**ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ (АССОЦИАЦИЯ)**

**«КИСЛОВОДСКИЙ ГУМАНИТАРНО-ТЕХНИЧЕСКИЙ ИНСТИТУТ»**

Факультет Инженерный

Кафедра Систем автоматического управления

Направление Управление в технических системах

К защите допустить:

Зав. кафедрой д.т.н., профессор Гайдук А.Р

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2017 г.

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**к выпускной квалификационной работе

на тему:

**РАЗРАБОТКА БАЗЫ ДАННЫХ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ ПОВЕРКИ ПРИБОРОВ УЧЁТА ЭЛЕКТРОЭНЕРГИИ**.

Руководитель работы: Титаренко С.П.

(должность, ученая степень и звание)

Консультанты:

по экономическому разделу к.э.н. Курданова Ю.Е.

по разделу безопасности и экологичности Сербулова Т.Н.

Студент: Костин А.А. гр. 241

(фамилия, имя, отчество, группа)

Кисловодск

2017

**ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ (АССОЦИАЦИЯ)**

**«КИСЛОВОДСКИЙ ГУМАНИТАРНО-ТЕХНИЧЕСКИЙ ИНСТИТУТ»**

Факультет Инженерный

Кафедра Систем автоматического управления

Направление Управление в технических системах

**ЗАДАНИЕ**

# на выпускную квалификационную работу

|  |
| --- |
| Костину Антону Александровичу |

|  |  |
| --- | --- |
| 1.Тема работы: | «Автоматизированная информационная система поверки приборов учета электроэнергии «Пятигорского центра стандартизации и метрологии» |

утверждена приказом по вузу № 9 от 15.01.2017 г.

|  |  |  |
| --- | --- | --- |
| 2. Срок сдачи студентом законченного работы | | 25.06.2017 г. |
| 3. Исходные данные к работе |  | |

3.1. Спроектировать автоматизированную информационную систему поверки приборов учета электроэнергии «Пятигорского центра стандартизации и метрологии», реализующую функции: разграничения прав доступа, проектирование главных кнопочных форм для каждого типа сотрудников и просмотр/изменение справочников и документов; ведение справочников, оформление приема и выдачи приборов, формирование отчетности.

|  |
| --- |
| 3.2. Реализовать автоматизированную информационную систему поверки приборов |
| 3.3. Тестировать автоматизированную информационную систему поверки приборов. |

4. Содержание пояснительной записки

|  |
| --- |
| 1.Введение. |
| 2. Характеристика предприятия |
| 3. Проектирование автоматизированной информационной системы. |
| 4. Разработка автоматизированной информационной системы |
| 5. Расчет экономической эффективности системы |
| 6. Безопасность и экологичность системы. |
| 7. Заключение. |
| 8. Список использованных источников. |

5. Перечень графического материала (с точным указанием обязательных чертежей)

|  |
| --- |
| 1. Исследование ФБУ «Пятигорский ЦСМ» как объекта автоматизации (1 слайд). |
| 2. Диаграммы проекта АИС поверки приборов (1 слайд). |
| 3. Разработка базы данных АИС поверки ФБУ «Пятигорский ЦСМ»  (1 слайд). |
| 4. Интерфейс АИС поверки ФБУ «Пятигорский ЦСМ»(1 слайд). |
| 5. ТЕХНИКО - ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ (1 слайд). |
| 6. БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ СИСТЕМЫ (1 слайд). |
| 6. Консультации по проекту (с указанием относящихся к ним разделов проекта)  Технико-экономическое обоснование к.э.н. /Курданов М.Д./  (подпись)  Безопасность и экологичность /Сербулова Т.Н./  (подпись)   |  | | --- | | 7. Дата выдачи задания  15 декабря 2016 г. | |

|  |
| --- |
| **Руководитель** к.т.н., доцент/Титаренко С.П../ |

(подпись)

|  |  |
| --- | --- |
| Задание принял к исполнению | 15 декабря 2016 г. |

(дата)

|  |
| --- |
| Подпись студента /Костин А.А./ |

(подпись)

УДК 65.011.56

Автоматизированная информационная система

поверки приборов учета электроэнергии

Выпускная квалификационная работа

Костин Антон

Кисловодск, КГТИ, 2017 г.

**Реферат**

Целью выпускной квалификационной работы является проектирование автоматизированной информационной системы поверки приборов учета электроэнергии «Пятигорского центра стандартизации и метрологии».

В процессе проектирования подробно исследован процесс поверки приборов учета электроэнергии в лаборатории «Пятигорского центра стандартизации и метрологии». В результате были сформулированы требования к функциональным характеристикам системы, разработана структура базы данных и интерфейс системы. Реализация системы выполнена с использованием системы управления базами данных MicrosoftAccess.

Кроме того, в выпускной работе проведено исследование безопасности и экологичности системы, а также оценена экономическая эффективность системы.

**Оглавление**

[Введение 5](#_Toc485386350)

[1. ХАРАКТЕРИСТИКА ПРЕДПРИЯТИЯ 6](#_Toc485386351)

[1.1. Организационная структура предприятия 6](#_Toc485386352)

[1.2 Программное обеспечение, используемое предприятием 8](#_Toc485386353)

[2. ПРОЕКТИРОВАНИЕ АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ 9](#_Toc485386354)

[3.8 Шифрование базы данных с помощью пароля 81](#_Toc485386355)

[При открытии зашифрованной базы данных появится диалоговое окно Необходимо ввести пароль. Ввести пароль в поле и нажать кнопку ОК. 81](#_Toc485386356)

[4.1 Общая информация о системе 83](#_Toc485386357)

[4.2 План анализа экономической эффективности 83](#_Toc485386358)

[4.3 Технико-экономическое обоснование разработки программного обеспечения 84](#_Toc485386359)

[5.1 Анализ условий труда исследователя 92](#_Toc485386360)

[5.2. Мероприятия по улучшению условий труда 96](#_Toc485386361)

[5.3. Пожарная опасность 100](#_Toc485386362)

[5.4. Экологические достоинства и недостатки разрабатываемого продукта для природной среды 101](#_Toc485386363)

## Введение

Выпускная квалификационная работа посвящена автоматизации поверки электросчётчиков на примере Федерального бюджетного учреждения «Государственный региональный центр стандартизации метрологии и испытаний в  
 г. Пятигорск», в дальнейшем сокращенно ФБУ«Пятигорский ЦСМ».

ФБУ«Пятигорский ЦСМ»является федеральным центром по поверке электросчётчиков. Автоматизации учета приборов в нем определена большим числом поверок электросчётчиков и большим числом контролёров, для более удобного поиска электросчётчиков нуждающихся в поверке и своевременного предупреждения о ней, а также для более удобного создания отчетности.

Автоматизация учета позволит:оцифровать весь бумажный документооборот; быстрее создавать документацию и копии информации, быстрее составлять отчетность, увеличить производительность труда.

Целью выпускной квалификационной работы является создание автоматизированной информационной системы для федерального бюджетного учреждения.

Цель работы определила следующие задачи:

- выявление связей и сущностей,

- установление структуры предприятия,

- исследование всей документацииучреждения,

- создание требований к разрабатываемому программному обеспечению (ПО),

- анализ услуг, существующих вучреждение,

- создание автоматизированной информационной системы,

- расчёт экономической эффективности разработки,

- оценка экологичности и безопасности для окружающих.

# ХАРАКТЕРИСТИКА ПРЕДПРИЯТИЯ

ФБУ «Пятигорский ЦСМ» является современным, развивающимся учреждение, основанный в городе Пятигорске в 1935 году, которая ведет свою работу в области метрологии и стандартизации, осуществляет: поверку приборов, выпускаемых на производстве, поверку приборов, выпускаемых из ремонта, поверку приборов, находящихся в обращении и поверку образцовых приборов предприятий; контрольно-ревизионная работа.

Контрольно-ревизионная работа заключалась в осуществлении государственного надзора за измерительными приборами, находящимися в применении во всех отраслях народного хозяйства, и проводилась в виде ревизий мер и измерительных приборов.

К основным услугам, предоставляемым организацией, относятся:

– Поверка электросчётчиков и поверяющей техники.

– Контроль за электросчётчиками и поверяющей техникой.

– Выдача сертификата и наклейки поверки.

– Выезд контролёра на дом.

Клиенты ФБУ «Пятигорский ЦСМ» имеют возможность вызвать контролёра на место установки электросчётчика, путем подачи заявки через диспетчерскую службу сервисном центре, а также привезти поверяющую технику в лабораторию самостоятельно. Также проходят плановые проверки электросчётчиков в цели профилактики приборов.

Сертифицированные специалисты центра стандартизации метрологии и испытаний, проходят плановые проверки с целью подтверждения квалификации, а также проходят повышения квалификации.

## Организационная структура предприятия

ФБУ «Пятигорский ЦСМ» работает в Пятигорске и имеет один филиал в Ессентуках. Поэтому её организационная структура проста.

Организационная структура ФБУ «Пятигорский ЦСМ» представлена на рисунке 1.

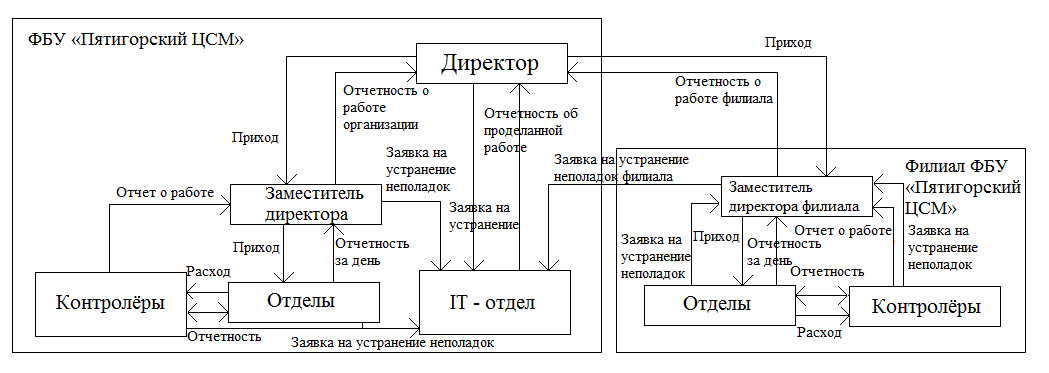


Рисунок 1.1– Организационная структура учреждения

К задачам и обязательствам Отделов ФБУ «Пятигорский ЦСМ» относятся:

– Прием и регистрация заявок на поверку;

– Консультация клиентов по телефону и электронной почте;

– Прием и выдача оборудование после поверки;

– Контроль и координация с работой контролёров;

– Выдача сертификата о поверке;

– Своевременно сообщать о неполадках IT-отделу;

– Предоставление отчетности о выполненных работах.

К функциям и задачам IT-отдела относятся:

– Консультация по полученным заявкам;

– Устранения неполадок;

– Обеспечение стабильной работы учреждения и филиала;

– Предоставление отчетности о выполненных работах Директору;

– Проведение профилактических работ и уведомление об этом;

– Выезд в филиал для работ.

Работой учреждения и филиала руководят Заместитель директора и Заместитель директора филиала соответственно, который назначается на должность и освобождается от должности директором организации. Они выполняют следующие функции и задачи:

– Взаимодействие с клиентами и финансовыми организациями.

– Оформление и сдача бухгалтерской, налоговой и управленческой отчетности деятельности организации.

– Ведение бухгалтерского, налогового и управленческого учета деятельности организации.

– Контроль за работой своего учреждения.

– Взаимодействие с государственными налоговыми и иными органами.

Руководит организацией директор, который несет полную ответственность за результаты принимаемых решений.

## 1.2 Программное обеспечение, используемое предприятием

На предприятии используются все средства Microsoft, распространена операционная система WindowsServer 2016 EnterpriseEdition.

WindowsServer 2016 EnterpriseEdition была создана для предприятий большого и среднего бизнеса. Создана для серверов, работающих с сетевыми программами, почтовыми программами, базами данных, обслуживания пользователей и системами управления запасами. Является периодически обновляемой системой, это позволяет оставаться в безопасности. Это решает проблемы с критическими ошибками системы. Данное программное обеспечение высоконадежно, производительно и экономически эффективно.

Почтовые программы:

OutlookExpress — программа почтовый клиент, а также позволяет получать рассылки от разработчика.

OutlookExpress является системной программой Windows, начиная с Windows 95 OSR 2.5 и Windows NT.

OutlookExpress является облегченной версией MicrosoftOutlook, которая тоже работает с почтой, но при этом владеет большим функционалом. Но на самом деле, между эти программы отличаются друг от друга. Например, MicrosoftOutlook, в отличие от OutlookExpress, не может работать с группами рассылками.

Антивирусные программы, используемые предприятием:

- Kaspersky — это семейство антивирусов, нацеленных для защиты от червей, троянских программ, файловых вирусов, вирусов-баннеров, которые крадут данные пользователя, а также вредят деятельности пользователей, шпионского программного обеспечения, вирусов-похитителей паролей, вероятно опасного программного обеспечения, хакерских программ, программ-люков, программ-шуток и других вредоносных объектов, а также от спама. Преимущество его в том, что можно купить лицензию на несколько компьютеров по существенной низкой цене.

Задача автоматизации поверки электросчётчиков актуальна в связи с большой номенклатурой и большим количеством поверяемого оборудования в лаборатории ФБУ «Пятигорский ЦСМ».

# ПРОЕКТИРОВАНИЕ АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ

**2.1 Описание предметной области**

При разработке концептуальной модели все усилия исполнителя должны быть сосредоточены в основном на структуризации данных и выявлении связей между ними без анализа особенностей реализации и вопросов результативности обработки. Проектирование концептуальной модели основано на анализе решаемых в организации задач по обработке данных. Концептуальная модель содержит в себе описания объектов и их связей между собой, представляющих интерес в анализируемой предметной области и выявляемых в результате рассмотрения данных. Подразумеваются данные, используемые не только в уже разработанных прикладных программах, но и в тех, которые только будут созданы.

Информационная система будет эксплуатироваться в ФБУ «Пятигорский ЦСМ», который занимается поверкой электроприборов и выдача сертификатов.

В настоящее время используется следующие подходы к проектированию информационных систем: структурный, функциональный, объектно-ориентированный и метод с использованием UML - диаграмм, который сочетает все выше перечисленные подходы.

**2.2 Обоснование выбора метода проектирования**

Сущность структурного подхода к разработке ИС заключается в ее декомпозиции (разбиении) на автоматизируемые функции: система разбивается на функциональные подсистемы, которые в свою очередь делятся на подфункции, подразделяемые на задачи и так далее. Процесс разбиения продолжается вплоть до конкретных процедур. При этом автоматизируемая система сохраняет целостное представление, в котором все составляющие компоненты взаимоувязаны. При разработке системы "снизу-вверх" от отдельных задач ко всей системе целостность теряется, возникают проблемы при информационной стыковке отдельных компонентов.

Все наиболее распространенные методологии структурного подхода базируются на ряде общих принципов. В качестве двух базовых принципов используются следующие:

• принцип "разделяй и властвуй" - принцип решения сложных проблем путем их разбиения на множество меньших независимых задач, легких для понимания и решения;

• принцип иерархического упорядочивания - принцип организации составных частей проблемы в иерархические древовидные структуры с добавлением новых деталей на каждом уровне.

В структурном анализе используются в основном две группы средств, иллюстрирующих функции, выполняемые системой и отношения между данными. Каждой группе средств соответствуют определенные виды моделей (диаграмм), наиболее распространенными среди которых являются следующие:

• SADT (StructuredAnalysisandDesignTechnique) модели и соответствующие функциональные диаграммы;

• DFD (DataFlowDiagrams) диаграммы потоков данных;

• ERD (Entity-Relationship Diagrams) диаграммы "сущность-связь".

На стадии проектирования ИС модели расширяются, уточняются и дополняются диаграммами, отражающими структуру программного обеспечения: архитектуру ПО, структурные схемы программ и диаграммы экранных форм.

Перечисленные модели в совокупности дают полное описание ИС независимо от того, является ли она существующей или вновь разрабатываемой. Состав диаграмм в каждом конкретном случае зависит от необходимой полноты описания системы.

Методология функционального проектированияSADT разработана Дугласом Россом. На ее основе разработана методология IDEF0 (IcamDEFinition), которая является основной частью программы ICAM (Интеграция компьютерных и промышленных технологий), проводимой по инициативе ВВС США.

Методология SADT представляет собой совокупность методов, правил и процедур, предназначенных для построения функциональной модели объекта какой-либо предметной области. Функциональная модель SADT отображает функциональную структуру объекта, т.е. производимые им действия и связи между этими действиями. Основные элементы этой методологии основываются на следующих концепциях:

• графическое представление блочного моделирования. Графика блоков и дуг SADT-диаграммы отображает функцию в виде блока, а интерфейсы входа/выхода представляются дугами, соответственно входящими в блок и выходящими из него. Взаимодействие блоков друг с другом описываются посредством интерфейсных дуг, выражающих "ограничения", которые в свою очередь определяют, когда и каким образом функции выполняются и управляются;

• строгость и точность. Выполнение правил SADT требует достаточной строгости и точности, не накладывая в то же время чрезмерных ограничений на действия аналитика. Правила SADT включают:

• ограничение количества блоков на каждом уровне декомпозиции (правило 3-6 блоков);

• связность диаграмм (номера блоков);

• уникальность меток и наименований (отсутствие повторяющихся имен);

• синтаксические правила для графики (блоков и дуг);

• разделение входов и управлений (правило определения роли данных).

• отделение организации от функции, т.е. исключение влияния организационной структуры на функциональную модель.

Методология SADT может использоваться для моделирования широкого круга систем и определения требований и функций, а затем для разработки системы, которая удовлетворяет этим требованиям и реализует эти функции. Для уже существующих систем SADT может быть использована для анализа функций, выполняемых системой, а также для указания механизмов, посредством которых они осуществляются.

Объектно-ориентированный подход к проектированию ИС отличается от функционально-ориентированного подхода лучшей способностью отражать динамическое поведение системы в зависимости от возникающих событий. ООП - рассматривает модель предметной области как совокупность взаимодействующих во времени объектов. Каждый объект рассматривается как экземпляр определенного класса, образующих иерархию классов. Конкретный процесс обработки информации формируется в виде последовательности взаимодействий объектов. Одна операция обработки данных может рассматриваться как результат одного взаимодействия объектов.

Конечным результатом процесса объектно-ориентированного проектирования должно стать множество классов объектов с присоединенными методами обработки атрибутов. Объектно-ориентированный подход предполагает совместное моделирование данных и процессов.

Объектно-ориентированный подход помогает справиться с такими сложными проблемами, как:

• уменьшение сложности программного обеспечения;

• повышение надежности программного обеспечения;

• обеспечение возможности модификации отдельных компонентов программного обеспечения без изменения остальных его компонентов;

• обеспечение возможности повторного использования отдельных компонентов программного обеспечения.

Систематическое применение объектно-ориентированного подхода позволяет разрабатывать хорошо структурированные, надежные в эксплуатации, достаточно просто модифицируемые программные системы. Объектно-ориентированное проектирование программного обеспечения связано с применением объектно-ориентированной методологии (технологии) разработки программных систем, а также инструментальных средств, поддерживающие эти технологии.

Объектно-ориентированная разработка может начаться на самом первом этапе жизненного цикла; она не связана с языком программирования, на котором предполагается реализовать разрабатываемую программную систему: этот язык может и не быть объектно-ориентированным. На этапе разработки объекты - это некоторые формальные конструкции (например, прямоугольники с закругленными углами, с помощью которых они изображаются на схемах), никак пока не связанные с их будущей реализацией на одном из языков программирования.

Объектно-ориентированная разработка программного обеспечения связана с применением объектно-ориентированных методологий (технологий). Обычно эти объектно-ориентированные методологии поддерживаются инструментальными программными средствами, но и без таких средств они полезны, так как позволяют хорошо понять различные аспекты и свойства разрабатываемой программной системы, что в последующем существенно облегчает ее реализацию, тестирование, сопровождение, разработку новых версий и более существенную модификацию.

В настоящее время для объектно-ориентированного моделирования проблемной области широко используется унифицированный язык моделирования UML.

Унифицированный язык объектно-ориентированного моделирования UnifiedModelingLanguage (UML) является универсальным современным средством проектирования информационных систем,которыйподерживается достаточно большим количеством инструментальных средств.UML поддерживает полный жизненный цикл информационных систем и, одновременно,  является достаточно гибким для настройки и поддержки специфики командной разработки.

UML основывается на технологии объектно-ориентированного програмирования. Объединяет в себе всю мощь объектно-ориентированного подхода и дает четкую модель системы, отражающую все ее значимые стороны. К середине девяностых явными лидерами в этой области стали методы Booch (GradyBooch), OMT-2 (JimRumbaugh), OOSE — Object-Oriented SoftwareEngineering (IvarJacobson). Однако эти три метода имели свои сильные и слабые стороны: OOSE был лучшим на стадии анализа проблемной области и анализа требований к системе, OMT-2 был наиболее предпочтителен на стадиях анализа и разработки информационных систем, Booch лучше всего подходил для стадий дизайна и разработки.

Создание UML началось в октябре 1994 г., когда Джим Рамбо и Гради Буч из Rational Software Corporation стали работать над объединением своих методов OMT и Booch. Осенью 1995 г. увидела свет первая черновая версия объединенной методологии, которую они назвали Unified Method 0.8. После присоединения в конце 1995 г. к Rational Software CorporationАйвара Якобсона и его фирмы Objectory, усилия трех создателей наиболее распространенных объектно-ориентированных методологий были объединены и направлены на создание UML.

UML представляет собой объектно-ориентированный язык моделирования, обладающий следующими основными характеристиками:

* является языком визуального моделирования, который обеспечивает разработку репрезентативных моделей для организации взаимодействия заказчика и разработчика ИС, различных групп разработчиков ИС;
* содержит механизмы расширения и специализации базовых концепций языка.

UML — это стандартная нотация визуального моделирования программных систем, принятая консорциумом ObjectManaging Group (OMG) осенью 1997 г., и на сегодняшний день она поддерживается многими объектно-ориентированными CASE-продуктами.

UML включает внутренний набор средств моделирования ("ядро"), которые сейчас приняты во многих методах и средствах моделирования. Эти концепции необходимы в большинстве прикладных задач. 

Пользователям языка предоставлены возможности:

* строить модели на основе средств ядра, без использования механизмов расширения для большинства типовых приложений;
* добавлять при необходимости новые элементы и условные обозначения, если они не входят в ядро, или специализировать компоненты, систему условных обозначений (нотацию) и ограничения для конкретных предметных областей.

Сравнительный анализ выше перечисленных подходов к проектированию ИС показывает, что наиболее универсальным, наиболее полно описывающим все стороны функционирования, и поддерживаемым широким кругом современных инструментальных средств являются универсальный язык моделирования UML, который будет использоваться для проектирования автоматизированной системы поверки приборов учета расхода электроэнергии.

***2.2.1Этапы проектирования ИС с применением UML***

UML обеспечивает поддержку всех этапов жизненного цикла ИС и предоставляет для этих целей ряд графических средств – диаграмм.

На этапе создания концептуальной модели для описания бизнес-деятельности используются модели бизнес-прецедентов и диаграммы видов деятельности, для описания бизнес-объектов – модели бизнес-объектов и диаграммы последователь-ностей.

На этапе создания логической модели ИС описание требований к системе задается в виде модели и описания системных прецедентов, а предварительное проектирование осуществляется с использованием диаграмм классов, диаграмм последовательностей и диаграмм состояний.

На этапе создания физической модели детальное проектирование выполняется с использованием диаграмм классов, диаграмм компонентов, диаграмм развертывания.

Ниже приводятся определения и описывается назначение перечисленных диаграмм и моделей применительно к задачам проектирования ИС (в скобках приведены альтернативные названия диаграмм, использующиеся в современной литературе).

Диаграммы прецедентов (диаграммы вариантов спользования, usecase diagrams) – это обобщенная модель функционирования системы в окружающей среде.

Диаграммы видов деятельности (диаграммы деятельностей, activitydiagrams) – модель бизнес-процесса или поведения системы в рамках прецедента.

Диаграммы взаимодействия (interaction diagrams) – модель процесса обмена сообщениями между объектами, представляется в виде диаграмм последовательностей (sequencediagrams) или кооперативных диаграмм (collaboration diagrams).

Диаграммы состояний (statechartdiagrams) – модель динамического поведения системы и ее компонентов при переходе из одного состояния в другое.

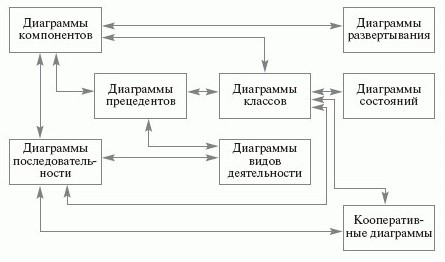
Диаграммы классов (classdiagrams) – логическая модель базовой структуры системы, отражает статическую структуру системы и связи между ее элементами.

Диаграммы базы данных (database diagrams) — модель структуры базы данных, отображает таблицы, столбцы, ограничения и т.п.

Диаграммы компонентов (component diagrams) – модель иерархии подсистем, отражает физическое размещение баз данных, приложений и интерфейсов ИС.

Диаграммы развертывания (диаграммы размещения, deployment diagrams) – модель физической архитектуры системы, отображает аппаратную конфигурацию ИС.

На рисунок 2.1 показаны отношения между различными видами диаграмм UML. Указатели стрелок можно интерпретировать как отношение "является источником входных данных для..." (например, диаграмма прецедентов является источником данных для диаграмм видов деятельности и последовательности). Приведенная схема является наглядной иллюстрацией итеративного характера разработки моделей с использованием UML.

  
Рисунок 2.1 Взаимосвязи между диаграммами UML

Ниже приводятся описания последовательных этапов проектирования ИС с использованием UML.

**2.2.2Использование основных объектов UML в проектирование информационной системы ФБУ «Пятигорский ЦСМ»**

Классы

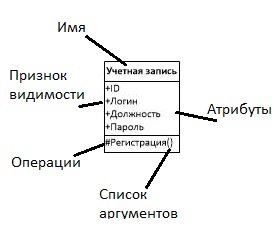
Классы — это базовые элементы любой  объектно- ориентированной системы. Классы представляют собой описание совокупностей однородных объектов с присущими им свойствами — атрибутами, операциями, отношениями и семантикой.

В рамках модели каждому классу присваивается уникальное имя – в нашем проекте это Учетная запись.

Атрибут — это свойство класса, которое может принимать множество значений. Множество допустимых значений атрибута образует домен. Атрибут имеет имя и отражает некоторое свойство моделируемой сущности, общее для всех объектов данного класса. Класс может иметь произвольное количество атрибутов. Например, для класса объектов Учетная запись информационной системы ФБУ «Пятигорский ЦСМ» определены атрибуты ID – уникальный идентификатор пользователя, Логин, Должность, Пароль.

Операция — реализация функции, которую можно запросить у любого объекта класса. Операция показывает, что можно сделать с объектом. Исполнение операции часто связано с обработкой и изменением значений атрибутов объекта, а также изменением состояния объекта. Для класса объектов Учетная запись информационной системы ФБУ «Пятигорский ЦСМ» определена операция Регистрация.

На рисунке 2.2 приведено графическое изображение класса "Учетная запись" в нотации UML.

  
Рисунок. 2.2 Изображение класса в UML другое.

Синтаксис UML для свойств классов (в отдельных программных средствах, например, в IBM UML Modeler, порядок записи параметров может быть иным):

<признак видимости><имя атрибута> : <тип данных>

= <значение по умолчанию>

<признак видимости><имя операции><(список аргументов)>

Для класса объектов Учетная запись описание атрибута ID иметь следующий вид:

publicID: int = 0001

Для класса объектов Учетная запись описание операции Регистрация иметь следующий вид:

protected Регистрация: (int ID, string Логин, string\* Должность, string Пароль)

Видимость свойства указывает на возможность его использования другими классами. Один класс может "видеть" другой, если тот находится в области действия первого и между ними существует явное или неявное отношение. Например, в данной информационной системы ФБУ «Пятигорский ЦСМ» атрибуты и операции доступны для класса объекта Поверка.

В языке UML определены три уровня видимости:

* public (общий) — любой внешний класс, который "видит" данный, может пользоваться его общими свойствами. Обозначаются знаком " + " перед именем атрибута или операции, как в данном проекте для атрибута ID; Например, в объектеАдминистратороперацияУправление правами имеет общую видимость.
* protected (защищенный) — только любой потомок данного класса может пользоваться его защищенными свойствами. Обозначаются знаком " # ", как в данном проекте для операции Регистрация; Например, в объекте Учетная запись операция Регистрация имеет защищенную видимость.
* private (закрытый) — только данный класс может пользоваться этими свойствами. Обозначаются символом " - ". В данной системе нет операции с типом закрытый.

Еще одной важной характеристикой атрибутов и операций классов является область действия. Область действия свойства указывает, будет ли оно проявлять себя по-разному в каждом экземпляре класса, или одно и то же значение свойства будет совместно использоваться всеми экземплярами:

* instance (экземпляр) — у каждого экземпляра класса есть собственное значение данного свойства;
* classifier (классификатор) — все экземпляры совместно используют общее значение данного свойства (выделяется на диаграммах подчеркиванием).

В Базе Данных все классы имеют значение instance.

В данном проекте свойства атрибутов не изменяется при использование в разных классов.

Возможное количество экземпляров класса называется его кратностью. В UML можно определять следующие разновидности классов:

* не содержащие ни одного экземпляра — тогда класс становится служебным ( Abstract );
* содержащие ровно один экземпляр ( Singleton );
* содержащие заданное число экземпляров;
* содержащие произвольное число экземпляров.

В Базе Данных все классы имеют ровно один экземпляр.

Принципиальное назначение классов характеризуют стереотипы. Это, фактически, классификация объектов на высоком уровне, позволяющая определить некоторые основные свойства объекта (пример стереотипа — класс " действующее лицо "). Механизм стереотипов является также средством расширения словаря UML за счет создания на основе существующих блоков языка новых, специфичных для решения конкретной проблемы.

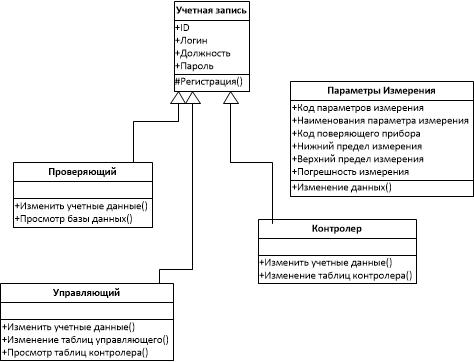
Диаграммы классов

Классы в UML изображаются на диаграммах классов, которые позволяют описать систему в статическом состоянии — определить типы объектов системы и различного рода статические связи между ними.

Классы отображают типы объектов системы.

Между классами возможны различные отношения, представленные на рисунок 2.3:

* зависимости, которые описывают существующие между классами отношения использования;
* обобщения, связывающие обобщенные классы со специализированными;
* ассоциации, отражающие структурные отношения между объектами классов.

  
Рисунок 2.3 Отображение связей между классами

Зависимостью называется отношение использования, согласно которому изменение в спецификации одного элемента (например, класса " Контролер ") может повлиять на использующий его элемент ( класс " Учетная запись "). Часто зависимости показывают, что один класс использует другой в качестве аргумента.

Обобщение — это отношение между общей сущностью (родителем — класс " Администратор ") и ее конкретным воплощением (потомком — классы " Учетная запись "). Объекты класса -потомка могут использоваться всюду, где встречаются объекты класса -родителя, но не наоборот. При этом он наследует свойства родителя (его атрибуты и операции). Операция потомка с той же сигнатурой, что и у родителя, замещает операцию родителя; это свойство называют полиморфизмом. Класс, у которого нет родителей, но есть потомки, называется корневым. Класс, у которого нет потомков, называется листовым.



Рисунок 2.4 – Отношение обобщения

Ассоциация — это отношение, показывающее, что объекты одного типа неким образом связаны с объектами другого типа (" Поверка " может сделать " Учетная запись "). Если между двумя классами определена ассоциация, то можно перемещаться от объектов одного класса к объектам другого. При необходимости направление навигации может задаваться стрелкой. Допускается задание ассоциаций на одном классе. В этом случае оба конца ассоциации относятся к одному и тому же классу. Это означает, что с объектом некоторого класса можно связать другие объекты из того же класса. Ассоциации может быть присвоено имя, описывающее семантику отношений. Каждая ассоциация имеет две роли, которые могут быть отражены на диаграмме (рисунок 2.5). Роль ассоциации обладает свойством множественности, которое показывает, сколько соответствующих объектов может участвовать в данной связи.

  
Рисунок 2.5 Свойства ассоциации

Рисунок 2.5 иллюстрирует модель формирования заказа на поверку счетчика. Каждый заказ может быть создан единственным клиентом (множественность   
роли 1...1). Каждый клиент может создать один и более заказов (множественность роли 1..n). Направление навигации показывает, что каждый заказ должен быть " привязан " к определенному клиенту.

Такого рода ассоциация является простой и отражает отношение между равноправными сущностями, когда оба класса находятся на одном концептуальном уровне и ни один не является более важным, чем другой. Если приходится моделировать отношение типа "часть-целое", то используется специальный тип ассоциации — агрегирование. В такой ассоциации один из классов имеет более высокий ранг (целое — класс " Поверка ") и состоит из нескольких меньших по рангу классов (частей — класс " Операция поверки "). В UML используется и более сильная разновидность агрегации — композиция, в которой объект-часть может принадлежать только единственному целому. В композиции жизненный цикл частей и целого совпадают, любое удаление целого обязательно захватывает и его части.

Для ассоциаций можно задавать атрибуты и операции, создавая по обычным правилам UML классы ассоциаций.

Диаграммы использования

Диаграммы использования описывают функциональность ИС, которая будет видна пользователям системы. "Каждая функциональность" изображается в виде "прецедентов использования" (usecase) или просто прецедентов. Прецедент — это типичное взаимодействие пользователя с системой, которое при этом:

* описывает видимую пользователем функцию,
* может представлять различные уровни детализации,
* обеспечивает достижение конкретной цели, важной для пользователя.

Прецедент обозначается на диаграмме овалом, связанным с пользователями, которых принято называть действующими лицами (актеры, actors). Действующие лица используют систему (или используются системой) в данном прецеденте. Действующее лицо выполняет некоторую роль в данном прецеденте. На диаграмме изображается только одно действующее лицо, однако реальных пользователей, выступающих в данной роли по отношению к ИС, может быть много. Список всех прецедентов фактически определяет функциональные требования к ИС, которые лежат в основе разработки технического задания на создание системы.

На диаграммах прецедентов, кроме связей между действующими лицами и прецедентами, возможно использование еще двух видов связей между прецедентами: "использование" и "расширение" ( рисунок 2.6). Связь типа "расширение" применяется, когда один прецедент подобен другому, но несет несколько большую функциональную нагрузку. Ее следует применять при описании изменений в нормальном поведении системы. Связь типа "использование" позволяет выделить некий фрагмент поведения системы и включать его в различные прецеденты без повторного описания.

На рисунок 2.6 показано, что при исполнении прецедента " Прием на поверку " возможно использование информации из предыдущего заказа, что позволит не вводить все необходимые данные. А при исполнении прецедентов " Регистрация " необходимо выполнить одно и то же действие — создать запись в таблицы "Поверяемое оборудование".

  
Рисунок 2.6 – Связи на диаграммах прецедентов

Динамические аспекты поведения системы отражаются приведенными ниже диаграммами.

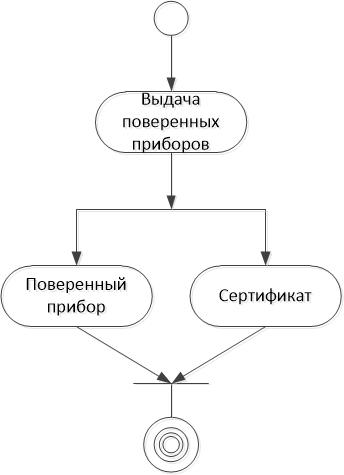
В отличие от некоторых подходов объектного моделирования, когда и состояние, и поведение системы отображаются на диаграммах классов, UML отделяет описание поведения в диаграммы взаимодействия. В UML диаграммы классов не содержат сообщений, которые усложняют их чтение. Поток сообщений между объектами выносится на диаграммы взаимодействия. Как правило, диаграмма взаимодействия охватывает поведение объектов в рамках одного варианта использования.

Прямоугольники на диаграмме представляют различные объекты и роли, которые они имеют в системе, а линии между классами отображают отношения (или ассоциации) между ними. Сообщения обозначаются ярлыками возле стрелок, они могут иметь нумерацию и показывать возвращаемые значения.

Существуют два вида диаграмм взаимодействия: диаграммы последова-тельностей и кооперативные диаграммы.

Диаграммы последовательностей

Этот вид диаграмм используется для точного определения логики сценария выполнения прецедента. Диаграммы последовательностей отображают типы объектов, взаимодействующих при исполнении прецедентов, сообщения, которые они посылают друг другу, и любые возвращаемые значения, ассоциированные с этими сообщениями. Эллипсы на вертикальных линиях показывают "время жизни" объекта. Линии со стрелками и надписями названий методов означают вызов метода у объекта.

  
Рисунок 2.7 – Диаграмма Выдача поверенных приборов

* вводятся строки выдачи с поверки;
* по каждой строке проверяется наличие поверки;
* если прибор поверен — то он выдается с сертификатом о поверке, клиенту;
* если прибор не поверен — инициируется поверка прибора.

Кооперативные диаграммы

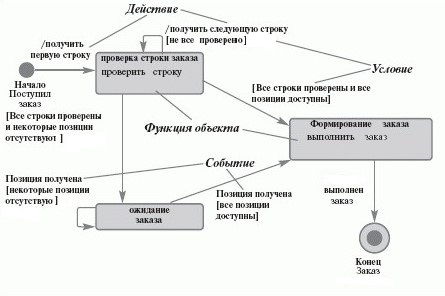
На кооперативных диаграммах объекты (или классы ) показываются в виде прямоугольников, а стрелками обозначаются сообщения, которыми они обмениваются в рамках одного варианта использования. Временная последовательность сообщений отражается их нумерацией.

Диаграммы состояний

Диаграммы состояний используются для описания поведения сложных систем. Они определяют все возможные состояния, в которых может находиться объект, а также процесс смены состояний объекта в результате некоторых событий.Эти диаграммы обычно используются для описания поведения одного объекта в нескольких прецедентах.

Эллипсами представляются состояния, через которые проходит объект во время своего поведения. Состояниям соответствуют определенные значения атрибутов объектов. Стрелки представляют переходы от одного состояния к другому, которые вызываются выполнением некоторых функций объекта. Имеется также два вида псевдо-состояний: начальное состояние, в котором находится только что созданный объект, и конечное состояние, которое объект не покидает, как только туда перешел.

Переходы имеют метки, которые синтаксически состоят из трех необязательных частей (см. рисунок 2.8):

  
Рисунок 2.8Диаграмма состояний объекта «заказ»

<Событие><[Условие]>< / Действие>.

На диаграммах также отображаются функции, которые выполняются объектом в определенном состоянии. Синтаксис метки деятельности:

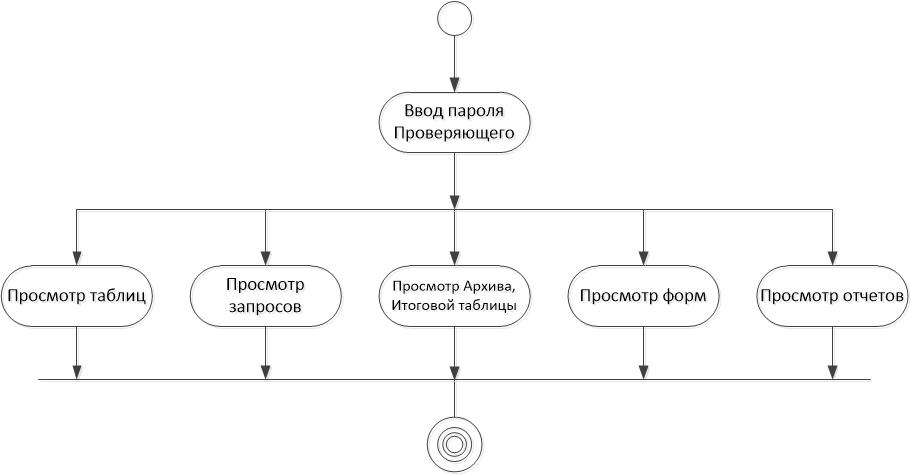
выполнить/< деятельность >.

Диаграммы деятельности

Диаграмма деятельности — это частный случай диаграммы состояний. На диаграмме деятельности представлены переходы потока управления от одной деятельности к другой внутри системы. Этот вид диаграмм обычно используется для описания поведения, включающего в себя множество параллельных процессов.

Основными элементами диаграмм деятельности являются (рисунок 2.9):

* овалы, изображающие действия объекта;
* линейки синхронизации, указывающие на необходимость завершить или начать несколько действий (модель логического условия "И");
* ромбы, отражающие принятие решений по выбору одного из маршрутов выполнения процесса (модель логического условия "ИЛИ");
* стрелки — отражают последовательность действий, могут иметь метки условий.

  
Рисунок 2.9**–** Диаграмма деятельности — обработка заказа

На диаграмме деятельности могут быть представлены действия, соответствующие нескольким вариантам использования. На таких диаграммах появляется множество начальных точек, поскольку они отражают теперь реакцию системы на множество внешних событий. Таким образом, диаграммы деятельности позволяют получить полную картину поведения системы и легко оценивать влияние изменений в отдельных вариантах использования на конечное поведение системы.

Любая деятельность может быть подвергнута дальнейшей декомпозиции и представлена в виде отдельной диаграммы деятельности или спецификации (словесного описания).

Диаграммы компонентов

Диаграммы компонентов позволяют изобразить модель системы на физическом уровне.

Элементами диаграммы являются компоненты — физические замещаемые модули системы. Каждый компонент является полностью независимым элементом системы. Разновидностью компонентов являются узлы. Узел — это элемент реальной (физической) системы, который существует во время функционирования программного комплекса и представляет собой вычислительный ресурс, обычно обладающий как минимум некоторым объемом памяти, а часто еще и способностью обработки. Узлы делятся на два типа:

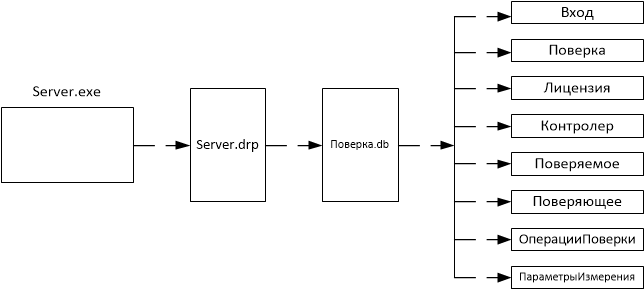
* устройства — узлы системы, в которых данные не обрабатываются.
* процессоры — узлы системы, осуществляющие обработку данных.

Для различных типов компонентов предусмотрены соответствующие стереотипы в языке UML.

Компонентом может быть любой достаточно крупный модульный объект, такой как таблица или экстент базы данных, подсистема, бинарный исполняемый файл, готовая к использованию система или приложение. Таким образом, диаграмму компонентов можно рассматривать как диаграмму классов в более крупном (менее детальном) масштабе. Компонент, как правило, представляет собой физическую упаковку логических элементов, таких как классы, интерфейсы и кооперации.

Основное назначение диаграмм компонентов — разделение системы на элементы, которые имеют стабильный интерфейс и образуют единое целое. Это позволяет создать ядро системы, которое не будет меняться в ответ на изменения, происходящие на уровне подсистем.

На рисунок 2.10 показана упрощенная схема элементов фрагмента серверной части базы данных. Прямоугольники представляют собой компоненты — приложения или внутренние подсистемы. Пунктирные линии отражают зависимости между компонентами.

  
Рисунок 2.10 Диаграмма компонентов фрагмента КИС

Каждый компонент диаграммы при необходимости документируется с помощью более детальной диаграммы компонентов, диаграммы сценариев или диаграммы классов.

Пакеты UML

Пакеты представляют собой универсальный механизм организации элементов в группы. В пакет можно поместить диаграммы различного типа и назначения. В отличие от компонентов, существующих во время работы программы, пакеты носят чисто концептуальный характер, то есть существуют только во время разработки. Изображается пакет в виде папки с закладкой, содержащей, как правило, только имя и иногда — описание содержимого.

Диаграмма пакетов содержит пакеты классов и зависимости между ними. Зависимость между двумя пакетами имеет место в том случае, если изменения в определении одного элемента влекут за собой изменения в другом. По отношению к пакетам можно использовать механизм обобщения (см. выше раздел " Диаграммы классов ").

**2.3Диаграмма вариантов использования**

Описание функционального назначения информационной системы ФБУ «Пятигорский ЦСМ» представим диаграммой вариантов использования UML. Диаграмма строится на основе анализа предметной области, является начальным концептуальным представлением системы в процессе её проектирования и разработки. В дальнейшем она детализируется в форме логических и физических моделей.

При построении диаграммы использования информационной системы ФБУ «Пятигорский ЦСМ» в качестве сущностей или актёров, используется пользователи системы: Администратор, Контролёр, Управляющий и Проверяющий. Вариантами использования являются: Регистрация и редактирование поверяющего оборудования; Внесение и редактирование данных о лицензиях и контролеров; Проверка деятельности контролера; Отчеты о поверяющих приборов и о поверки; Создание и выдача сертификатов; Регистрация и редактирование поверяемых приборов; Внесение данных о поверках; Отчеты о поверяемых приборов и о поверки; Создание и выдача наклеек и сертификатов; Просмотр данных в Базе Данных; Редактирование Базы Данных; Редактирование данных входа; Редактирование классов доступов в Базу Данных; Просмотр данных в отчетах; Просмотр Итоговой таблицы и Архива данных; Просмотр методов регистрации и редактирования Базы Данных.

Для Администратора системы определены следующие варианты использования: Просмотр данных в Базе Данных; Редактирование Базы Данных; Редактирование данных входа; Редактирование классов доступов в Базу Данных.

Для Проверяющего определены следующие варианты использования: Просмотр данных в Базе Данных; Просмотр данных в отчетах; Просмотр Итоговой таблицы и Архива данных; Просмотр методов регистрации и редактирования Базы Данных.

Для Проверяющего определены следующие варианты использования: Регистрация и редактирование поверяющего оборудования; Внесение и редактирование данных о лицензиях и контролеров; Проверка деятельности контролера; Отчеты о поверяющих приборов и о поверки; Создание и выдача сертификатов.

Для Проверяющего определены следующие варианты использования: Регистрация и редактирование поверяемых приборов; Внесение данных о поверках; Отчеты о поверяемых приборов и о поверки; Создание и выдача наклеек и сертификатов.

Как видно из диаграммы использования вариант использования Просмотр данных в Базе Данных является общим для актёров Администратор и Проверяющий.

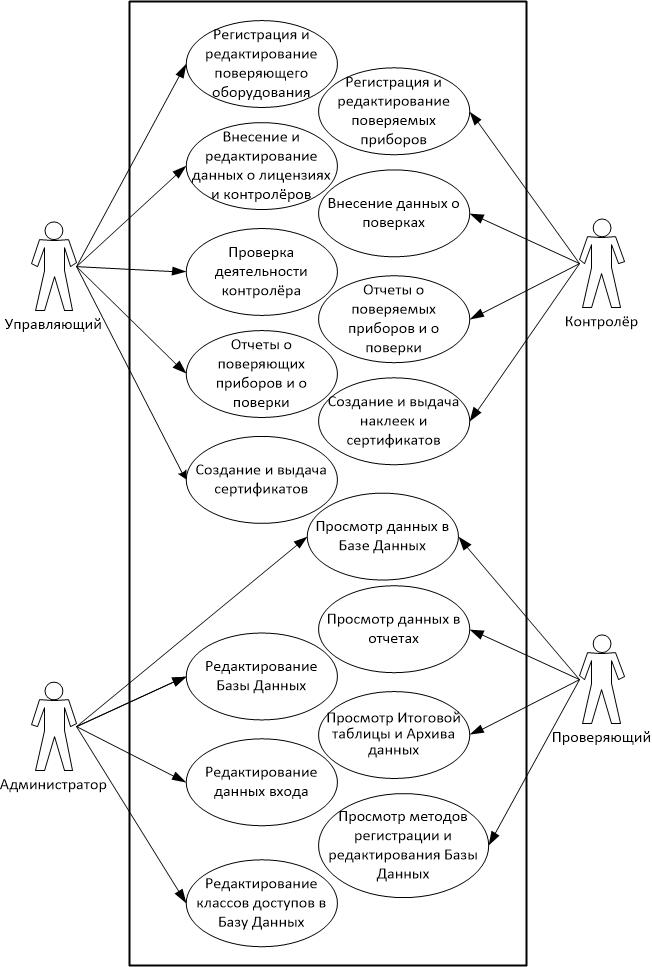


Рисунок2.11 – Диаграмма вариантов использования

Диаграмма использования в данной работе определит интерфейс информационной системы ФБУ «Пятигорский ЦСМ», а именно, состав форм для ввода информации о поверяемых приборов, регистрации новых пользователей, формирование запросов на отчёты системы.

**2.4 Диаграмма Сущность-связь**

Диаграмма "сущность-связь" (ERD) нацелена на определения отношений между моделями данных и их типов.

Фактически с помощью диаграммы "сущность-связь" осуществляется детализация хранилищ данных проектируемой информационной системы, а также фиксируются сущности системы и способы их взаимодействия, включая определение объектов, важных для предметной области, то есть сущностей, свойств этих объектов, то есть атрибутов, и их отношений с другими объектами, то есть связей.

Эта диаграммаиспользуется в нашем случае дляпроектирования сетевой базы данных.

Сущностями являются Администратор, Диспечер, Управляющий, Проверяющий,Прием на поверку, Выдача поверенных приборов.

Все пользователиподтверждают свою личность в системе, путем ввода индивидуального логина и пароля.

Администратор осуществляет Редактирование классов доступов в Базу Данных и редактирование данных входа в неё.

Проверяющий осуществляет проверку деятельности Контролера и Управляющего.

Управляющий осуществляет проверку деятельности Контролера. А также может создавать любое количествоПрием на поверку и Выдача поверенных приборов. Прием на поверкусодержитназвание приборов осуществляющие поверку, а также информацию о предыдущей поверки, информацию о производителе и дате ввода в эксплуатацию.Выдача поверенных приборовсодержитинформацию о выданном поверяющем приборе и сертификат подтверждающий поверку оборудования и всю краткую информацию о нем.

Контролер может создавать любое количествоПрием на поверку и Выдача поверенных приборов. Прием на поверкусодержитназвание электросчетчика, принесенного поверку, а также информацию о предыдущей поверки, информацию о производителе и дате ввода в эксплуатацию.Выдача поверенных приборовсодержитинформацию о выданном электросчетчике и сертификат подтверждающий поверку оборудования и всю краткую информацию о нем.

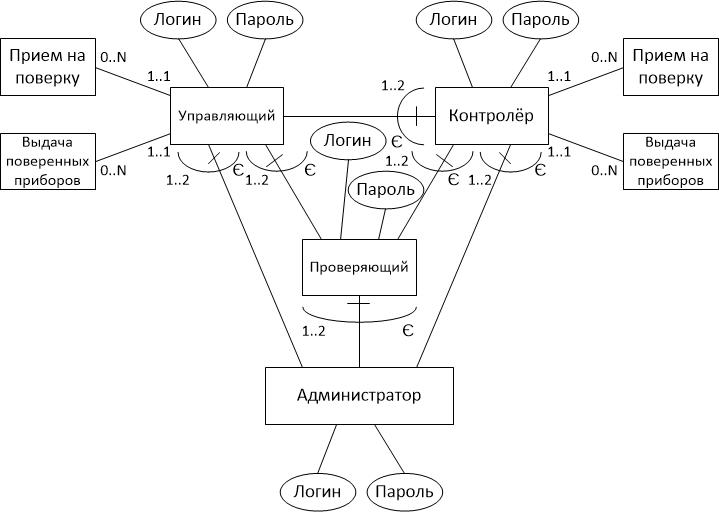


Рисунок 2.12 – Диаграмма сущность-связь

**2.5 Диаграмма развертывания**

Диаграмма развертывания - метод объектно-ориентированного проектирования, отражающий физические взаимосвязи между программными и аппаратными составляющими системы.

Диаграмма развертывания предназначена для отображения компонентов и элементов программы, имеющихся в Базе Данных. На этой диаграмме отображаются все компоненты, входящие в информационную систему ФБУ «Пятигорский ЦСМ».

Диаграмма развертывания содержит графическое изображение всех сети, которая пользуется проектируемой Базой данных. На ней изображены все элементы сети, без которой не сможет функционировать нормально система.

Узлами-экземплярами в системе учета выработки являются автоматизированное рабочее место Администратора, автоматизированное рабочее место Управляющего, с имеющимся на нем компонентом Client.exe; сервер базы данных с имеющимся на нем компонентами Server.exe и Поверка.db. В составПоверка.db входя такие компоненты как: Вход, Поверка, Лицензия, Контролер, Поверенные приборы, Поверяющие приборы, Операции поверки, Параметры Измерения, расположенные на сервере.

Соединения являются разновидностью ассоциации и изображаются отрезками линий. Присутствие такой линии указывает на необходимость организации физического канала для обмена информацией между соответствующими узлами.

Для связи с Поверяющими и Контролерами используется интернет(Internet, TCP\IP, Wi-Fi) для того, что эти пользователи имели доступ к информационной системе ФБУ «Пятигорский ЦСМ» в любой точке мира. А также это сделанно для того, что Проверяющий мог в любой момент войти в базу данных и проверить деятельность учереждения.

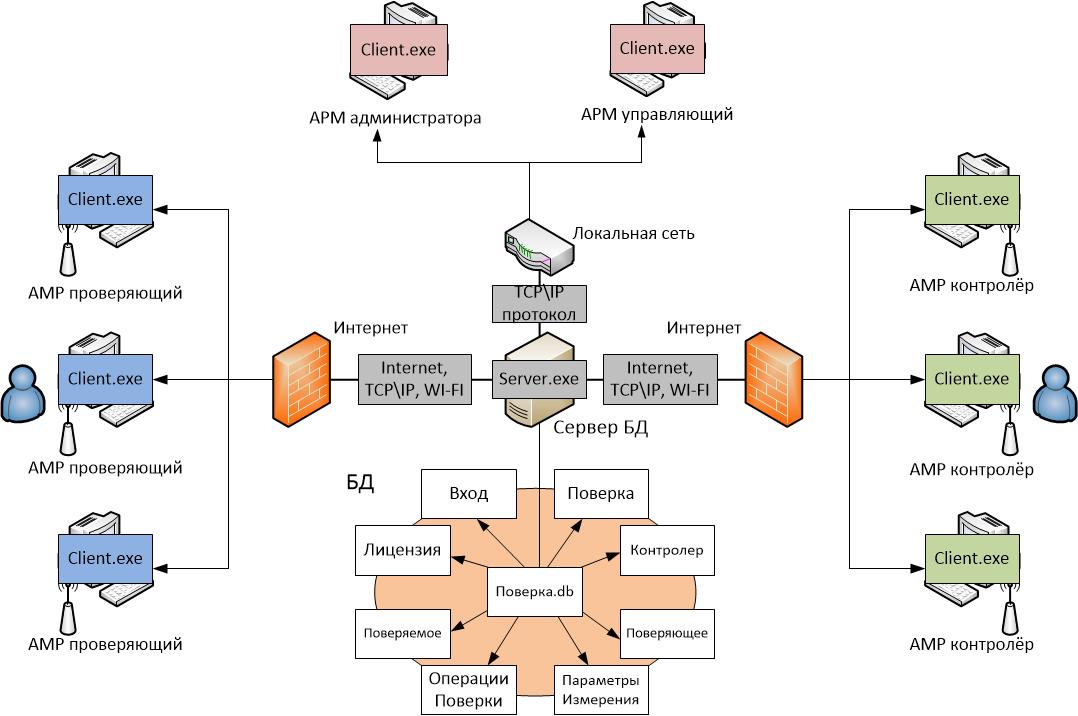


Рисунок 2.13 – Диаграмма развертывания

**2.6 Диаграмма компонентов**

Диаграмма компонентов - метод объектно-ориентированного проектирования, позволяющая устанавливать подчиненности между компонентами, определяющая архитектуру разрабатываемой системы.

Server.exe и Client.exe–это компоненты являющиеся исполнимыми модулями.

Поверка.drp и Server.drp, Поверка.db - компоненты-рабочие продукты.

Администратор, Контролер, Управляющий, Поверяющий,тКонтролер, Запрос за текущий год, Запрос за текущий месяц, Запрос на Итоговую таблицу, Запрос на количества использования прибора, Запрос не используемых приборов, Многотабличный Запрос,Неповерятые счетчики, Запрос\* на Даты поверок,Запрос на таблицу Даты поверок, СЗапрос на вид поверки, ТЛицензия, Обновление дат, Удаление счетчиков, Карточка счетчика,Форма Архив,Подчиненная форма Карточки контролера, Карточка контролера, Форма Регистрации новой поверки, Форма Регистрации нового счетчика,тОперацииПоверки, Главная кнопочная форма Контролёра, Главная кнопочная форма Управляющего, Главная кнопочная форма Проверяющего, Карточки Поверяющего Оборудования,Наклейки Поверки, Отчет за текущий год, Отчет за текущий месяц, Отчет о Поверки,Отчет Поверяющее Оборудование,тПараметрыИзмерения, Отчет Сертификата, Форма Вход, Форма Регистрации,тПоверка,тПоверяемоеОборудование,тПоверяющееОборудование,тВход- компоненты развертывания, которые обеспечивают непосредственное выполнение системой своих функций.

Компоненты связаны отношением зависимости.

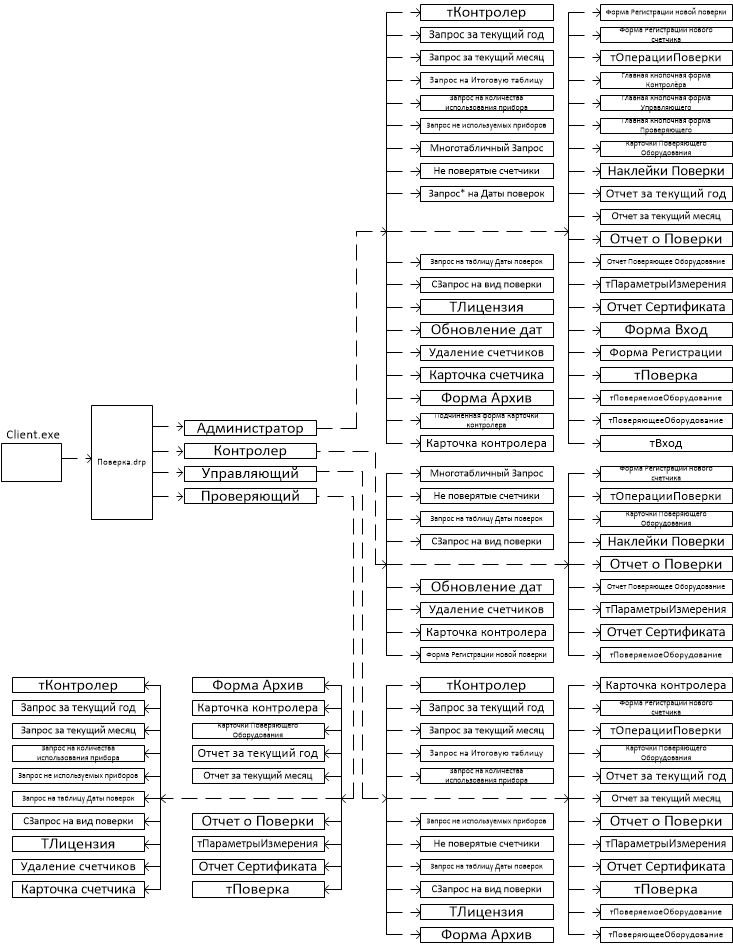


Рисунок 2.14 – Диаграмма компонентов для клиентской части

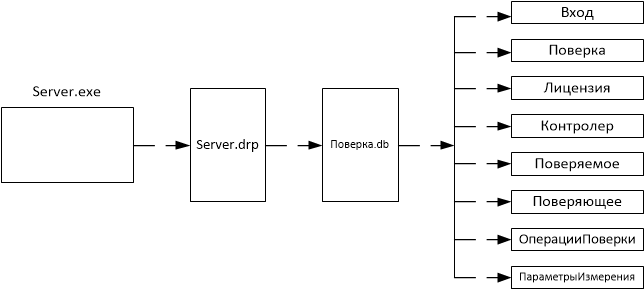


Рисунок 2.15 – Диаграмма компонентов для серверной части

**2.7 Диаграмма активности**

Диаграммы активности можно использовать для описания потоков событий в вариантах использования. С помощью текстового описания можно довольно детально рассказать о потоке событий, но в сложных и запутанных потоках с массой альтернативных ветвей будет трудно понять логику событий. Диаграммы активности в наглядной графической форме предоставляют ту же информацию, что и текстовое описание потока событий.



Рисунок 2.16 – Диаграмма активности для оформления прихода

При действии «Прием на поверку» наблюдается Электросчетчики.

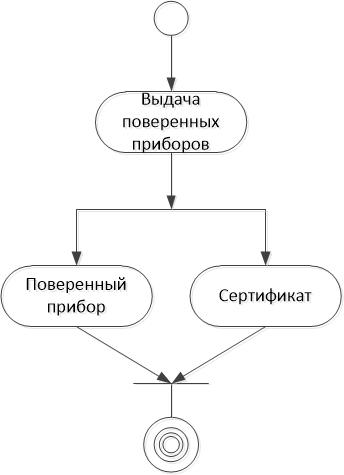


Рисунок 2.17 – Диаграмма активности для оформления расхода

При действии «Выдача поверенных приборов» наблюдается разделение на Поверенный прибор, Сертификат. В дальнейшем происходит слияние потоков управления.

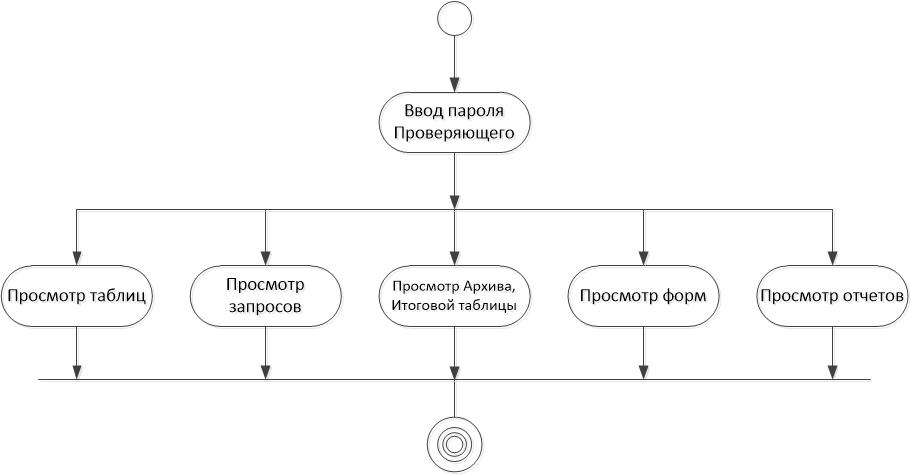


Рисунок 2.18 – Диаграмма активности для Поверяющего

Для Поверяющего после авторизации с помощью пароля наблюдается разделение деятельности на Просмотр таблиц, Просмотр запросов, Просмотр Архива и Итоговой таблицы,а также просмотр форм и отчетов.

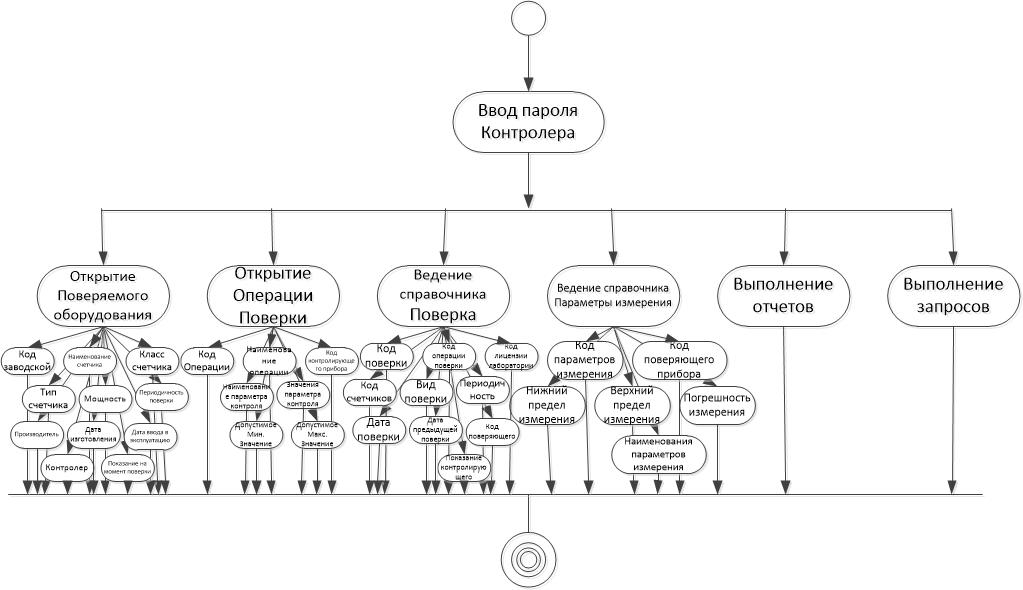


Рисунок 2.19 – Диаграмма активности для Контролера

Для Контролера после авторизации с помощью пароля наблюдается разделение деятельности на Открытие Поверяемого оборудования, Открытие Операции Поверки, Ведение справочника Поверка, Ведение справочника Параметры измерения, а также выполнения запросов, отчетов.

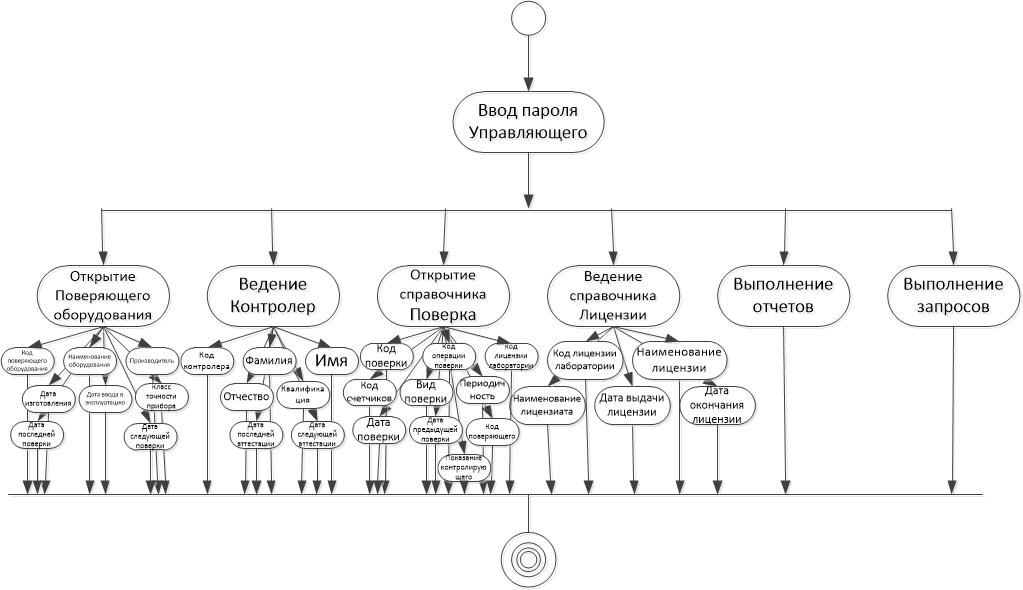


Рисунок 2.20 – Диаграмма активности для Управляющего

Для Управляющего после авторизации с помощью пароля наблюдается разделение деятельности на Открытие Поверяющего оборудования, Ведение Контролер, Открытие справочника Поверка, Ведение справочника Лицензии, а также выполнения запросов, отчетов.

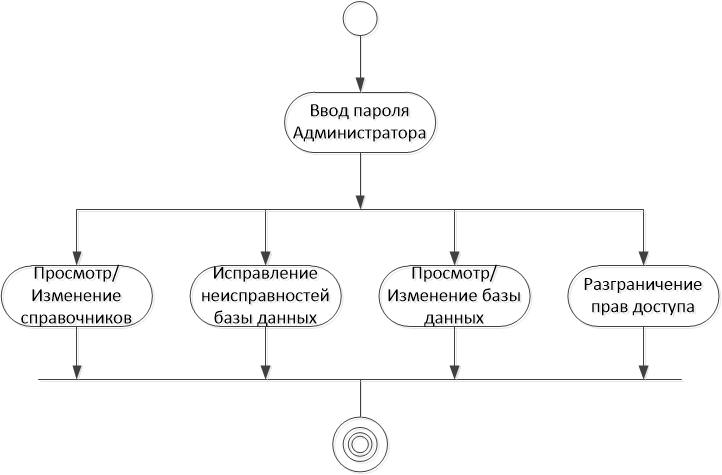


Рисунок 2.21 – Диаграмма активности для администратора

Для Администратора после авторизации с помощью пароля наблюдается разделение деятельности на Просмотр/Изменение справочников, Исправление неисправностей базы данных, Просмотр/Изменение базы данных, Разграничения прав доступа.

**2.8 Диаграмма классов**

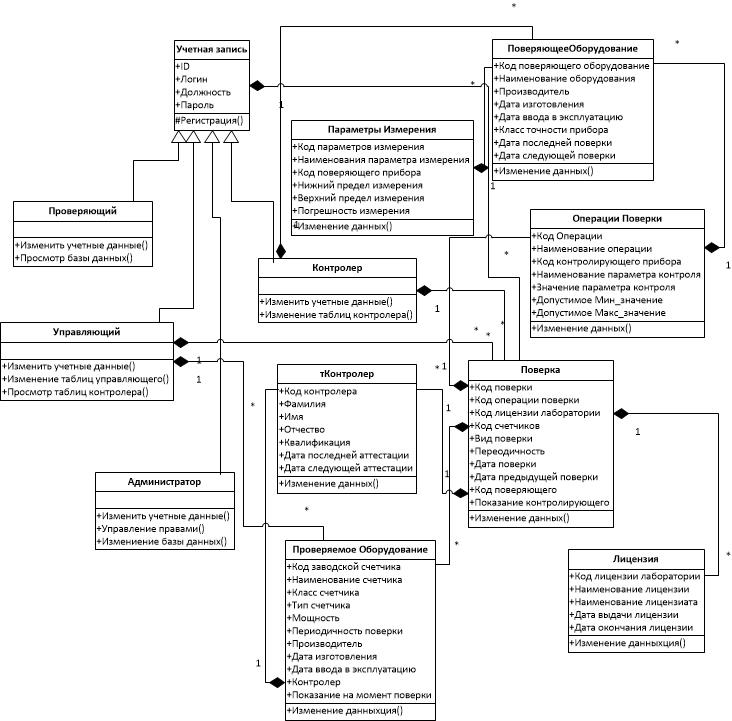


Рисунок 2.22 – Диаграмма классов

На диаграмме представлены классы Проверяющий, Администратор, Управляющий, Контролер, Лицензия, Поверяемое Оборудование, Поверяющее Оборудование, Поверка, тКонтролер, Операции Поверки, Параметр Измерения, Учетная запись. В каждом классе указаны его значения, а также действия, возможные для данного класса.

Классы Проверяющий, Администратор, Управляющий, Контролер связаны отношением обобщения с классом Учетная запись, так как входят в систему, используя логин и пароль.

Все остальные классы связаны отношением композиция.

**3 РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ**

**3.1Основные сведенья оинформационной системе**

Информационная система (ИС) — [система](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0), предназначенная для хранения, поиска и обработки [информации](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F), и соответствующие организационные ресурсы (человеческие, технические, финансовые и т. д.), которые обеспечивают и распространяют информацию.

К компонентаминформационной системы относится:

* База данных
* Концептуальная схема
* Информационный процессор, образующие вместе систему хранения и манипулирования данными

База данных – это информационная система, требующая создания в памяти ЭВМ динамически обновляемой модели внешнего мира с использованием единого хранилища.

Преимущества централизованного управления данными в базе данных:

* Сокращение избыточности хранимых данных благодаря однократному хранению каждого сообщения в базе данных;
* Совместное использование хранимых данных всеми пользователями ИС;
* Стандартизацию представления данных, упрощающую проблемы эксплуатации БД и обмена данными между ИС;
* Обеспечение процедур проверки достоверности информации обеспечение процедур ограничения доступа к данным;
* Совмещение требований к использованию БД со стороны различных пользователей ИС.

Концептуальная схема – это (от слова concept - понятие) представляет собой описание структуры всех единиц информации, хранящихся в БД.

Под структурой понимается вхождение одних единиц информации в состав других единиц информации.

Информационный процессор – это механизм, который в ответ на получение команды выполняет операции с БД и концептуальной схемой. Информационный процессор состоит из вычислительной системы, системы управления базой данных – СУБД, а также включает администратора базы данных.

Системой управления базой данных называется комплекс программ, обеспечивающий централизованное хранение, накопление, модификацию и выдачу данных, входящих в БД.

Обычно современная СУБД содержит следующие компоненты:

* Ядро, которое отвечает за управление данными во внешней и оперативной памяти и журнализацию;
* Процессор языка базы данных, обеспечивающий оптимизацию запросов на извлечение и изменение данных и создание, как правило, машинно-независимого исполняемого внутреннего кода;
* Подсистему поддержки времени исполнения, которая интерпретирует программы манипуляции данными, создающие пользовательский интерфейс с СУБД;
* Сервисные программы (внешние утилиты), обеспечивающие ряд дополнительных возможностей по обслуживанию информационной системы.

Основным функциям СУБД относится:

* Управление данными во внешней памяти (на дисках);
* Управление данными в оперативной памяти;
* Журнализация изменений;
* Восстановление базы данных после сбоев (механизм транзакций);
* Поддержание языков БД (язык определения данных, язык манипулирования данными).

Для реализации автоматизированной системы поверки приборов учета расхода электроэнергии выбрана СУБД Access, которая является наиболее доступной, имеет развитый инструментарий разработки баз данных, по своим возможностям является достаточной, но не избыточной, СУБД для решения поставленной задачи в данной работе.

**3.2 Архитектура MicrosoftAccess**

MicrosoftAccess обладает всеми чертами классической СУБД. Access – это не только мощная, гибкая и простая в использовании СУБД, но и система для разработки приложений баз данных. К числу наиболее мощных средств Accessотносятся средства разработки объектов – мастера, которые можно использовать для создания таблиц, запросов, различных типов форм и отчетов. В MicrosoftAccess включены мастера, помогающие производить анализ структуры данных, импортировать электронные таблицы и текстовые данные, повышать быстродействие приложения, создавать и настраивать одно из более, чем двадцати типов приложений с использованием встроенных шаблонов. Чтобы полностью автоматизировать работу приложения, можно использовать макросы для связывания данных с формами и отчетами. Большинство приложений можно создать, не написав ни единой строки программного кода. Однако при необходимости построения действительно сложного приложения можно использовать язык программирования – VisualBasic для приложений.

MicrosoftAccess называет объектами все, что может иметь имя (в смысле Access). В базе данных Access основными объектами являются таблицы, запросы, формы, отчеты, макросы и модули. В других СУБД, как правило, термин база данных обычно относится только к файлам, в которых хранятся данные. В MicrosoftAccess база данных включает в себя все объекты, связанные с хранимыми данными, в том числе и те, которые определяются для автоматизации работы с ними. Ниже приведен список основных объектов базы данных Access.

Таблица. Объект, который определяется и используется для хранения данных. Каждая таблица включает информацию об объекте определенного типа, например, о клиентах. Таблица содержит поля (столбцы), в которых хранятся различного рода данные и записи (которые называются также строками). В записи собрана вся информация о некотором объекте. Для каждой таблицы можно определить первичный ключ (одно или несколько полей, содержащих уникальные для каждой записи значения) и один или несколько индексов, помогающих ускорить доступ к данным.

Запрос. Объект, который позволяет пользователю получить нужные данные из одной или нескольких таблиц. Для создания запроса можно использовать бланк QBE (запрос по образцу) или инструкции SQL (структурированный язык запросов). Можно создать запросы на выборку, обновление, удаление или добавление данных. С помощью запросов можно также создавать новые таблицы, используя данные из одной или нескольких существующих таблиц.

Форма. Объект, предназначенный в основном для ввода данных, отображения их на экране или управления работой приложения. Формы используются для того, чтобы реализовать требования пользователя к представлению данных из запросов или таблиц. Формы можно также распечатать. С помощью формы можно в ответ на некоторое событие, например изменение значения определенных данных, запустить макрос или процедуру VBA.

Отчет. Объект, предназначенный для создания документа, который впоследствии может быть распечатан или включен в документ другого приложения.

Макрос. Объект, представляющий собой структурированное описание одного или нескольких действий, которые должен выполнить Access в ответ на определенное событие. Например, можно определить макрос, который в ответ на выбор некоторого элемента в основной форме открывает другую форму. С помощью другого макроса можно осуществлять проверку значения некоторого поля при изменении его содержимого. В макрос можно включить дополнительные условия для выполнения или невыполнения тех или иных указанных в нем действий. Из одного макроса можно также запустить другой макрос или процедуру VBA.

Модуль.Объект, содержащий программы, написанные на языке VisualBasic для приложений. Модули могут быть независимыми объектами, содержащими функции, вызываемые из любого места приложения, но они могут быть и непосредственно «привязаны» к отдельным формам или отчетам для реакции на те или иные происходящие в них изменения.

Концептуальные взаимосвязи объектов Access показаны на рисунке3.1. В таблицах хранятся данные, которые можно извлекать с помощью запросов. Используя формы, можно выводить данные на экран или изменять их. Формы и отчеты получают данные как непосредственно из таблиц, так и через запросы. Для выполнения нужных вычислений и преобразования данных запросы могут использовать встроенные функции или функции, созданные с помощью VisualBasic для приложений. События, происходящие в формах или отчетах, могут запускать макросы или процедуры VBA. Событие – любое изменение состояния объекта MicrosoftAccess. Например, событием является открытие формы, закрытие формы, ввод новой строки в форму, изменение содержимого текущей записи или элемента управления (объекта формы или отчета, который может содержать данные). Для обработки события можно создать макрос или процедуру VisualBasic для приложений.

С помощью макросов и модулей можно изменять ход выполнения приложения; открывать, фильтровать и изменять данные в формах и отчетах; выполнять запросы и создавать новые таблицы. Используя VisualBasic для приложений, можно создать, модифицировать и удалить любой объект Access, обрабатывать данные по строкам или по столбцам, а также каким-либо другим способом. Можно также вызывать процедуры из библиотек динамической компоновки (DLL) MicrosoftWindows, чтобы использовать в своем приложении не только встроенные в Access функции, но и возможности Windows.

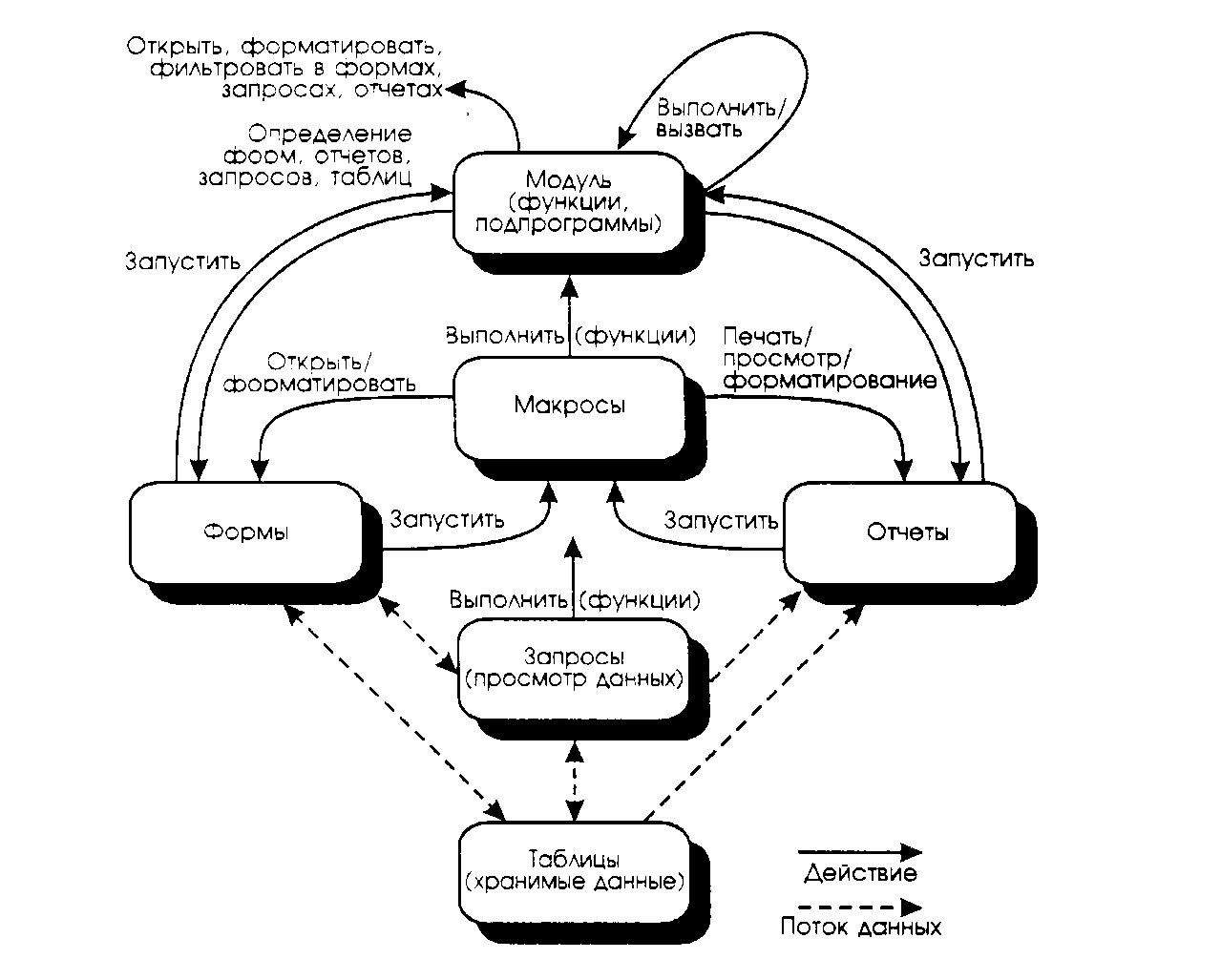


Рисунок3.1– Взаимосвязи основных объектов в MicrosoftAccess

**3.3Структурная схема таблиц**

В таблице “тКонтролер” фиксируются данные контролера, его квалификация и даты последней и следующей аттестации. Ключевым полем является “Код контролера”, представленная типом данных “Счетчик”. Для полей: “Фамилия”, “Имя”, “Отчество”, “Квалификация” - выбран тип данных “Короткий текст (256 символов)”. Для полей: “Дата последней аттестации”, “Дата следующей аттестации” - выбран тип данных “Дата и время”. Структура таблицы “тКонтролер” имеет следующий вид:

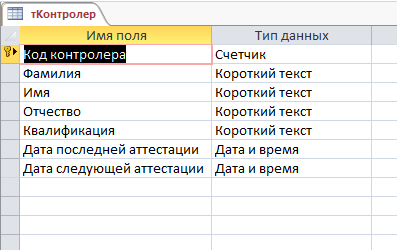


Рисунок 3.2 – Структура таблицы тКонтролер

В таблице “тЛицензия” фиксируются данные лаборатории поверки, её данные о лицензии и даты действия лицензии. Ключевым полем является “Код лицензии лаборатории”, представленная типом данных “Счетчик”. Для полей: “Наименование лицензии”, “Наименование лицензиата” - выбран тип данных “Короткий текст (256 символов)”. Для полей: “Дата выдачи лицензии”, “Дата окончания лицензии” - выбран тип данных “Дата и время”. Структура таблицы “тЛицензия” имеет следующий вид:

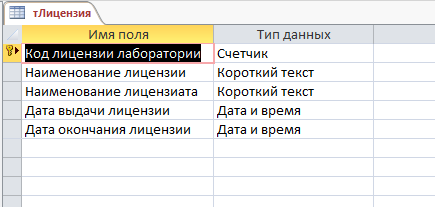


Рисунок 3.3 – Структура таблицы тЛицензия

В таблице характеристик приборов поверки “тПоверяющееОборудование” фиксируются данные поверяющего оборудования, его точность, название, производитель и даты последней и следующей поверки, а также даты изготовления и ввода в эксплуатацию. Ключевым полем является “Код поверяющего оборудования”, представленная типом данных “Счетчик”. Для полей: “Наименование оборудования”, “Производитель”, “Класс точности прибора” - выбран тип данных “Короткий текст (256 символов)”. Для полей: “Дата изготовления”, “Дата ввода в эксплуатацию”, “Дата последней поверки”, “Дата следующей поверки” - выбран тип данных “Дата и время”. Структура таблицы “тПоверяющееОборудование” имеет следующий вид:

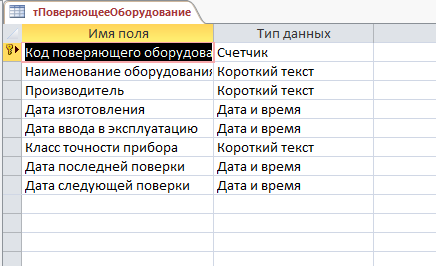


Рисунок 3.4– Структура таблицы тПоверяющееОборудование

В таблице характеристик приборов “тПараметрыИзмерения” фиксируются параметры измерения поверяющего оборудования, наименование прибора, его параметры измерения и диапазон измерения, а также погрешность прибора измерения. Ключевым полем является “Код параметров измерения”, представленная типом данных “Счетчик”. Для поля “Наименование параметра измерения” выбран тип данных “Короткий текст (256 символов)”. Для полей: “Нижний предел измерения”, “Верхний предел измерения”, “Погрешность измерения” - выбран тип данных “Числовой”. Поле “Код поверяющего прибора” является полем подстановки ключевого поля из таблицы “тПоверяющееОборудование”. Структура таблицы “тПараметрыИзмерения” имеет следующий вид:

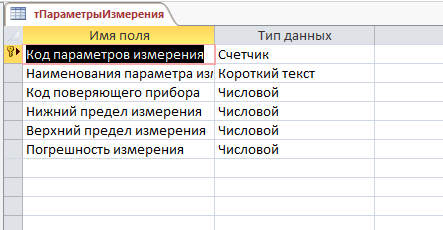


Рисунок 3.5– Структура таблицы тПараметрыИзмерения

В таблице измерений “тОперацииПоверки” фиксируются измерения, сделанные контролером, прибор, производящий измерения, и его параметр измерения, название операции и допустимый диапазон. Ключевым полем является “Код Операции”, представленная типом данных “Счетчик”. Для полей: “Наименование операции”, “Наименование параметра контроля” - выбран тип данных “Короткий текст (256 символов)”. Для полей: “Значение параметра контроля”, “Допустимое Мин\_значение”, “Допустимое Макс\_значение” - выбран тип данных “Числовой”. Поле “Код контролирующего прибора” является полем подстановки ключевого поля из таблицы “тПоверяющееОборудование”. Структура таблицы “тОперацииПоверки” имеет следующий вид:

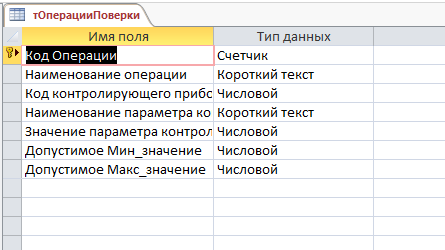


Рисунок 3.6 – Структура таблицы тОперацииПоверки

В таблице характеристик приборов “тПоверяемоеОборудование” фиксируются характеристики счетчика, периодичность поверки, имя контролера, результат измерения (контроля) и даты изготовления и ввода в эксплуатацию. Ключевым полем является “Код заводской счетчика”, представленная типом данных “Счетчик”. Для полей: “Наименование счетчика”, “Класс счетчика”, “Мощность”, “Периодичность поверки”, “Производитель”, “Показания на момент поверки” - выбран тип данных “Короткий текст (256 символов)”. Для полей: “Дата изготовления”, “Дата ввода в эксплуатацию” - выбран тип данных “Дата и время”. Поле “Контролер” является полем подстановки ключевого поля из таблицы “тКонтролер”. Поле “Тип счетчика” является полем подстановки из раннее созданного списка выбора. Структура таблицы “тПоверяемоеОборудование” имеет следующий вид:

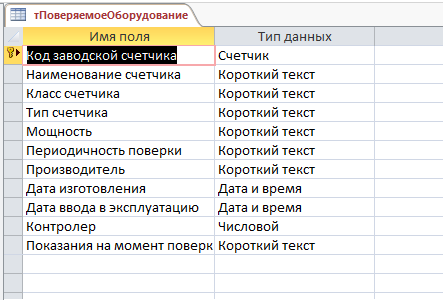


Рисунок 3.7 – Структура таблицы тПоверяемоеОборудование

В главной таблице “тПоверка” фиксируются все данные о поверки: лицензия, на основе которой проходит поверка, наименование операции поверки, имя контролера, производящего поверку, и показание его прибора, название и характеристики счетчика, вид поверки и даты последней и следующей поверки электросчетчика. Ключевыми полями является “Код поверки” и “Код операции поверки”, представленные типами данных “Счетчик”. Для полей: “Периодичность”, “Показания контролирующего” - выбран тип данных “Числовой”. Для полей: “Дата поверки”, “Дата предыдущей поверки” - выбран тип данных “Дата и время”. Ключевое поле “Код операции поверки” является полем подстановки ключевого поля из таблицы “тОперацииПоверки”. Поле “Код лицензии лаборатории” является полем подстановки соответствующего ключевого поля из таблицы “тЛицензия”. Поле “Код счетчиков” является полем подстановки ключевого поля из таблицы “тПоверяемоеОборудование”. Поле “Код поверяющего” является полем подстановки ключевого поля из таблицы “тКонтролер”. Поле “Вид поверки” является полем подстановки из раннее созданного списка выбора. Структура таблицы “тПоверка” имеет следующий вид:

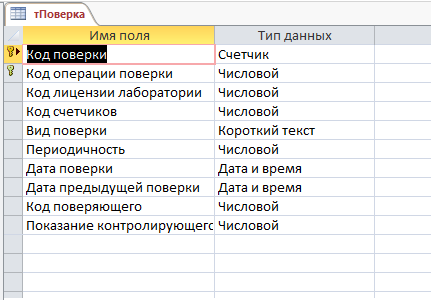


Рисунок 3.8– Структура таблицы тПоверка

В таблице данных пользователей БД “тВход” фиксируются данные пользователя такие как уникальное имя пользователя “Логин”, должность, занимаемая в учреждение, а также уникальный пароль защищающий запись от посторонних, и уникальный индикатор присваивающий системой. Ключевым полем является “ИД”, представленная типом данных “Счетчик”. Для полей: “Логин” и “Пароль” - выбран тип данных “Короткий текст (256 символов)”. Поле “Должность” является полем подстановки из раннее созданного списка выбора. Структура таблицы “тВход” имеет следующий вид:

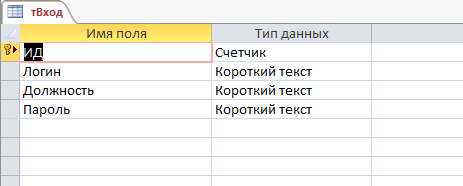


Рисунок 3.9 – Структура таблицы тВход

**3.4 Структурная схема базы данных**

Структура БД представляется схемой взаимосвязей таблиц БД. В этих связях отображаются виды связей(‘один к одному’, ‘один ко многим’), характер связей(правая, левая, внутренняя связи). Структура базы данных представлена на Рисунке 3.10:

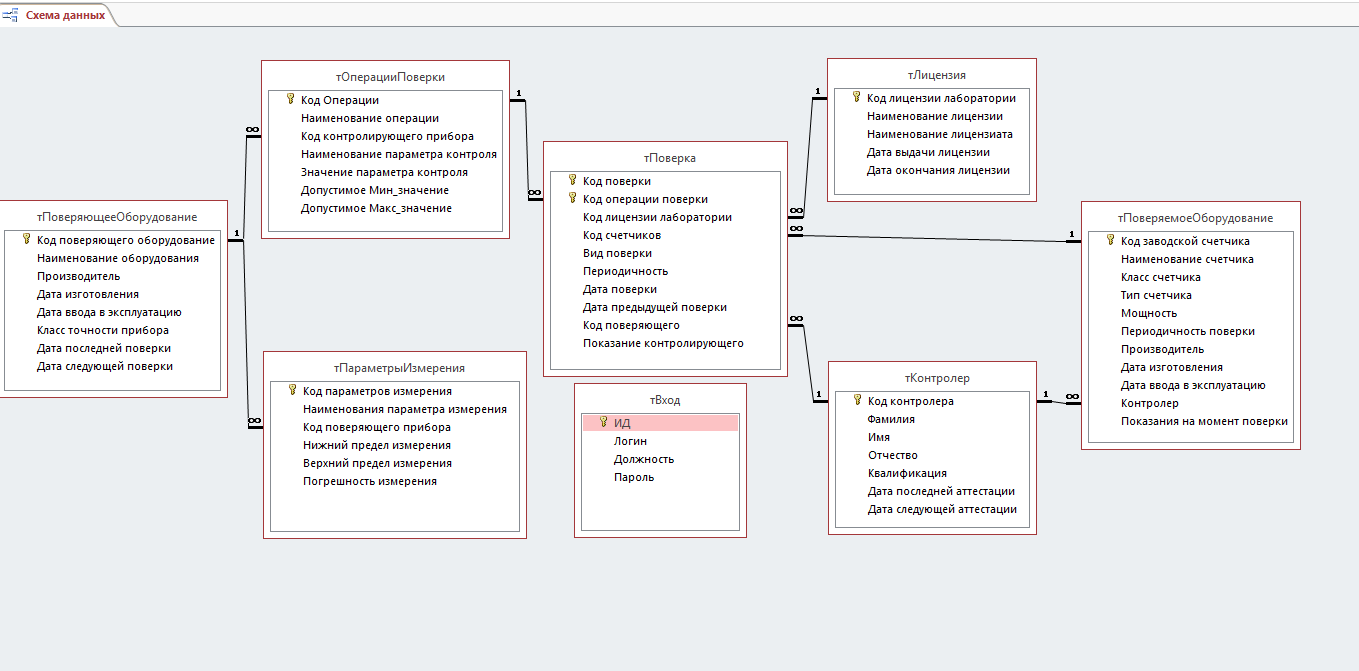


Рисунок 3.10 – Схема данных базы данных

В этой схеме отображены взаимосвязи таблиц БД.

Главной таблицей, объединяющей все остальные, является таблица “тПоверка”, в которой фиксируются “Код поверки”, “Код операции поверки”, “Код лицензии лаборатории”, “Код счетчиков”, “Вид поверки”, “Периодичность”, “Дата поверки”, “Дата предыдущей поверки”, “Код поверяющего”, “Показания контролирующего”. Через поле “Код счетчиков” в таблицу подставляются названия поверяемых приборов из таблицы “тПоверяемоеОборудование” через поле подстановки “Код заводской счетчиков”.

В свою очередь в таблицу “тПоверяемоеОборудование” подставляются данные проверяющих из таблицы “тКонтролер” такие как: “Код контролера”, “Фамилия”, “Имя”, “Отчество”, “Квалификация”, “Дата последней аттестации”, “Дата следующей аттестации” - через поле подстановки “Контролер”. Эти же поля таблицы “тКонтролер” подставляются в таблицу “тПоверка”. Также в таблицу “тПоверка” подставляются данные о лицензиях из таблицы “тЛицензия” через поле “Код лицензии лаборатории” и включает такие поля как: “Код лицензии лаборатории”, “Наименование лицензии”, “Наименование лицензиата”, “Дата выдачи лицензии”, “Дата окончания лицензии”.

Результаты поверки счетчика, наименование измерительного прибора и их параметров фиксируются в таблицах “тОперацииПоверки”, “тПоверяемоеОборудование”, “тПараметрыИзмерения”.

Схема БД отражает вид связей между таблицами: связь “тПоверяемоеОборудование” и “тПоверка” определена как “ один-ко-многим”, что означает, что в системе предусмотрена поверка множества поверяемых приборов; в свою очередь связь “ один-ко-многим” между таблицами поверки “тПоверка” и “тКонтролер” показывает, что предусмотрено использование в поверке множество контролеров, которые, как и поверяемые приборы связаны отношением “многие-к-одному” с таблицей характеристик приборов тПоверяемоеОборудование”, что определяет возможность описание множества данных для этих приборов.

Таблица “тВход” является обособленной таблицей в базе данных и взаимодействует только с формами “Вход”, “Регистрация” и “Главная кнопочная форма Администратор”. Что обеспечивает защиту этих данных от других пользователей, в виду отсутствия прямого доступа к данным.

В схеме БД отображена возможность использование в поверке множеством операций поверок, множеством поверяющего оборудования, которые имеют множества параметров (отношения “один-ко-многим” и “многие-к-одному”). Все поля связи на стороне “много” являются полями подстановки соответствующих ключевых полей на стороне один. Например, поле “Код Операции” таблицы “тПараметрыИзмерения” является полем подстановки “Код операции поверки” таблицы “тПоверка”.

**3.5Структурная схема запросов**

Запрос“Запрос за текущий год” фильтрует информацию о электросчетчиках, которые надо поверить в текущем году. Структура запроса “Запрос за текущий год” представлена следующей схемой:

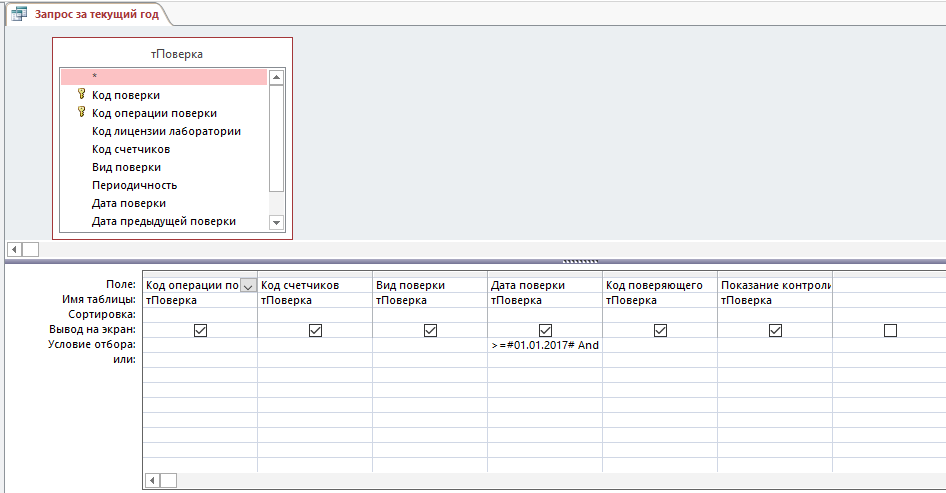


Рисунок 3.11– Схема запроса “Запрос за текущий год”

Представление запроса “Запрос за текущий год” в форме SQL имеет следующий вид:

SELECT тПоверка.[Код операции поверки], тПоверка.[Код счетчиков], тПоверка.[Вид поверки], тПоверка.[Дата поверки], тПоверка.[Код поверяющего], тПоверка.[Показание контролирующего]

FROM тПоверка

WHERE (((тПоверка.[Дата поверки])>=#1/1/2017# And (тПоверка.[Дата поверки])<=#12/31/2017#));

Здесь перечислены наименования полей, включаемые в запрос, указывается таблица, из которой они извлекаются, а также описываются условия выборки, участвующий в формировании запроса. Запрос WHERE показывает начало условия в SQL коде.

Запрос“Запрос за текущий месяц” фильтрует информацию о электросчетчиках, которые надо поверить в текущем месяце. Структура запроса “Запрос за текущий месяц” представлена следующей схемой:

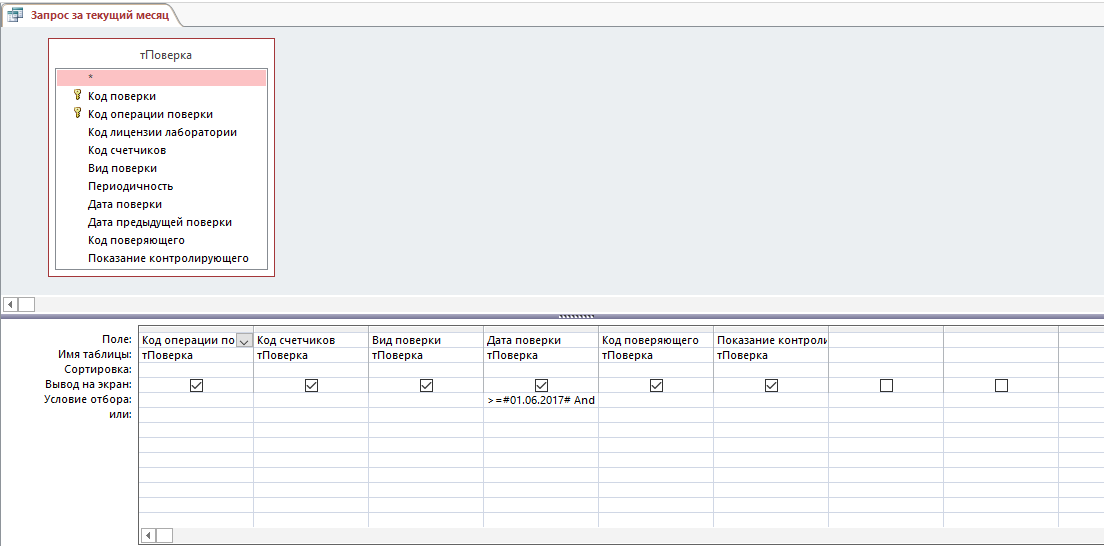


Рисунок 3.12– Схема запроса “Запрос за текущий месяц”

Представление запроса “Запрос за текущий месяц” в форме SQL имеет следующий вид:

SELECT тПоверка.[Код операции поверки], тПоверка.[Код счетчиков], тПоверка.[Вид поверки], тПоверка.[Дата поверки], тПоверка.[Код поверяющего], тПоверка.[Показание контролирующего]

FROM тПоверка

WHERE (((тПоверка.[Дата поверки])>=#6/1/2017# And (тПоверка.[Дата поверки])<=#6/30/2017#));

Здесь перечислены наименования полей, включаемые в запрос, указывается таблица, из которой они извлекаются, а также описываются условия выборки, участвующий в формировании запроса. Запрос WHERE показывает начало условия в SQL коде.

Запрос“Запрос за Итоговую таблицу”выводит полную информацию о электросчетчиках. Структура запроса “Запрос за Итоговую таблицу” представлена следующими схемами:

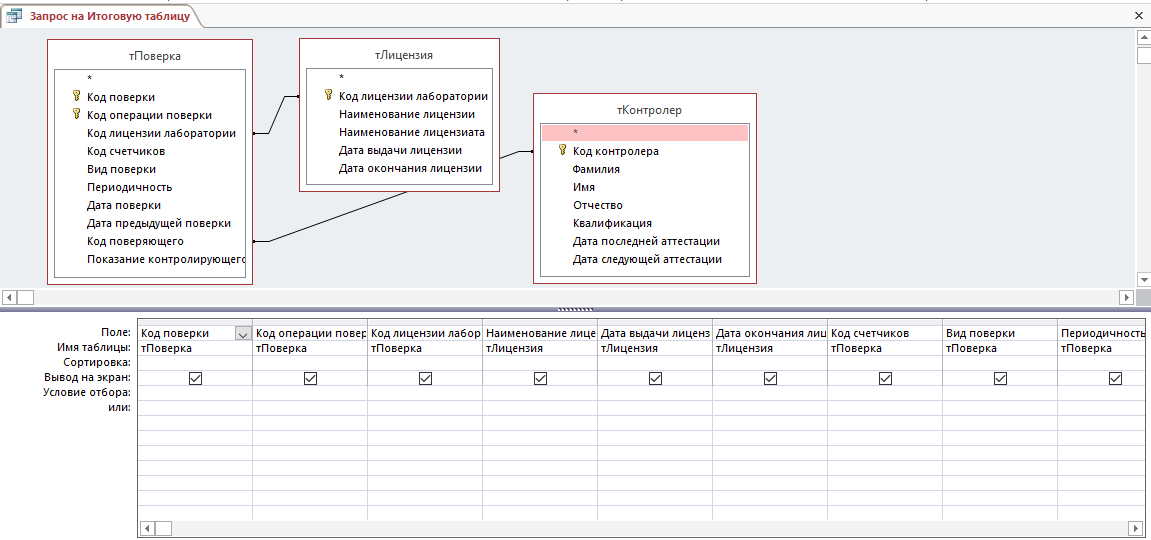


Рисунок 3.13а– Схема запроса “Запрос за Итоговую таблицу”

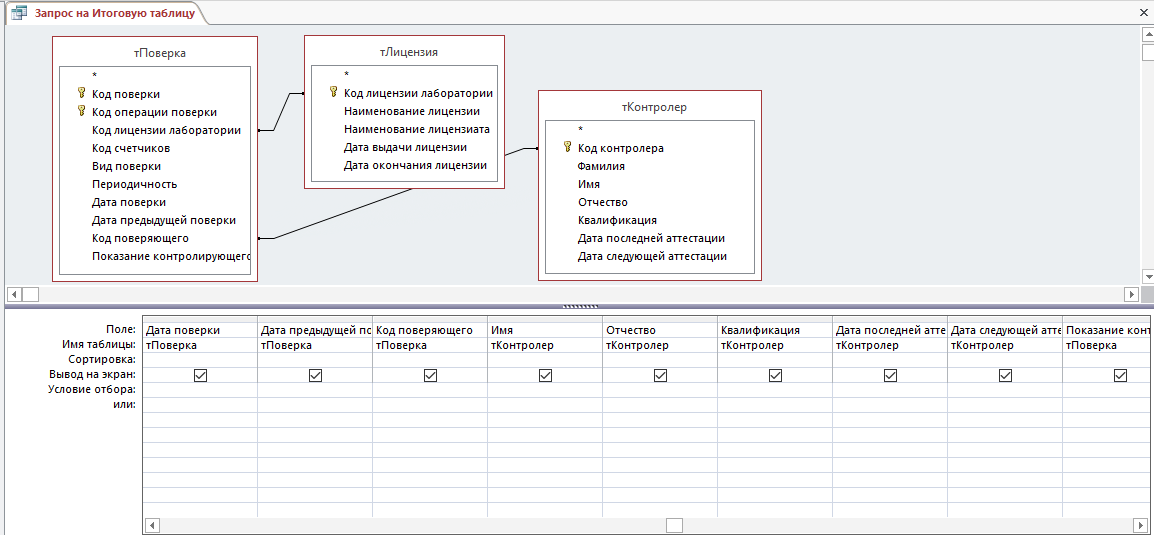


Рисунок 3.13б– Схема запроса “Запрос за Итоговую таблицу”

Представление запроса “Запрос за Итоговую таблицу” в форме SQL имеет следующий вид:

SELECT тПоверка.[Код поверки], тПоверка.[Код операции поверки], тПоверка.[Код лицензии лаборатории], тЛицензия.[Наименование лицензиата], тЛицензия.[Дата выдачи лицензии], тЛицензия.[Дата окончания лицензии], тПоверка.[Код счетчиков], тПоверка.[Вид поверки], тПоверка.Периодичность, тПоверка.[Дата поверки], тПоверка.[Дата предыдущей поверки], тПоверка.[Код поверяющего], тКонтролер.Имя, тКонтролер.Отчество, тКонтролер.Квалификация, тКонтролер.[Дата последней аттестации], тКонтролер.[Дата следующей аттестации], тПоверка.[Показание контролирующего]

FROM тКонтролер INNER JOIN (тЛицензия INNER JOIN тПоверка ON тЛицензия.[Код лицензии лаборатории] = тПоверка.[Код лицензии лаборатории]) ON тКонтролер.[Код контролера] = тПоверка.[Код поверяющего];

Здесь перечислены наименования полей, включаемые в запрос, указываются таблицы, из которой они извлекаются, а также описываются условия выборки, участвующий в формировании запроса. Вид связи INNERJOIN показывает, что в запросе объединяются только те записи таблиц “тПоверка”, “тЛицензия” и “тКонтролер”, в которых связанные поля обеих таблиц совпадают.

Запрос“Запрос на количество использования прибора” фильтрует информацию о электросчетчиках, которые хоть раз использовались, и выводит статистику их использований. Структура запроса “Запрос на количество использования прибора” представлена следующей схемой:

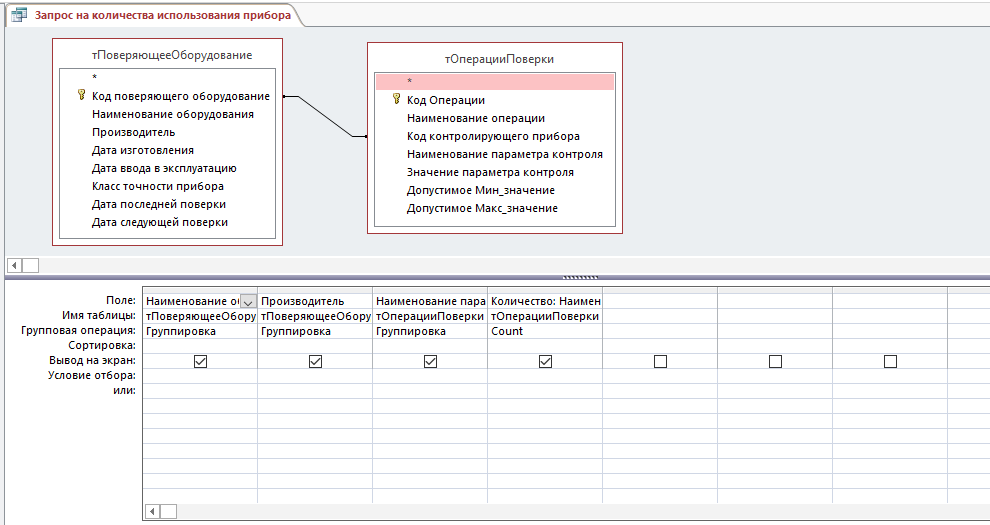


Рисунок 3.14– Схема запроса “Запрос на количество использования прибора”

Представление запроса “Запрос на количество использования прибора” в форме SQL имеет следующий вид:

SELECT тПоверяющееОборудование.[Наименование оборудования], тПоверяющееОборудование.Производитель, тОперацииПоверки.[Наименование параметра контроля], Count(тОперацииПоверки.[Наименование параметра контроля]) AS Количество

FROM тПоверяющееОборудование INNER JOIN тОперацииПоверки ON тПоверяющееОборудование.[Код поверяющего оборудование] = тОперацииПоверки.[Код контролирующего прибора]

GROUP BY тПоверяющееОборудование.[Наименование оборудования], тПоверяющееОборудование.Производитель, тОперацииПоверки.[Наименование параметра контроля];

Здесь перечислены наименования полей, включаемые в запрос, указываются таблицы, из которой они извлекаются, а также описываются условия выборки, участвующий в формировании запроса. Запрос GROUPBYпоказывает, что в запросе группируются записи.

Запрос“Запрос не используемых приборов” фильтрует информацию о приборах поверки, которые ни разу не использовались. Структура запроса “Запрос не используемых приборов” представлена следующей схемой:

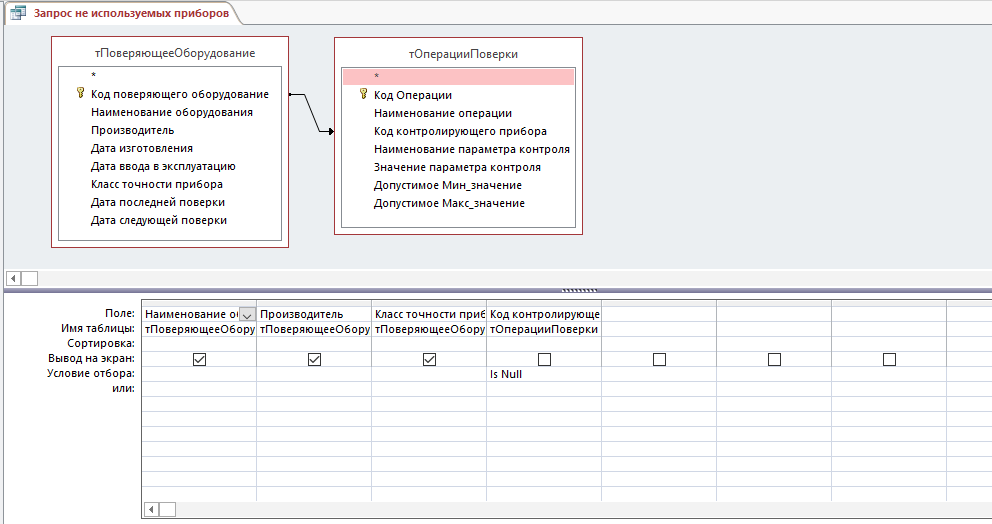


Рисунок 3.15– Схема запроса “Запрос не используемых приборов”

Представление запроса “Запрос не используемых приборов” в форме SQL имеет следующий вид:

SELECT тПоверяющееОборудование.[Наименование оборудования], тПоверяющееОборудование.Производитель, тПоверяющееОборудование.[Класс точности прибора]

FROM тПоверяющееОборудование LEFT JOIN тОперацииПоверки ON тПоверяющееОборудование.[Код поверяющего оборудование] = тОперацииПоверки.[Код контролирующего прибора]

WHERE (((тОперацииПоверки.[Код контролирующего прибора]) IsNull));

Здесь перечислены наименования полей, включаемые в запрос, указываются таблицы, из которой они извлекаются, а также описываются условия выборки, участвующий в формировании запроса. Запрос WHERE показывает начало условия в SQL коде, а условие IsNullпоказывает то, что ищутся записи, которые не используются в двух таблицах одновременно.

Запрос “Запрос на таблицу Даты поверок” создает наглядную таблицу информации о датах поверки и данных контролеров, которая создается на базе подчинённого запроса. Структура запроса “Запрос на таблицу Даты поверок” представлена следующей схемой:

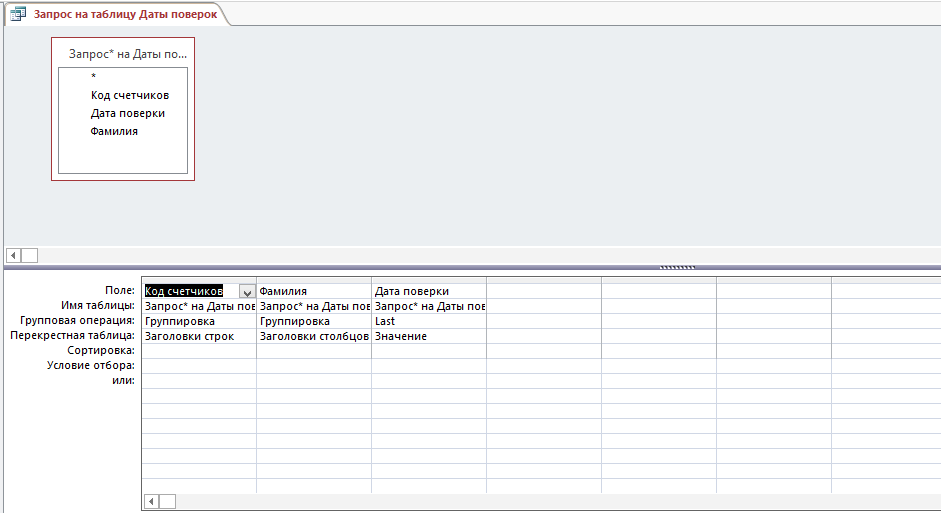


Рисунок 3.16– Схема запроса “Запрос на таблицу Даты поверок”

Представление запроса “Запрос на таблицу Даты поверок” в форме SQL имеет следующий вид:

TRANSFORM Last([Запрос\* на Даты поверок].[Дата поверки]) AS [Last-Дата поверки]

SELECT [Запрос\* на Даты поверок].[Код счетчиков]

FROM [Запрос\* на Даты поверок]

GROUP BY [Запрос\* на Даты поверок].[Код счетчиков]

PIVOT [Запрос\* на Даты поверок].Фамилия;

Здесь создана таблица с помощью команды TRANSFORM. А наименования его полей, формируются на основе полей подчиненного запроса “Запрос\* на Даты поверок”. И на основе этого создается наглядная таблица.

Структура запроса “Запрос\* на Даты поверок” представлена следующей схемой:

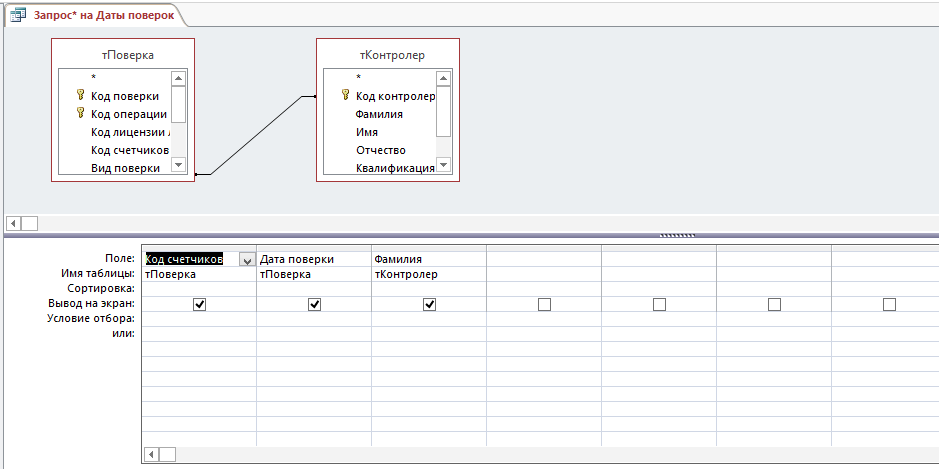


Рисунок 3.17– Схема запроса “Запрос\* на Даты поверок”

Представление запроса “Запрос\* на Даты поверок” в форме SQL имеет следующий вид:

SELECT тПоверка.[Код счетчиков], тПоверка.[Дата поверки], тКонтролер.Фамилия

FROM тКонтролер INNER JOIN тПоверка ON тКонтролер.[Код контролера] = тПоверка.[Код поверяющего];

Здесь перечисленные наименование полей включаемые в запрос, указываются таблицы, из которых они извлекаются, а также описываются взаимосвязи таблиц, участвующие в формирование запроса. Вид связи INNERJOIN показывает, что в запросе объединяются только те записи таблиц “тПоверка” и “тКонтролер”, в которых связанные поля обеих таблиц совпадают.

Запрос“Многотабличный Запрос” фильтрует информацию о электросчетчиках и кто поверял их. Структура запроса “Многотабличный Запрос” представлена следующей схемой:

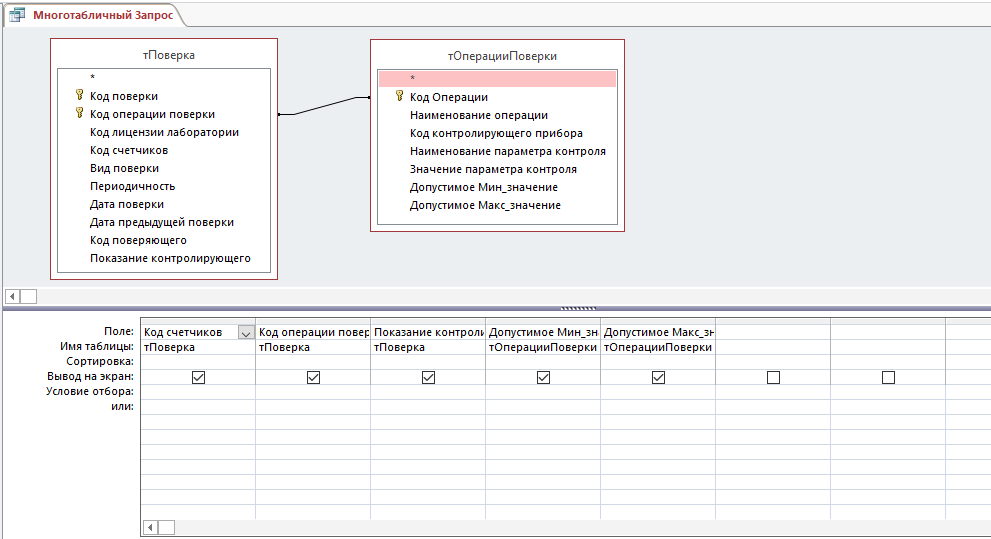


Рисунок 3.18– Схема запроса “Многотабличный Запрос”

Представление запроса “Многотабличный Запрос” в форме SQL имеет следующий вид:

SELECT тПоверка.[Код счетчиков], тПоверка.[Код операции поверки], тПоверка.[Показание контролирующего], тОперацииПоверки.[Допустимое Мин\_значение], тОперацииПоверки.[Допустимое Макс\_значение]

FROM тОперацииПоверки INNER JOIN тПоверка ON тОперацииПоверки.[Код Операции] = тПоверка.[Код операции поверки];

Здесь перечислены наименования полей, включаемые в запрос, указываются таблицы, из которой они извлекаются, а также описываются условия выборки, участвующий в формировании запроса. Вид связи INNER JOIN показывает, что в запросе объединяются только те записи таблиц “тПоверка” и “тОперацииПоверки”, в которых связанные поля обеих таблиц совпадают.

Запрос“Неповерятые счетчики” фильтрует информацию о электросчетчиках, которые еще предстоит проверить. Структура запроса “Не поверятые счетчики” представлена следующей схемой:

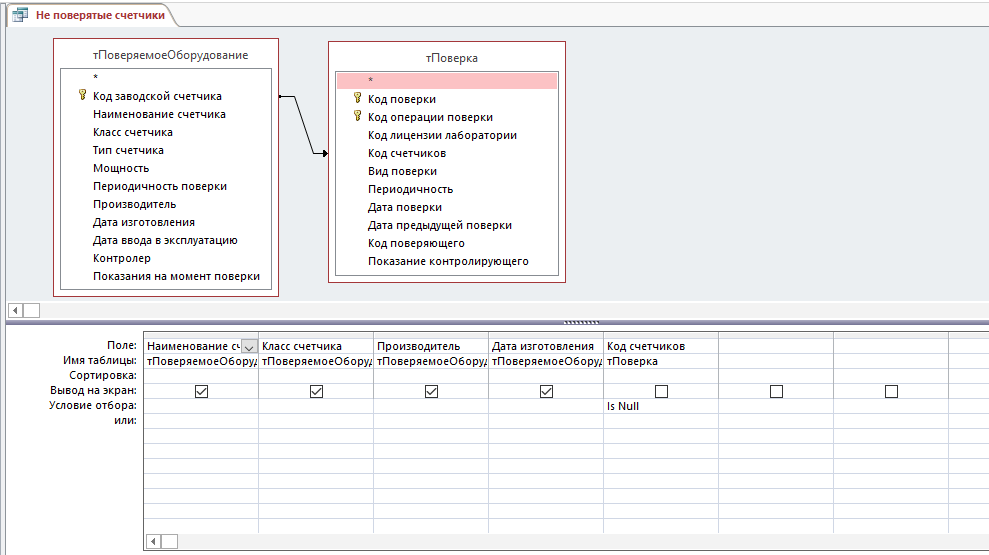


Рисунок 3.19 – Схема запроса “Не поверятые счетчики”

Представление запроса “Не поверятые счетчики” в форме SQL имеет следующий вид:

SELECT тПоверяемоеОборудование.[Наименование счетчика], тПоверяемоеОборудование.[Класс счетчика], тПоверяемоеОборудование.Производитель, тПоверяемоеОборудование.[Дата изготовления]

FROM тПоверяемоеОборудование LEFT JOIN тПоверка ON тПоверяемоеОборудование.[Код заводской счетчика] = тПоверка.[Код счетчиков]

WHERE (((тПоверка.[Код счетчиков]) IsNull));

Здесь перечислены наименования полей, включаемые в запрос, указываются таблицы, из которой они извлекаются, а также описываются условия выборки, участвующий в формировании запроса. Запрос WHERE показывает начало условия в SQL коде, а условие IsNull показывает то, что ищутся записи, которые не используются в двух таблицах одновременно.

Запрос“СЗапрос на вид поверки” фильтрует информацию о видах поверкиэлектросчетчиков, которымимы конкретно в данный момент интересуемся. Структура запроса “СЗапрос на вид поверки” представлена следующей схемой:

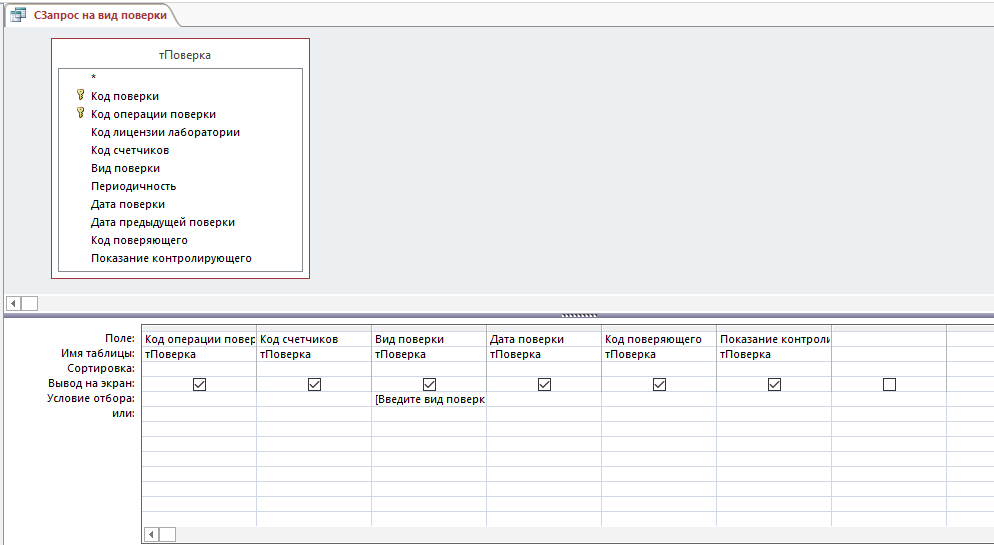


Рисунок 3.20– Схема запроса “СЗапрос на вид поверки”

Представление запроса “СЗапрос на вид поверки” в форме SQL имеет следующий вид:

SELECT тПоверка.[Код операции поверки], тПоверка.[Код счетчиков], тПоверка.[Вид поверки], тПоверка.[Дата поверки], тПоверка.[Код поверяющего], тПоверка.[Показание контролирующего]

FROM тПоверка

WHERE (((тПоверка.[Вид поверки])=[Введите вид поверки]));

Здесь перечислены наименования полей, включаемые в запрос, указывается таблица, из которой они извлекаются, а также описываются условия выборки, участвующий в формировании запроса. Запрос WHERE показывает начало условия в SQL коде, в котором создается окно запроса для ввода ответа.

Запрос “Обновление дат” обновляет информацию о поверках электросчетчиков. Структура запроса “Обновление дат” представлена следующей схемой:

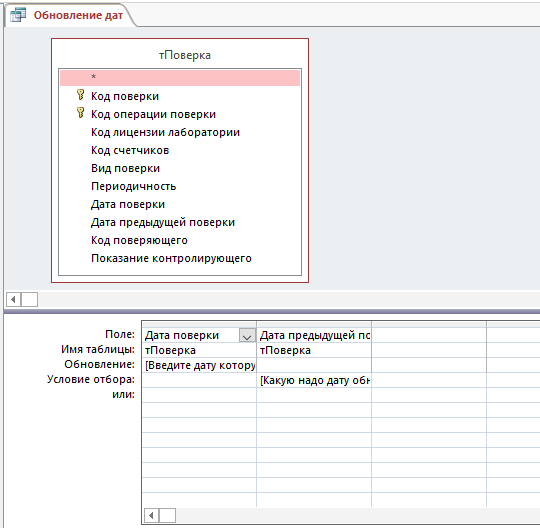


Рисунок 3.21– Схема запроса “Обновление дат”

Представление запроса “Обновление дат” в форме SQL имеет следующий вид:

UPDATE тПоверка SET тПоверка.[Дата поверки] = [Введите дату которую надо установить]

WHERE (((тПоверка.[Дата предыдущей поверки])=[Какую надо дату обновить]));

Запрос UPDATE показывает начало условия по поиску записей, в которой надо обновить поле в SQL коде.

Запрос “Удаление счетчиков” удаляет записи о электросчетчиках. Структура запроса “Удаление счетчиков” представлена следующей схемой:

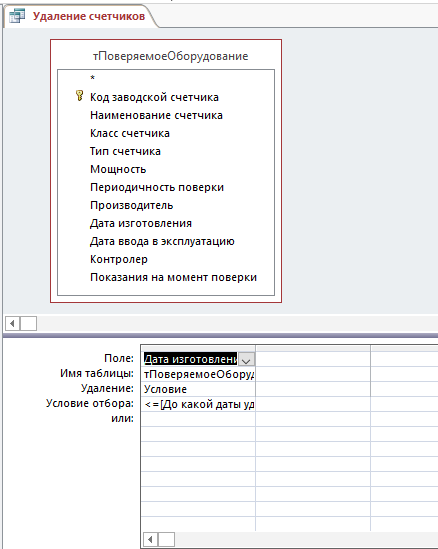


Рисунок 3.22– Схема запроса “Удаление счетчиков”

Представление запроса “Удаление счетчиков” в форме SQL имеет следующий вид:

DELETE тПоверяемоеОборудование.[Дата изготовления]

FROM тПоверяемоеОборудование

WHERE (((тПоверяемоеОборудование.[Дата изготовления])<=[До какой даты удалить счетчики]));

Запрос DELETE показывает начало условия по удалению в SQL коде.

**3.6Структурная схема форм**

Главная форма для администратора, котораясделана так, чтоб легко было взаимодействовать со всей базой данных, и редактировать ее. Структура формы “Главная кнопочная форма Администратора” представлена следующей схемой:

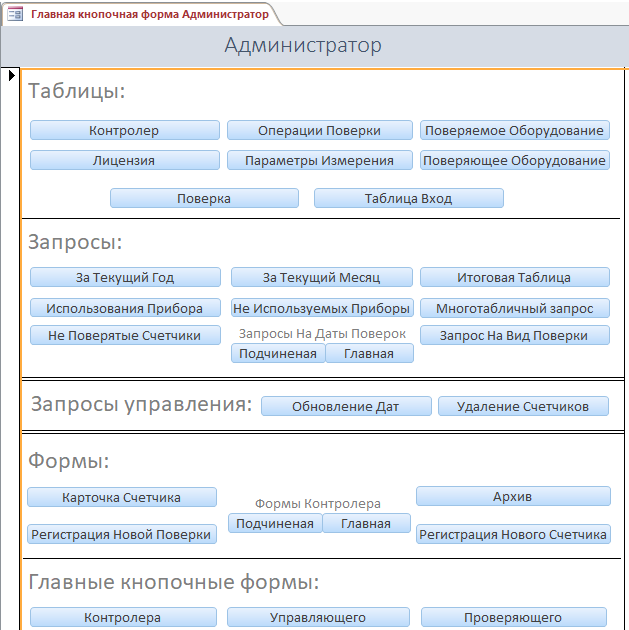
****

Рисунок 3.23– Схема формы “Главная кнопочная форма Администратора”

Главная форма для контролера, котораясделана так, чтоб легко было взаимодействовать базой данных контролеру, и редактировать ее. Структура формы “Главная кнопочная форма Контролера” представлена следующей схемой:

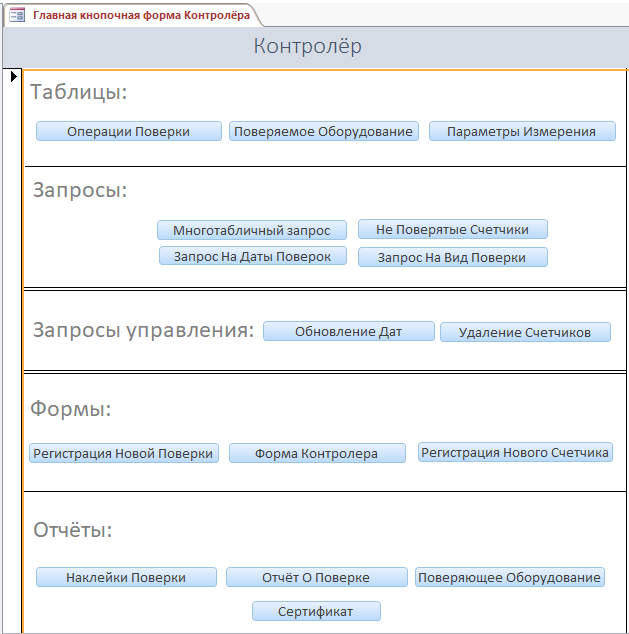
****

Рисунок 3.24– Схема формы “Главная кнопочная форма Контролера”

Главная форма для проверяющего, котораясделана так, чтоб легко было взаимодействовать базой данных проверяющего, и следить с помощью ее за деятельностью учреждения. Структура формы “Главная кнопочная форма Проверяющего” представлена следующей схемой:

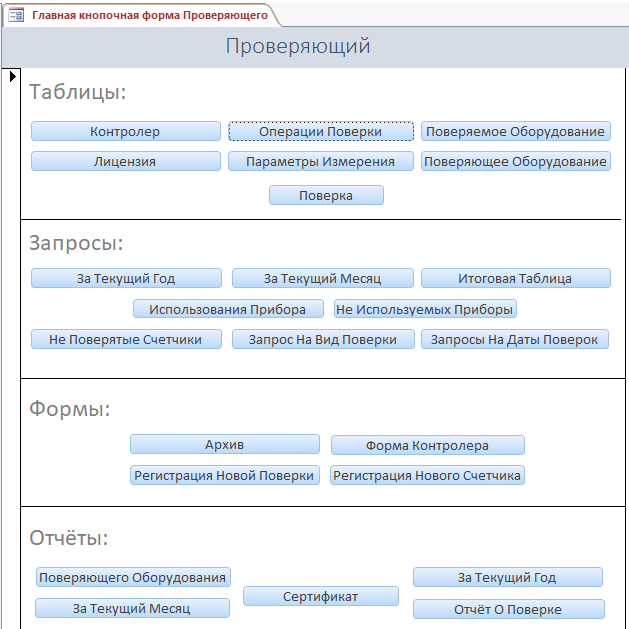
****

Рисунок 3.25– Схема формы “Главная кнопочная форма Проверяющего”

Главная форма для управляющего, котораясделана так, чтоб легко было взаимодействовать базой данных управляющего, и редактировать ее, а также проверять деятельность контролера. Структура формы “Главная кнопочная форма Управляющего” представлена следующей схемой:

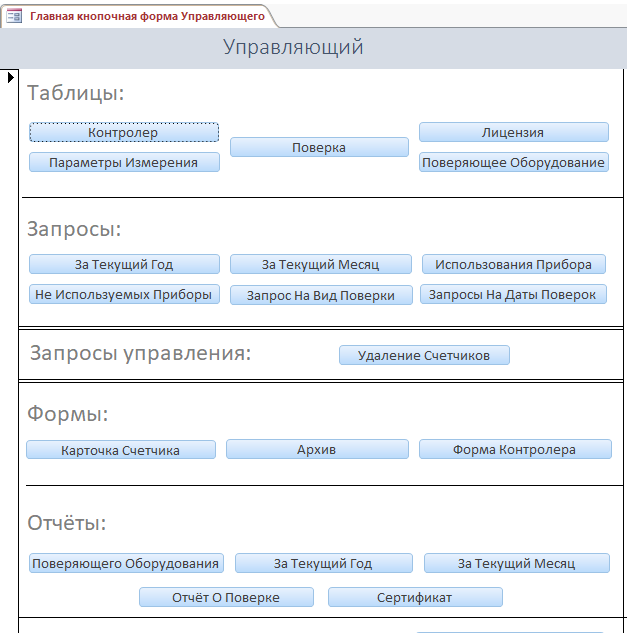
****

Рисунок 3.26– Схема формы “Главная кнопочная форма Управляющего”

Форма фильтрует информацию в зависимости от выбранного контролера и выводит его статистику о поверках, которые расположены в более удобной форме. Структура формы “Карточка контролера” представлена следующей схемой:

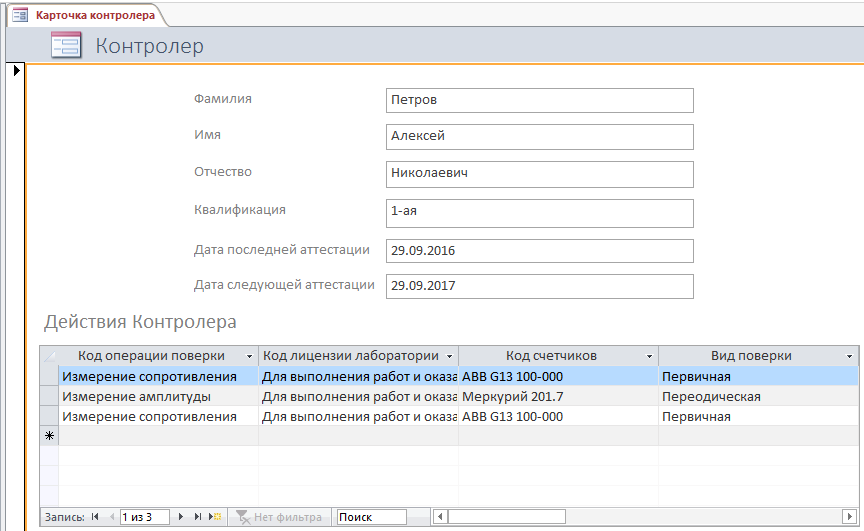


Рисунок 3.27– Схема формы “Карточка контролера”

Форма построена с помощью вспомогательной формы, которая представлена ниже. Подчиненная форма представлена в виде зависимой формы, которая выводит информацию, связанную с информацией на главной форме.

Структура формы “подчиненная форма Карточки контролера” представлена следующей схемой:

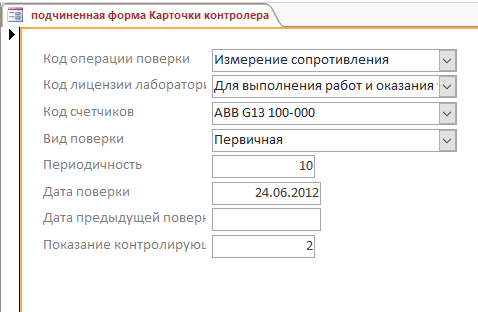


Рисунок 3.28– Схема формы “подчиненная форма Карточки контролера”

Здесь информация о счетчики представлена самой важной информацией о них.Вся информация представлена в компактном виде.

Форма выводит всю имеющуюся информацию из таблиц, имеющихся в БД. Структура формы “Карточка счетчика” представлена следующими схемами:

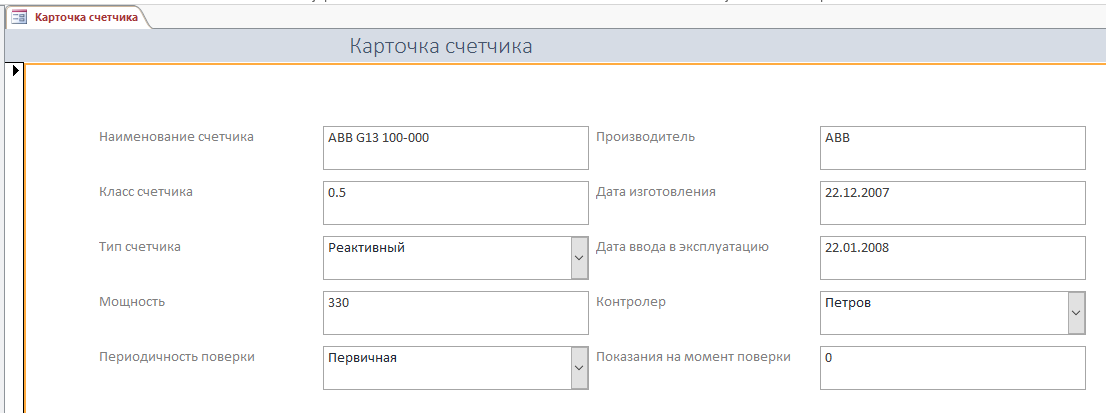


Рисунок 3.29– Схема формы “Карточка счетчика”

Здесь перечислены все наименования полей, хоть как-то связанных с электросчетчиками хранящимися в таблицах БД. Карточки счетчиков можно перелистывать с помощью полосы прокрутки, что позволяет быстро находить информацию.

Форма фильтрует информацию о все таблица имеющих в БД, которые расположены в более удобной форме. Структура формы “Форма Архив” представлена следующими схемами:

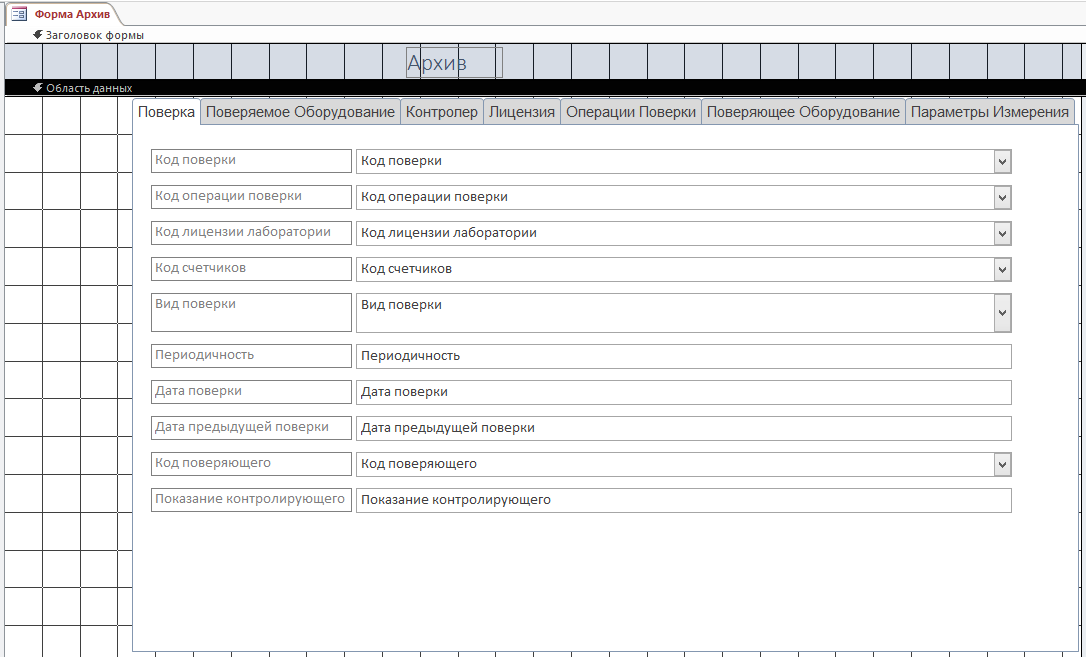


Рисунок 3.30а – Схема формы “Форма Архив”

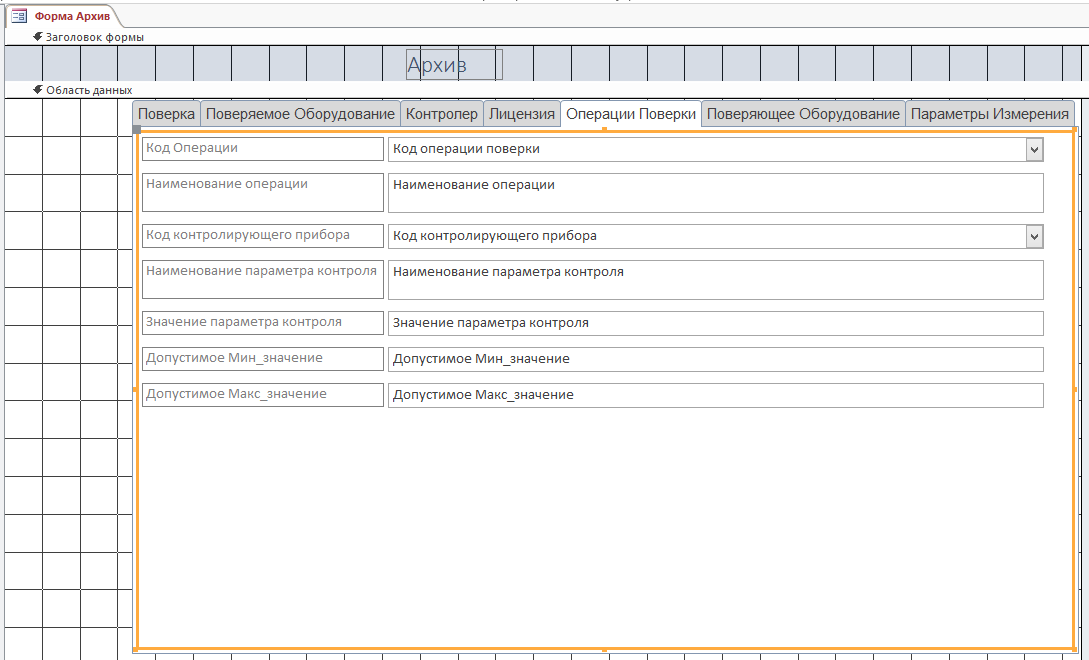


Рисунок 3.30б – Схема формы “Форма Архив”

Здесь перечислены наименования полей, отображают всю доступную информацию хранящихся в таблицах БД. А вкладки позволяют осуществлять более легкое перемещение между формами отдельных таблиц. И позволяют быстрее находить информацию.

Форма создана для того, что более удобно можно было разграничивать права пользователей в базе данных. Структура формы “Форма Вход” представлена следующими схемами:

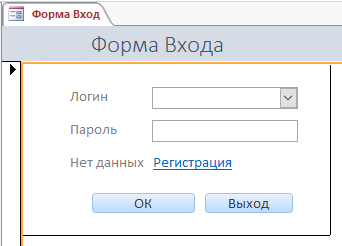


Рисунок 3.31– Схема формы “Форма Вход”

С помощью этого интерфейса можно введя свои данные войти в базу данных или сначала зарегистрироваться в ней, а потом войти в базу данных. В базе данных первоначально уже присутствует один зарегистрированный класс администратор.

Форма создана для удобной регистрации новых пользователей в базе данных и определения своего уровня прав в базе. Структура формы “Форма Регистрация” представлена следующими схемами:

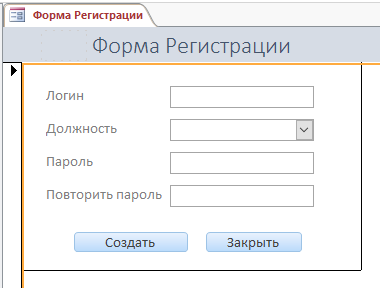


Рисунок 3.32– Схема формы “Форма Регистрация”

С помощью этого интерфейса можно легко пройти регистрацию. Единственное, что нельзя зарегистрировать с помощью этой формы нового администратора базы данных.

Форма позволяет легко внести в БД информацию о новом счетчики. Структура формы “Форма Регистрации нового счетчика” представлена следующими схемами:

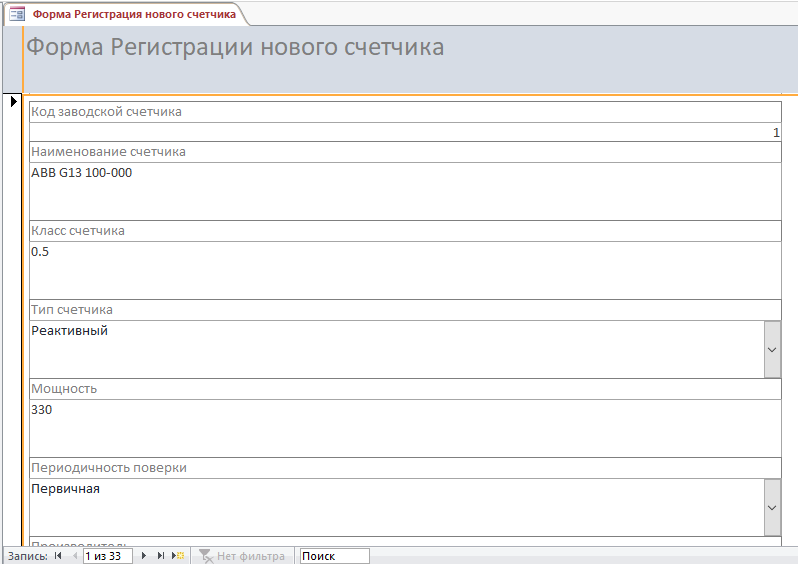


Рисунок 3.33– Схема формы “Форма Регистрации нового счетчика”

Здесь перечислены наименования полей, которые необходимо заполнить, чтоб зарегистрировать новый счетчикв БД. Это форма облегчает эту задачу.

Форма фильтрует информацию о все таблица имеющих в БД, которые расположены в более удобной форме. Структура формы “Форма Регистрации новой поверки” представлена следующими схемами:

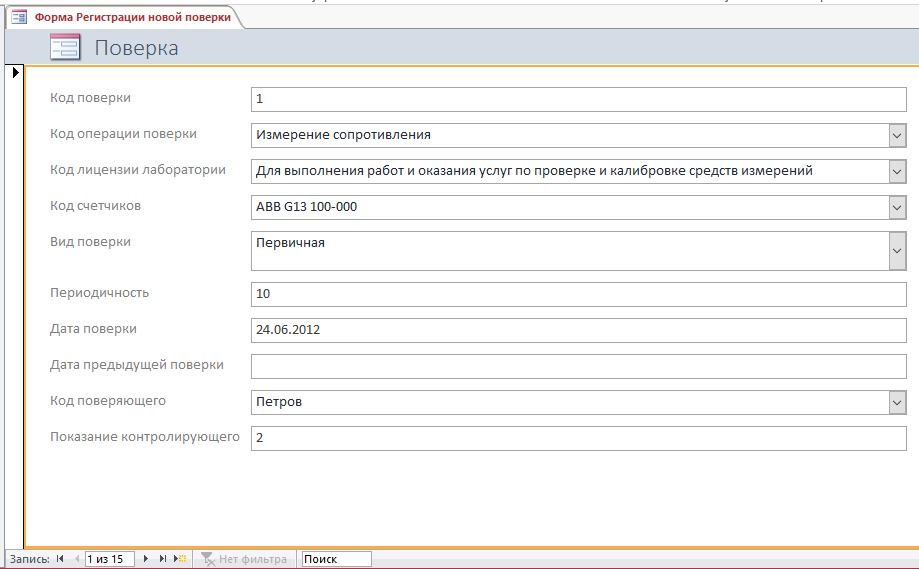


Рисунок 3.34– Схема формы “Форма Регистрации новой поверки”

Здесь перечислены наименования полей, которые необходимо заполнить, чтоб зарегистрировать новую поверкув БД. Это форма облегчает эту задачу.

**3.7Структурная схема отчетов**

Отчет выводит информацию о поверяющем оборудование, котороеимеется в учреждение. Структура запроса “Карточки Поверяющего Оборудования” представлена следующей схемой:

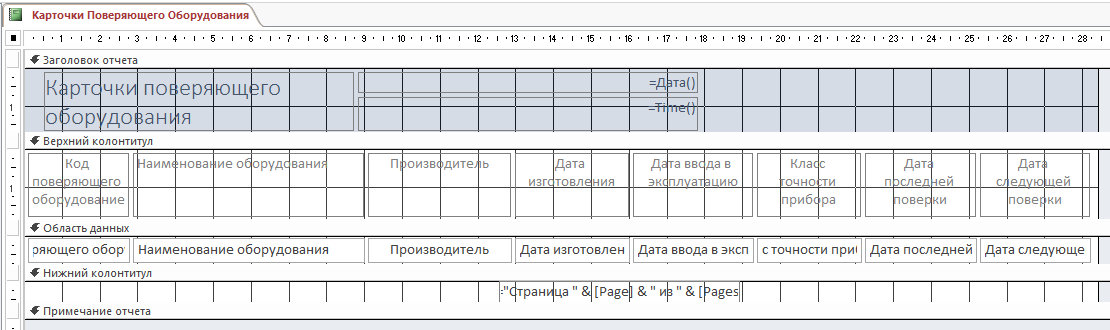


Рисунок 3.35– Схема отчета в конструкторе “Карточки Поверяющего Оборудования”

В форме представление отчета “Карточки Поверяющего Оборудования” имеет следующий вид:

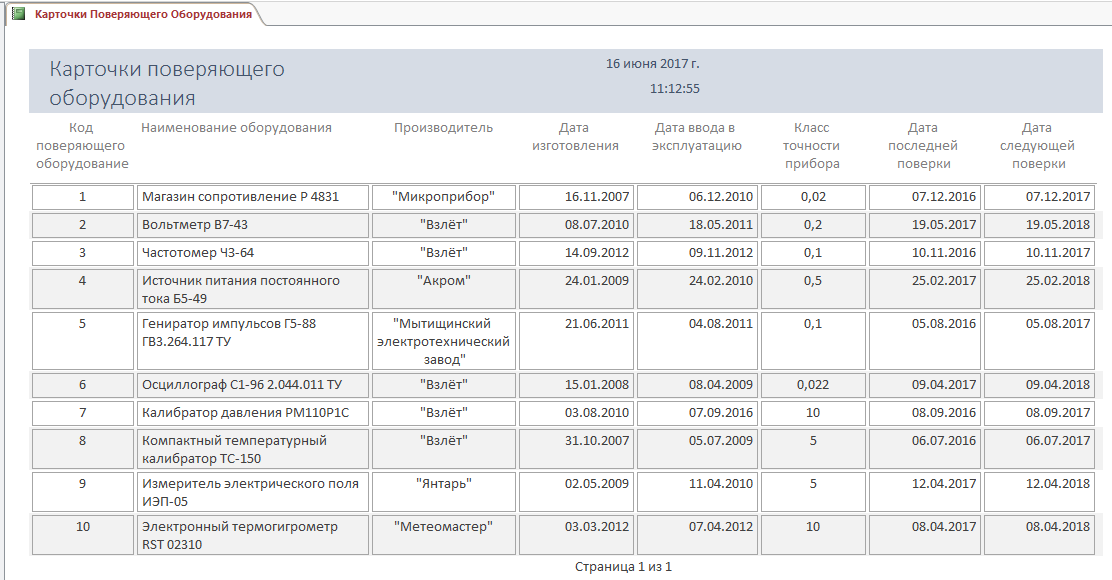
****

Рисунок 3.36– Схема в форме представления отчета “Карточки Поверяющего Оборудования”

Здесь перечислены наименования полей, взятые из таблицы “тПоверяющееОборудование”, и представленные в виде актуального на текущее время отчета “Карточки Поверяющего Оборудования”. Отчет представлен таким образом, чтоб всю информацию можно был легко напечатать в удобной форме. С полезной сопутствующей информацией такой как: время распечатки, чтоб знать о дате формирования текущего отчета, количество страниц и текущая страница, чтоб было легко собрать весь отчет вместе или найти быстро интересующую страницу.

Отчет создает краткую информацию о поверке электросчетчика, котораяраспечатается и клеится на электросчетчики. Структура запроса “Наклейки Поверка” представлена следующей схемой:

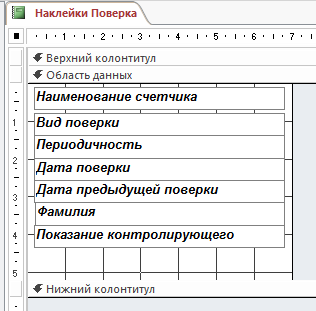


Рисунок 3.37– Схема отчета в конструкторе “Наклейки Поверка”

В форме представление отчета “Наклейки Поверка” имеет следующий вид:

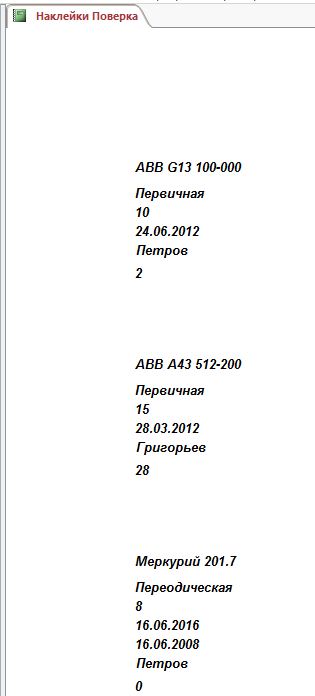


Рисунок 3.38– Схема в форме представления отчета “Наклейки Поверка”

Здесь перечислены наименования полей, взятые из запроса “Наклейки Поверка”, и представленные в виде актуального на текущее время отчета “Наклейки Поверка”. Отчет представлен таким образом, чтоб всю информацию было легко напечатать и поместить на электроприбор.

Отчет фильтрует информацию о электросчетчиках, которые надо поверить в текущем году. Структура запроса “Отчет за текущий год” представлена следующей схемой:

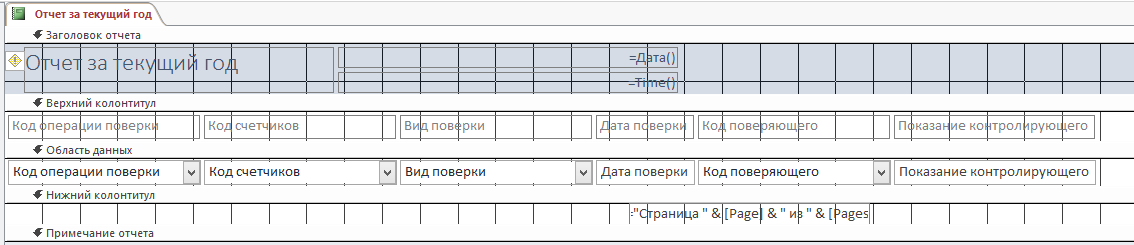


Рисунок 3.39– Схема отчета в конструкторе “Отчет за текущий год”

В форме представление отчета “Отчет за текущий год” имеет следующий вид:

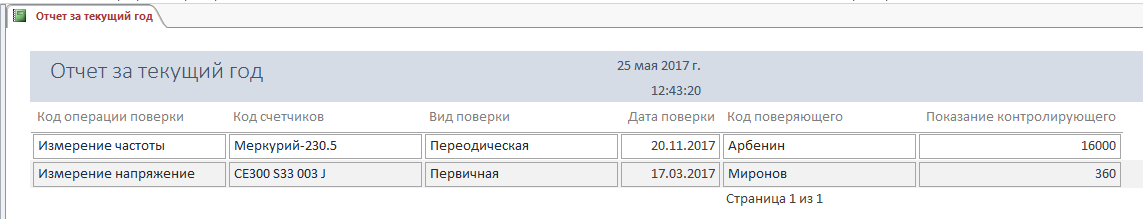


Рисунок 3.40– Схема в форме представления отчета “Отчет за текущий год”

Здесь перечислены наименования полей, взятые из запроса “Запрос за текущий год”, и представленные в виде актуального на текущее время отчета “Отчет за текущий год”. Отчет представлен таким образом, чтоб всю информацию из одноименного запроса было легко напечатать в удобной форме. С полезной сопутствующей информацией такой как: время распечатки, чтоб знать о дате формирования текущего отчета, количество страниц и текущая страница, чтоб было легко собрать весь отчет вместе или найти быстро интересующую страницу.

Отчет фильтрует информацию о электросчетчиках, которые надо поверить в текущем месяце. Структура запроса “Отчет за текущий месяц” представлена следующей схемой:

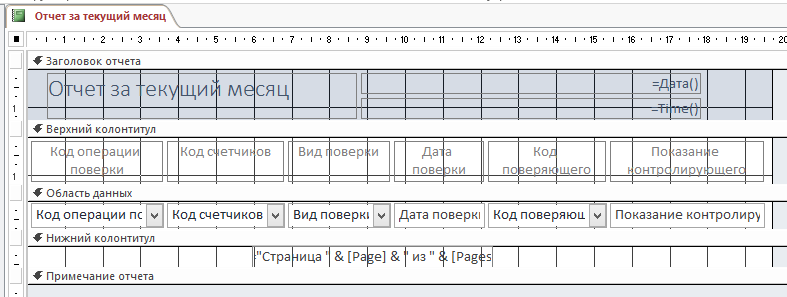


Рисунок 3.41– Схема отчета в конструкторе “Отчет за текущий месяц”

В форме представление отчета “Отчет за текущий месяц” имеет следующий вид:

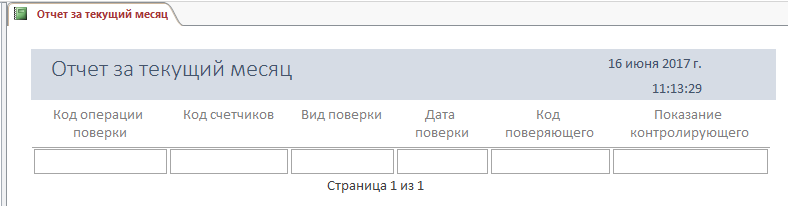


Рисунок 3.42– Схема в форме представления отчета “Отчет за текущий месяц”

Здесь перечислены наименования полей, взятые из запроса “Отчет за текущий месяц”, и представленные в виде актуального на текущее время отчета “Отчет за текущий месяц”. Отчет представлен таким образом, чтоб всю информацию из одноименного запроса было легко напечатать в удобной форме. С полезной сопутствующей информацией такой как: время распечатки, чтоб знать о дате формирования текущего отчета, количество страниц и текущая страница, чтоб было легко собрать весь отчет вместе или найти быстро интересующую страницу.

Отчет создает информацию о поверке электросчетчика, которую потом можно распечатать. Структура запроса “Отчет о поверке” представлена следующей схемой:

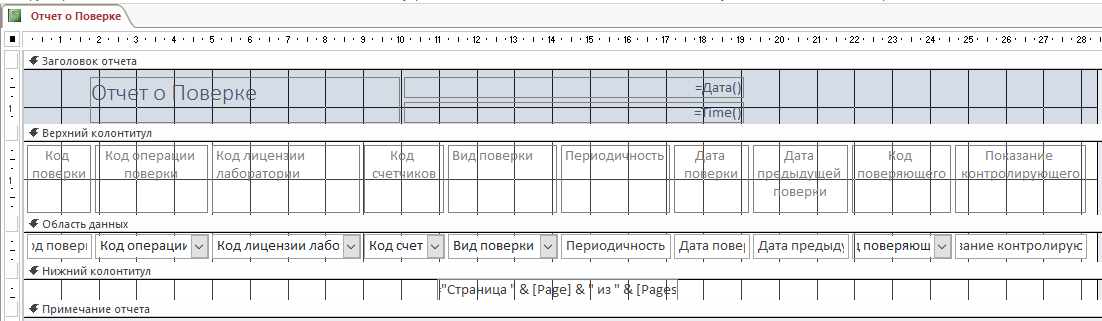


Рисунок 3.43– Схема отчета в конструкторе “Отчет о поверке”

В форме представление отчета “Отчет о поверке” имеет следующий вид:

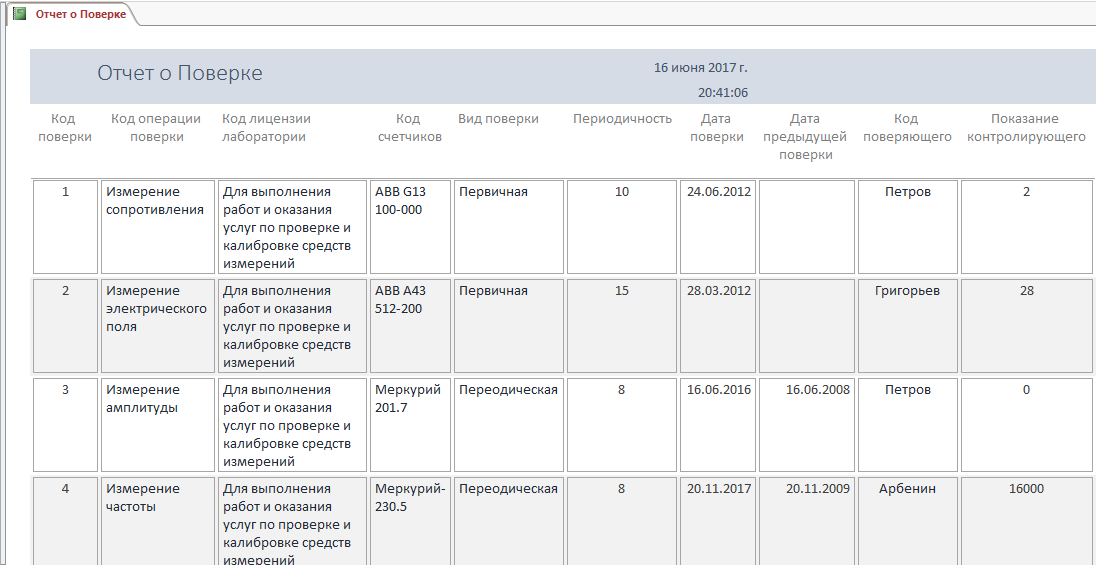


Рисунок 3.44– Схема в форме представления отчета “Отчет о поверке”

Здесь перечислены наименования полей, взятые из запроса “Запрос за Итоговую таблицу”, и представленные в виде актуального на текущее время отчета “Отчет о поверке”. Отчет представлен таким образом, чтоб всю информацию из одноименного запроса было легко напечатать в удобной форме. С полезной сопутствующей информацией такой как: время распечатки, чтоб знать о дате формирования текущего отчета, количество страниц и текущая страница, чтоб было легко собрать весь отчет вместе или найти быстро интересующую страницу.

Отчет выводит краткую информацию о поверяющем оборудование, котороеимеется в учреждение. Структура запроса “Отчет Поверяющее Оборудование” представлена следующей схемой:

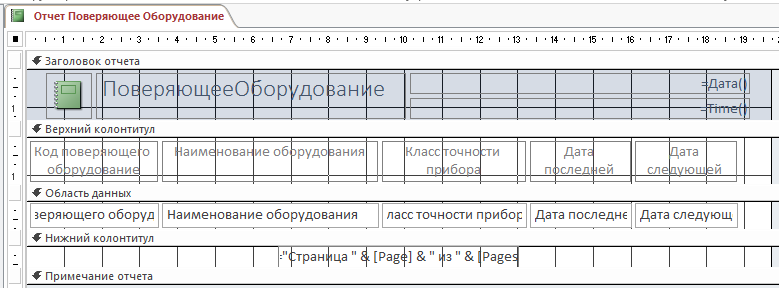


Рисунок 3.45– Схема отчета в конструкторе “Отчет Поверяющее Оборудование”

В форме представление отчета “Отчет Поверяющее Оборудование” имеет следующий вид:

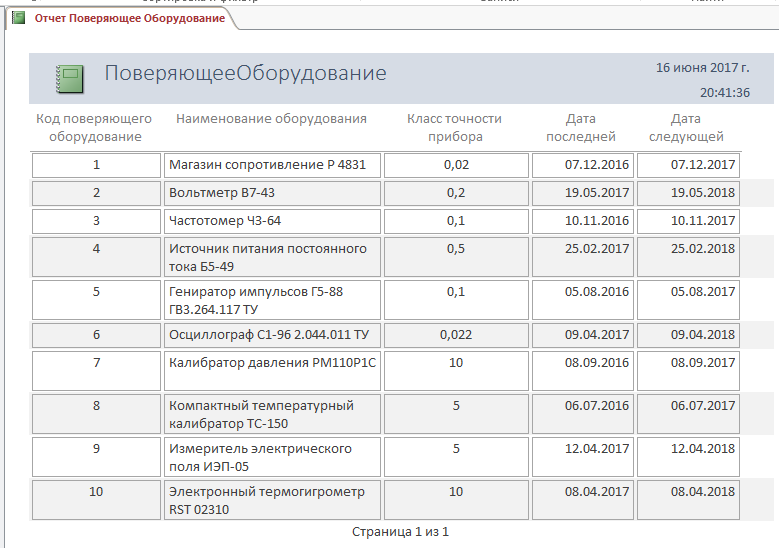


Рисунок 3.46– Схема в форме представления отчета “Отчет Поверяющее Оборудование”

Здесь перечислены наименования полей, взятые из таблицы “тПоверяющееОборудование”, и представленные в виде краткого актуального на текущее время отчета “Отчет Поверяющее Оборудование”. Отчет представлен таким образом, чтоб всю информацию из одноименного запроса было легко напечатать в удобной форме. С полезной сопутствующей информацией такой как: время распечатки, чтоб знать о дате формирования текущего отчета, количество страниц и текущая страница, чтоб было легко собрать весь отчет вместе или найти быстро интересующую страницу.

Отчет выводит информацию о поверке электросчетчика, которые оформлены в виде сертификата. Структура запроса “Отчет Сертификата” представлена следующей схемой:

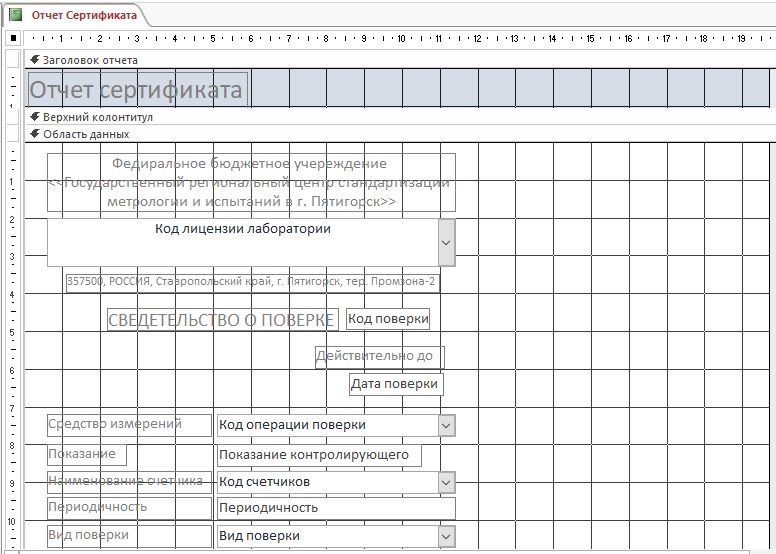


Рисунок 3.47– Схема отчета в конструкторе “Отчет Сертификата”

В форме представление отчета “Отчет Сертификата” имеет следующий вид:

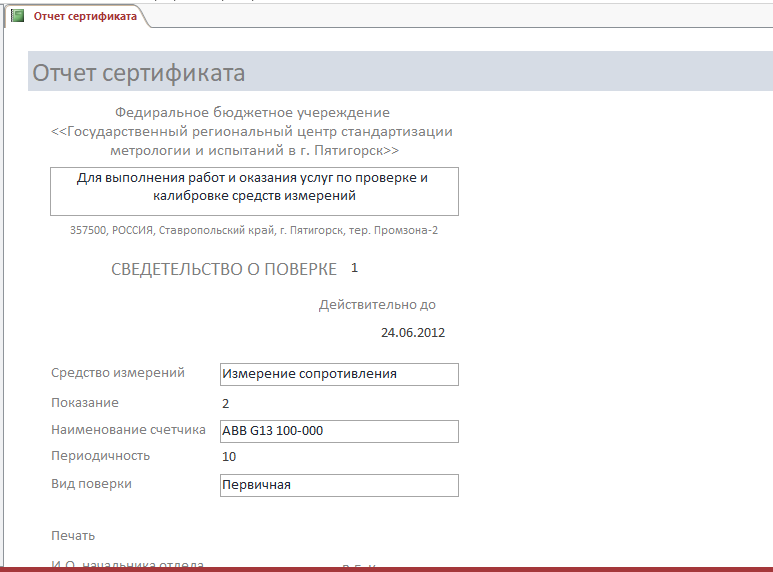


Рисунок 3.48– Схема в форме представления отчета “Отчет Сертификата”

Здесь перечислены наименования полей, взятые из таблицы “тПоверка”, и представленные в виде актуального на текущее время отчета “Отчет Сертификата”. Отчет представлен таким образом, чтоб всю информацию из одноименного запроса было легко напечатать в удобной форме. С полезной сопутствующей информацией такой как: время распечатки, чтоб знать о дате формирования текущего отчета, количество страниц и текущая страница, чтоб было легко собрать весь отчет вместе или найти быстро интересующую страницу. Этот сертификат подтверждает факт поверки.

3.8 Защита базы данных с помощью логина и пароля

Чтобы предотвратить несанкционированный доступ к базе данных Access, ее защищают уникальные данные такие, как Логин и Пароль известные только владельцу данных. Чтобы войти в базу данных и начать работать с базой данных, сотрудник вводит свои данные и проходит индикацию данных. Если данные совпадают, то ему доступ разрешен, и он получает доступ к данным соответствующий его классу доступа.

Чтобы получить логин и пароль для базы данных, необходимо зарегистрироваться в базе данных, как новый пользователь.

При открытии базы данных появится диалоговое окно,в которую надо ввести логин и пароль и нажать кнопку OK.

Пример открытия базы данных показан на рисунке 3.43.

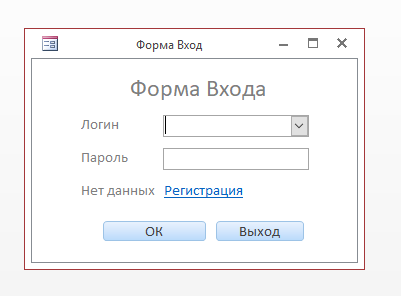


Рисунок 3.49 –Диалоговое окнооткрытия базы данных.

**4 ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ СИСТЕМЫ**

**4.1 Общая информация о системе**

Главной целью данного проекта является организация базы данных в ФБУ«Пятигорский ЦСМ» с целью предоставления более удобного доступа к данным учреждения.

Основой экономической эффективности базы данных является небольшая стоимость, простота внедрения, удобный доступ к данным, возможность редактирования информации, – эти преимущества делаютбазу данных однойиз более привлекательным способом хранения и передачи информации.

Основными целями, которые ставит перед собой руководство компании, являются:

* создать удобную и простую базу данных;
* получение самоокупаемую систему.

**4.2 План анализа экономической эффективности**

После завершения работ по проектированию и реализации, автоматизированная информационная система готова для внедрения в организации заказчика. Для последующего развития информационной системы необходимо рассчитать экономическую эффективность проекта.

Произведем расчет экономической эффективности проекта. План анализа экономической эффективности при создании программного обеспечения следующий:

* Технико-экономическое обоснование разработки программного обеспечения
* Общие расходы на реализацию системы
* Эффективность внедрения для организации

**4.3 Технико-экономическое обоснование разработки программного обеспечения**

Таблица 4.1 – Фазы проектирования и разработки

|  |  |  |  |
| --- | --- | --- | --- |
| Фаза | Содержание работ | Трудоемкость | |
| дни | % |
| Начальная стадия | Сбор информации, анализ требований | 19 | 0,2 |
| Стадия проектирования | Анализ требований и проектирование системы, планирование необходимых ресурсов | 27 | 0,29 |
| Разработка | Разработка информационной системы | 40 | 0,43 |
| Внедрение | Тестирование, обучение пользователей | 7 | 0,08 |
| Итого |  | 93 | 100,00 |

Общая трудоемкость разработки программного обеспечения рассчитывается по формуле:

 (4.1)

где Тоб– общая трудоемкость разработки, дни;

Т*i* – трудоемкость по стадиям, дн*;*

*n* – количество стадий разработки.

В итоге на проектирование и разработку информационной системы было потрачено 93 рабочих дня.

**4.4. Общие расходы на реализацию системы**

Затраты на эксплуатацию данной системы определяются по формуле:

 (4.2)

где Р – расходы на разработку;

Е – единовременные выплаты;

Н – накладные затраты;

***4.4.1Общие расходы на реализацию системы***

Текущиерасходы на разработку данной системы рассчитываются по формуле:

 (4.3)

где Фот – фонд оплаты труда;

ОС – отчисления на соц. нужды;

Ао – амортизационные отчисления;

Э – электроэнергия для производственных нужд;

Фонд оплаты труда

Затраты по оплате труда состоят из основной и дополнительной заработных плат и рассчитываются по формуле:

 (4.4)

где Зосн - основная заработная плата,

Здоп - дополнительная заработная плата.

Заработная плата при проектировании и разработке включает зарплату всех сотрудников, принимающих непосредственное участие в разработке программного обеспечения. В данном случае необходимо учитывать заработную плату разработчика.

Среднедневная заработная плата разработчика определена из расчета 27500руб. в месяц и равна:

Зср. дн. р =27 500/22=1250руб./день

Получаем, заработная плата при проектировании и разработке равна:

Зосн =1 250\*93 = 116250руб.

Дополнительная заработная плата составляет 10% от основной заработной платы и рассчитывается по формуле:

 (4.5)

Здоп=0,1\*116 250= 11 625руб.

Общий фонд оплаты труда за год составит:

Фот=116250 + 11625 =127875руб.

Социальный налог (30,2)

Отчисления на социальные нужды составляют на сегодняшний день 30.2%от общего фонда заработной платы, следовательно:

Ос= Фот\*0,302 = 127875\*0,302 = 38 618,25руб. (4.6)

Расчет затрат на амортизацию

Общие амортизационные отчисления берутся исходя из суммы всех амортизаций:

 (4.7)

где Ао–общая амортизации;

Ам–машинная амортизации;

Аи–инструментальная амортизации;

Тогда машинные амортизационные отчисления составляют:

 (4.8)

где Ам – амортизационные отчисления, руб.;

Оф – стоимость ЭВМ и оборудования, руб.;

Нам – норма амортизации (принято 20%), %;

Тм – время использования оборудования, дни

АМ = (30 000 \* 20 \* 67) / (365 \* 100) = 40 200 000 / 36 500 = 1 101,37руб.

Стоимость инструментальных средств включает стоимость системного программного обеспечения, примененного при разработке программного обеспечения, в размере износа за период использования.

Норма амортизации для системного программного обеспечения 30%, а время применения 67 дня.

Использованные средства представлены в таблице 4.2.

Таблица 4.2 – Инструментальные средства

|  |  |
| --- | --- |
| Наименование продукта | Стоимость (руб.) |
| MicrosoftOffice 2016 | 18 900 |
| Итого | 18 900 |

Рассчитаем амортизацию инструментального средства:

 (4.9)

где Ам – амортизационные отчисления, руб.;

Оф – стоимость ЭВМ и оборудования, руб.;

Нам – норма амортизации (принято 30%), %;

Тм – время использования оборудования, дни

АИ = ((18 900 \* 30) / (365 \* 100)) \* 67 = (567 000 / 36 500) \* 67 = 15,53 \* \* 67 = 1 040,51руб.

Следовательно,

Ао = 1 101,37 + 1 040,51 = 2 141,88руб.

Расчет затрат на электроэнергию

Затраты на электроэнергию для производственных нужд, включают в себя расходы электроэнергии на оборудование и дополнительные нужды и рассчитываются по формуле:

, (4.10)

где Зэл.обор. – затраты на электроэнергию для оборудования;

Здоп.нуж. – затраты на дополнительные нужды;

Затраты электроэнергии на оборудование рассчитывается следующим образом:

Стоимость затрат электрооборудованияЗэл.обор. зависит от себестоимости машино-часа работы ЭВМ *СМЧ*, а также времени работы на ЭВМ *ТЭВМ*.

Себестоимость машино-часа ЭВМ равна:

СЭВМ=0,8 кВт/час \* 4,16руб/кВт = 3,33руб./час

Время использования оборудования:

ТЭВМ=0,32\*Тобщ+0,7\*Тпроект+0,91\*Треал+0,48\*Твнед=0,32\*19++0,7\*27+0,91\*40+0,48\*7=6,08 + 18,9 + 36,4 +3,36=64,74(дней) = =64,74 \*8= 517,92(часов).

Затраты на электроэнергию:

Зэл.обор.= 517,92часов \* 3,33руб./час=1724,67руб.

Затраты на дополнительные нужды составляют 5% от затрат на электроэнергию оборудования и рассчитываются по формуле:

 (4.11)

где ЗЭЛ.ОБОР - затраты на электроэнергию для оборудования;

Затраты на электроэнергию для дополнительных нужд:

Здоп.нуж. = 0,05 \* 1724,67= 86,23руб.

Тогда суммарные затраты на электроэнергию будут равны:

Э = 1724,67+ 86,23 = 1810,9руб.

Итак, план затрат на разработку приведен в таблице 4.3.

Таблица 4.3 – Разделение затрат по видам.

|  |  |
| --- | --- |
| Вид затрат | Затраты (руб.) |
| Заработная плата | 127875 |
| Отчисления на социальные нужды | 38 618,25руб. |
| Амортизационные отчисления | 2 141,88 |
| Затраты на электроэнергию | 1 810,9 |
| Итого | 170 446,03 |

***4.4.2Стоимость внедрения программного обеспечения организацией***

К единовременным затратам пользователя программного обеспечения относятся затраты на оплату:

* программного обеспечения Цпо;
* инструментальных средств Цис;
* обучение сотрудников Косв.

Текущие расходы на разработку данной системы рассчитываются по формуле:

 (4.12)

Стоимость программного обеспечения.

Стоимость программного обеспечения равна сумме себестоимости и прибыли разработчика (обычно составляет 20% себестоимости), а также налог на добавленную стоимость, который составляет 18%. Для расчета можно использовать следующую формулу

Цпо= Спо + П + НДС, (4.13)

где Спо- себестоимость программного обеспечения,

П - прибыль разработчика,

НДС - налог на добавленную стоимость.

Следовательно, стоимость программного обеспечения составляет:

Спо=170 446,03руб.

НДС = 170 446,03\*18% =30 680,29руб.

П = 170 446,03\* 20% = 34 089,21руб.

Цпо= 170 446,03+ 30 680,29 + 34 089,27 = 235 215,59руб.

Стоимость инструментальных средств

Стоимость инструментальных средств, необходимых для функционирования системы включает стоимость операционных систем. В организации уже установлены и используются все необходимые инструментальные средства. Поэтому при внедрении не предусматривается расходов по данным статьям.

Так как в организации установлено также и все необходимое техническое обеспечение, то расходы по данной статье не предусматриваются.

Стоимость обучения сотрудников организации.

Расчет производится по следующей формуле:

Косв=Чпп\*Сосв\*tосв, (4.14)

где Чпп - численность сотрудников на обучение,

Сосв - стоимость обучения одного сотрудника в день,

tосв - время обучения.

Фактически в организации информационной системой будут пользоваться 20 сотрудников. Время необходимое для обучения приблизительно оценивается в семь рабочих дней для каждого. Стоимость обучения одного сотрудника в день составляет 1 250 рублей. В итоге получается затраты на обучение персонала:

Косв= 20 \* 1 250 \* 7 = 175 000 руб.

Суммарные затраты для организации представлены в таблице 4.4.

Таблица 4.4 – Суммарные затраты для организации

|  |  |
| --- | --- |
| Вид затрат | Затраты (руб.) |
| Цена программного обеспечения | 235 215,59 |
| Обучение персонала | 175 000 |
| Итого | 410 215,59 |

***4.4.3Накладные расходы, общие затраты и инвестиционный план***

Накладные расходы составляют 75 % от всех затрат и рассчитываются по формуле:

 (4.15)

Тогда накладные затраты составят:

Н=0,75\*(235 215,59 + 175 000)= 0,75 \* 410 215,59 = 307 661,7руб.

Результаты расчета общий расход по проектированию системы, представлен в таблице 4.5

Таблица 4.5 – Расход по проектированию системы

|  |  |
| --- | --- |
| Показатель | Сумма руб. |
| Расходы на разработку | 170 446,03 |
| Единовременные выплаты | 410 215,59 |
| Накладные расходы | 307 661,7 |
| ИТОГО | 888 323,32 |

Составим инвестиционный план, учитывая время реализации проекта, стадии проектирования, затраты на разработку.

Таблица 4.6 – Этапы реализации проекта

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Этапы реализации | Январь | Февраль | Март | Апрель | Май |
| Начальная стадия | 17 | 2 |  |  |  |
| Стадия проектирования |  | 16 | 11 |  |  |
| Разработка |  |  | 11 | 20 | 9 |
| Внедрение |  |  |  |  | 7 |

Составим инвестиционный план, с более детальными денежными инвестициями.Таблица 4.6 – Инвестиционный план

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Этапы реализации | Январь | Февраль | Март | Апрель | Май |
| Начальная стадия | 162 381,68 | 19 103,72 | 0 | 0 | 0 |
| Стадия проектирования | 0 | 152 829,76 | 105 070,46 | 0 | 0 |
| Разработка | 0 | 0 | 105070,46 | 191 037,2 | 85 966,74 |
| Внедрение | 0 | 0 |  |  | 66 863,02 |
| Итого | 162 381,68 | 171 933,51 | 210 140,92 | 191 037,2 | 152 829,76 |

**4.5Эффективность внедрения для организации**

Оценивая повседневную работу ФБУ«Пятигорский ЦСМ», постараемся оценить экономический эффект от внедрения информационной системы учета расходных материалов и комплектующих.

Диспетчер сервисного центра ежедневно в среднем оформляет 15 заявок в день. Стоимость таких запросов составляет в среднем 1 400руб. в день, что составит в месяц 21000руб. Безусловно, количество совершаемых ежедневно заявок и их стоимость не может быть постоянной величиной.

Возьмем 21 000руб. как среднее ежедневное значение, тогда за месяц будет:

Д = 21 000 \* 22 = 462 000руб.

Срок окупаемости информационной системы по учету расходных материалов и комплектующих при затратах на разработку и внедрение 908 040,05 рублей.

Определим срок окупаемости по формуле:

Сок=888 323,32/462 000 = 1,9 месяца.

Получаем, что через 1,9 месяца информационная система полностью себя окупит, значит, применение разработки является эффективным.

В ходе вычислений были получены результаты:

* Рассчитаны затраты на разработку – 888 323,32 рублей;
* Рассчитан экономический эффект от внедрения - 462 000 рублей;
* Срок окупаемости системы составляет 1,9 месяца.

**5. БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ СИСТЕМЫ**

5.1 Анализ условий труда исследователя

Данная система была исследована на персональном компьютере типа IBMPC. Рассмотрим условия труда в помещении, характеризующиеся совокупностью факторов производственной сферы, оказывающих определенное воздействие на здоровье и работоспособность человека в процессе труда. Под условиями работы подразумевают комплекс физических, химических и психофизиологических факторов, установленных стандартами по безопасности труда (ССБТ).

Общая площадь помещения 21м2. Рассматриваемое помещение рассчитано на три рабочих места, из которых одно оснащено персональным компьютером. На одного работающего приходится 7м2  площади и 17,5м3 объема, что соответствует установленному стандарту (не менее 6 м2 площади и 15м3 объема).

Наибольший допустимый уровень шума в вычислительной лаборатории согласно равен 50дБА. В помещении, где проводилась разработка проекта, допустимый уровень шума не превышается и составляет 28дБА.

По тяжести физической работы, согласно ГОСТ 12.1.005–88, производственную деятельность исследователя можно отнести к категории 1а, 2б (легкие физические работы). Для этой категории считаются оптимальными: температура воздуха – 22-240С (холодное время года) и 23–250С (теплое время года); скорость ветра – 0,1м/с; относительная влажность воздуха 40–60%. В холодный и переходный период года при температуре наружного воздуха ниже +10°С эффективно-эквивалентная температура воздуха в помещении в пределах +16…20°С, относительная влажность воздуха 50–60%, скорость движения воздуха менее 0,1м/с.

В теплый период года при температуре наружного воздуха выше +10°С эффективно-эквивалентная температура воздуха в помещении в пределах +22…24°С, относительная влажность воздуха 40–50%, скорость движения воздуха менее 0,1м/с, что соответствует требованиям ГОСТ 12.1.005–88.ССБТ.

Температура воздуха в помещении поддерживается в летнее время с помощью вентиляции. В данном помещении используется естественная вентиляция. Перемещение воздуха осуществляется за счет естественных процессов (разности температур в помещении и вне помещения, ветрового напора), в зимнее время – с помощью радиаторов централизованной системы отопления и теплоэлектрических нагревателей.

Работник, работающий за компьютером, подвергается облучению электромагнитными полями (ЭМП) двух диапазонов: критически низких частот (КНЧ) (генерируется током сети 50-60 Гц и кадровой разверткой   
50-85 Гц) и низких частот (генерируется строчным трансформатором).

При компоновке компьютеризированного рабочего места следует учесть, что электромагнитное поле обыкновенного дисплея может составлять от 0.5 до 2.3мкТ на расстоянии10 см от экрана и от 0.4 до 1.4мкТ на расстоянии30 см. Так как источником КНЧ - излучения является фокусирующая система дисплея, то соответственно интенсивность поля сзади и сбоку от дисплея гораздо выше, чем спереди. Немаловажно и то, что КНЧ-поля практически ничем не экранируются: ни стенами, ни свинцовыми перегородками.

Идеальным вариантом защиты от КНЧ - электромагнитных полей является использование дисплеев без электронно-лучевой трубки (ЭЛТ). Такими дисплеями являются жидкокристаллические дисплеи. По сравнению с дисплеями на ЭЛТ они не излучают пе­ременного электромагнитного поля.

Экран монитора создает в непосредственной близости от себя электростатическое поле. Предельно допустимый уровень электростатического поля не более 20кВ/м. С целью устранения зарядов статического электричества в лабораториях вычислительной техники применяют: заземление электропроводных частей (защитных экранов), общее или местное увлажнение воздуха, нейтрализаторы статического электричества.

Конструкция рабочей мебели обеспечивает возможность индивидуальной регулировки соответственно росту работающего и создает удобную позу. Экран видеомонитора находится от глаз работника на оптимальном расстоянии – 600–700мм, но не менее 500мм.

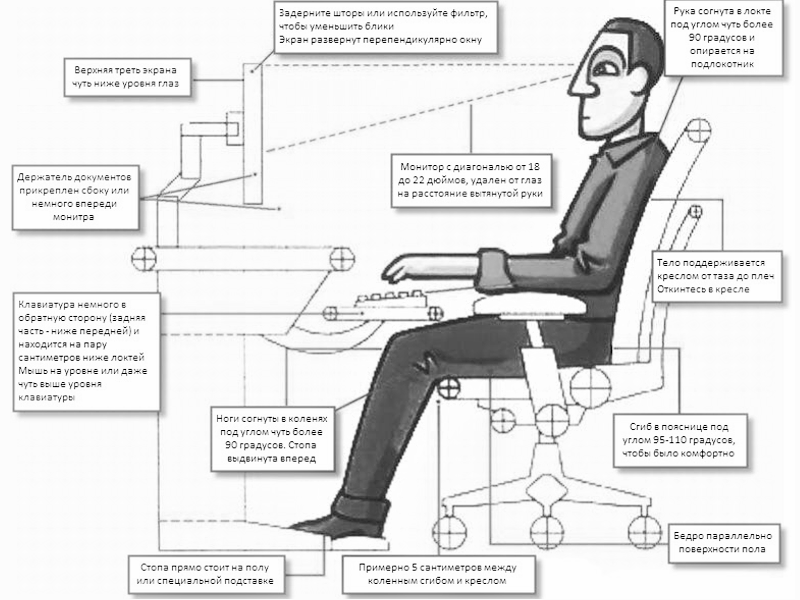


Рисунок 5.1 – Положение оператора за ЭВМ

Работник получает статическую физическую нагрузку второго вида (поддержание рабочей позы). Так как приходится вводить данные в ЭВМ, то число мелких движений и информационных сигналов в час не превышает, соответственно, 360 и 75. Для исследователя характерна монотонность труда первой категории (однообразие рабочих действий и частое их повторение). Профилактикой перенапряжения нервно-мышечного аппарата является рациональный режим труда и отдыха.

Сведем все исследованные факторы и показатели в единую таблицу   
(см. табл. 5.1).

На основании анализа всех рассмотренных факторов получим такой вывод: общая оценка условий труда составляет 3,2 – вредный напряженный труд первой степени, т.е. условия труда характеризуются такими отклонениями уровней вредных факторов от гигиенических нормативов, которые вызывают функциональные изменения, восстанавливающиеся, как правило, при более длительном (чем к началу следующей смены) прерывании контакта с вредными факторами и увеличивают риск повреждения здоровья.

Вредные условия труда характеризуются наличием вредных производственных факторов, превышающих гигиенические нормативы и оказывающих неблагоприятное действие на организм работающего и/или его потомство.

Таблица 5.1. Факторы и показатели работы работника

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Фактор | | | Фактическое значение | Нормативное значение | Класс |
| факторы вредности и опасности работы | | | | | |
| Уровень шума, дБА | | | 28 | 50 | 2 |
| Напряженность электростатического поля, кВ/м | | | 10 | 20 | 2 |
| Микро-климат | Температура воздуха, 0С | | 24 | 23…25 | 1 |
| Скорость движения воздуха, м/с | | <0,1 | 0,1 | 1 |
| Относительная влажность воздуха, % | | 51 | 40…60 | 1 |
| Тепловое излучение, Вт/м2 | | 30 | 30 | 1 |
| Коэффициент естественной освещенности, % | | | 6 | 5 | 1 |
| Освещенность рабочей поверхности, лк | | | 300 | 300 | 1 |
| показатели тяжести работы | | | | | |
| Статическая физическая нагрузка | | | Для поддержания рабочей позы. | | 1 |
| Рабочая поза | | | Периодическое нахождение в неудобной фиксированной позе до 25% времени. | | 2 |
| показатели напряженности работы | | | | | |
| Содержание работы | | Творческая деятельность, требующая решения. | | | 3.2 |
| Восприятие информации и ее оценка | | Восприятие информации с последующей комплексной оценкой связанных параметров. | | | 3.2 |
| Характер работы | | Работа по установленному графику с его коррекцией по ходу деятельности. | | | 2 |
| Наблюдение за экраном (ч в смену) | | Более 4 | | до 2 | 3.2 |
| Фактическая продолжительность рабочего дня | | 8—9ч | | 6—7ч | 2 |
| Наличие регламентированных перерывов | | Регламентированные перерывы достаточной продолжительности. | | | 1 |
| **Общая оценка условий труда – 3,2** | | | | | |

5.2. Мероприятия по улучшению условий труда

При работе работника происходит периодическое нахождение в неудобной фиксированной позе при введении данных в базу данных. Поэтому необходимо подобрать удобную и практичную мебель: специальный компьютерный стол и кресло. Деятельность работника является эмоционально напряженной. Длительность работы может привести к нервному перенапряжению. Поэтому при его работе необходимо предусмотреть меры психологической релаксации. Также необходимо предусмотреть обязательные физические нагрузки для общего расслабления организма.

Учитывая все выше изложенное, можно порекомендовать следующие параметры организации рабочего места:

* экран дисплея должен находиться не ближе40 см от лица работника, так как на этом расстоянии интенсивность поля падает до безопасной величины;
* необходимо, чтобы сзади и сбоку от дисплея не организовывались рабочие места на расстоянии менее1.22 м. При организации рабочих мест необходимо учесть дисплеи, находящиеся в соседних помещениях;
* при перерывах в работе с компьютером, если нет надобности его выключать, необходимо выключать дисплей;
* при работе работника с компьютером в течение длительного времени необходимо один раз в час на 15 минут прерывать работу с компьютером. Хронометраж перерывов при восьмичасовом рабочем дне (с 8.00 до 17.00):

9.00 – 9.15; 10.15 – 10.30; 11.30 – 11.45; 12.00 – 13.00 – обеденный перерыв; 14.00 – 14.15; 15.15- 15.30; 16.30 – 16.45.

* дисплей не должен быть повернут экраном в сторону окна. В случае его расположения возле окна, необходимо расположить его перпендикулярно стеклу, для предотвращения возникновения бликов на экране;
* свет от осветительных ламп не должен падать на дисплей с углом более 60 градусов от вертикали;
* освещенность рабочего места необходимо поддерживать в пределах двух третьих от нормальной освещенности (порядка 300);
* при освещении рабочего места неприемлемо использование мигающих источников света, например люминесцентных ламп;
* интерьер, на фоне которого установлен дисплей, должен быть не ярким. Блестящие предметы необходимо из него исключить, заменив их, по возможности, на матовые. Освещенность интерьера должна быть примерно такая же, как и у дисплея. Соотношение яркости экрана и окружения не должно превышать 3:1;
* при работе использовать специальные противобликовые фильтры, надеваемые на экран дисплея.

В настоящее время, в связи с все более убыстряющимися темпами компьютери­зации общества, возрастает актуальность проблемы влияния компьютера на гомеостаз организма работника. Использование ПЭВМ без принятия мер безопасности может привести к повышению уровня заболеваемости пользователей вследствие негативного влияния факторов, описанных ниже. В соответствии с СанПиН 2.2.2.542-96 негативные факторы этой группы обуславливаются воздействием излучений работающего монитора компьютера. Их можно разделить на три группы:

* воздействие рентгеновского излучения;
* воздействие электростатических полей;
* воздействие электромагнитных полей.

Исследования показали, что воздействие рентгеновского излучения влияет на человека работника незначительно. Так мощность дозы рентгеновского излучения на расстоянии 5см от экрана дисплея составляет 0.5мР/ч. Далее она уменьшается пропорционально квадрату расстояния от экрана.

Воздействие электростатических полей в своем проявленном аспекте состоит в следующем. Положительные ионы и частички пыли, попадая в статическое поле экрана, разгоняются в нем и выбрасываются навстречу работнику. В этот поток обычно попадает лицо, в результате чего у работника может возникнуть не проходящая сыпь на лице и повреждения глаз. Данного воздействия можно практически полностью избежать, установив на монитор специальный антистатический экранный фильтр.

Воздействие электромагнитных полей являются наиболее опасными из рассматриваемой группы. Работник, работающий за компьютером, подвергается облучению электромагнитными полями (ЭМП) двух диапазонов: критически низких частот (КНЧ) (генерируется током сети 50-60 Гц и кадровой разверткой 50-85 Гц) и низких частот (генерируется строчным трансформатором).

При компоновке компьютеризированного рабочего места следует учесть, что электромагнитное поле обыкновенного дисплея может составлять от 0.5 до 2.3мкТ на расстоянии10 см от экрана и от 0.4 до 1.4мкТ на расстоянии30 см. Так как источником КНЧ - излучения является фокусирующая система дисплея, то соответственно интенсивность поля сзади и сбоку от дисплея гораздо выше, чем спереди. Немаловажно и то, что КНЧ-поля практически ничем не экранируются: ни стенами, ни свинцовыми перегородками.

Идеальным вариантом защиты от КНЧ - электромагнитных полей является использование дисплеев без электронно-лучевой трубки (ЭЛТ). Такими дисплеями являются жидкокристаллические дисплеи. По сравнению с дисплеями на ЭЛТ они не излучают пе­ременного электромагнитного поля.

Воздействие данного фактора проявляется в нарушении нормального функционирования опорно-двигательного аппарата человека-работника. Работая с компьютером, работник может часами находится в фиксированных позах, совершая движения лишь пальцами рук, работая на клавиатуре и с мышью. В результате чего у него может возникнуть, в той или иной форме, синдром длительных статических нагрузок (СДСН).

Одной из форм СДСН является так называемый запястный синдром, который мо­жет появиться при длительной работе с клавиатурой. СДСН также может возникнуть из-за длительного нахождения работник в сидячем положении. В этом случае возникает перенапряжение мышц спины и ног. Данная форма СДСН проявляется в виде болезненных ощущений в нижней части спины. У работника, которым приходится долго работать с мышью, может возникнуть СДСН, поражающий плечо и руку. При неправильном расположении дисплея могут страдать мышцы шеи и спины.

Предотвратить появление СДСН может правильная эргономическая организация компьютеризированного рабочего места.

Данные воздействия проявляются у работника в виде астенопии (быстрой утомляемости глаз) после продолжительной работы за компьютером. Кроме астенотопии у работника также возможно появление трудностей, связанных с фокусировкой зрения. Важным элементом профилактики зрительных заболеваний является правильный выбор видеоадаптера и видеомонитора, служащего для отображения информации. Эргономические исследования показали, что при прямом контрасте (символ темнее фона) частота кадра должна быть не менее 80 Гц, а при обратном контрасте (фон темнее символа) 50-60 Гц.

Немаловажным фактором, влияющим на зрение работника, является организация программного интерфейса. Неправильно организованный программный интерфейс может воздействовать на зрение двумя путями: прямо и косвенно. Прямое воздействие отражается в излишнем перенапряжении глаз от перегрузки (информационной или цветовой) экрана дисплея. Механизм косвенного воздействия гораздо сложнее. Разные люди имеют разное цветовосприятие. Цвет приятный одним людям, может быть дискомфортен для других. Если работник работает с программным продуктом, в интерфейсе которого широко используется неприемлемый для него цвет, то у него может возникнуть искусственная астенопия, которая является защитной реакцией мозга на внешний раздражитель. Устранить такое косвенное воздействие можно используя на рабочем месте монохромный видеомонитор или при проектировании программы предусмотреть возможность изменения пользователем цвета.

Шумовые воздействия, особенно длительные, могут в значительной степени влиять на работника приводить к снижению работоспособности.

Какую бы тревогу ни вызывали некоторые из отчетов и статистических данных, следует иметь в виду, что многие болезни, связанные с работой на компьютере, можно полностью предотвратить, изменив привычные методы работы и устройство рабочего места.

5.3. Пожарная опасность

Все осветительные сети рабочего помещения защищены от коротких замыканий. Защита сетей осуществляется плавкими предохранителями. Номинальные токи аппаратов защиты выбирались по возможности наименьшими по расчетным токам защищаемых участков сети, но с учетом того, чтобы аппараты защиты не отключали линии от пусковых токов источников света. Аппараты защиты устанавливают в цепях всех нормально незаземленных полюсов и фаз.

Пожарная безопасность осветительных приборов обеспечивается также выбором комплектующих изделий и материалов с тепловыми характеристиками, соответствующими тепловому режиму работы осветительных приборов.

Но все, же существует небольшая вероятность возникновения пожара, а как следствие, и опасность отравления работника продуктами горения и плавления пластмассы. Рассмотрим некоторые причины, ведущие к такому исходу. Корпус ПК содержит большое количество составляющих, выполненных из металла и пластика. При возникновении пожара или короткого замыкания, а также из-за неисправностей линий передач, возможно возгорание данных составляющих и их тления. Выделяющиеся в атмосферу вредные газы могут нанести ущерб здоровью работника путем его отравления. Использованные в процессе разработке и изготовление элементов ПК пластмасс и других горючих компонентов провоцирует указанный исход событий.

Согласно классификации по пожарной опасности (СНиП 2.09.02-85) лабораторию можно отнести к категории "В". К этой категории относятся производства, связанные с обработкой сгораемых веществ, материалов в холодном состоянии. По (СНиП 2.01.02-85) помещение можно отнести к II степени огнестойкости, т. к. все конструкции (стены, перекрытия) выполнены из негорючих материалов с пределом огнестойкости от 0.25 до 2.5 часов.

При тушении пожара применяют самые разнообразные огнетушительные вещества, самым дешевым и доступным из которых является вода. В административных зданиях для тушения пожара широко применяют огнетушители, которые располагаются в общедоступных местах чаще всего в коридорах зданий на каждом этаже. Наибольшее применение при тушении пожара находят химические пенные огнетушители ОХП–10 и ОХПВ–10. Для тушения пожара электрооборудования и радиоэлектронной аппаратурой следует применять углекислотно-бромэтиловые огнетушители ОУБ–3.

5.4. Экологические достоинства и недостатки разрабатываемого продукта для природной среды

Разработки в области прикладной науки и техники, помимо решения прямых задач проекта должны предусматривать экономию финансовых, энергетических или людских ресурсов.

Использование данной разработки предусматривает значительную экономию времени за счет того, что отпадает необходимость в расчете погрешностей по определенным формулам, и упрощается процесс управления. Это обеспечивает существенную экономию финансовых и энергетических средств и ресурсов пользователя, а также избавляет его от дополнительных нервно-эмоциональных нагрузок, связанных с проведением поверки.

Освободившиеся финансовые, энергетические и людские ресурсы могут быть направлены на решение других, возможно, более важных проблем.

Разработанный программный продукт для проведения моделирования систем не оказывает вредного воздействия на окружающую среду, так как для разработки использовалась ПЭВМ. Влияние таких негативных факторов как инфракрасное, тепловое, ионизирующее излучение, шум и вибрация ограничивается помещением, в котором находится ПЭВМ. Исходя из всего вышеописанного, можно заключить, что данный проект оказывает положительный экологический эффект на людей, его использующих и не несет вреда окружающей среде.

**ЗАКЛЮЧЕНИЕ**

В результате квалификационной работы была спроектирована информационнаясистема поверки электросчётчиков на примере Федерального бюджетного учреждения«Государственный региональный центр стандартизации метрологии и испытаний в г. Пятигорск».

Данная информационная система была разработана с целью повышения точности учета поверяемого оборудования,автоматизированного формирования плана поверки в соответствие со сроками, обязательных очередных поверок. Работа доведена до действующего макета информационной системы, выполнено комплексное тестирование системы. Показана работа способность и эффективность информационной системы поверки электрооборудования. Работа имеет перспективу развития для использования системы в филиальной сети.

Был проведен анализ экономической эффективности проекта. В результате сделаны выводы: затраты на разработку составят888 323,32рублей, а срок окупаемости около двух месяцев.

**СПИСОК ИСТОЧНИКОВ**

1. ГОСТ 8.259-2004 ГСИ.«Межгосударственная система стандартизации. Стандарты межгосударственные, правила и рекомендации по межгосударственной стандартизации. Порядок разработки, применения, обновления и отмены».

2. Федеральный Закон РФ от 23.09.1992 г. № 3523-I (в редакции от 24.12.2002 № 177-ФЗ) О правовой охране программ для электронных вычислительных машин и баз данных.

3. Леоненков А.В. Самоучитель UML. – СПб.: БХВ-Петербург, 2014. – 304 стр.: ил.

4. Принципы проектирования и разработки программного обеспечения. Учебный курс MCSD: Скотт Ф. Уилсон, Брюс Мэйплс, Тим Лэндгрейв. – М: Русская редакция, 2012. – 736стр.

5. Проектирование экономических информационных систем: Учебник/Г.Н.Смирнова, А.А.Сорокин, Ю.Ф.Тельнов. – М: Финансы и статистика, 2013. – 512стр.

6. Теория и практика построения баз данных: Д. Крёнке. – Питер, 2013. – 800стр.

7. Экономика, разработка и использование программного обеспечения ЭВМ / В.А.Благодатских, М.А.Енгибарян, Е.В. Ковалевская и др. - М.: Финансы и статистика, 2015.

8. Мишенин А. И.. Теория экономических информационных систем. М.: Финансы и статистика. 2012г., 240 стр.

9. Диго С.М. Базы данных: проектирование и использование: Учебник. - М.: Финансы и статистика, 2015, 592 стр

10. Сайт в сервисном центре Microsoft. WWW http:\\office.microsoft.com.

11. Сербулова Т.Н. Методические указания к выполнению раздела «Безопасность и экологичность» в дипломном проекте (работе), Кисловодск, КГТИ, 2014 г.