive... [Конструктор]  Form2.cs [Конструктор]  rating2.cs [Конструктор]  Form1.Desig
WindowsFormsApplication1
WindowsFormsApplication1.DB3
InitializeComponent();
}
private void CreateColumns()
{
    dataGridView1.Columns.Add("id", "ID активного сетевого оборудования");
    dataGridView1.Columns.Add("typePRODUCT_of", "Тип оборудования");
    dataGridView1.Columns.Add("countPRICE_of", "Цена");
    dataGridView1.Columns.Add("postavka", "Модель");
    dataGridView1.Columns.Add("raspise", "Источники");
}
private void ReadSingleRow(DataGridView dvw, IDataRecord record)
{
    dvw.Rows.Add(record.GetInt32(0), record.GetString(1), record.GetInt32(2), record.GetString(3), RowState.Modified);
}
private void RefreshDataGridView(DataGridView dvw)
{
    dvw.Rows.Clear();
    string queryString = $"select * from localDB";
    SqlCommand command = new SqlCommand(queryString, database.GetConnection());
    database.OpenConnection();
    SqlDataReader reader = command.ExecuteReader();
    while (reader.Read())
    {
        ReadSingleRow(dvw, reader);
    }
    reader.Close();
}
private void menuItem1_ItemClicked(object sender, ToolStripItemClickedEventArgs e)
{
}
private void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e)
{
    selectedRow = e.RowIndex;
    if (e.RowIndex >= 0)
    {
        DataGridViewRow row = dataGridView1.Rows[selectedRow];
        textBox_PRODUCT.Text = row.Cells[0].Value.ToString();
        textBox_PRICE.Text = row.Cells[1].Value.ToString();
        textBox_postavka.Text = row.Cells[2].Value.ToString();
        textBox_id.Text = row.Cells[3].Value.ToString();
    }
}
```

Рисунок 30 – Подключение БД

Первые две таблицы содержат данные о закупленном сетевом оборудовании на данный момент. В третьей таблице справа находятся данные о кабеле за 1 метр в зависимости от вариации прокладки ЛВС (рис 31).

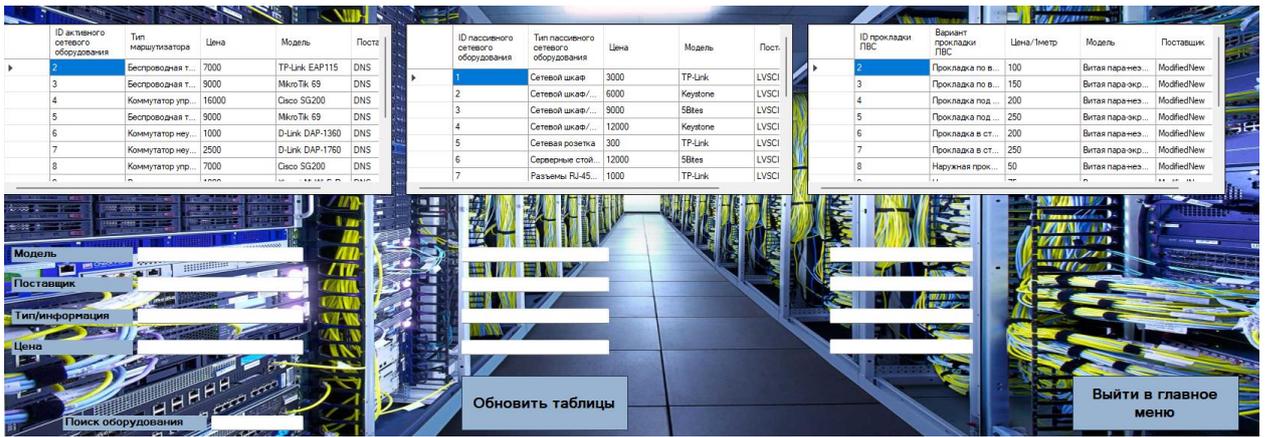


Рисунок 31 – Ассортимент оборудования

После успешного подключения базы данных визуализируем три таблицы в созданном окне «Ассортимент оборудования» добавим возможность клика по какому-либо столбцу таблицы для вывода полной информации для более полного ознакомления с оборудованием (рис 32).

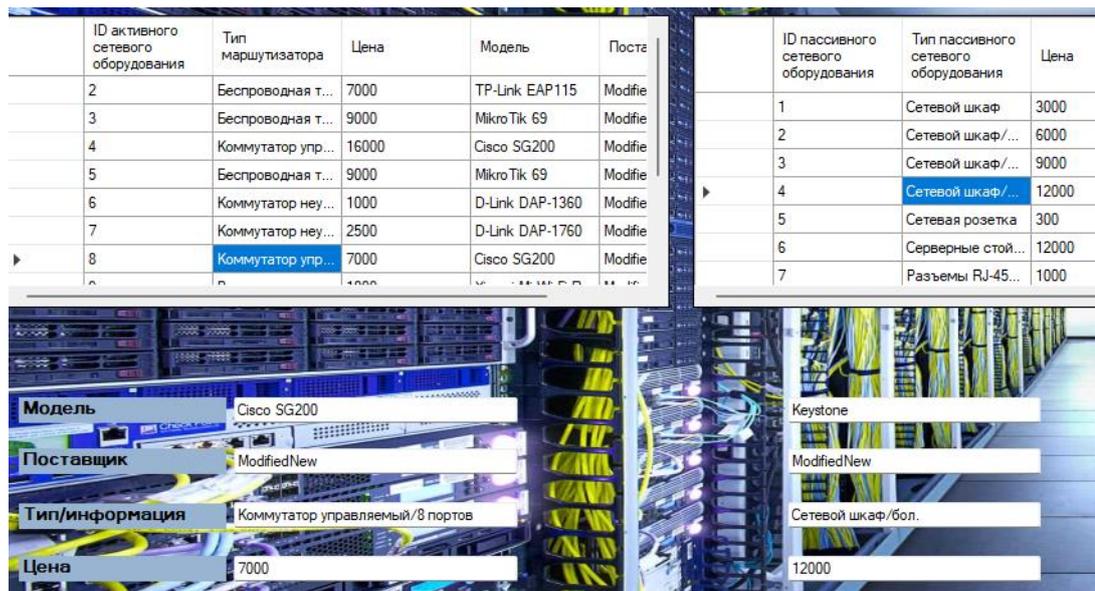


Рисунок 32 – Информация по щелчку

Добавим способ обновления базы данных, если вдруг поступило сетевое оборудование в реальном времени в созданные таблицы (рис 33).

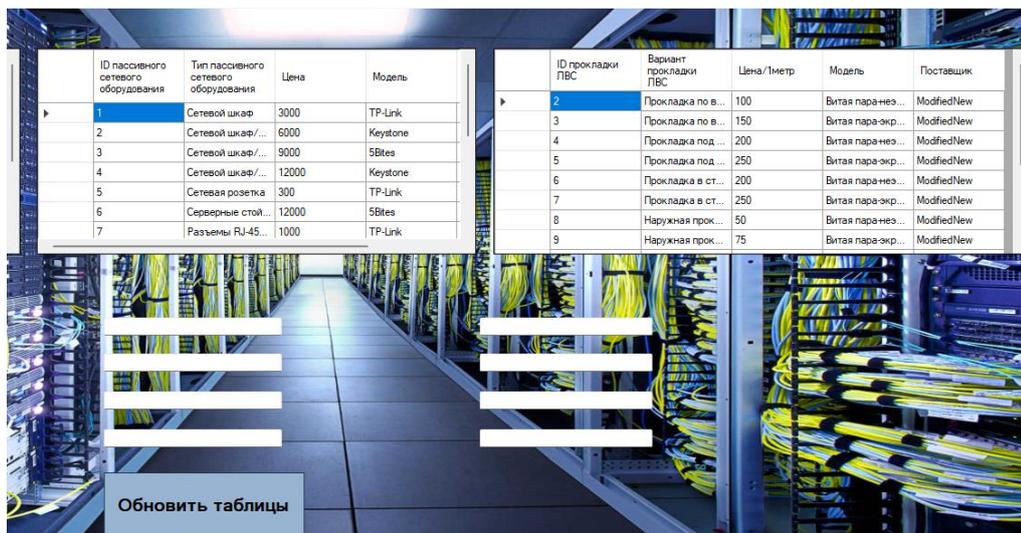


Рисунок 33 – Обновление БД

Разработаем возможность поиска информации (рис.34).

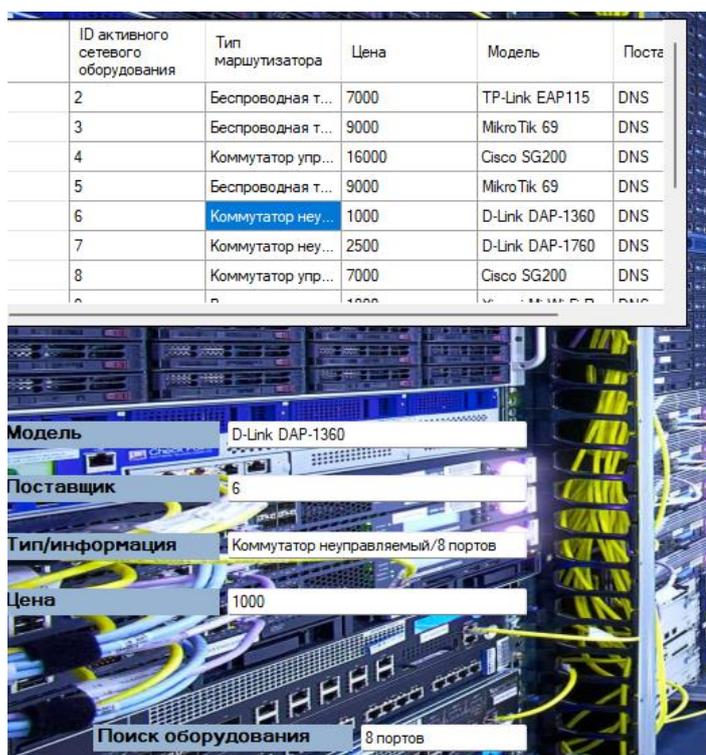


Рисунок 34 – Поиск

Добавим способ выхода в главное меню после ознакомления с ассортиментом оборудования. Для этого создадим кнопку «Выйти в главное меню» и внедрим данную возможность (рис 35).

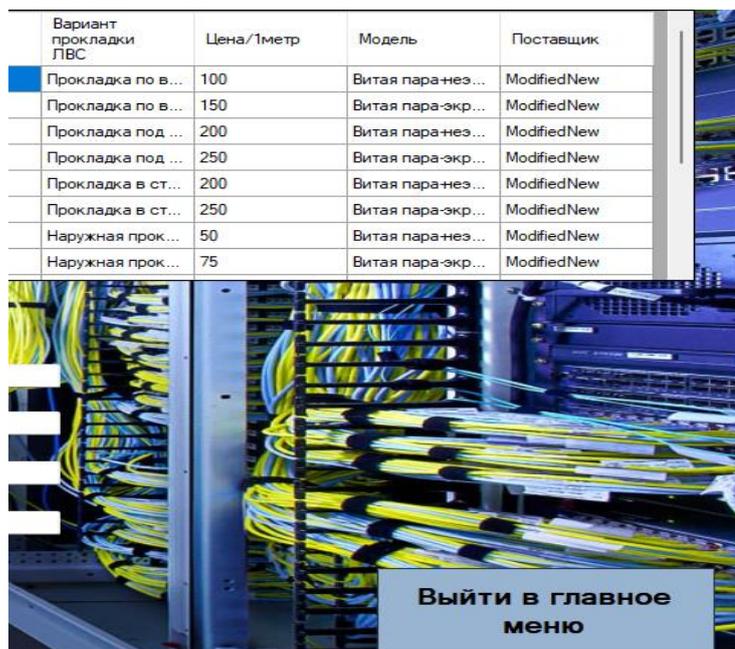


Рисунок 35 – Выход в главное меню

После успешного ознакомления с ассортиментом оборудования выйдем в главное меню. С помощью кнопки «Оценка и выбор оборудования» переходим в основную часть программы. В данном окне начинается оценка стоимости проектирования ЛВС в зависимости от выбора оборудования (рис 36).



Рисунок 36 – Выбор прокладки сети

Так как программа будет использоваться для проектирования больших и объемных локально вычислительных сетей, состоящих из

множества вариаций прокладок сети внутри одного проекта, разработаем возможность добавления дополнительных вариантов прокладки ЛВС (рис 1). После нажатия на кнопку окно с выбором типа кабеля повторит цикл, сохранив введенные данные о первом оборудовании. Таким образом, добавляем в оценку стоимости множество вариаций прокладки сети (рис 37).



Рисунок 37 – Дополнительный тип прокладки

Учтем возможность того, что пользователю не нужно проектировать полную ЛВС, а нужна пара позиций или вообще нет нужды выбирать и оценивать ЛВС. При помощи данной кнопки разработаем расчет прокладки сети, не доходя до конечной формы (рис.38).

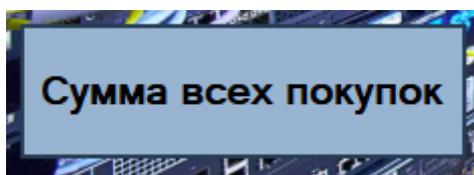


Рисунок 38 – Сумма покупок

После перехода по данной кнопке появится окно – сумма всех покупок, которое будет обозначать не до конца оцененную локально вычислительную сеть (рис.39). Чтобы рассчитать оценочную стоимость для проектирования ЛВС, нужно пройти все окна с выбором оборудования.

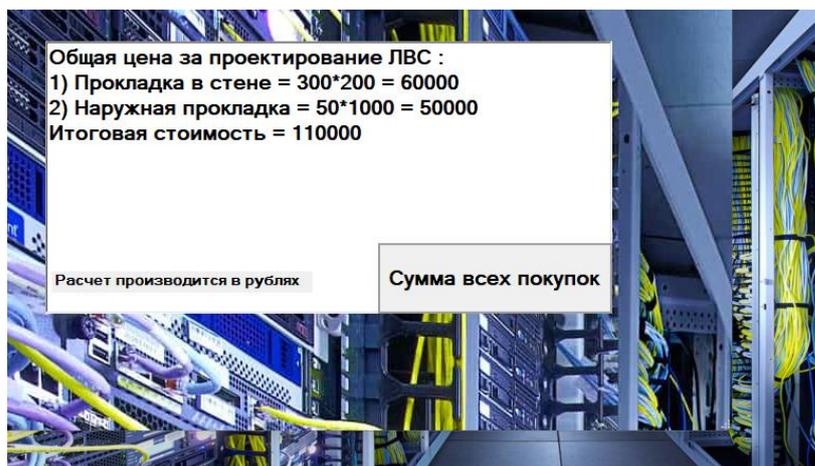


Рисунок 39 – Сумма покупок 2

Если пользователь не начал ни один из оценочных расчетов и хочет узнать сумму всех покупок, то, переходя по кнопке получит информацию позиция не выбрана (рис.40).

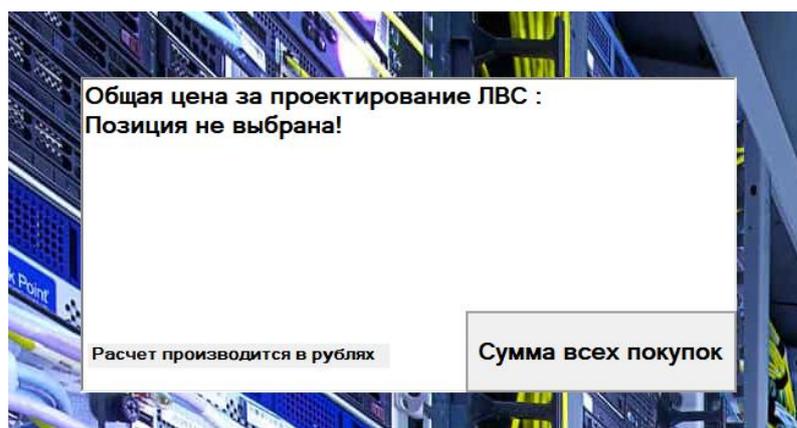


Рисунок 40 – Сумма покупок 3

Добавим возможность ввода метража кабеля и выбор оборудования из таблицы, цену за кабель вписывать вручную необязательно. Достаточно кликнуть по выбранной позиции, цена внесется во второе поля для дальнейшего расчета (рис.41).

ID прокладки ЛВС	Вариант прокладки ЛВС	Цена(руб)/1метр	Модель	Поставщик
2	Прокладка по в...	100	Витая пара-нез...	ModifiedNew
3	Прокладка по в...	150	Витая пара-экр...	ModifiedNew
4	Прокладка под ...	200	Витая пара-нез...	ModifiedNew
5	Прокладка под ...	250	Витая пара-экр...	ModifiedNew
6	Прокладка в ст...	200	Витая пара-нез...	ModifiedNew
7	Прокладка в ст...	250	Витая пара-экр...	ModifiedNew
8	Наружная прок...	50	Витая пара-нез...	ModifiedNew
9	Наружная прок...	75	Витая пара-экр...	ModifiedNew
10	Наружная прок...	50	Витая пара-нез...	ModifiedNew
11	Беспроводная ...	15	Wifi	ModifiedNew
12	Прокладка по в...	100	Оптическое вол...	ModifiedNew

Рисунок 41 – Ввод метража

Если решаем продолжить расчет ЛВС, следует перейти в следующее окно для выбора активного сетевого оборудования. Внедрим переход в новое окно с помощью кнопки и назовем ее продолжить (рис.42).

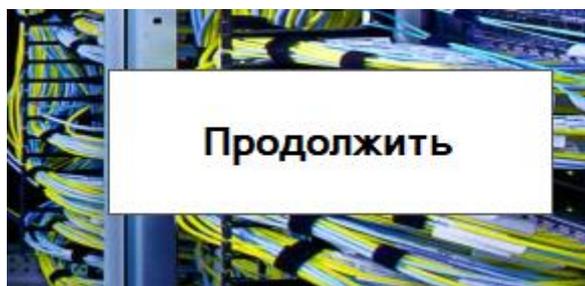


Рисунок 42 – Продолжить

Создадим новую форму схожую по функционалу с предыдущим окном. Новая аналогичная форма отличается тем, что подключена таблица с маршрутизаторами и в данном окне выбираем активное сетевое оборудования вместо вариации прокладки сети (рис.43).

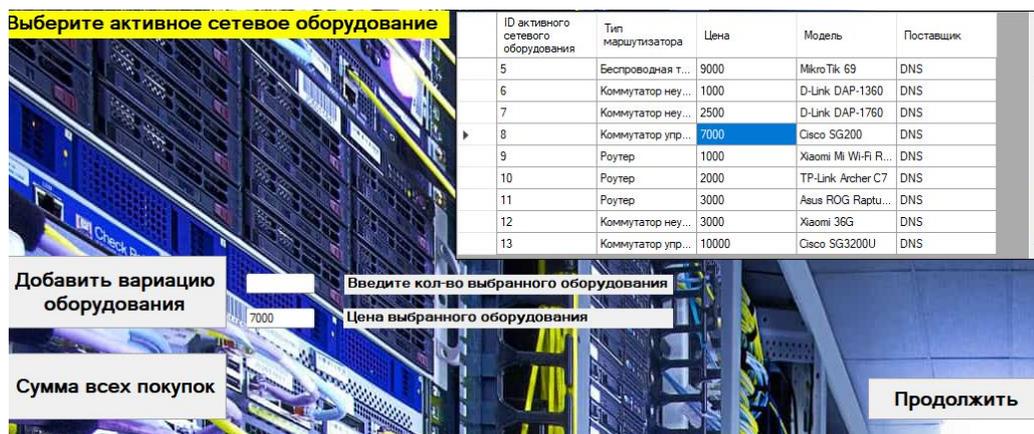


Рисунок 43 – Выбор активного сетевого оборудования

Добавим возможность ввода количества выбранных маршрутизаторов, цену за активное сетевое оборудование вписывать вручную необязательно. Достаточно кликнуть по выбранной позиции из таблицы, цена внесется во второе поля для дальнейшего расчета стоимости (рис.44).

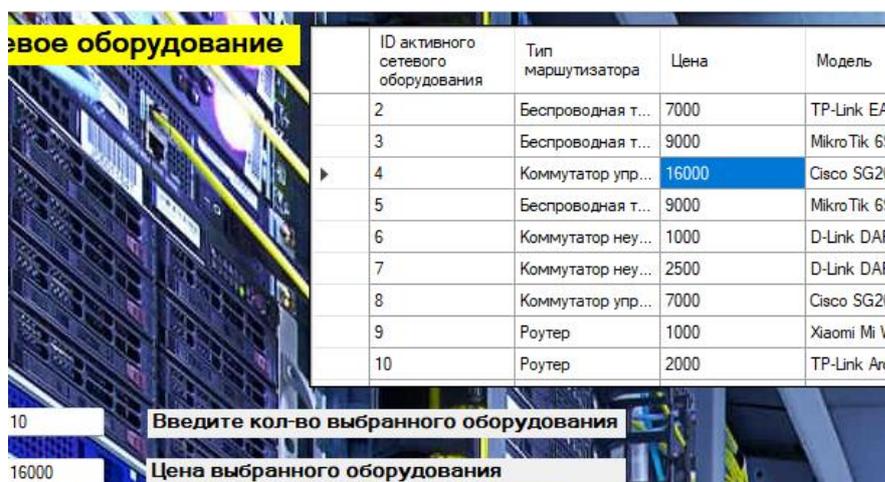


Рисунок 44 – Выбор маршрутизаторов

Создадим новую форму схожую по функционалу с предыдущими двумя окнами. Новая аналогичная форма отличается тем, что в данном окне выбираем пассивное сетевое оборудование, которое не участвует напрямую в обработке и передаче данных, а служит для организации физической инфраструктуры сети (рис.45).



Рисунок 45 – Выбор пассивного сетевого оборудования

Добавим аналогично возможность ввода количества выбранного пассивного сетевого оборудования. Цену вписывать вручную необязательно, достаточно кликнуть по выбранной позиции из таблицы, цена внесется во второе поля для дальнейшего расчета стоимости (рис.46).

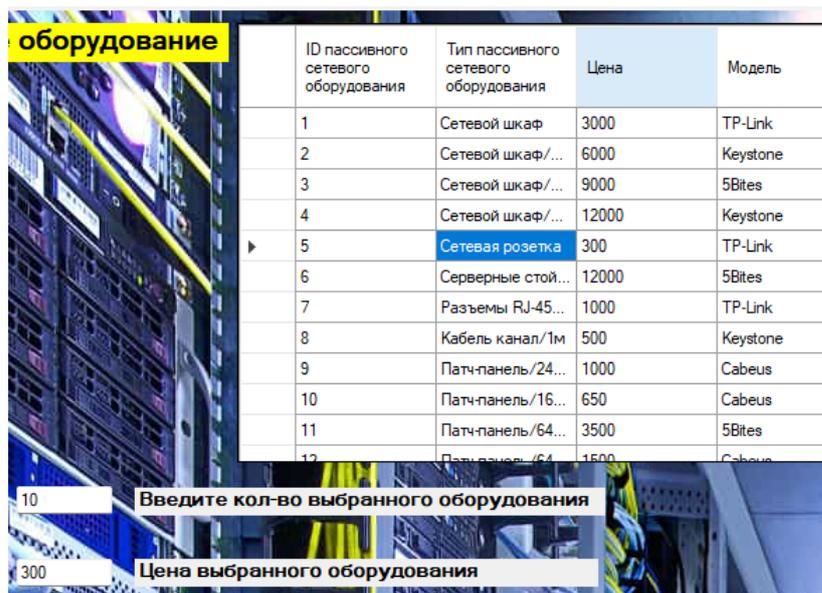


Рисунок 46 – Выбор пассивного оборудования

После прохождения всех форм получается собрать все необходимые данные о выбранном оборудовании для расчета стоимости локально вычислительной сети. Разработаем в конечной форме возможность вывода общей стоимости за проектирование ЛВС. Получим итоговый результат (рис.47).

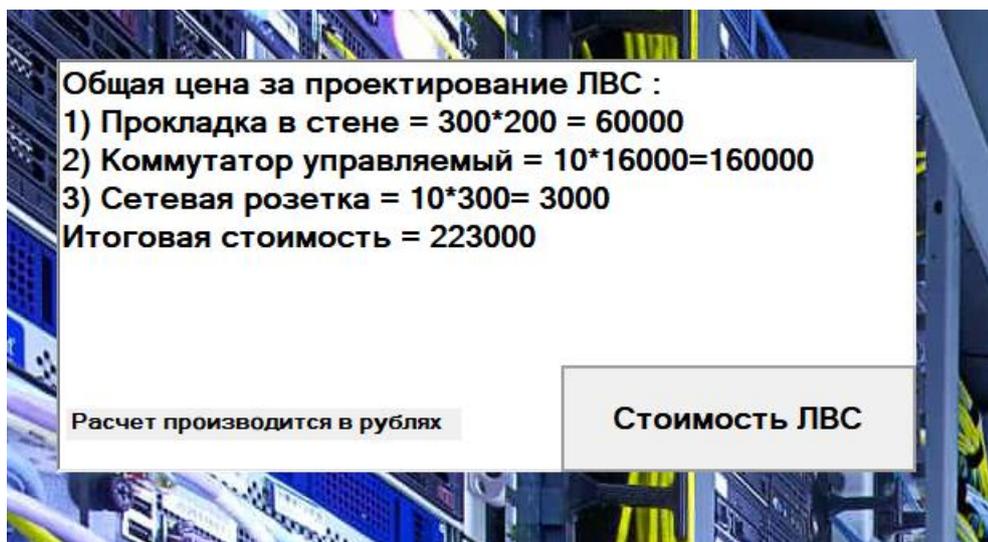


Рисунок 47 – Общая цена за проектирование

#### 2.4. Оценка общей стоимости оборудования для проектирования ЛВС на примере ООО «Сфера Ай Ти»

Перейдем в разработанное приложение оценки и выбора необходимого оборудования (рис 48).



Рисунок 48 – Запуск приложения

Согласно спроектированной схеме исследования ООО «Сфера Ай Ти» понадобится два управляемых коммутатора на 24 порта. Подберём нужное оборудование со списка поставщика (рис 49).

ID активного сетевого оборудования	Тип маршрутизатора	Цена	Модель	Поставщик
7	Коммутатор неу...	2500	D-Link DAP-1760	DNS
8	Коммутатор упр...	7000	Cisco SG200	DNS
9	Роутер	1000	Xiaomi Mi Wi-Fi R...	DNS
10	Роутер	2000	TP-Link Archer C7	DNS
11	Роутер	3000	Asus ROG Raptu...	DNS
12	Коммутатор неу...	3000	Xiaomi 36G	DNS
13	Коммутатор упр...	10000	Cisco SG3200U	DNS

Рисунок 49 – Выбор коммутаторов

Выберем два серверных шкафа небольшого размера. (рис. 50).

ID пассивного сетевого оборудования	Тип пассивного сетевого оборудования	Цена	Модель	Поставщик
1	Сетевой шкаф	3000	TP-Link	LVSCITY
2	Сетевой шкаф/...	6000	Keystone	LVSCITY
3	Сетевой шкаф/...	9000	5Bites	LVSCITY
4	Сетевой шкаф/...	12000	Keystone	LVSCITY
5	Сетевая розетка	300	TP-Link	LVSCITY
6	Серверные стой...	12000	5Bites	LVSCITY
7	Разъемы RJ-45...	1000	TP-Link	LVSCITY

Рисунок 50 – Выбор серверных шкафов

Понадобится сорок метров кабель-канала для упорядоченного монтажа и организации кабельных соединений (рис. 51).

ID пассивного сетевого оборудования	Тип пассивного сетевого оборудования	Цена	Модель	Поставщик
7	Разъемы RJ-45...	1000	TP-Link	LVSCITY
8	Кабель канал/1м	500	Keystone	LVSCITY
9	Патч-панель/24...	1000	Cabeus	LVSCITY
10	Патч-панель/16...	650	Cabeus	LVSCITY
11	Патч-панель/64...	3500	5Bites	LVSCITY
12	Патч-панель/64...	1500	Cabeus	LVSCITY
13	Патч-корд/1м	200	Cabeus	LVSCITY



Рисунок 51 – Выбор кабель-канала

Найдем для закупки шестнадцать сетевых розеток к каждому рабочему месту (рис. 52).

ID пассивного сетевого оборудования	Тип пассивного сетевого оборудования	Цена	Модель	Поставщик
4	Сетевой шкаф/...	12000	Keystone	LVSCITY
5	Сетевая розетка	300	TP-Link	LVSCITY
6	Серверные стой...	12000	5Bites	LVSCITY
7	Разъемы RJ-45...	1000	TP-Link	LVSCITY
8	Кабель канал/1м	500	Keystone	LVSCITY
9	Патч-панель/24...	1000	Cabeus	LVSCITY
10	Патч-панель/16...	650	Cabeus	LVSCITY



Рисунок 52 – Выбор сетевых розеток

Выберем сто штук разъемов RJ-45 (рис. 53).

ID пассивного сетевого оборудования	Тип пассивного сетевого оборудования	Цена	Модель	Поставш
4	Сетевой шкаф/...	12000	Keystone	LVSCITY
5	Сетевая розетка	300	TP-Link	LVSCITY
6	Серверные стой...	12000	5Bites	LVSCITY
7	Разъемы RJ-45...	1000	TP-Link	LVSCITY
8	Кабель канал/1м	500	Keystone	LVSCITY
9	Патч-панель/24...	1000	Cabeus	LVSCITY
10	Патч-панель/16...	650	Cabeus	LVSCITY



Рисунок 53 – Выбор разъемов RJ-45

Кабинеты расположены недалеко друг от друга поэтому принимаем решение прокинуть наружным способом прокладки негерметизированный медный кабель UTP. UTP–кабели обычно дешевле и легче в установке по сравнению с другими типами кабелей, поддерживают скорость передачи данных от 10 Мбит/с до 100 Гбит/с, передают поток даже в условиях электромагнитного излучения. По результатам замеров понадобится около двух ста метров кабеля (рис. 54).

ID прокладки ЛВС	Вариант прокладки ЛВС	Цена/1метр	Модель	Поставщик
5	Прокладка под ...	250	Витая пара-экр...	ModifiedNew
6	Прокладка в ст...	200	Витая пара-нез...	ModifiedNew
7	Прокладка в ст...	250	Витая пара-экр...	ModifiedNew
8	Наружная прок...	50	Витая пара-нез...	ModifiedNew
9	Наружная прок...	75	Витая пара-экр...	ModifiedNew
10	Наружная прок...	50	Витая пара-нез...	ModifiedNew
11	Беспроводная ...	15	Wifi	ModifiedNew



Рисунок 54 – Вариация прокладки сети 1

Выберем оптоволоконно – многомодовый кабель. Выбранный вариант прокладки – под полом. По результатам замеров понадобится около десяти метров кабеля (рис. 55).

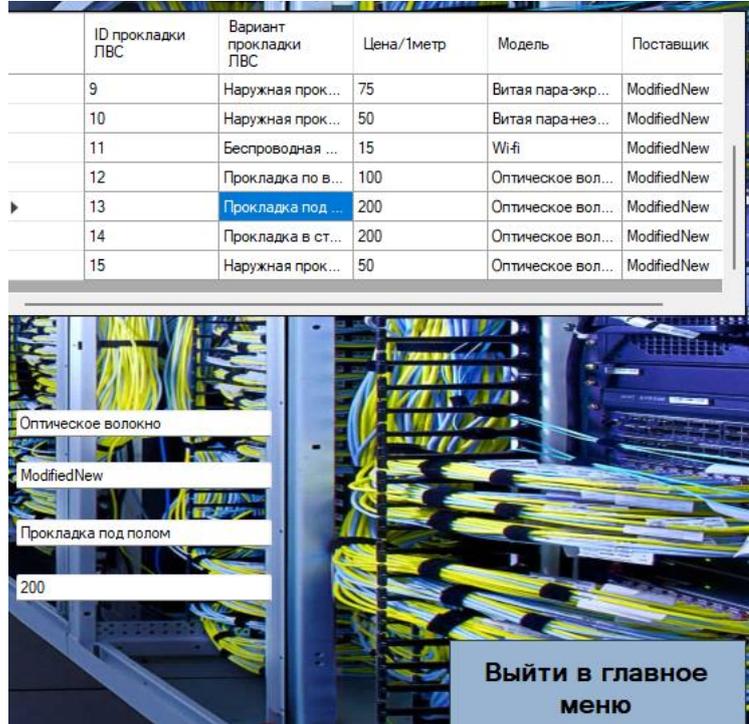


Рисунок 55 – Вариация прокладки сети 2

После подбора ассортимента вернемся в главное меню и приступим к выбору и оценке стоимости для проектирования ЛВС (рис. 56).

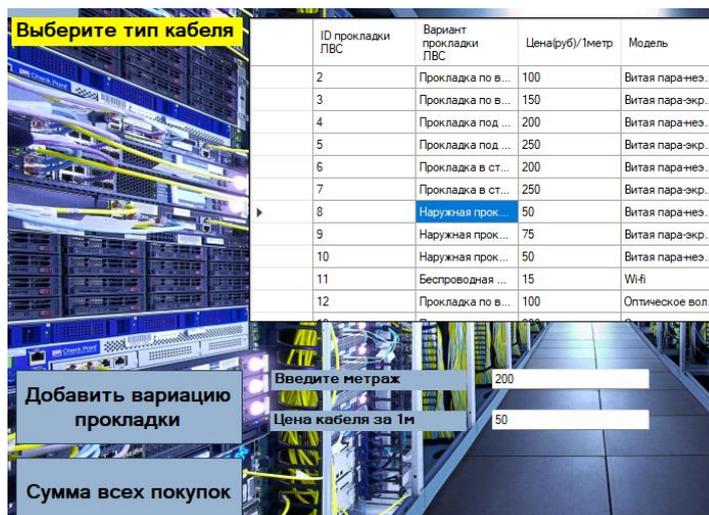
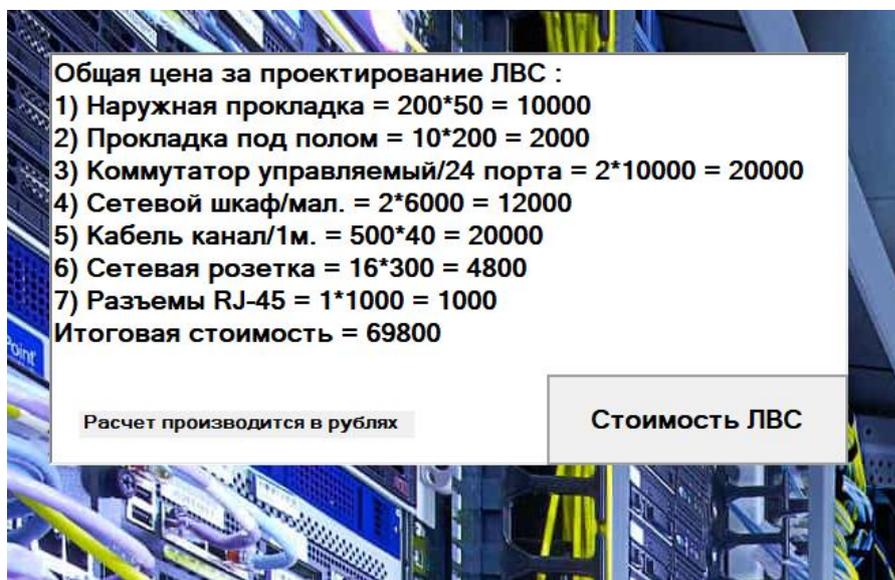


Рисунок 56 – Выбор оборудования 2

Внесем все замеры после исследования объекта в приложение, соответствуя спроектированной схеме ЛВС организации (рис.47). Получаем оценочную стоимость проектирования сети в ООО «Сфера Ай Ти» (рис. 57).



Общая цена за проектирование ЛВС :

- 1) Наружная прокладка =  $200 \cdot 50 = 10000$
- 2) Прокладка под полом =  $10 \cdot 200 = 2000$
- 3) Коммутатор управляемый/24 порта =  $2 \cdot 10000 = 20000$
- 4) Сетевой шкаф/мал. =  $2 \cdot 6000 = 12000$
- 5) Кабель канал/1м. =  $500 \cdot 40 = 20000$
- 6) Сетевая розетка =  $16 \cdot 300 = 4800$
- 7) Разъемы RJ-45 =  $1 \cdot 1000 = 1000$

Итоговая стоимость = 69800

Расчет производится в рублях

Стоимость ЛВС

Рисунок 57 – Результат выбора и оценки

## ЗАКЛЮЧЕНИЕ

В ходе выполнения бакалаврской работы был проведен анализ предметной области. На его основе удалось определить инструментальные средства и технологии разработки. Разработано приложение для выбора и оценки стоимости сетевого оборудования, которое используется для проектирования локально вычислительных сетей. К приложению удалось подключить базу данных со всей необходимой информацией о закупках оборудования. В данном приложении разработано все, что поможет инженерам, проектирующим большие и объемные локально вычислительные сети. В программе есть поиск информации об оборудовании, характеристиках, актуальных ценах и поставщиках, обновления базы данных в режиме реального времени в случае новой поставки. Разработан поэтапный алгоритм оценки и выбора оборудования, который позволяет получить оценочную стоимость для проектирования ЛВС. Алгоритм позволяет не пропустить какой-либо организационный момент, который повлечет за собой ошибку расчета. Получилось провести исследование организации ООО «Сфера Ай Ти». При анализе были получены все необходимые данные, которые помогли на практике подобрать и оценить общую стоимость сетевого оборудования для проектирования ЛВС выбранной организации.

Таким образом, была достигнута главная цель – снизить нагрузку инженеров и пользователей, уменьшить допустимые ошибки расчета до минимума за счёт разработанного приложения.

## СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Сергеев А.Н., Сергеев А.Н. Основы локальных компьютерных сетей. – М.: Лань, 2021. – 44 с.
2. Пролетарский А.В., Смирнова Е.В. Беспроводные сети Wi-Fi. – М.: МГТУ им. Н.Э. Баумана, 2021. – 178 с.
3. Таненбаум Э., Фимстер Н., Уэзеролл Д. Компьютерные сети. 6-е издание – Спб.: Питер, 2024. – 396 с.
4. Сколяров О.К. Волоконно–оптические сети и системы связи. Учебное пособие. – М.: Лань, 2023. – 200 с.
5. Холмс Д., Рест Н., Рест Д. Настройка Active Directory. Windows Server 2022. – М.: «Русская редакция», 2021. – 660 с.
6. Чекмарев Ю.В. Локальные вычислительные сети. Учебное пособие. – М.: ДМК–Пресс, 2021. – 199 с.
7. Новиков Ю.В., Кондратенко С.В. Основы локальных сетей. – М.: Эком, 2024. – 308 с.
8. Стерлинг Д.Д. Техническое руководство по волоконной оптике. 4-е издание – М.: Лори, 2022. – 195 с.
9. Visual Studio приложение Windows forms(.NET Framework). [Электронный ресурс]. URL: <https://learn.microsoft.com/ru-ru/visualstudio/get-started/csharp/tutorial-windows-forms-picture-viewer-layout?view=vs-2022> (Дата обращения 27.05.2024).
10. Определение Active Directory. [Электронный ресурс]. URL: <https://hackware.ru/?p=16316> (Дата обращения 27.05.2024).
11. Спилман Д., Хадсон К., Крафт М. Планирование, внедрение и поддержка инфраструктуры Active Directory Windows Server 2018. – М.: «Русская редакция», Спб.: «Питер», 2021. – 656 с.
12. Олифер В., Олифер Н. Компьютерные сети. Принципы, технологии. Юбилейное издание – Спб.: Питер, 2021. – 308 с.
13. Эндрю С., Дженнифер Г. Изучаем C#. – М.: Фолио, 2022. – 96 с.

14. Букатов А.А., Гуда С.А. Компьютерные сети: расширенный начальный курс. – СПб.: Питер, 2021. – 576 с.
15. Джон. М. С# 2-е издание. – М.: Бином, 2021. – 186 с.
16. Поляк–Брагинский А.В. Локальные сети. Модернизация и поиск неисправностей. – СПб.: BHV, 2021. – 640 с.
17. Поляк–Брагинский А. В. Сеть своими руками. 5-е издание. – СПб.: BHV, 2021. – 640 с.
18. Веб-приложение для создания диаграмм. [Электронный ресурс]. URL: <https://app.diagrams.net/> (Дата обращения 30.05.2024).
19. Технология ethernet и коммутаторы. [Электронный ресурс]. URL: <https://proglib.io/p/ethernet> (Дата обращения 20.05.2024).
20. Подключение к SQL Server Database [Электронный ресурс]. URL: <https://metanit.com/sharp/mvc/5.2.php> (Дата обращения 18.05.2024).

## ПРИЛОЖЕНИЕ

### Приложение «Оценивания общей стоимости проектирования ЛВС».

#### ЛИСТИНГ ПРОГРАММЫ

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Configuration;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using DB_LVS;
using WindowsFormsApplication1;
namespace LVS
{
    public partial class Form1 : Form
    {
        DataBase dataBase = new DataBase();
        public Form1()
        {
            InitializeComponent();
        }
        private void Form1_Load(object sender, EventArgs e)
        {
            Form2 form2 = new Form2();
            form2.ShowDialog();
        }

        private void Выход(object sender, EventArgs e)
        {
            Close();
        }
        // SqlConnection = new
        SqlConnection(ConfigurationManager.ConnectionStrings[""])
        private void button1_Click(object sender, EventArgs e)
        {
            int price = 0;
            rating2 form2 = new rating2(price);
            form2.ShowDialog();
            Form2 form3 = new Form2();
            form3.ShowDialog();
            Form4__passive form4 = new Form4__passive();
            form4.ShowDialog();
        }
    }
}
```

```

        Form3 form5 = new Form3();
        form5.richTextBox1.Text = String.Format("Общая цена
за проектирование ЛВС: 1) Прокладка в стене = {0} * {1} = 2)
Коммутатор управляемый = {2} * {3} 3) Сетевая розетка = {4} *
{5} Итоговая стоимость = {0} * {1} + {2} * {3} + {4} * {5} = ",
        form2.textBox1.Text, form2.textBox2.Text,
form3.textBox1.Text, form3.textBox2.Text, form4.textBox1.Text,
form4.textBox2.Text);
        form5.ShowDialog();
    }

    private void Form1_Load_1(object sender, EventArgs e)
    {
    }
    private void button2_Click(object sender, EventArgs e)
    {
        DB3 form3 = new DB3();
        form3.ShowDialog();
    }
    private void button3_Click(object sender, EventArgs e)
    {
        Close();
    }
}
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using DB_LVS;
using static
System.Windows.Forms.VisualStyleElement;
namespace WindowsFormsApplication1
{
    public partial class rating2 : Form
    {
        DataBase dataBase = new DataBase();
        int selectedRow;
        int provodPrice;
        public rating2(int provodPrice)
        {
            InitializeComponent();
            this.provodPrice = provodPrice;
            this.label2.Text = String.Format("{0} рублей",
provodPrice);
        }
    }
}

```

```

private void button1_Click(object sender, EventArgs e)
{
    this.Close();
}
// - таблица три
private void CreateColumns_4()
{
    dataGridView34.Columns.Add("id", "ID прокладки
ЛВС");
    dataGridView34.Columns.Add("typeSKS_of", "Вариант
прокладки ЛВС");
    dataGridView34.Columns.Add("countPRICE_of",
"Цена (руб) /1метр");
    dataGridView34.Columns.Add("postavka", "Модель");
    dataGridView34.Columns.Add("zakupka", "Поставщик");
}
private void ReadSingleRow_4(DataGridView dvw_3,
IDataRecord record_3)
{
    dvw_3.Rows.Add(record_3.GetInt32(0),
record_3.GetString(1), record_3.GetInt32(2),
record_3.GetString(3), RowState.MODIFIED);
}
private void RefreshDataGrid_4(DataGridView dvw_4)
{
    dvw_4.Rows.Clear();
    string queryString_4 = $"select * from sksDB";
    SqlCommand command_4 = new SqlCommand(queryString_4,
dataBase.GetConnection());
    dataBase.OpenConnection();
    SqlDataReader reader_4 = command_4.ExecuteReader();
    while (reader_4.Read())
    {
        ReadSingleRow_4(dvw_4, reader_4);
    }
    reader_4.Close();
}
private void rating2_Load_1(object sender, EventArgs e)
{
    CreateColumns_4();
    RefreshDataGrid_4(dataGridView34);
}
private void dataGridView34_CellClick(object sender,
DataGridViewCellEventArgs e)
{
    selectedRow = e.RowIndex;
    if (e.RowIndex >= 0)
    {
        DataGridViewRow row =
dataGridView34.Rows[selectedRow];
        textBox1.Text = row.Cells[2].Value.ToString();
    }
}

```

```

        }
    }
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using DB_LVS;
using static
System.Windows.Forms.VisualStyles.VisualStyleElement;
namespace WindowsFormsApplication1
{
    enum RoWState
    {
        Existed,
        REG,
        Modified,
        ModifiedNew,
        Deleted
    }
    public partial class DB3 : Form
    {
        DataBase dataBase = new DataBase();
        int selectedRow;
        public DB3()
        {
            InitializeComponent();
        }
        private void CreateColumns()
        {
            dataGridView1.Columns.Add("id", "ID активного
сетевое оборудование");
            dataGridView1.Columns.Add("typePRODUCT_of", "Тип
маршрутизатора");
            dataGridView1.Columns.Add("countPRICE_of", "Цена");
            dataGridView1.Columns.Add("postavka", "Модель");
            dataGridView1.Columns.Add("zakupka", "Поставщик");
        }
        private void ReadSingleRow(DataGridView dvw, IDataRecord
record)
        {
            dvw.Rows.Add(record.GetInt32(0),
record.GetString(1), record.GetInt32(2), record.GetString(3),
RoWState.MODIFIED);
        }
        private void RefreshDataGrid(DataGridView dvw)
        {

```

```

        dvw.Rows.Clear();
        string queryString = $"select * from localDB";
        SqlCommand command = new SqlCommand(queryString,
dataBase.GetConnection());
        dataBase.OpenConnection();
        SqlDataReader reader = command.ExecuteReader();
        while (reader.Read())
        {
            ReadSingleRow(dvw, reader);
        }
        reader.Close();
    }
    private void menuStrip1_ItemClicked(object sender,
ToolStripItemClickedEventArgs e)
    {
    }
    private void dataGridView1_CellClick(object sender,
DataGridViewCellEventArgs e)
    {
        selectedRow = e.RowIndex;
        if (e.RowIndex >= 0)
        {
            DataGridViewRow row =
dataGridView1.Rows[selectedRow];

            textBox_PRODUCT.Text =
row.Cells[0].Value.ToString();
            textBox_PRICE.Text =
row.Cells[1].Value.ToString();
            textBox_postavka.Text =
row.Cells[2].Value.ToString();
            textBox_id.Text = row.Cells[3].Value.ToString();
        }
    }
    // - таблица два
    private void CreateColumns_2()
    {
        dataGridView2.Columns.Add("id", "ID пассивного
сетевого оборудования");
        dataGridView2.Columns.Add("countLOCKER_of", "Тип
пассивного сетевого оборудования");
        dataGridView2.Columns.Add("countPRICE_of", "Цена");
        dataGridView2.Columns.Add("postavka", "Модель");
        dataGridView2.Columns.Add("zakupka", "Поставщик");
    }
    private void ReadSingleRow_2(DataGridView dvw_2,
IDataRecord record_2)
    {
        dvw_2.Rows.Add(record_2.GetInt32(0),
record_2.GetString(1), record_2.GetInt32(2),
record_2.GetString(3), RowState.REG);
    }
    private void RefreshDataGrid_2(DataGridView dvw_2)

```

```

        {
            dvw_2.Rows.Clear();
            string queryString_2 = $"select * from lockerDB";
            SqlCommand command_2 = new SqlCommand(queryString_2,
dataBase.GetConnection());
            dataBase.OpenConnection();
            SqlDataReader reader_2 = command_2.ExecuteReader();
            while (reader_2.Read())
            {
                ReadSingleRow_2(dvw_2, reader_2);
            }
            reader_2.Close();
        }
        private void dataGridView2_CellClick(object sender,
DataGridViewCellEventArgs e)
        {
            selectedRow = e.RowIndex;
            if (e.RowIndex >= 0)
            {
                DataGridViewRow row =
dataGridView2.Rows[selectedRow];

                textBox_LOCKER.Text =
row.Cells[0].Value.ToString();
                textBoxPRICE_2.Text =
row.Cells[1].Value.ToString();
                textBoxpostavka_2.Text =
row.Cells[2].Value.ToString();
                textBox_id_2.Text =
row.Cells[3].Value.ToString();
            }
        }
        //
        // - таблица три
        private void CreateColumns_3()
        {
            dataGridView3.Columns.Add("id", "ID прокладки ЛБС");
            dataGridView3.Columns.Add("typeSKS_of", "Вариант
прокладки ЛБС");
            dataGridView3.Columns.Add("countPRICE_of",
"Цена/1метр");
            dataGridView3.Columns.Add("postavka", "Модель");
            dataGridView3.Columns.Add("zakypka", "Поставщик");
        }
        private void ReadSingleRow_3(DataGridView dvw_3,
IDataRecord record_3)
        {
            dvw_3.Rows.Add(record_3.GetInt32(0),
record_3.GetString(1), record_3.GetInt32(2),
record_3.GetString(3), RowState.ModifiedNew);
        }
        private void RefreshDataGrid_3(DataGridView dvw_3)
        {

```

```

        dvw_3.Rows.Clear();
        string queryString_3 = $"select * from sksDB";
        SqlCommand command_3 = new SqlCommand(queryString_3,
dataBase.GetConnection());
        dataBase.OpenConnection();
        SqlDataReader reader_3 = command_3.ExecuteReader();
        while (reader_3.Read())
        {
            ReadSingleRow_3(dvw_3, reader_3);
        }
        reader_3.Close();
    }
    private void dataGridView3_CellClick(object sender,
DataGridViewCellEventArgs e)
    {
        selectedRow = e.RowIndex;
        if (e.RowIndex >= 0)
        {
            DataGridViewRow row =
dataGridView3.Rows[selectedRow];

            textBoxtypeSKS.Text =
row.Cells[0].Value.ToString();
            textBoxPRICE_3.Text =
row.Cells[1].Value.ToString();
            textBoxpostavka_3.Text =
row.Cells[2].Value.ToString();
            textBox_id_3.Text =
row.Cells[3].Value.ToString();
        }
    }
    //
    private void DB3_Load_1(object sender, EventArgs e)
    {
        // TODO: данная строка кода позволяет загрузить
данные в таблицу "lvSDataSet2.sksDB". При необходимости она
может быть перемещена или удалена.
        this.sksDBTableAdapter.Fill(this.lvSDataSet2.sksDB);
        // TODO: данная строка кода позволяет загрузить
данные в таблицу "lvSDataSet1.lockerDB". При необходимости она
может быть перемещена или удалена.

this.lockerDBTableAdapter.Fill(this.lvSDataSet1.lockerDB);
        // TODO: данная строка кода позволяет загрузить
данные в таблицу "lvSDataSet.localDB". При необходимости она
может быть перемещена или удалена.

this.localDBTableAdapter.Fill(this.lvSDataSet.localDB);
        CreateColumns();
        RefreshDataGrid(dataGridView1);
        CreateColumns_2();
        RefreshDataGrid_2(dataGridView2);
        CreateColumns_3();
    }

```

```

        RefreshDataGrid_3(dataGridView3);
    }
    private void search(DataGridView dvw)
    {
        dvw.Rows.Clear();
        string searchstring = $"select * from localDB where
contact (id,typePRODUCT_of,countPRICE_of,postavka) like '%" +
textBox_search1.Text + "%'";
        SqlCommand com = new SqlCommand(searchstring,
dataBase.GetConnection());
        dataBase.OpenConnection();
        SqlDataReader read = com.ExecuteReader();
        while (read.Read())
        {
            ReadSingleRow(dvw, read);
        }
        read.Close();
    }
    private void textBox_search1_TextChanged(object sender,
EventArgs e)
    {
        search(dataGridView1);
    }
    private void Update(DataGridView dvw)
    {
        dataBase.OpenConnection();
        for(int index =0; index<
dataGridView1.Rows.Count; index++)
        {
            var rowState =
(RowState)dataGridView1.Rows[index].Cells[5].Value;
            if (rowState == RowState.Existed)
                continue;
            if (rowState == RowState.Deleted)
            {
                var id =
Convert.ToInt32(dataGridView1.Rows[index].Cells[0].Value);
                var deleteQuery = $"delete from
localDB where id = {id}";
                var command = new
SqlCommand(deleteQuery, dataBase.GetConnection());
                command.ExecuteNonQuery();
            }
            if(rowState == RowState.ModifiedNew)
            {
                var ID =
dataGridView1.Rows[index].Cells[0].Value.ToString();
            }
        }
    }
    private void textBox1_TextChanged(object sender,
EventArgs e)
    {

```

```

        /// Update(dataGridView1);
    }
    private void button_refresh_1_Click(object
sender, EventArgs e)
    {
        int n, k,m, res;
        n = int.Parse(textBox_PRICE.Text);
        k = int.Parse(textBoxPRICE_2.Text);
        m = int.Parse(textBoxPRICE_3.Text);
        res = n + k + m;
        textBox_search1.Text = res;
    }
    private void button1_Click(object sender, EventArgs
e)
    {
        rating2 form2 = new rating2(123);
        form2.ShowDialog();
    }
    private void файдToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        RefreshDataGrid(dataGridView1); */
    }
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using DB_LVS;
using static
System.Windows.Forms.VisualStyleElement;
namespace WindowsFormsApplication1
{
    public partial class Form2 : Form
    {
        DataBase dataBase = new DataBase();
        int selectedRow;
        public Form2()
        {
            InitializeComponent();
        }
        private void CreateColumns()
        {
            dataGridView1.Columns.Add("id", "ID активного
сетевого оборудования");

```

```

        dataGridView1.Columns.Add("typePRODUCT_of", "Тип
маршрутизатора");
        dataGridView1.Columns.Add("countPRICE_of", "Цена");
        dataGridView1.Columns.Add("postavka", "Модель");
        dataGridView1.Columns.Add("zakupka", "Поставщик");
    }
    private void ReadSingleRow(DataGridView dvw, IDataRecord
record)
    {
        dvw.Rows.Add(record.GetInt32(0),
record.GetString(1), record.GetInt32(2), record.GetString(3),
RowState.MODIFIED);
    }
    private void RefreshDataGrid(DataGridView dvw)
    {
        dvw.Rows.Clear();
        string queryString = $"select * from localDB";
        SqlCommand command = new SqlCommand(queryString,
dataBase.GetConnection());
        dataBase.OpenConnection();
        SqlDataReader reader = command.ExecuteReader();
        while (reader.Read())
        {
            ReadSingleRow(dvw, reader);
        }
        reader.Close();
    }
    private void button1_Click(object sender, EventArgs e)
    {
        this.Close();
    }
    private void dataGridView1_CellClick(object sender,
DataGridViewCellEventArgs e)
    {
        selectedRow = e.RowIndex;
        if (e.RowIndex >= 0)
        {
            DataGridViewRow row =
dataGridView1.Rows[selectedRow];
            textBox2.Text = row.Cells[2].Value.ToString();
        }
    }
    private void Form2_Load(object sender, EventArgs e)
    {
        CreateColumns();
        RefreshDataGrid(dataGridView1);
    }
}
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;

```

```

using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using DB_LVS;
using static
System.Windows.Forms.VisualStyleElement;
namespace WindowsFormsApplication1
{
    public partial class Form4__passive : Form
    {
        DataBase dataBase = new DataBase();
        int selectedRow;
        public Form4__passive()
        {
            InitializeComponent();
        }
        // - таблица два
        private void CreateColumns_2()
        {
            dataGridView2.Columns.Add("id", "ID пассивного
сетевое оборудование");
            dataGridView2.Columns.Add("countLOCKER_of", "Тип
пассивного сетевое оборудование");
            dataGridView2.Columns.Add("countPRICE_of", "Цена");
            dataGridView2.Columns.Add("postavka", "Модель");
            dataGridView2.Columns.Add("zakupka", "Поставщик");
        }
        private void ReadSingleRow_2(DataGridView dvw_2,
IDataRecord record_2)
        {
            dvw_2.Rows.Add(record_2.GetInt32(0),
record_2.GetString(1), record_2.GetInt32(2),
record_2.GetString(3), RowState.MODIFIED);
        }
        private void RefreshDataGrid_2(DataGridView dvw_2)
        {
            dvw_2.Rows.Clear();
            string queryString_2 = $"select * from lockerDB";
            SqlCommand command_2 = new SqlCommand(queryString_2,
dataBase.GetConnection());
            dataBase.OpenConnection();
            SqlDataReader reader_2 = command_2.ExecuteReader();
            while (reader_2.Read())
            {
                ReadSingleRow_2(dvw_2, reader_2);
            }
            reader_2.Close();
        }
        private void Form4__passive_Load(object sender,
EventArgs e)

```

```

        {
            CreateColumns_2();
            RefreshDataGrid_2(dataGridView2);
        }
        private void dataGridView2_CellClick(object sender,
DataGridViewCellEventArgs e)
        {
            selectedRow = e.RowIndex;
            if (e.RowIndex >= 0)
            {
                DataGridViewRow row =
dataGridView2.Rows[selectedRow];
                textBox1.Text = row.Cells[2].Value.ToString();
            }
        }
    }
}
namespace LVS
{
    partial class Form1
    {
        /// <summary>
        /// Требуется переменная конструктора.
        /// </summary>
        private System.ComponentModel.IContainer components =
null;
        /// <summary>
        /// Освободить все используемые ресурсы.
        /// </summary>
        /// <param name="disposing">истинно, если управляемый
ресурс должен быть удален; иначе ложно.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }
        #region Код, автоматически созданный конструктором форм
Windows
        /// <summary>
        /// Обязательный метод для поддержки конструктора - не
изменяйте
        /// содержимое данного метода при помощи редактора кода.
        /// </summary>
        private void InitializeComponent()
        {
            this.button1 = new System.Windows.Forms.Button();
            this.button2 = new System.Windows.Forms.Button();
            this.button3 = new System.Windows.Forms.Button();
            this.SuspendLayout();
            //

```

```

        // button1
        //
        this.button1.FlatStyle =
System.Windows.Forms.FlatStyle.Popup;
        this.button1.Font = new
System.Drawing.Font("Microsoft Sans Serif", 14.25F,
System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte) (204)));
        this.button1.Location = new
System.Drawing.Point(522, 100);
        this.button1.Name = "button1";
        this.button1.Size = new System.Drawing.Size(177,
62);

        this.button1.TabIndex = 0;
        this.button1.Text = "Оценка и выбор оборудования";
        this.button1.UseVisualStyleBackColor = true;
        this.button1.Click += new
System.EventHandler(this.button1_Click);
        //
        // button2
        //
        this.button2.FlatStyle =
System.Windows.Forms.FlatStyle.Popup;
        this.button2.Font = new
System.Drawing.Font("Microsoft Sans Serif", 14.25F,
System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte) (204)));
        this.button2.Location = new
System.Drawing.Point(522, 200);
        this.button2.Name = "button2";
        this.button2.Size = new System.Drawing.Size(177,
62);

        this.button2.TabIndex = 1;
        this.button2.Text = "Ассортимент оборудования";
        this.button2.UseVisualStyleBackColor = true;
        this.button2.Click += new
System.EventHandler(this.button2_Click);
        //
        // button3
        //
        this.button3.FlatStyle =
System.Windows.Forms.FlatStyle.Popup;
        this.button3.Font = new
System.Drawing.Font("Microsoft Sans Serif", 14.25F,
System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte) (204)));
        this.button3.Location = new
System.Drawing.Point(522, 300);
        this.button3.Name = "button3";
        this.button3.Size = new System.Drawing.Size(177,
57);

        this.button3.TabIndex = 2;
        this.button3.Text = "Выход из программы";

```

```

        this.button3.UseVisualStyleBackColor = true;
        this.button3.Click += new
System.EventHandler(this.button3_Click);
        //
        // Form1
        //
        this.BackColor =
System.Drawing.SystemColors.ActiveCaption;
        this.BackgroundImage =
global::WindowsFormsApplication1.Properties.Resources.ЛВС_и_СКС_
4_min;
        this.BackgroundImageLayout =
System.Windows.Forms.ImageLayout.Stretch;
        this.ClientSize = new System.Drawing.Size(1184,
561);
        this.Controls.Add(this.button3);
        this.Controls.Add(this.button2);
        this.Controls.Add(this.button1);
        this.Name = "Form1";
        this.ResumeLayout(false);
    }
#endregion
private System.Windows.Forms.Button button1;
private System.Windows.Forms.Button button2;
private System.Windows.Forms.Button button3;
}
}
namespace WindowsFormsApplication1
{
    partial class Form2
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components =
null;
        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources
should be disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }
#region Windows Form Designer generated code
        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.

```

```

    /// </summary>
    private void InitializeComponent()
    {
        this.textBox1 = new System.Windows.Forms.TextBox();
        this.textBox2 = new System.Windows.Forms.TextBox();
        this.button1 = new System.Windows.Forms.Button();
        this.label1 = new System.Windows.Forms.Label();
        this.label2 = new System.Windows.Forms.Label();
        this.dataGridView1 = new
System.Windows.Forms.DataGridView();
        this.label3 = new System.Windows.Forms.Label();
        this.button2_second_type_cable = new
System.Windows.Forms.Button();
        this.button1_Summa = new
System.Windows.Forms.Button();

        ((System.ComponentModel.ISupportInitialize)(this.dataGridView1))
        .BeginInit();
            this.SuspendLayout();
            //
            // textBox1
            //
            this.textBox1.Location = new
System.Drawing.Point(243, 274);
            this.textBox1.Margin = new
System.Windows.Forms.Padding(2);
            this.textBox1.Name = "textBox1";
            this.textBox1.Size = new System.Drawing.Size(68,
20);

            this.textBox1.TabIndex = 0;
            //
            // textBox2
            //
            this.textBox2.Location = new
System.Drawing.Point(243, 309);
            this.textBox2.Margin = new
System.Windows.Forms.Padding(2);
            this.textBox2.Name = "textBox2";
            this.textBox2.Size = new System.Drawing.Size(68,
20);

            this.textBox2.TabIndex = 1;
            //
            // button1
            //
            this.button1.FlatStyle =
System.Windows.Forms.FlatStyle.Popup;
            this.button1.Font = new
System.Drawing.Font("Microsoft Sans Serif", 14.25F,
System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(204)));
            this.button1.Location = new
System.Drawing.Point(865, 379);
            this.button1.Name = "button1";

```

```

        this.button1.Size = new System.Drawing.Size(179,
42);
        this.button1.TabIndex = 2;
        this.button1.Text = "Продолжить";
        this.button1.UseVisualStyleBackColor = true;
        this.button1.Click += new
System.EventHandler(this.button1_Click);
        //
        // label1
        //
        this.label1.FlatStyle =
System.Windows.Forms.FlatStyle.Popup;
        this.label1.Font = new
System.Drawing.Font("Microsoft Sans Serif", 9.75F,
System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte) (204)));
        this.label1.Location = new System.Drawing.Point(340,
309);
        this.label1.Name = "label1";
        this.label1.Size = new System.Drawing.Size(330, 20);
        this.label1.TabIndex = 3;
        this.label1.Text = "Цена выбранного оборудования";
        //
        // label2
        //
        this.label2.FlatStyle =
System.Windows.Forms.FlatStyle.Popup;
        this.label2.Font = new
System.Drawing.Font("Microsoft Sans Serif", 9.75F,
System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte) (204)));
        this.label2.Location = new System.Drawing.Point(341,
275);
        this.label2.Name = "label2";
        this.label2.Size = new System.Drawing.Size(329, 20);
        this.label2.TabIndex = 4;
        this.label2.Text = "Введите кол-во выбранного
оборудования";
        //
        // dataGridView1
        //
        this.dataGridView1.AllowUserToAddRows = false;
        this.dataGridView1.AllowUserToDeleteRows = false;
        this.dataGridView1.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;
        this.dataGridView1.Location = new
System.Drawing.Point(452, 9);
        this.dataGridView1.Name = "dataGridView1";
        this.dataGridView1.ReadOnly = true;
        this.dataGridView1.Size = new
System.Drawing.Size(592, 251);
        this.dataGridView1.TabIndex = 5;

```

```

        this.dataGridView1.CellClick += new
System.Windows.Forms.DataGridViewCellEventHandler(this.dataGridViewV
iew1_CellClick);
        //
        // label3
        //
        this.label3.BackColor = System.Drawing.Color.Yellow;
        this.label3.FlatStyle =
System.Windows.Forms.FlatStyle.Popup;
        this.label3.Font = new
System.Drawing.Font("Microsoft Sans Serif", 14.25F,
System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(204)));
        this.label3.Location = new System.Drawing.Point(-2,
9);

        this.label3.Name = "label3";
        this.label3.Size = new System.Drawing.Size(448, 28);
        this.label3.TabIndex = 6;
        this.label3.Text = "Выберите активное сетевое
оборудование";
        //
        // button2_second_type_cable
        //
        this.button2_second_type_cable.FlatStyle =
System.Windows.Forms.FlatStyle.Popup;
        this.button2_second_type_cable.Font = new
System.Drawing.Font("Microsoft Sans Serif", 14.25F,
System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(204)));
        this.button2_second_type_cable.Location = new
System.Drawing.Point(2, 256);
        this.button2_second_type_cable.Name =
"button2_second_type_cable";
        this.button2_second_type_cable.Size = new
System.Drawing.Size(221, 73);
        this.button2_second_type_cable.TabIndex = 7;
        this.button2_second_type_cable.Text = "Добавить
вариацию оборудования";

this.button2_second_type_cable.UseVisualStyleBackColor = true;
        //
        // button1_Summa
        //
        this.button1_Summa.FlatStyle =
System.Windows.Forms.FlatStyle.Popup;
        this.button1_Summa.Font = new
System.Drawing.Font("Microsoft Sans Serif", 14.25F,
System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(204)));
        this.button1_Summa.Location = new
System.Drawing.Point(2, 354);
        this.button1_Summa.Name = "button1_Summa";

```

```

        this.button1_Summa.Size = new
System.Drawing.Size(221, 67);
        this.button1_Summa.TabIndex = 8;
        this.button1_Summa.Text = "Сумма всех покупок";
        this.button1_Summa.UseVisualStyleBackColor = true;
        //
        // Form2
        //
        this.AutoScaleDimensions = new
System.Drawing.SizeF(6F, 13F);
        this.AutoScaleMode =
System.Windows.Forms.AutoScaleMode.Font;
        this.BackgroundImage =
global::WindowsFormsApplication1.Properties.Resources.ЛВС_и_КС_
4_min1;
        this.ClientSize = new System.Drawing.Size(1073,
460);

        this.Controls.Add(this.button1_Summa);
        this.Controls.Add(this.button2_second_type_cable);
        this.Controls.Add(this.label3);
        this.Controls.Add(this.dataGridView1);
        this.Controls.Add(this.label2);
        this.Controls.Add(this.label1);
        this.Controls.Add(this.button1);
        this.Controls.Add(this.textBox2);
        this.Controls.Add(this.textBox1);
        this.Margin = new System.Windows.Forms.Padding(2);
        this.Name = "Form2";
        this.Text = "Form2";
        this.Load += new
System.EventHandler(this.Form2_Load);

((System.ComponentModel.ISupportInitialize)(this.dataGridView1))
.EndInit();
        this.ResumeLayout(false);
        this.PerformLayout();
    }
    #endregion
    public System.Windows.Forms.TextBox textBox1;
    public System.Windows.Forms.TextBox textBox2;
    private System.Windows.Forms.Button button1;
    private System.Windows.Forms.Label label1;
    private System.Windows.Forms.Label label2;
    private System.Windows.Forms.DataGridView dataGridView1;
    private System.Windows.Forms.Label label3;
    private System.Windows.Forms.Button
button2_second_type_cable;
    private System.Windows.Forms.Button button1_Summa;
}
}
namespace WindowsFormsApplication1
{
    partial class rating2

```

```

    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components =
null;
        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources
should be disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }
        #region Windows Form Designer generated code
        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.components = new
System.ComponentModel.Container();
            this.label1 = new System.Windows.Forms.Label();
            this.button1_Summa = new
System.Windows.Forms.Button();
            this.button2_second_type_cable = new
System.Windows.Forms.Button();
            this.dataGridView34 = new
System.Windows.Forms.DataGridView();
            this.lVSDataset21BindingSource = new
System.Windows.Forms.BindingSource(this.components);
            this.lVSDataset21 = new
WindowsFormsApplication1.LVSDataset2();
            this.label2 = new System.Windows.Forms.Label();
            this.textBox1 = new System.Windows.Forms.TextBox();
            this.textBox2 = new System.Windows.Forms.TextBox();
            this.button1 = new System.Windows.Forms.Button();
            this.label3 = new System.Windows.Forms.Label();
            this.label4 = new System.Windows.Forms.Label();

            ((System.ComponentModel.ISupportInitialize)(this.dataGridView34)
).BeginInit();

            ((System.ComponentModel.ISupportInitialize)(this.lVSDataset21Bin
dingSource)).BeginInit();

```

```

((System.ComponentModel.ISupportInitialize)(this.lVSDataset21)).
BeginInit();
    this.SuspendLayout();
    //
    // label1
    //
    this.label1.BackColor = System.Drawing.Color.Yellow;
    this.label1.FlatStyle =
System.Windows.Forms.FlatStyle.Popup;
    this.label1.Font = new
System.Drawing.Font("Microsoft Sans Serif", 14.25F,
System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(204)));
    this.label1.Location = new System.Drawing.Point(35,
23);

    this.label1.Name = "label1";
    this.label1.Size = new System.Drawing.Size(225, 28);
    this.label1.TabIndex = 2;
    this.label1.Text = "Выберите тип кабеля";
    this.label1.Click += new
System.EventHandler(this.label1_Click);
    //
    // button1_Summa
    //
    this.button1_Summa.FlatStyle =
System.Windows.Forms.FlatStyle.Popup;
    this.button1_Summa.Font = new
System.Drawing.Font("Microsoft Sans Serif", 14.25F,
System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(204)));
    this.button1_Summa.Location = new
System.Drawing.Point(39, 455);
    this.button1_Summa.Name = "button1_Summa";
    this.button1_Summa.Size = new
System.Drawing.Size(221, 67);
    this.button1_Summa.TabIndex = 1;
    this.button1_Summa.Text = "Сумма всех покупок";
    this.button1_Summa.UseVisualStyleBackColor = true;
    //
    // button2_second_type_cable
    //
    this.button2_second_type_cable.FlatStyle =
System.Windows.Forms.FlatStyle.Popup;
    this.button2_second_type_cable.Font = new
System.Drawing.Font("Microsoft Sans Serif", 14.25F,
System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(204)));
    this.button2_second_type_cable.Location = new
System.Drawing.Point(39, 369);
    this.button2_second_type_cable.Name =
"button2_second_type_cable";

```

```

        this.button2_second_type_cable.Size = new
System.Drawing.Size(221, 73);
        this.button2_second_type_cable.TabIndex = 3;
        this.button2_second_type_cable.Text = "Добавить
вариацию прокладки";

this.button2_second_type_cable.UseVisualStyleBackColor = true;
//
// dataGridView34
//
        this.dataGridView34.AllowUserToAddRows = false;
        this.dataGridView34.AllowUserToDeleteRows = false;
        this.dataGridView34.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.Auto
oSize;
        this.dataGridView34.Location = new
System.Drawing.Point(266, 23);
        this.dataGridView34.Name = "dataGridView34";
        this.dataGridView34.ReadOnly = true;
        this.dataGridView34.RowHeadersWidth = 62;
        this.dataGridView34.Size = new
System.Drawing.Size(858, 299);
        this.dataGridView34.TabIndex = 5;
        this.dataGridView34.CellClick += new
System.Windows.Forms.DataGridViewCellEventHandler(this.dataGridViewV
iew34_CellClick);
//
// LVSDataset21BindingSource
//
        this.lVSDataset21BindingSource.DataSource =
this.lVSDataset21;
        this.lVSDataset21BindingSource.Position = 0;
//
// LVSDataset21
//
        this.lVSDataset21.DataSetName = "LVSDataset2";
        this.lVSDataset21.SchemaSerializationMode =
System.Data.SchemaSerializationMode.IncludeSchema;
//
// label2
//
        this.label2.Location = new System.Drawing.Point(0,
0);

        this.label2.Name = "label2";
        this.label2.Size = new System.Drawing.Size(100, 23);
        this.label2.TabIndex = 13;
//
// textBox1
//
        this.textBox1.Location = new
System.Drawing.Point(503, 408);
        this.textBox1.Name = "textBox1";

```

```

        this.textBox1.Size = new System.Drawing.Size(153,
20);
        this.textBox1.TabIndex = 7;
        //
        // textBox2
        //
        this.textBox2.Location = new
System.Drawing.Point(503, 368);
        this.textBox2.Name = "textBox2";
        this.textBox2.Size = new System.Drawing.Size(153,
20);
        this.textBox2.TabIndex = 8;
        //
        // button1
        //
        this.button1.BackColor =
System.Drawing.SystemColors.ButtonHighlight;
        this.button1.FlatStyle =
System.Windows.Forms.FlatStyle.Popup;
        this.button1.Font = new
System.Drawing.Font("Microsoft Sans Serif", 14.25F,
System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte) (204)));
        this.button1.Location = new
System.Drawing.Point(903, 379);
        this.button1.Name = "button1";
        this.button1.Size = new System.Drawing.Size(221,
73);
        this.button1.TabIndex = 10;
        this.button1.Text = "Продолжить";
        this.button1.UseVisualStyleBackColor = false;
        this.button1.Click += new
System.EventHandler(this.button1_Click);
        //
        // label3
        //
        this.label3.FlatStyle =
System.Windows.Forms.FlatStyle.Popup;
        this.label3.Font = new
System.Drawing.Font("Microsoft Sans Serif", 9.75F,
System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte) (204)));
        this.label3.Location = new System.Drawing.Point(287,
409);
        this.label3.Name = "label3";
        this.label3.Size = new System.Drawing.Size(191, 20);
        this.label3.TabIndex = 12;
        this.label3.Text = "Цена кабеля за 1м";
        //
        // label4
        //
        this.label4.FlatStyle =
System.Windows.Forms.FlatStyle.Popup;

```

```

        this.label4.Font = new
System.Drawing.Font("Microsoft Sans Serif", 9.75F,
System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(204)));
        this.label4.Location = new System.Drawing.Point(288,
369);

        this.label4.Name = "label4";
        this.label4.Size = new System.Drawing.Size(190, 20);
        this.label4.TabIndex = 12;
        this.label4.Text = "Введите метраж";
        //
        // rating2
        //
        this.BackColor =
System.Drawing.SystemColors.ActiveCaption;
        this.BackgroundImage =
global::WindowsFormsApplication1.Properties.Resources.ЛВС_и_КС_
4_min;

        this.BackgroundImageLayout =
System.Windows.Forms.ImageLayout.Stretch;
        this.ClientSize = new System.Drawing.Size(1184,
561);

        this.Controls.Add(this.label4);
        this.Controls.Add(this.label3);
        this.Controls.Add(this.button1);
        this.Controls.Add(this.textBox2);
        this.Controls.Add(this.textBox1);
        this.Controls.Add(this.label2);
        this.Controls.Add(this.dataGridView34);
        this.Controls.Add(this.button2_second_type_cable);
        this.Controls.Add(this.button1_Summa);
        this.Controls.Add(this.label1);
        this.Name = "rating2";
        this.Load += new
System.EventHandler(this.rating2_Load_1);
        ((System.ComponentModel.ISupportInitialize)(this.dataGridView34)
).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.lvSDataSet21Bin
dingSource)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.lvSDataSet21)).
EndInit();

        this.ResumeLayout(false);
        this.PerformLayout();

    }
#endregion
    private System.Windows.Forms.Label label1;
    private System.Windows.Forms.Button button1_Summa;
    private System.Windows.Forms.Button
button2_second_type_cable;
    private System.Windows.Forms.DataGridView
dataGridView34;
    private System.Windows.Forms.Label label2;
    public System.Windows.Forms.TextBox textBox1;

```

```

        public System.Windows.Forms.TextBox textBox2;
        private System.Windows.Forms.Button button1;
        private System.Windows.Forms.BindingSource
lvSDataSet21BindingSource;
        private LVSDataset2 lvSDataSet21;
        private System.Windows.Forms.Label label4;
        public System.Windows.Forms.Label label3;
    }
}
namespace WindowsFormsApplication1
{
    partial class DB3
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components =
null;
        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources
should be disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }
        #region Windows Form Designer generated code
        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.components = new
System.ComponentModel.Container();
            this.menuStrip1 = new
System.Windows.Forms.MenuStrip();
            this.файлToolStripMenuItem = new
System.Windows.Forms.ToolStripItem();
            this.опоррамаToolStripMenuItem = new
System.Windows.Forms.ToolStripItem();
            this.dataGridView1 = new
System.Windows.Forms.DataGridview();
            this.localDBBindingSource = new
System.Windows.Forms.BindingSource(this.components);
            this.lvSDataSet = new
WindowsFormsApplication1.LVSDataset();

```

```

        this.textBox_id = new
System.Windows.Forms.TextBox();
        this.textBox_PRODUCT = new
System.Windows.Forms.TextBox();
        this.textBox_PRICE = new
System.Windows.Forms.TextBox();
        this.textBox_postavka = new
System.Windows.Forms.TextBox();
        this.localDBTableAdapter = new
WindowsFormsApplication1.LVSDatasetTableAdapters.localDBTableAda
pter();
        this.lVSDatasetBindingSource = new
System.Windows.Forms.BindingSource(this.components);
        this.dataGridView2 = new
System.Windows.Forms.DataGridview();
        this.lockerDBBindingSource = new
System.Windows.Forms.BindingSource(this.components);
        this.lVSDataset1 = new
WindowsFormsApplication1.LVSDataset1();
        this.textBox_id_2 = new
System.Windows.Forms.TextBox();
        this.lockerDBTableAdapter = new
WindowsFormsApplication1.LVSDataset1TableAdapters.lockerDBTableA
dapter();
        this.textBox_LOCKER = new
System.Windows.Forms.TextBox();
        this.textBoxPRICE_2 = new
System.Windows.Forms.TextBox();
        this.textBoxpostavka_2 = new
System.Windows.Forms.TextBox();
        this.dataGridView3 = new
System.Windows.Forms.DataGridview();
        this.sksDBBindingSource = new
System.Windows.Forms.BindingSource(this.components);
        this.lVSDataset2 = new
WindowsFormsApplication1.LVSDataset2();
        this.sksDBTableAdapter = new
WindowsFormsApplication1.LVSDataset2TableAdapters.sksDBTableAdap
ter();
        this.textBox_id_3 = new
System.Windows.Forms.TextBox();
        this.textBoxtypeSKS = new
System.Windows.Forms.TextBox();
        this.textBoxPRICE_3 = new
System.Windows.Forms.TextBox();
        this.textBoxpostavka_3 = new
System.Windows.Forms.TextBox();
        this.textBoxUpdate_1 = new
System.Windows.Forms.TextBox();
        this.label3 = new System.Windows.Forms.Label();
        this.button2_second_type_cable = new
System.Windows.Forms.Button();
        this.label1 = new System.Windows.Forms.Label();

```

```

        this.label2 = new System.Windows.Forms.Label();
        this.label4 = new System.Windows.Forms.Label();
        this.label5 = new System.Windows.Forms.Label();
        this.button1 = new System.Windows.Forms.Button();
        this.menuStrip1.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.dataGridView1))
.BeginInit();

((System.ComponentModel.ISupportInitialize)(this.localDBBindingS
ource)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.lvsDataSet)).Be
ginInit();

((System.ComponentModel.ISupportInitialize)(this.lvsDataSetBindi
ngSource)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.dataGridView2))
.BeginInit();

((System.ComponentModel.ISupportInitialize)(this.lockerDBBinding
Source)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.lvsDataSet1)).B
eginInit();

((System.ComponentModel.ISupportInitialize)(this.dataGridView3))
.BeginInit();

((System.ComponentModel.ISupportInitialize)(this.sksDBBindingSou
rce)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.lvsDataSet2)).B
eginInit();
        this.SuspendLayout();
        //
        // menuStrip1
        //
        this.menuStrip1.ImageScalingSize = new
System.Drawing.Size(24, 24);
        this.menuStrip1.Items.AddRange(new
System.Windows.Forms.ToolStripItem[] {
            this.файдToolStripMenuItem,
            this.оПрограммеToolStripMenuItem});
        this.menuStrip1.Location = new
System.Drawing.Point(0, 0);
        this.menuStrip1.Name = "menuStrip1";
        this.menuStrip1.Size = new System.Drawing.Size(1686,
24);
        this.menuStrip1.TabIndex = 0;
        this.menuStrip1.Text = "menuStrip1";

```

```

        this.menuStrip1.ItemClicked += new
System.Windows.Forms.ToolStripItemClickedEventHandler(this.menuS
trip1_ItemClicked);
        //
        // файлToolStripMenuItem
        //
        this.файлToolStripMenuItem.Name =
"файлToolStripMenuItem";
        this.файлToolStripMenuItem.Size = new
System.Drawing.Size(48, 20);
        this.файлToolStripMenuItem.Text = "Файл";
        this.файлToolStripMenuItem.Click += new
System.EventHandler(this.файлToolStripMenuItem_Click);
        //
        // оПрограммеToolStripMenuItem
        //
        this.оПрограммеToolStripMenuItem.Name =
"оПрограммеToolStripMenuItem";
        this.оПрограммеToolStripMenuItem.Size = new
System.Drawing.Size(94, 20);
        this.оПрограммеToolStripMenuItem.Text = "О
программе";
        //
        // dataGridView1
        //
        this.dataGridView1.AllowUserToAddRows = false;
        this.dataGridView1.AllowUserToDeleteRows = false;
        this.dataGridView1.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.Aut
oSize;
        this.dataGridView1.Location = new
System.Drawing.Point(0, 68);
        this.dataGridView1.Name = "dataGridView1";
        this.dataGridView1.ReadOnly = true;
        this.dataGridView1.RowHeadersWidth = 62;
        this.dataGridView1.Size = new
System.Drawing.Size(517, 229);
        this.dataGridView1.TabIndex = 1;
        this.dataGridView1.CellClick += new
System.Windows.Forms.DataGridViewCellEventHandler(this.dataGridViewV
iew1_CellClick);
        //
        // localDBBindingSource
        //
        this.localDBBindingSource.DataMember = "localDB";
        this.localDBBindingSource.DataSource =
this.lVSDataset;
        //
        // lVSDataset
        //
        this.lVSDataset.DataSetName = "LVSDataset";
        this.lVSDataset.SchemaSerializationMode =
System.Data.SchemaSerializationMode.IncludeSchema;

```

```

        //
        // textBox_id
        //
        this.textBox_id.Location = new
System.Drawing.Point(179, 367);
        this.textBox_id.Name = "textBox_id";
        this.textBox_id.Size = new System.Drawing.Size(220,
20);

        this.textBox_id.TabIndex = 2;
        //
        // textBox_PRODUCT
        //
        this.textBox_PRODUCT.Location = new
System.Drawing.Point(179, 407);
        this.textBox_PRODUCT.Name = "textBox_PRODUCT";
        this.textBox_PRODUCT.Size = new
System.Drawing.Size(220, 20);
        this.textBox_PRODUCT.TabIndex = 3;
        //
        // textBox_PRICE
        //
        this.textBox_PRICE.Location = new
System.Drawing.Point(179, 449);
        this.textBox_PRICE.Name = "textBox_PRICE";
        this.textBox_PRICE.Size = new
System.Drawing.Size(220, 20);
        this.textBox_PRICE.TabIndex = 4;
        //
        // textBox_postavka
        //
        this.textBox_postavka.Location = new
System.Drawing.Point(179, 491);
        this.textBox_postavka.Name = "textBox_postavka";
        this.textBox_postavka.Size = new
System.Drawing.Size(220, 20);
        this.textBox_postavka.TabIndex = 5;
        //
        // localDBTableAdapter
        //
        this.localDBTableAdapter.ClearBeforeFill = true;
        //
        // lVSDatasetBindingSource
        //
        this.lVSDatasetBindingSource.DataSource =
this.lVSDataset;
        this.lVSDatasetBindingSource.Position = 0;
        //
        // dataGridView2
        //
        this.dataGridView2.AllowUserToAddRows = false;
        this.dataGridView2.AllowUserToDeleteRows = false;

```

```

        this.dataGridView2.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;
        this.dataGridView2.Location = new
System.Drawing.Point(535, 68);
        this.dataGridView2.Name = "dataGridView2";
        this.dataGridView2.ReadOnly = true;
        this.dataGridView2.RowHeadersWidth = 62;
        this.dataGridView2.Size = new
System.Drawing.Size(514, 229);
        this.dataGridView2.TabIndex = 6;
        this.dataGridView2.CellClick += new
System.Windows.Forms.DataGridViewCellEventHandler(this.dataGridView2_CellClick);
        //
        // lockerDBBindingSource
        //
        this.lockerDBBindingSource.DataMember = "lockerDB";
        this.lockerDBBindingSource.DataSource =
this.lVSDataset1;
        //
        // lVSDataset1
        //
        this.lVSDataset1.DataSetName = "LVSDataset1";
        this.lVSDataset1.SchemaSerializationMode =
System.Data.SchemaSerializationMode.IncludeSchema;
        //
        // textBox_id_2
        //
        this.textBox_id_2.Location = new
System.Drawing.Point(609, 367);
        this.textBox_id_2.Name = "textBox_id_2";
        this.textBox_id_2.Size = new
System.Drawing.Size(196, 20);
        this.textBox_id_2.TabIndex = 7;
        //
        // lockerDBTableAdapter
        //
        this.lockerDBTableAdapter.ClearBeforeFill = true;
        //
        // textBox_LOCKER
        //
        this.textBox_LOCKER.Location = new
System.Drawing.Point(609, 407);
        this.textBox_LOCKER.Name = "textBox_LOCKER";
        this.textBox_LOCKER.Size = new
System.Drawing.Size(196, 20);
        this.textBox_LOCKER.TabIndex = 8;
        //
        // textBoxPRICE_2
        //
        this.textBoxPRICE_2.Location = new
System.Drawing.Point(609, 449);

```

```

        this.textBoxPRICE_2.Name = "textBoxPRICE_2";
        this.textBoxPRICE_2.Size = new
System.Drawing.Size(196, 20);
        this.textBoxPRICE_2.TabIndex = 9;
        //
        // textBoxpostavka_2
        //
        this.textBoxpostavka_2.Location = new
System.Drawing.Point(609, 491);
        this.textBoxpostavka_2.Name = "textBoxpostavka_2";
        this.textBoxpostavka_2.Size = new
System.Drawing.Size(196, 20);
        this.textBoxpostavka_2.TabIndex = 10;
        //
        // dataGridView3
        //
        this.dataGridView3.AllowUserToAddRows = false;
        this.dataGridView3.AllowUserToDeleteRows = false;
        this.dataGridView3.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.Auto
oSize;
        this.dataGridView3.Location = new
System.Drawing.Point(1068, 68);
        this.dataGridView3.Name = "dataGridView3";
        this.dataGridView3.ReadOnly = true;
        this.dataGridView3.RowHeadersWidth = 62;
        this.dataGridView3.Size = new
System.Drawing.Size(555, 229);
        this.dataGridView3.TabIndex = 11;
        this.dataGridView3.CellClick += new
System.Windows.Forms.DataGridViewCellEventHandler(this.dataGridViewV
iew3_CellClick);
        //
        // sksDBBindingSource
        //
        this.sksDBBindingSource.DataMember = "sksDB";
        this.sksDBBindingSource.DataSource =
this.lVSDataset2;
        //
        // lVSDataset2
        //
        this.lVSDataset2.DataSetName = "LVSDataset2";
        this.lVSDataset2.SchemaSerializationMode =
System.Data.SchemaSerializationMode.IncludeSchema;
        //
        // sksDBTableAdapter
        //
        this.sksDBTableAdapter.ClearBeforeFill = true;
        //
        // textBox_id_3
        //
        this.textBox_id_3.Location = new
System.Drawing.Point(1098, 367);

```

```

        this.textBox_id_3.Name = "textBox_id_3";
        this.textBox_id_3.Size = new
System.Drawing.Size(190, 20);
        this.textBox_id_3.TabIndex = 12;
        //
        // textBoxtypeSKS
        //
        this.textBoxtypeSKS.Location = new
System.Drawing.Point(1098, 406);
        this.textBoxtypeSKS.Name = "textBoxtypeSKS";
        this.textBoxtypeSKS.Size = new
System.Drawing.Size(190, 20);
        this.textBoxtypeSKS.TabIndex = 13;
        //
        // textBoxPRICE_3
        //
        this.textBoxPRICE_3.Location = new
System.Drawing.Point(1098, 449);
        this.textBoxPRICE_3.Name = "textBoxPRICE_3";
        this.textBoxPRICE_3.Size = new
System.Drawing.Size(190, 20);
        this.textBoxPRICE_3.TabIndex = 14;
        //
        // textBoxpostavka_3
        //
        this.textBoxpostavka_3.Location = new
System.Drawing.Point(1098, 490);
        this.textBoxpostavka_3.Name = "textBoxpostavka_3";
        this.textBoxpostavka_3.Size = new
System.Drawing.Size(190, 20);
        this.textBoxpostavka_3.TabIndex = 15;
        //
        // textBoxUpdate_1
        //
        this.textBoxUpdate_1.Location = new
System.Drawing.Point(277, 592);
        this.textBoxUpdate_1.Name = "textBoxUpdate_1";
        this.textBoxUpdate_1.Size = new
System.Drawing.Size(122, 20);
        this.textBoxUpdate_1.TabIndex = 18;
        this.textBoxUpdate_1.TextChanged += new
System.EventHandler(this.textBox1_TextChanged);
        //
        // label3
        //
        this.label3.FlatStyle =
System.Windows.Forms.FlatStyle.Popup;
        this.label3.Font = new
System.Drawing.Font("Microsoft Sans Serif", 9.75F,
System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte) 204));
        this.label3.Location = new System.Drawing.Point(80,
592);

```

```

        this.label3.Name = "label3";
        this.label3.Size = new System.Drawing.Size(191, 20);
        this.label3.TabIndex = 19;
        this.label3.Text = "Поиск оборудования";
        //
        // button2_second_type_cable
        this.button2_second_type_cable.FlatStyle =
System.Windows.Forms.FlatStyle.Popup;
        this.button2_second_type_cable.Font = new
System.Drawing.Font("Microsoft Sans Serif", 14.25F,
System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(204)));
        this.button2_second_type_cable.Location = new
System.Drawing.Point(609, 539);
        this.button2_second_type_cable.Name =
"button2_second_type_cable";
        this.button2_second_type_cable.Size = new
System.Drawing.Size(221, 73);
        this.button2_second_type_cable.TabIndex = 20;
        this.button2_second_type_cable.Text = "Обновить
таблицы";
        this.button2_second_type_cable.UseVisualStyleBackColor
= true;
        //
        // label1
        //
        this.label1.FlatStyle =
System.Windows.Forms.FlatStyle.Popup;
        this.label1.Font = new
System.Drawing.Font("Microsoft Sans Serif", 9.75F,
System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(204)));
        this.label1.Location = new System.Drawing.Point(12,
367);
        this.label1.Name = "label1";
        this.label1.Size = new System.Drawing.Size(161, 20);
        this.label1.TabIndex = 21;
        this.label1.Text = "Модель";
        //
        // label2
        //
        this.label2.FlatStyle =
System.Windows.Forms.FlatStyle.Popup;
        this.label2.Font = new
System.Drawing.Font("Microsoft Sans Serif", 9.75F,
System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(204)));
        this.label2.Location = new System.Drawing.Point(12,
407);
        this.label2.Name = "label2";
        this.label2.Size = new System.Drawing.Size(161, 20);
        this.label2.TabIndex = 22;
        this.label2.Text = "Поставщик";

```

```

        //
        // label4
        //
        this.label4.FlatStyle =
System.Windows.Forms.FlatStyle.Popup;
        this.label4.Font = new
System.Drawing.Font("Microsoft Sans Serif", 9.75F,
System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte) (204)));
        this.label4.Location = new System.Drawing.Point(12,
450);

        this.label4.Name = "label4";
        this.label4.Size = new System.Drawing.Size(161, 20);
        this.label4.TabIndex = 23;
        this.label4.Text = "Тип/информация";
        //
        // label5
        //
        this.label5.FlatStyle =
System.Windows.Forms.FlatStyle.Popup;
        this.label5.Font = new
System.Drawing.Font("Microsoft Sans Serif", 9.75F,
System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte) (204)));
        this.label5.Location = new System.Drawing.Point(12,
491);

        this.label5.Name = "label5";
        this.label5.Size = new System.Drawing.Size(161, 20);
        this.label5.TabIndex = 24;
        this.label5.Text = "Цена";
        //
        // button1
        //
        this.button1.FlatStyle =
System.Windows.Forms.FlatStyle.Popup;
        this.button1.Font = new
System.Drawing.Font("Microsoft Sans Serif", 14.25F,
System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte) (204)));
        this.button1.Location = new
System.Drawing.Point(1420, 539);
        this.button1.Name = "button1";
        this.button1.Size = new System.Drawing.Size(221,
73);

        this.button1.TabIndex = 25;
        this.button1.Text = "ВЫЙТИ в главное меню";
        this.button1.UseVisualStyleBackColor = true;
        //
        // DB3
        //
        this.BackColor =
System.Drawing.SystemColors.ActiveCaption;

```

```

        this.BackgroundImage =
global::WindowsFormsApplication1.Properties.Resources.ЛВС_и_КС_
4_min;
        this.BackgroundImageLayout =
System.Windows.Forms.ImageLayout.Stretch;
        this.ClientSize = new System.Drawing.Size(1686,
624);

        this.Controls.Add(this.button1);
        this.Controls.Add(this.label5);
        this.Controls.Add(this.label4);
        this.Controls.Add(this.label2);
        this.Controls.Add(this.label1);
        this.Controls.Add(this.button2_second_type_cable);
        this.Controls.Add(this.label3);
        this.Controls.Add(this.textBoxUpdate_1);
        this.Controls.Add(this.textBoxpostavka_3);
        this.Controls.Add(this.textBoxPRICE_3);
        this.Controls.Add(this.textBoxtypeSKS);
        this.Controls.Add(this.textBox_id_3);
        this.Controls.Add(this.dataGridView3);
        this.Controls.Add(this.textBoxpostavka_2);
        this.Controls.Add(this.textBoxPRICE_2);
        this.Controls.Add(this.textBox_LOCKER);
        this.Controls.Add(this.textBox_id_2);
        this.Controls.Add(this.dataGridView2);
        this.Controls.Add(this.dataGridView1);
        this.Controls.Add(this.textBox_postavka);
        this.Controls.Add(this.textBox_PRICE);
        this.Controls.Add(this.textBox_PRODUCT);
        this.Controls.Add(this.textBox_id);
        this.Controls.Add(this.menuStrip1);
        this.MainMenuStrip = this.menuStrip1;
        this.Name = "DB3";
        this.Load += new
System.EventHandler(this.DB3_Load_1);
        this.menuStrip1.ResumeLayout(false);
        this.menuStrip1.PerformLayout();

((System.ComponentModel.ISupportInitialize)(this.dataGridView1))
.EndInit();

((System.ComponentModel.ISupportInitialize)(this.localDBBindings
ource)).EndInit();

((System.ComponentModel.ISupportInitialize)(this.lVSDataset)).En
dInit();

((System.ComponentModel.ISupportInitialize)(this.lVSDatasetBindi
ngSource)).EndInit();

((System.ComponentModel.ISupportInitialize)(this.dataGridView2))
.EndInit();

```

```

((System.ComponentModel.ISupportInitialize)(this.lockerDBBinding
Source)).EndInit();

((System.ComponentModel.ISupportInitialize)(this.lvsDataSet1)).E
ndInit();

((System.ComponentModel.ISupportInitialize)(this.dataGridView3))
.EndInit();

((System.ComponentModel.ISupportInitialize)(this.sksDBBindingSou
rce)).EndInit();

((System.ComponentModel.ISupportInitialize)(this.lvsDataSet2)).E
ndInit();
        this.ResumeLayout(false);
        this.PerformLayout();
    }
#endregion
private System.Windows.Forms.MenuStrip menuStrip1;
private System.Windows.Forms.DataGridView dataGridView1;
private System.Windows.Forms.TextBox textBox_id;
private System.Windows.Forms.TextBox textBox_PRODUCT;
private System.Windows.Forms.TextBox textBox_PRICE;
private System.Windows.Forms.TextBox textBox_postavka;
private LVSDataset lvsDataSet;
private System.Windows.Forms.BindingSource
localDBBindingSource;
private LVSDatasetTableAdapters.localDBTableAdapter
localDBTableAdapter;
private System.Windows.Forms.BindingSource
lvsDataSetBindingSource;
private System.Windows.Forms.DataGridView dataGridView2;
private System.Windows.Forms.TextBox textBox_id_2;
private LVSDataset1 lvsDataSet1;
private System.Windows.Forms.BindingSource
lockerDBBindingSource;
private LVSDataset1TableAdapters.lockerDBTableAdapter
lockerDBTableAdapter;
private System.Windows.Forms.TextBox textBox_LOCKER;
private System.Windows.Forms.TextBox textBoxPRICE_2;
private System.Windows.Forms.TextBox textBoxpostavka_2;
private System.Windows.Forms.DataGridView dataGridView3;
private LVSDataset2 lvsDataSet2;
private System.Windows.Forms.BindingSource
sksDBBindingSource;
private LVSDataset2TableAdapters.sksDBTableAdapter
sksDBTableAdapter;
private System.Windows.Forms.TextBox textBox_id_3;
private System.Windows.Forms.TextBox textBoxtypeSKS;
private System.Windows.Forms.TextBox textBoxPRICE_3;
private System.Windows.Forms.TextBox textBoxpostavka_3;
private System.Windows.Forms.TextBox textBoxUpdate_1;

```

```
        private System.Windows.Forms.ToolStripMenuItem  
ToolStripMenuItem;  
        private System.Windows.Forms.ToolStripMenuItem  
ToolStripMenuItem;  
        public System.Windows.Forms.Label label3;  
        private System.Windows.Forms.Button  
button2_second_type_cable;  
        public System.Windows.Forms.Label label11;  
        public System.Windows.Forms.Label label12;  
        public System.Windows.Forms.Label label14;  
        public System.Windows.Forms.Label label15;  
        private System.Windows.Forms.Button button1;  
    }  
}
```